

M83-SERIES MODEL 8/32 MAINTENANCE MANUAL

CONSISTS OF:

GENERAL DESCRIPTION	
General Description	29-394A12
PROCESSOR	
Maintenance Specification	01-078R01A21
WRITABLE CONTROL STORE	
Installation Specification	35-555F01A20
Maintenance Specification	35-555F01A21
MEMORY	
Main Memory System Maintenance Specification	35-535R01A21
32KB Memory Maintenance Manual	29-370R03
EXTENDED SELECTOR CHANNEL	
Installation Specification	02-328R01A20
Maintenance Specification	02-328A21
DISPLAY PANEL	
Hexadecimal Display Panel Information Specification	09-065R01A12
DRAWINGS	
Back Panel Map	01-078R03D08
Processor CPU-A Schematic	35-536R17D08
Processor CPU-A Assembly	35-536R12E03
Processor CPU-B Schematic	35-537R09D08
Processor CPU-B Assembly	35-537R09E03
Processor CPU-C Schematic	35-555R06D08
Processor CPU-C Assembly	35-555R04E03
Processor IOU Schematic	35-539R15D08
Processor IOU Assembly	35-539R11E03
Processor ALU Schematic	35-538R05D08
Processor ALU Assembly	35-538R05E03
Memory Bus Controller (MBC) Schematic	35-535R11D08
Memory Bus Controller (MBC) Assembly	35-535R04E03
Local Memory Interface (LMI) Schematic	35-534R06D08
Local Memory Interface (LMI) Assembly	35-534R04E03
Extended Selector Channel Schematic	02-328M02R01D08
Extended Selector Channel Assembly	35-508M02R01E03
Hexadecimal Display Panel Schematic	09-065R01D08
Hexadecimal Display Panel Assembly	35-519R03D03
DMA Terminator Schematic	35-548C08
DMA Terminator Assembly	35-548B03
DMA Terminator Assembly	35-572B03
DMA Cable Assembly	17-336B03
Processor Bus Terminator Schematic	35-569D08
Processor Bus Terminator Assembly	35-569R01C03
LMB Terminator Assembly	35-578B03
I/O Bus Terminator Schematic	35-433B08
I/O Bus Terminator Assembly	35-433R02B03
Power Supply Schematic	34-024R02D08
Power Supply Assembly	34-024R01D03

THIS MANUAL CONTAINS PROPRIETARY INFORMATION AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF USING AND MAINTAINING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY AUTHORIZED IN WRITING.



Subsidiary of PERKIN-ELMER
Oceanport, New Jersey 07757, U.S.A.

M83 SERIES

MODEL 8/32 PROCESSOR

GENERAL DESCRIPTION

1. INTRODUCTION

The Model 8/32 Processor is a 32 bit micro-programmed minicomputer. By combining advanced circuits, packaging, and micro-programming, INTERDATA gives the user a price/performance optimized machine. The Model 8/32 was developed because of a need for a high-speed 32 bit minicomputer. Because of INTERDATA's experience with 16 bit minicomputers and a user instruction format that is readily suited for a 32 bit machine, INTERDATA is able to provide a 32 bit machine. The Model 8/32 is upward compatible with current INTERDATA Processors. Through micro-programming, the Model 8/32 Processor is able to provide present and future owners of INTERDATA's 16 bit minicomputers the ability to grow into a 32 bit Model 8/32 without having to regenerate all the programs that were created on their 16 bit Processors.

The Model 8/32 has 148 instructions defined which include arithmetic and logical, operational, list processing, floating point, cyclic redundancy checking, and bit and byte manipulation instructions. Double indexing is also allowed, along with a multitude of branch instructions. There are 40 extended branch instructions (Mnemonics) defined which brings the total instructions to 180. Through these instructions and direct addressing, coding and debugging time is reduced to a minimum.

The Model 8/32 offers 32 General Registers, each 32 bits wide in two sets of 16, (optionally expandable to 8 sets). Stack set selection is controlled by bits in the Program Status Word. The multi-stack organization offers fast and simple context switching without the necessity of storing and restoring register stacks. See 32 Bit Series Reference Manual, Publication Number 29-365.

The Model 8/32 provides a flexible input/output system in addition to the conventional means of programmed I/O. The Model 8/32 can have up to 1,024 auto driver channels. These provide fast automatic character input/output operations, including automatic conversion from one character to another. Each character is transferred into or out of memory without any effect on a running program except for a small amount of stolen time.

The Model 8/32 is capable of directly addressing up to 1,048,576 bytes of memory. Memory is constructed of 32KB memory modules. Memory is addressable to the eight-bit byte level. No paging or indirect addressing is required by the user instruction sets.

The Memory Access Controller (MAC) uses sixteen 32 bit hardware registers to allow segmentation, relocation, and memory protection of user programs. There are five different ways memory may be protected.

1. Address invalid—not accessible to user
2. Non-present address
3. Write-protected; read only
4. Write then interrupt
5. Instruction execution not allowed

The 8/32 Memory Bus Controller provides access to the extended Direct Memory Access Bus. The extended Direct Memory Access Bus is a high quality, high speed bus that may have up to seven extended Direct Memory Access (DMA) devices on the bus.

2. SCOPE

This specification is intended to enable the digital technician to understand the INTERDATA documentation system. Number Notation, the Part Numbering System, and the Drawing System are described. A cross reference between INTERDATA part numbers and standard industry part numbers for Integrated Circuits, Transistors, etc., is found in Appendix 1. A list of other publications which may be useful in the programming or trouble shooting of the Model 8/32 system is found in Table 1.

TABLE 1. RELATED PUBLICATIONS

Title	Publication Number
32KB Memory Maintenance Manual	29-370*
Multiplexor Bus Buffer Instruction Manual	29-267
Loader Storage Unit Users Manual	29-308
Paper Tape Reader/Punch Instruction Manual	29-291
Universal Logic Module Instruction Manual	29-311
1600 BPI Magnetic Tape Instruction Manual	29-309
Cassette Users Manual	29-298
7 Inch Teletype Manual	29-288

*Will be included in this Manual.

3. BLOCK DIAGRAM

A simplified block diagram of the Model 8/32 system is shown in Figure 1. The Processor logic is contained on three circuit boards. The Memory Bus Controller (MBC), Local Memory Interface (LMI) and memory are contained on separate boards. The Arithmetic/Logic Unit (ALU) and Input/Output Unit (IOU) are also separate boards.

<u>Part Number</u>	<u>Description</u>	<u>Card File Position</u>
35-534	Local Memory Interface (LMI)	3T, 6T
35-535	Memory Bus Controller (MBC)	1T
35-536	Control Processor A (CPA)	0T
35-537	Control Processor B (CPB)	7B
35-538	Arithmetic Logic Unit (ALU)	4B
35-539	Input/Output Unit (IOU)	3B
35-555	Control Processor C (CPC)	6B
--	Input/Output	2B, 1B, 0B
--	Memory	7T, 5T, 4T, 2T
--	Options	5B

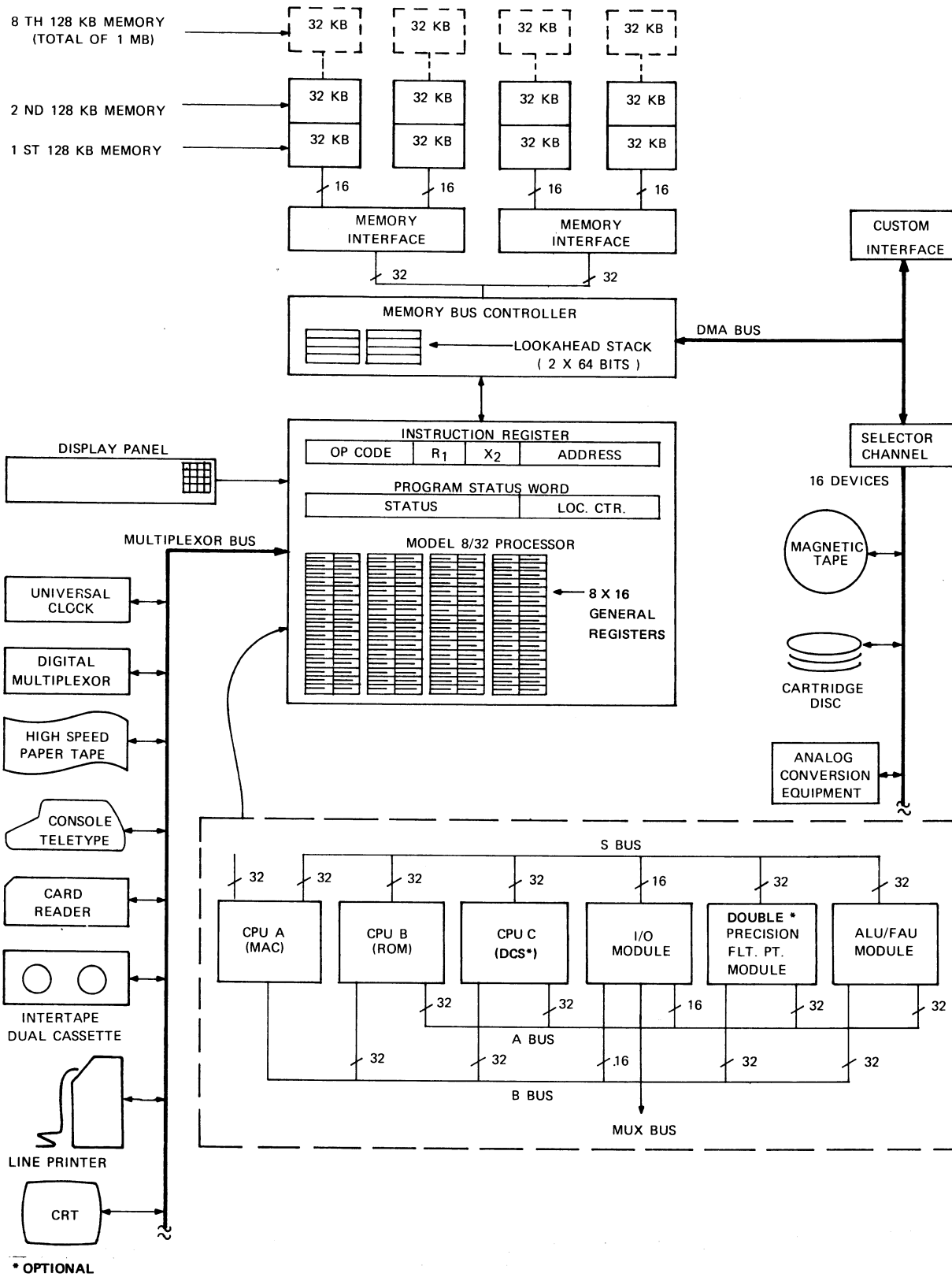


Figure 1. 8/32 Processor Block Diagram

4. DOCUMENTATION

This section describes the style and conventions used with INTERDATA documentation.

4.1 Number Notation

The most common form of number notation used in INTERDATA documentation is hexadecimal notation. In this system, groups of four binary digits are represented by a single hexadecimal digit. Table 2 lists the hexadecimal characters employed.

TABLE 2. HEXADECIMAL NOTATION DATA

Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal
0000	0	0	0110	6	6	1100	12	C
0001	1	1	0111	7	7	1101	13	D
0010	2	2	1000	8	8	1110	14	E
0011	3	3	1001	9	9	1111	15	F
0100	4	4	1010	10	A			
0101	5	5	1011	11	B			

To differentiate between decimal and hexadecimal numbers, hexadecimal numbers are preceded by the letter "X", and the number is enclosed in single quotation marks. Examples of hexadecimal numbers are: X'1234', X'2EC6', X'A340', X'EEFA', and X'10B9'.

4.2 Part Numbering System

INTERDATA parts, drawings, and publications employ a common numbering system. The part number and drawing numbers for drawings which describe the part are related. The publication number is also often related to the part number of the device or program described. Figure 2 shows the format used for INTERDATA part numbers. The fields are described in the following paragraphs.

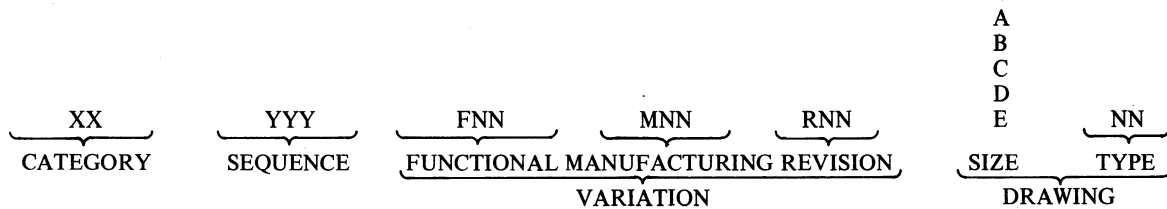


Figure 2. Part Number Format

4.2.1 Category Field. The two-digit Category number indicates the broad class or category to which a part belongs. Typical examples of category number assignments are:

- 01 - Basic Hardware Systems
- 02 - Basic Hardware Expansions
- 03 - Basic Software Systems
- 04 - Software Packages
- 05 - Micro-programs
- 06 - Test Programs
- 07 - Subroutines of General Utility
- 10 - Spare Parts Packages
- 12 - Card File Assemblies
- 13 - Panels
- 17 - Wire and Cables
- 19 - Integrated Circuits
- 20 - Transistors
- 27 - Peripheral Equipment
- 29 - Manuals
- 34 - Power Supplies
- 35 - Assembled Printed Circuit Boards
- 36 - Electro-Mechanical Devices

4.2.2 Sequence Field. The Sequence number identifies a particular item within the category. Sequence numbers are assigned serially, and have no other significance.

NOTE

The Sequence Field, like all other part number fields, may be lengthened as required. The field lengths shown on Figure 2 are minimum lengths (insignificant zeros must be added to maintain these minimums).

A part number must contain a Category number and a Sequence number. All other fields are optional.

4.2.3 Functional Variation Field. The optional Functional Variation Field consists of the letter "F" followed by two digits. The F field is used to distinguish between parts which are not necessarily electrically or mechanically equivalent, but which are described by the same set of drawings. For example, a power supply may be strapped internally to operate on either 110 VAC or 220 VAC. Except for this strap, all power supplies of this type are identical. The strapping option is easily described by a note on the assembly and test specification drawings. Therefore, this is a functional variation.

4.2.4 Manufacturing Variation Field. The optional Manufacturing Variation Field consists of the letter "M" followed by two digits.

The M Field is used to distinguish between parts which are electrically and mechanically equivalent (interchangeable), but which vary in method of manufacture. For example, if leads are welded instead of soldered on an assembly, the M Field changes.

An important exception to the meaning of the M Field exists for categories related to software. In software, the M Field number, when used, indicates the form in which a particular program is presented. For example, define a program as a set of machine instructions. These same identical instructions may be presented on punched cards, paper tape, or magnetic tape; and for any of these they could be in symbolic form or in relative or absolute binary form. Thus, there are many ways to present the same identical program.

The format for the M field and its meaning for software is:

Mxy

where x identifies the media selection (i.e., paper tape, mag tape, cassette, etc.) and y identifies object or source and the format.

<u>Meaning of x</u>		<u>Meaning of y</u>
Paper tape	1	1 Object program standard format 32 bit Processor
Cassette	2	4 Memory Image
Mag tape (800)	3	6 Object program standard format 16 bit Processor
Cards	4	7 Object non-standard format
Disc (2.5)	5	8 Object established task
		9 Source program

The above numbers refer to the physical program placed on an approved media for INTERDATA Software. A paper tape object program in standard format and for a 16 bit Processor has an M16 identifier. A magnetic tape object program in standard format and for a 32 bit Processor has an M31 identifier.

In addition to the above, there are three unique M numbers which have special meaning:

M99 always refers to a documentation package.

M00 always refers to a conceptual object program divorced from any media. This reference is used for all parts lists when object programs may be on any media.

M09 always refers to a conceptual source program and is used on all parts lists where any media may be used.

NOTE

M00 and M09 may only be used on parts lists and never identify a physical program on any media.

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

4.2.5 Revision Field. The optional Revision Field consists of the letter "R" followed by two digits. The R Field is used to indicate minor electrical or mechanical changes to a part which do not change the part's original character. R Field changes often reflect improvements. A part with a revision level HIGHER than the one specified will work. A part with a revision level LOWER than specified should not be used.

4.2.6 Drawing Field. The optional Drawing Field consists of a letter from "A" to "E" followed by two digits. The letter indicates the size of the original drawing. The sizes for each letter are:

A - 8½" X 11"

B - 11" X 17"

C - 17" X 22"

D - 22" X 34"

E - 34" X 44"

The two digits indicate the drawing type as follows:

01 - Parts List	13 - Program Listing
02 - Machine Details	14 - Abstracts
03 - Assembly Details	15 - Program Description
05 - Art Details	16 - Operating Instructions
06 - Wire Run List	17 - Program Design Specifications
08 - Schematic	18 - Flow Charts
09 - Test Specification	19 - Product Specification
10 - Purchase Specification	20 - Installation Specification
11 - Bill of Material	21 - Maintenance Specification
12 - Information	22 - Programming Specification

4.2.7 Examples. The following list provides examples of the part numbering system. The numbers were arbitrarily selected, and in most cases are fictitious.

35-060	The 60th printed-circuit board assigned a part number under this system.
35-060M01	A printed circuit board electrically and mechanically interchangeable with the 35-060, but differing in method of manufacture.
35-060F01	A printed-circuit board not electrically and mechanically interchangeable with the 35-060, but described by the same set of drawings.
35-060R01	A revised 35-060 printed-circuit board. Probably supercedes the 35-060.
35-060A01	The 8½ by 11 inch parts list for a 35-060.
35-060B08	The 11 by 17 inch schematic for a 35-060.
06-072	The 72nd utility program assigned a part number.
06-072A13	An 8½ by 11 inch listing of the 06-072 program.
06-072M03	An absolute binary deck of punched cards for the 06-072 program.
06-072A12	An 8½ by 11 inch information drawing on the 06-072 program. Probably a part of the program.
29-060	The 60th manual assigned a number under this system. Note that this number is not referenced in any way to the part number of equipment described in the manual.

4.3 Drawing System

This section describes the drawings provided with INTERDATA equipment. Note that drawings provided with peripheral devices and other purchased items may vary from the system described in this section.

A digital system may be divided into a collection of functionally independent circuits such as memory, Processor, and I/O device controllers. These circuits may or may not be saleable units in their own right, but in the electrical sense they are essentially self contained and capable of performing their function with minimum dependence on other functional circuits in the system. Hence a functional circuit is treated as a building block. Each schematic contains a variety of information including type and location of discrete Integrated Circuits (IC's), pin connections, all interconnections within the schematic, connector pin numbers and connections to other schematics. Further, the schematics are drawn to reflect, in an orderly fashion, all logical operations performed by the circuits. Generally, symbols used on schematics conform to MIL-STD-806B.

Registers are named according to the following rules:

1. The register Mnemonic name has a maximum of three letters, excluding "I, O, Q, and Z".
2. Each bit in the register is numbered, usually starting at 00 on the left, or most significant positions, and continuing to N-1 on the right, where N is the number of bits in the register.
3. The 00 bit is the Most Significant Bit and the N-1 is the Least Significant Bit.

The IC's, mounted directly on the logic board, are represented on the schematic drawings by logic symbols. Each symbol contains the reference designation, device part number (category and sequence), and symbol Mnemonic designation. Refer to Figure 3.

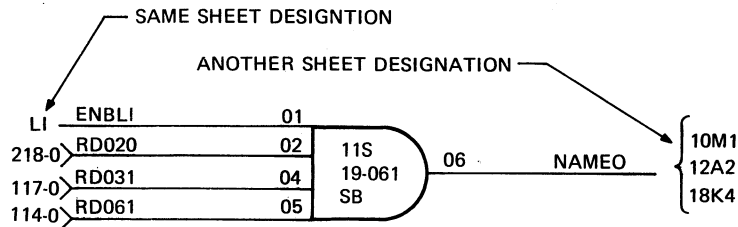


Figure 3. Example of a Schottky Buffer

The designations, numbers, and references shown in Figure 3 are:

11S - This indicates the component location on the logic board. Figure 4 illustrates the method generally used to determine component location on a logic board. With the logic board oriented so that the header connectors (Conn 0 and Conn 1) are on the right, the components are numbered from left to right starting in the upper left corner. That is, the first IC in the upper left corner is 00A and the first capacitor is C1. Test points are lettered right to left from A-Y (omitting I, O, L, E).

19-061 - The number 19 is the category number of ICs, and the 061 is the sequence number of the component.

SB - Indicates this component is a Schottky Buffer. Some other common designations used are:

- P - Power Gate
- SA - Schottky AND Gate
- SB - Schottky Buffer
- SG - Schottky Gate
- SGO - Schottky Gate, Open Collector
- HG - High Speed Gate
- HPO - High Speed Power Gate, Open Collector
- SFF - Schottky Flip-flop

L1 - This input lead is from area L1 on the same schematic sheet.

10M1, 12A2, 18K4 - Indicate outputs to another logic schematic sheet.

218-0, 117-0, 114-0, - Indicate inputs from Connector 0.

Note that the pin numbers (01, 02, 04, 05 and 06) correspond directly to the actual IC pin numbers.

Figure 4 also shows the locations of the header connectors (Conn 0 and Conn 1) and the cable connectors (Conn 2 and Conn 3). All logic boards always contain Header Connectors 0 and 1, however, any combination (either, both, or none) of cable connectors (Conn 2 and Conn 3) may be provided.

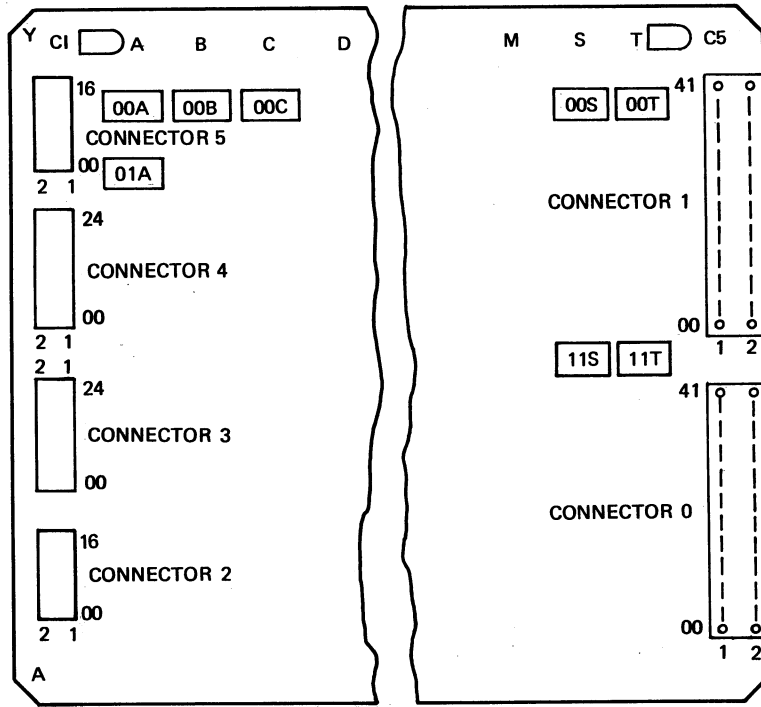


Figure 4. Example of a Logic Board Layout

Clocked devices, flip-flops and counters in particular, are drawn in a manner which indicates information concerning their inputs. An input which has a circle adjacent to the pin designation implies a low active signal is required to perform the specified operation. In addition, a rotated V at the clock input shows that the device changes state on an edge. Thus, if no circle is present the chip is positive edge triggered. Refer to Figure 5 for examples.

Figure 6 provides the pin numbering scheme for the header and cable connectors. Header connectors always have 2 rows of pins and 42 positions. Cable connectors always have 2 rows of pins but may vary in the number of positions.

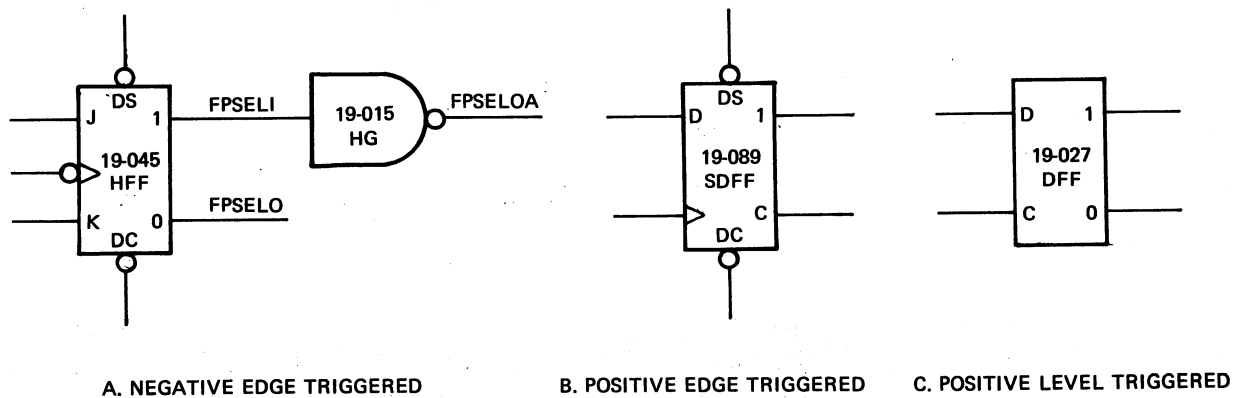


Figure 5. Examples of Clocked Devices

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

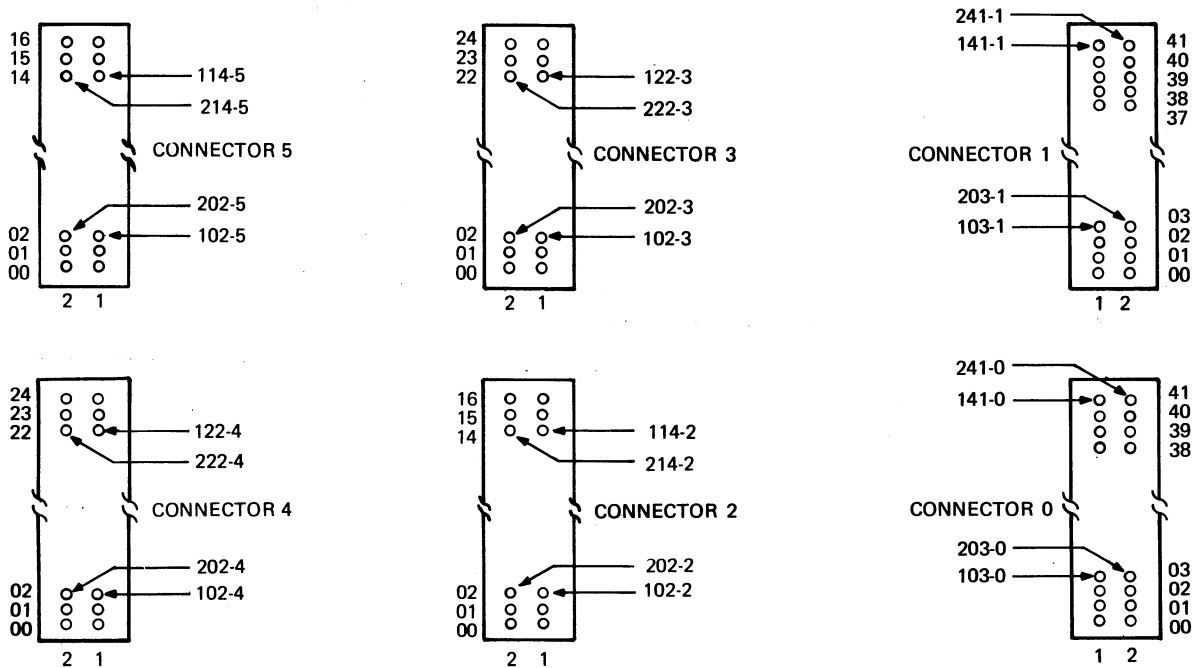


Figure 6. Connector Pin Numbering

A net is defined as an electrical connection between two or more points in a circuit. Ordinarily, a net has an originating end (usually an output where the signal is generated) and one or more terminating ends. Often it is convenient to assign descriptive mnemonic names to nets as a way of identifying them on schematics. Whether a net is named or not is sometimes arbitrary. However, a net is always assigned a name if:

1. The net is contained on one drawing sheet but is not shown as a complete solid line on that sheet.
2. Part of the net appears on more than one sheet.
3. Part of the net connects with a different schematic.
4. Part of the net leaves a logic board.

If a net is named, the following rules are observed.

1. All mnemonic names are a maximum of six characters.
2. All decimal digits and upper case letters are permitted.
3. No other characters permitted.
4. Where possible, Mnemonics are descriptive. However, it should be recognized that descriptive names are not always possible and a danger of misinterpreting a Mnemonic exists.
5. Mnemonic names are not repeated within a schematic.
6. Every Mnemonic is suffixed by a state indicator. This indicator consists of the digit "1" for the logically true state, or the digit "0" for the logically false state. For example, the set side of a flip-flop would have the "1" state indicator, while the reset side would have the "0" state indicator. The state indicator for a function changes each time that function is inverted. Thus, the state indicator permits assigning the same Mnemonic to functions that are identical except for an inversion.
7. When a logical function is inverted, an inversion indicator is added after the state indicator. This allows for functionally equivalent, but electrically different nets to have the same Mnemonic name. For example, assume a signal NAME1, NAME1 may be inverted to produce NAME0. If NAME0 is then inverted, NAME1A is produced. NAME1 and NAME1A are functionally equivalent, but physically different nets.

Sometimes a net fans-out to many sheets in a schematic. It is also possible for a net to fan-out to sheets in different schematics. In these situations, the net is assigned a mnemonic name. The net is also "zoned" from sheet to sheet to allow for properly identifying the originating and terminating ends of the net. The originating end of a net is defined as the collector at which a signal is generated. All other points to which the net connects are called terminating ends. When a lead leaves a sheet at the originating end, it is zoned to each and every sheet on which the net reappears, by indicating first the page number, followed by the schematic number that contains the page. For example, assume that the gate shown on Figure 3 is on a schematic, Sheet 20. The output NAME0, appears on Sheets 10, 12 and 18 of the schematic. Note that the schematic number is implied. When a net enters a sheet from another sheet, it is labeled, with the same Mnemonic name, and is zoned back to the originating end of the net only. Thus, on Figure 3, the ENBL1 may, however, have many other terminations in addition to the one shown. Generally, then, when a net leaves the sheet where it originates, it is zoned to every other sheet where the net terminates, while the terminating end is zoned only to the originating sheet. Note that in the Model 8/32 schematics, signals are co-ordinated between sheets only when the sheets are related to the same board. When a signal leaves a board, the Back Panel Map must be used.

When a lead leaves a logic board, it usually does so through a logic board back panel connector pin. These connector pins must be shown on the schematic even if the complete net is shown on one drawing sheet. Only the connector pin number need be indicated under the pin symbol, since the connector number itself is implied by the logic board location number in the logic symbol or in the footnote. Thus, on Figure 3, RD061 enters the logic board on Pin 114 of Header Connector 0.

Figure 7 is a typical schematic sheet with call-outs illustrating many of the conventions described in this section.

The schematic drawings for the basic Digital System and some of the more common expansions are commonly included in the rear of the appropriate Digital System Maintenance Manual. Schematic drawings for other expansions are included with the expansion or with the publications that describe the expansion.

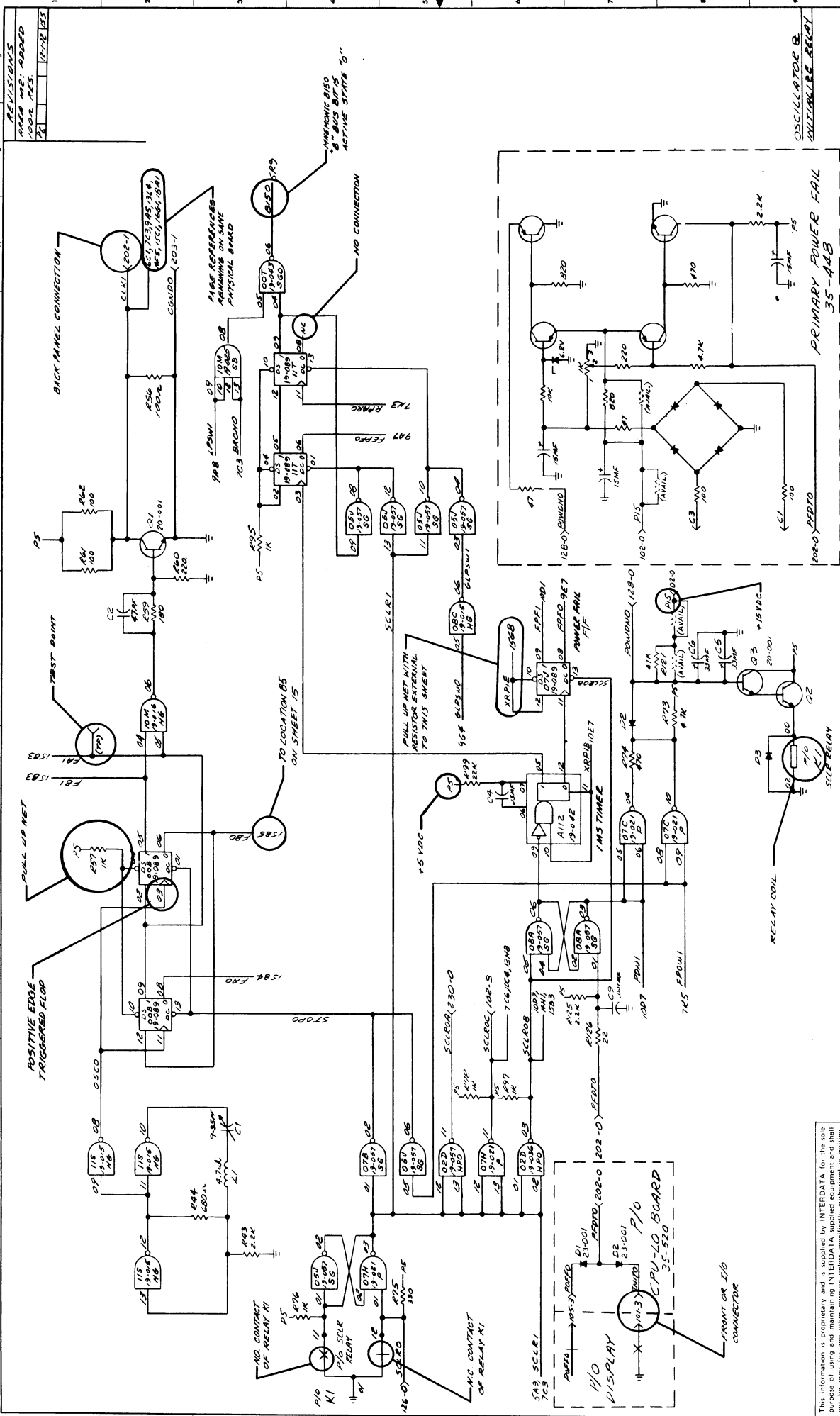


Figure 7. Functional Schematic Format Drawing

REVISIONS
 REVISION NO. DATE
 1 10/22/78
 2 11/12/78

FILE NO. 11-9121
 TITLE FUNCTIONAL SCHEMATIC
 MODEL 35-448
 PROJECT PROCESSOR
 DATE 11-9-78
 DRAWN BY E. MCGINLEY
 CHECKED BY
 DESIGNED BY
 APPROVED BY

ALL INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE USE OF THE CUSTOMER. INTERDATA ACCEPTS NO LIABILITY FOR THE USE OR MISUSE OF THIS INFORMATION FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY AUTHORIZED IN WRITING.

NOTES
 ALL APPROPRIATE ON THIS SHEET LOCATED ON CPU-I/O BOARD 35-448 UNLESS OTHERWISE SPECIFIED

APPENDIX 1

PART NUMBER CROSS REFERENCE TABLE

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
19-001	Dual 4 Input Nand DTL	15861
19-002	Triple 3 Input Nand DTL	15863
19-003	Quad 2 Input Nand DTL	15849N
19-004	Hex 1 Input Nand DTL	15837N
19-005	Dual Power Gate DOC	8633N
19-006	Dual Buffer DTL	1582N
19-007	Flip-Flop DTL	15848N
19-008	Gate Expander Dual 4 Input DTL	15833N
19-009	8 Bit Stack DTL	903059 (Fairchild)
19-010	Differential Compator LIN	72710L
19-012	Dual 4 Input Buffer TTL	74H40H
19-013	Quad 2 Input Nand DTL	15846
19-014	Dual J-K Flip-Flop DTL	158097N
19-015	Hex Inverter 1 Input	74H04H
19-016	Quad 2 Input TTL	74H00N
19-017	Triple 3 Input TTL	74H10N
19-018	Dual 4 Input TTL	74H20N
19-019	Single 8 Input TTL	MC3015 (Motorola)
19-020	Operational Amplifier LIN	MC1709C (Motorola)
19-021	Quad 2 Input Power DOC	15858N
19-022	Dual J-K Flip-Flop TTL	MC3061P (Motorola)
19-023	Selected Dual Buffer 19-006 with 20-30 nsec. delay DTL	15832N
19-024	Triple 3 Input AND TTL	74H11N
19-025	Dual 4 Input AND TTL	74H21N
19-026	2-2-2-3 Input AND-OR TTL	74H52

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
19-027	4 Bit Latch TTL	7475N
19-028	4 Bit Adder TTL	7483N
19-029	Quad Exclusive - OR TTL	7486N
19-030	4 Bit Shift Register TTL	7495N
19-031	One Shot TTL	74121N
19-032	1 out of 10 Decoder TOC	74145N 5445 7445
19-033	Sense Amplifier LIN	7524N
19-034	Retriggerable One Shot TTL	74122N
19-035	4 Bit Counter TTL	74193N
19-036	Quad 2 Input Open Collector TTL	7438N
19-037	High Performance Operational Amp	7748393 (Fairchild)
19-038	Dual 4 line to 1 line Mux TTL	74153
19-039	4 Bit ALU TTL	74181
19-040	Look Ahead Carry TTL	74182
19-041	4 x 4 Register Stack TTL	74170
19-042	Dual Retriggerable One Shot TTL	74123N
19-043	Quad 2 Input Open Collector TTI	74H01N
19-044	Hex Inverter Open Collector TTL	74H05N
19-045	Dual J-K Flip-Flop TTL	74H106
19-046	Quad RS-232C Line Driver	MC1488L (Motorola)
19-047	Quad RS-232C Line Receiver	MC1489AL (Motorola)
19-048	8 Bit Shifter	74198N
19-050	8 Input Nand TTL	74H30
19-051	1024 Bit PROM TTL	74187 (Fairchild)
19-055	Quad 2 Input Nand STTL	74S00
19-056	Quad 2 Input Nand Open Collector STTL	74S03
19-057	Hex 1 Input Inverter STTL	74S04
19-058	Triple 3 Input Nand STTL	74S10
19-059	Triple 3 Input AND STTL	74S11
19-060	Dual 4 Input Nand STTL	74S20

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
19-061	Dual 4 Input Buffer STTL	74S40
19-062	2-2-3-4 Input AND-OR Inverter STTL	74S64
19-063	Dual D Edge Triggered Flip-Flop STTL	74S74
19-064	Dual J-K Flip-Flop STTL	74S112
19-065	Quad 2:1 Mux Non-inverting STTL	74S157
19-066	Quad 2:1 Mux Inverting STTL	74S158
19-067	4 Bit ALU STTL	74S181
19-068	Carry Look Ahead STTL	74S182
19-069	8 line to 1 line Mux STTL	74151
19-070	4 Bit Synchronous Counter TTL	74161
19-071	Quad D Edge Triggered Flip-Flop	74175
19-072	4 Bit Left/Right Shift Register TTL	74194
19-073	Dual 4:1 Mux Tri-State TTL	8214 (National)
19-074	8 Bit Priority Encoder TTL	9318 (Fairchild)
19-075	16 x 4 Register Stack TTL	3101A (Intel)
19-076	1024 Bit Memory MOS	TM54062
19-077	256 Bit Memory TTL	6531 (Monolithic Memories)
19-078	Dual 4 Input Nand-OC	74S22
19-080	High-Speed PROM	82S29 (Signetics)
19-081	Univ. Asynchronous Receiver/Transmitters	TR1042A (Western Digital)
19-082	2-2-3-4 Input AND-OR Invert Open Collector STTL	74S65
19-083	9 Bit Parity Generator/Checker STTL	82S62 (Signetics)
19-085	Monolithic Timing Circuit	MC1555 (Motorola) NE555V (Signetics)
19-086	741 C DIP Operational Amplifier	U6A7741393 (Fairchild)
19-087	747 DIP Operational Amplifier	U7A774 (Fairchild)
19-088	737 C DIP Operational Amplifier	U6A773393 (Fairchild)
19-089	Dual D Edge Triggered Flip-Flop	74H74
19-090	High Speed (710) Differential Comparator DIP	U6A771093 (Fairchild)
19-091	Retriggerable Single One Shot	9600 (Fairchild)

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
19-092	Negative Voltages Regulator	MC1463R (Motorola)
19-093	Positive Voltages Regulator	MC1469R (Motorola)
19-094	Voltage Regulator	U6A7723393 (Fairchild) MC1723CL (Motorola)
19-095	Linear Positive Voltage Regulator	U9H7805393 (Fairchild)
19-096	First In-First Out Serial Memory 64 Word 4 Bit	3341 (Fairchild)
19-097	Amplifier	LH0002H (National)
19-098	Quad 2:1 Multiplexor Non-Inverting	74157
19-099	Dual Sense Amplifier	75234N
19-100	Driver	75452N
19-101	4-2 Input Buffer	7437N
19-102	6-1 Input Buffer OC	7407N
19-103	1 out of 10 Decoder	7442N
19-104	Current Switch	75325N
19-105	Dual Differential Driver	Fairchild 9614
19-106	Dual Differential Receiver	Fairchild 9615
19-107	Sense Amplifier	SN7520N
19-108	Quad 2 Input Nand	SN7400N
19-109	Hex Inverter Open Collector	SN7406N
19-110	Hex Inverter	SN7404N
19-111	Dual 4 Input Nand	SN7440N
19-112	Optically Coupled Isolator	TIL-111 4N25
19-113	360 Dual Transmitter	TI 75123
19-114	360 Triple Receiver	TI 75124
19-115	Quad 2 Input AND	74H08
19-116	Dual 4:2 Multiplexor STTL	74S153
19-117	4 Bit Magnitude Comparator STTL	74S85
19-118	Quad Bus Transceiver TTL	26S12A
19-119	Expandable AND-OR Invert TTL	74455
19-120	Dual Timer	Signetics NE556

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
19-121	Matched Pair 19-085 (P. S. Timing)	See 19-085
19-123	Dual Voltage Controlled Oscillator	74S124
19-124	Quad 2 Input Nand Buffer, STTL	74S37
19-125	O.C. Quad 2 Input Nand Buffer, STTL	74S38
19-126	Dual 2 Wide-2 input AOI, STTL	74S51
19-127	Quad Exclusive-OR, STTL	74S86
19-128	12 Input Nand-Tri State, STTL	74S134
19-129	3 to 8 Decoder, STTL	74S138
19-130	Dual 4 in. Nand 50 Ohm Driver, STTL	74S140
19-131	Quad D Flip Flop, STTL	74S175
19-132	Quad 2/1 Mux.-Tri State, STTL	74S258
19-133	4 Bit Binary Adder, TTL	74283
19-134	Hex Buffers/Inverts, TTL	8T98
19-135	4 Bit Binary Counter, STTL	93S16
20-001	Transistor NPN High Speed Switch	2N3646
20-002	Transistor PNP 500 MA	MPS6534 (Motorola)
20-003	Transistor	2N3902
20-004	Transistor NPN	2N5189
20-006	Transistor NPN 15 Amps 100W T03 case	2N3055 (RCA)
20-007	Transistor NPN 3 Amps	TIP31A
20-008	Transistor PNP 3 Amps	TIP32A
20-009	Transistor Triac 2 Amps 100V	A03001 (Electronic Control Corp).
20-010	Transistor NPN 500 MA Code Driver	2N5845
20-011	Transistor Photo	2N5777
20-012	Transistor PNP High Current Switch	2N2907
20-013	Transistor NPN	2N3303
20-014	Transistor NPN	2N4238
20-015	Transistor PNP	2N4235
20-016	Transistor PNP	2N3740

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
20-017	Transistor NPN	2N3766
20-018	Transistor, Power Silicon NPN	2N3054
20-019	Transistor NPN Fast PWR Switch	2N6308 (Motorola)
20-020	Transistor Switching 1 Amp T05 can	2N3725
20-021	Transistor NPN Silicon	MPS3646 (Motorola)
20-022	Transistor NPN	1N1711
20-023	Transistor PNP	2N2905A
20-024	Transistor Switch	2N3776
20-025	PNP HI SPEED Switch	2N3467
20-026	Transistor Module, Quad	MPQ3725
20-027	Transistor	2N2369
20-029	Transistor	
21-025F01	1K ohm-15 to Common DIP	898-1-1K ohm (Beckman)
21-025F02	470 ohm-15 to Common DIP	898-1-470 ohm (Beckman)
21-025F03	330 ohm-15 to Common DIP	898-1-330 ohm (Beckman)
23-001	Diode High Speed-High Current	1N914
23-002	Diode 5.1 V Zener	1M5.1ZS5 (Motorola)
23-003	Diode 10V Zener	1M10ZS5 (Motorola)
23-004	Diode 6.2 V Zener	1M6.2ZS5 (Motorola)
23-007	Diode Mot Bridge	MDA962-2 (Motorola)
23-008	Diode Int. Rectifier	40HF-5R
23-009	Diode	1N4735
23-010	Diode Int. Rectifier	S1Y1P
23-011	Diode Rectifier	2N681
23-012	Diode Thermister	KA31J1 (Fenwall)
23-013	Diode 9.4V	1N2163
23-014	Diode	1N3880
23-015	Diode	1N3889
23-016	Diode Bridge Recitifer	VS448 (Varo)
23-017	Diode	1N2070

<u>INTERDATA Part Number</u>	<u>Type</u>	<u>JEDIC Number</u>
23-018	Diode 18 V Zener	1N4746A
23-019	Diode	1N3615
23-020	Diode 8.2V Zener	1N756A
23-021	Diode 9.1 V Zener	1N757A
23-022	Diode 3.3V Zener	1N746A
23-023	Diode Bridge Rectifier	KBH2506 (General Instrument)
23-024	Diode, Power Fast Rec. 30 Amps.	1N3909
23-025	Diode, Power Fast Rec. 3 Amps.	A115A (General Electric)
23-026	Triac 600V 30 Amps	2N6162
23-027	Diac 32V	1N5761
23-028	Power SCR Thyristor	2N4441
23-029	Diode	1N4607
23-030	Diode	1N4156
23-031	Diode 6.6 V Zener	1N4736
23-032	Diode 8.8 V Zener	1N4739
23-033	16 Diode Array	45190 (Litton)
30-018	100 nsec. Delay Line 10 taps	30-018 (Princeton Advanced Eng.)
30-019	50 nsec. Delay Line 10 taps	30-018 (Princeton Advanced Eng.)
33-034	PR CKT Count Butt Contact Push Button Switch ¼ AMP SPDT	
33-035	8 Pole Dip Switch AMP Part No. 4351665	

MODEL 8/32 PROCESSOR MAINTENANCE SPECIFICATION

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Packaging	1
1.2 Processor	1
1.3 Control Store	1
1.4 Peripherals	1
2. INTERNAL ARCHITECTURE	1
2.1 Modules	2
2.2 Micro-Instructions	2
2.3 Interrupts	2
2.4 Registers	2
2.5 Processor Timing	2
3. FUNCTIONAL DESCRIPTION OF THE BASIC PROCESSOR	2
3.1 Processor Busses	2
3.2 Registers	4
3.3 Interrupts	4
3.4 Control Store Memory	4
3.5 Micro-Programming	5
3.6 Processor Block Diagram Analysis	13
4. CPA GENERAL DESCRIPTION	17
4.1 CPA Block Diagram Description	17
4.2 Memory Addressing	19
4.3 Memory Reference Operations	24
4.4 S Bus Operations	34
4.5 B Bus Operations	35
5. CPB FUNCTIONAL DESCRIPTION	36
5.1 State Counter and Control	36
5.2 Clock and Control	37
5.3 Control Store	38
5.4 Bus Selection	39
5.5 Interrupts	40
5.6 PSW Register	41
5.7 Branch Control	41
5.8 A, B, and S Gating	41
5.9 Test Aids	41

TABLE OF CONTENTS
(Continued)

6. CPC GENERAL DESCRIPTION	42
6.1 A and B Stacks (ASTK and BSTK)	43
6.2 S Buffer (SBUFF)	43
6.3 Stack Addressing	43
6.4 Read/Write Control	46
7. ALU	46
7.1 Arithmetic State Register (AS)	48
7.2 ALU Algorithms	48
7.3 ALU Clock	57
7.4 Arithmetic Iterative Counter (ACNT01:05)	58
7.5 Arithmetic Condition Code	58
7.6 Arithmetic Elements and ROM Control	59
7.7 MQ Register	59
7.8 AL Register and Shift Multiplexors	61
7.9 Exponent Arithmetic	61
8. I/O GENERAL DESCRIPTION	61
9. FUNCTIONAL DESCRIPTION	61
9.1 I/O Control Functions	61
9.2 Machine Malfunction and Power Fail Hardware	64
10. MULTIPLEXOR CHANNEL (MUX) BUS	64
10.1 Multiplexor Channel IOU	64
10.2 Multiplexor Channel Timing	66
10.3 Multiplexor Channel and Multiplexor Operations (MUX)	69
11. BYTE MANIPULATION AND AUXILIARY FUNCTIONS	75
11.1 Byte Manipulation Functions	75
11.2 Auxiliary Functions	78
12. DISPLAY COUNTER	79
12.1 Addressing Logic	79
12.2 Data Output	79
12.3 Data Input	79
12.4 Status Input	79
12.5 Control Logic	79
13. TELETYPE CONTROLLER	79
13.1 Block Diagram Analysis	80
13.2 Bus Communication and Address Circuits	80
13.3 Status and Commands	80
13.4 Timer Circuits	81
13.5 Data Output	81
13.6 Data Input	83
13.7 Interrupt Circuit	85
13.8 Initialization	85
13.9 TTY Timer Adjustment	85
13.10 Machine Control Register (MCR)	85
13.11 Power Monitor and System Initialize	85
13.12 Primary Power Fail Check	87
13.13 Start Timer	87

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

TABLE OF CONTENTS
(Continued)

14. IOU S AND D BUS ROM CONTROLLERS	88
14.1 IOU S Bus High ROM Controller (19-142F45)	88
14.2 IOU S Bus Low ROM Controller (19-142F46)	89
14.3 IOU Bus ROM Controller (19-142F47)	90
15. EXTENDER BOARD OPERATION	91
16. MNEMONICS	92
16.1 CPA Mnemonics, Schematic Drawing 35-536D08	92
16.2 CPB Mnemonics, Schematic Drawing 35-537D08	95
16.3 CPC Mnemonics, Schematic Drawing 35-555D08	99
16.4 ALU Mnemonics, Schematic Drawing 35-538D08	100
16.5 IOU Mnemonics, Schematic Drawing 35-539D08	104

ILLUSTRATIONS

Figure 1. Model 8/32 Module Concept	3
Figure 2. Control and Module Instructions	5
Figure 3. CPU (CPA, CPB, and CPC) Block Diagram	15
Figure 4. CPA Block Diagram	18
Figure 5A. Instruction Read, RR or SF Formats	27
Figure 5B. Instruction Read RX1, RX2 or RI1 Formats	28
Figure 5C. Instruction Read RX3 or RI2 Formats	29
Figure 6. Second Halfword Clock Timing	30
Figure 7. Data Read Operation	31
Figure 8. Data Write Operations	32
Figure 9. CD Bus Interface	33
Figure 10. CPU State Diagram	36
Figure 11. Simplified Clock Circuit	37
Figure 12. Simplified Control Store Diagram	39
Figure 13A. Control Store Address Gating Low (CSA10:15)	40
Figure 13B. Control Store Address Gating High (CSA4:9)	40
Figure 14. CPC Block Diagram	42
Figure 15. A Stack Timing Diagram, 32-Bit Write	44
Figure 16A. SBUFF Timing, 32-Bit Write	44
Figure 16B. SBUFF Timing, 64-Bit Write	44
Figure 17. Stack Addressing Scheme	45
Figure 18. ALU Functional Block Diagram	47
Figure 19. ALU State Transitions	48
Figure 20. ALU Bus Timing – Immediate Response Functions (FSEL000)	49
Figure 21. IOU Block Diagram	63
Figure 22. Multiplexor Channel Timing	66
Figure 23. Multiplexor Channel (Input) Timing ADRS and SR/DR	67
Figure 24. Multiplexor Channel (Output) Timing ADRS and CMD/DA	68
Figure 25. Multiplexor Circuit Generation Description	70/71
Figure 26. Cycle Counter	72
Figure 27. Multiplexor Channel Timing, ACK	74
Figure 28. D Bus ROM Controller Data Gating for WD and WDA	77
Figure 29. ROM Controller Data Gating for RDH and RDHA	78
Figure 30. Serial ASCII Code U (Even Parity)	79
Figure 31. Teletype Controller Block Diagram	82
Figure 32. Write Mode (Output) Timing, Teletype	83
Figure 33. Read Mode (Input) Timing, Teletype	84

TABLE OF CONTENTS
(Continued)

TABLES

TABLE 1. FUNCTION CODES FOR CPU INSTRUCTIONS	6
TABLE 2. MC FIELD	8
TABLE 3. INTERRUPT TRAPS	9
TABLE 4. EXTERNAL INTERRUPT ENABLE	9
TABLE 5. REGISTER ADDRESSING	10
TABLE 6. MODULE 1 (FIXED POINT) AND MODULE 3 (FLOATING POINT) OPERATION	11
TABLE 7. I/O CONTROL FUNCTIONS	14
TABLE 8. STARTING LOCATION JUMPERS	21
TABLE 9. MEMORY ACCESS INTERRUPT SIGNALS	23
TABLE 10. MEMORY ACCESS INTERRUPTS	24
TABLE 11. MICRO COMMANDS	24
TABLE 12. INCREMENT MICRO-COMMANDS	34
TABLE 13. S BUS SELECT DECODING	35
TABLE 14. CONTROL STATE LOGIC IMPLEMENTATION	37
TABLE 15. ALU FUNCTION CODES	47
TABLE 16. STATE REGISTER LOGIC	49
TABLE 17. ALU ROM CONTROL	60
TABLE 18. FUNCTION MNEMONICS	62
TABLE 19. I/O MODULE FUNCTION GATING	76
TABLE 20. STB INSTRUCTIONS	76
TABLE 21. TELETYPE STATUS AND COMMAND BYTE	80
TABLE 22. MCR BIT ASSIGNMENT	86

MODEL 8/32 PROCESSOR

MAINTENANCE SPECIFICATION

1. INTRODUCTION

The INTERDATA Model 8/32 Digital System is a low cost, general purpose system, versatile enough to perform a wide range of industrial control, data processing, and scientific computation. The Model 8/32 is well suited to the real-time scanning of hundreds of instrument readings, process alarms, and pulse trains. It is particularly useful where larger amounts of main Processor time are needed for computation.

1.1 Packaging

The Model 8/32 Processor is contained in a 19 x 14 inch RETMA card file allowing 16 card positions. The basic Processor with 128KB of core memory uses 12 card positions and allows three positions for I/O expansion plus one for planned Processor options.

1.2 Processor

The basic Processor configuration is Model 8/32 with 128KB memory, Product Number M83-023. Other features such as parity, additional memory, etc. are optional.

The Model 8/32 uses a technique commonly referred to as "emulation" to implement the standard INTERDATA user repertoire. This technique requires a micro-processor, or sub-processor, not apparent to the user, employing one or more of its micro-instructions in sequence to implement one user level instruction. The basic micro-program is contained in 1,280 words of Read-Only-Memory (ROM).

The Model 8/32 employs a 32-bit micro-instruction word and 32 bit internal bussing. The basic instruction time of the micro-processor is 240 nanoseconds per micro-instruction.

1.3 Control Store

The basic Model 8/32 uses 1,280 words of control store which is mounted on the CPB board. The control store may be further expanded by the Writable Control Store (WCS). Three user instructions are used for manipulating the WCS.

1.4 Peripherals

The Model 8/32 interfaces to, and is compatible with, all standard INTERDATA peripheral controllers and controllers designed to the standard INTERDATA Multiplexor Bus. Any number of devices up to 1,024 can be accommodated, but a maximum of 16 can be interfaced directly to the Multiplexor Bus or to the Selector Channel Bus.

2. INTERNAL ARCHITECTURE

The architecture of the Model 8/32 encompasses a principle of modules communicating over a common bussing system, directed by instructions from a control memory which specify the module to which an instruction is directed and the function to be performed. In theory, the function of any module is arbitrary and the significance of various instructions take meaning only when applied to a specific module. Thus, a computer achieves a capability and personality determined by what functions can be performed by its complement of modules.

2.1 Modules

The Model 8/32 architecture accommodates eight modules which communicate over four Processor busses. The basic Processor is comprised of three modules.

1. Central Processing Unit (CPA, CPB, and CPC). The CPU contains the Processor registers. This module controls the user memory, control memory, register gating, and sequencing of instructions.
2. Arithmetic Logic Unit (ALU). The ALU provides the basic arithmetic/logical capability of the Processor.
3. Input/Output Unit (IOU). The IOU provides the I/O capability of the Processor by generating the standard INTERDATA Multiplexor (I/O) Bus for peripheral communications. It is capable of various byte manipulations of data presented on the busses. In addition, the I/O module contains the Display Console controller, the Teletypewriter controller, the Power monitor, and the System Initialize circuits.

The architecture accommodates five additional arbitrary modules such as floating point, Boolean manipulators, or special nature designs.

2.2 Micro-Instructions

The micro-instruction word is 32-bits long. In addition to the branch and write instructions, there are three types of instructions to the modules. These minimally encoded instructions provide 112 combinations of module/function commands. The micro-instruction can simultaneously direct two operands and a result independently on three of the computer's busses; generate 12-bit immediate field operands; select the address of the next micro-instruction; perform encoded micro control of the computer's functions such as reading/writing main memory; incrementing user location and memory address registers; controlling the user status register; and decoding the next user instruction.

2.3 Interrupts

The Model 8/32 has nine hardware priority interrupts, most of which can be masked by various bits of the Program Status Word (PSW). The occurrence of a recognized interrupt causes the micro-program to trap to one of nine specific control store locations associated with the interrupts.

2.4 Registers

The Model 8/32 can have up to 8 sets of 16 general registers, of which 15 in each set may be used as index registers. In addition, there are 16 floating point registers, 8 additional general purpose registers, plus 5 registers associated with the user level machine control that are available to the micro-programmer.

2.5 Processor Timing

Communications between modules are request/response. Timing is completely asynchronous (rather than quantized) to achieve maximum speeds. In addition, interlocks are provided between the control memories and the CPU to facilitate programming the micro machine. The control module operates on a 120 nanosecond clock, allowing a minimum instruction execution in 240 nanoseconds. Internal timing within the other modules can be selected to best suit the needs of the module.

3. FUNCTIONAL DESCRIPTION OF THE BASIC PROCESSOR

3.1 Processor Busses

The functional characteristics of the Processor can best be described in terms of its registers, busses and related gating. There are four busses which are the key to the modular design philosophy of the Model 8/32 architecture. Refer to Figure 1. An understanding of the bus structure is necessary to determine how each module of the Processor interrelates, and how the registers and gating of each module contribute to the function of the module it is designed to serve.

3.1.1 Control Bus. The Control Bus of the computer is commanded by the control module and is, in essence, a reflection of that segment of the micro-instruction selecting the function and module to be addressed, plus timing to effect data transfers. Also included is a means for a module to transfer data to the Condition Code of the PSW. The Control Bus signals are described in the following paragraphs:

Module Select Lines (MDSEL00:02). These three lines contain the address of a module for which the current micro-instruction is intended. One of the eight arbitrary modules can be selected by the instruction to perform some function. These three lines reflect Bits 0:2 of the micro-instruction.

Function Select Lines (FSEL00:03). These four lines reflect Bits 16:19 of the micro-instruction and normally select 1 of 16 arbitrary functions to be performed by the selected module.

Start (STRT). STRT signals the modules that data on the busses is valid. It is, in effect, a request from the control module for a response to a micro-instruction. Data is held static on the A and B Busses while STRT is active. The control module holds STRT active until it recognizes a response on the Module Finished (MFIN) line and has stored the results presented on the S Bus.

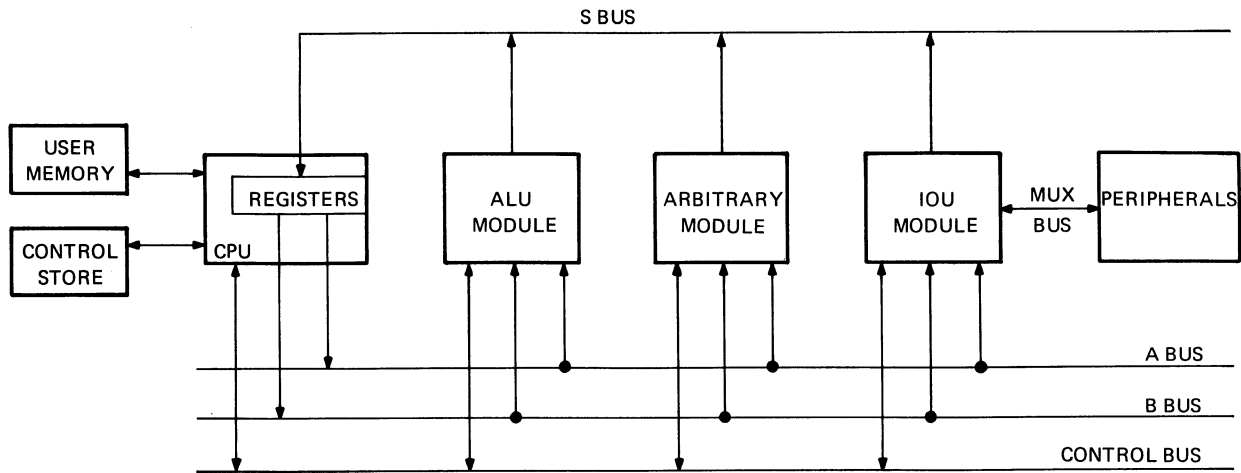


Figure 1. Model 8/32 Module Concept

Module Finished (MFIN). MFIN is a response to the control module from a selected module indicating that it has recognized STRT and completed the selected function. The selected module gates data and other responses onto the S Bus prior to returning MFIN. The data and responses must be held on the busses until the control module removes STRT. This time is indefinite and depends on events within the control module.

Module Signal (MSIG). This is a control signal manipulated by the selected module to indicate some arbitrary condition to the control module. It may be tested by the control module during a normal micro-instruction to the selected device, to control a conditional branch in the micro-program.

Condition Code Bus (SCC, VCC, CCC, GCC, and LCC). SCC signals the control module that the selected module wishes to manipulate the Condition Code of the Program Status Word. If the micro-programmer has enabled this manipulation, the Condition Code is forced to a status specified by the selected module. The status is unconditionally forced into the CPU flags. This is done concurrent with a normal instruction to the selected module. VCC, CCC, GCC, and LCC specify the status forced into CPU flags and the Condition Code of the Program Status Word, and represent overflow, carry, greater than, and less than, respectively.

3.1.2. **A, B, and S Busses.** The A, B, and S Busses are the primary data links between the control module and the selected module. Gating of data to/from each of these busses is controlled by the micro-instruction. Most of the registers of the control module can be gated to/from these busses.

Data is selected by the micro-instruction from two independent sources and transmitted to a selected module over the A and B Busses. The module is thus presented simultaneously with two operands. The resulting data is returned to the control module via the S Bus. The destination of the S Bus is selected by the micro-instruction.

3.1.3 **Typical Bus Exchange.** The use of the A, B, and S Busses can be summarized by the following example.

1. The micro-instruction selects a module (MDSEL00:02) and directs it to perform some function (FSEL00:03).
2. The operands are selected from somewhere in the control module and gated onto the A and B Busses.
3. The control module informs the selected module that all data on the busses is valid and that it may begin (STRT).
4. The selected module performs the function $(S) = (A) F (B)$ and gates the results to the S Bus.
5. The selected module may manipulate the Condition Code via SCC, VCC, GCC, LCC, and CCC.
6. The selected module may signal, for example, that the result of the operation is zero by activating MSIG.
7. The selected module activates MFIN to signal the CPU module that the operation is complete and the results are presented on the S Bus.
8. The control module recognizes MFIN, gates the S Bus to the destination specified by the micro-instruction, and then removes STRT.
9. The selected module deactivates itself when STRT is removed.

3.2 Registers

The following registers are part of the control module.

3.2.1 A Stack/B Stack. The A stack and B stack are redundant sets of register banks containing the general purpose registers of the CPU. The registers are duplicated to allow simultaneous gating of any register in the stack onto either the A Bus or the B Bus. These registers are gated onto the A and B Busses and are loaded from the S Bus under control of the micro-instruction.

3.2.2 Memory Data Register (MDR). This register provides the data buffer between the CPU and the user level memory. The MDR can be gated onto the B Bus and loaded from the S Bus under control of the micro-instruction. It is, of course, also loaded under control of the memory when a memory read cycle is requested. Hardware interlocks are employed to synchronize the memory to the CPU.

3.2.3 Memory Location Register (MLC). The MLC is a general purpose register which can be gated to the B Bus and loaded from the S Bus, and can be incremented by the length of the emulated instruction to facilitate the emulation of the user level repertoire. This register keeps track of the current instruction location of the emulated machine.

3.2.4 Memory Address Register (MAR). This register contains the address of the user memory that the micro-programmer is reading or writing. The MAR can be loaded from the S Bus under control of the micro-instruction, or incremented by four under micro-control. The least significant bit of the MAR is used to control byte steering for the byte-oriented instructions of the user repertoire (refer to I/O Section). As in the MDR, timing conflicts are resolved by hardware interlocks.

3.2.5 Program Status Word (PSW). The Program Status Word is an 18-bit register which may be gated onto the A Bus and loaded from the S Bus under control of the micro-instruction. Various bits of the PSW are used to enable associated hardware interrupts. PSW Bits 28:31 contain the Condition Code of the user level computer. These bits may be compared and tested against corresponding bits of the user instruction under Module 0 micro-instructions to emulate user branch instructions. In addition, they can be manipulated by any module designed to do so, if they are enabled by the micro-programmer.

3.2.6. User Destination Register, User Source Register (YD, YS). These two control registers store Bits 08:11 and 12:15 respectively, of the current user level instruction being emulated, and allow the micro-programmer to indirectly reference the general registers selected by the user instruction. The YD is compared to the PSW Condition Code on certain micro-instructions to emulate user level branches. These registers can be examined by gating them onto the A and B Busses under micro-instruction control. The YD can also be loaded from the S Bus.

3.2.7 User Instruction Register (UIR), Memory User Destination Register (UDR), and Memory User Source Register (USR). These three registers are loaded with Bits 0:7, 8:11, and 12:15, respectively, of the next user level instruction to be emulated. The 8-bit op-code stored in the UIR is used to vector to the emulation sequence for the next user instruction. It is also used to interrogate a ROM which has been configured to decode privileged and illegal user level instructions. The contents of the UDR and USR are transferred to the YD and YS at the beginning of the next emulation.

3.2.8 ROM Location Register (RLR). This register stores the current address of the control store instruction. It is loaded from the ROM Address Gates (RAG) at the beginning of every instruction except interrupt trap instructions and execute type instructions (explained in the section on micro-programming). The RLR is a 12-bit register allowing direct addressing of the control store up to 4K instructions.

3.2.9 ROM Instruction Register (RIR). This 32-bit register stores the current micro-instruction. The RIR is the focus of control of the CPU.

3.3 Interrupts

The hardware of the computer provides nine priority interrupts. Each interrupt has a unique control store trap location associated with it. Recognition of an interrupt causes the micro-instruction stored at its respective trap location to be performed. The RLR contents are preserved to allow the address of the interrupted sequence to be saved, if desired, so that control can be returned at the completion of the interrupt routine. Certain interrupts are enabled/disabled by bits of the PSW.

3.4 Control Store Memory

The Model 8/32 can accommodate a maximum of 4K x 32 bits of control store memory. The computer allows data as well as instructions to be retrieved from its control memory. This capability expands its versatility by allowing data such as sine tables, translation tables, and matrices to be stored and operated upon efficiently by the micro-programmer.

On models so equipped, the Processor can alter its control store (write into its memory). This capability to store and retrieve data provides the power of a hardware computer at micro-instruction speeds.

3.5 Micro-programming

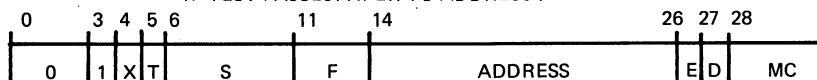
The control store of the Model 8/32 is a 32-bit word memory which may be read indirectly by an instruction to retrieve data, and may be written into by an instruction if it is a writable memory. The basic Model 8/32 contains a 1,280 x 32 ROM array containing the user repertoire and support programs.

The basic instruction format provides the computer with a three address capability, but various options of the repertoire can modify this to range from two to four. Figure 2 displays the different types of instructions and their modifiers.

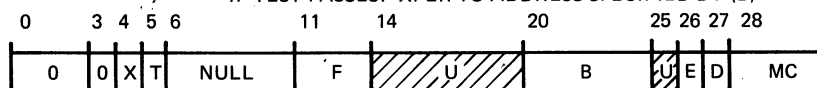
The format of the micro-instruction specifies which module is to be addressed, allowing only one module of the computer to be addressed at any one time. All other modules must ignore the communications in process. Bits 0:2 of the instruction selects the module to which the instruction is addressed.

CONTROL INSTRUCTIONS

ADDRESS LINK: FLAGS ARE TESTED AS PER F AND T, RLC → (S)
IF TEST PASSES: XFER TO ADDRESS.

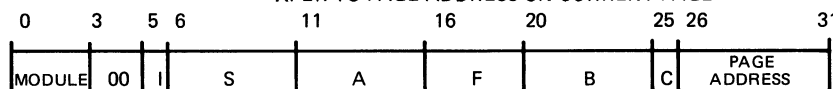


REGISTER BRANCH: FLAGS ARE TESTED AS PER F AND T
IF TEST PASSES: XFER TO ADDRESS SPECIFIED BY (B)

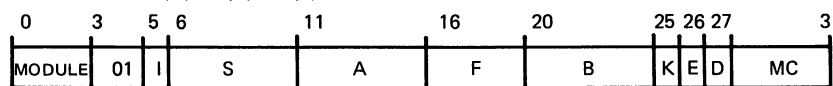


MODULE INSTRUCTIONS

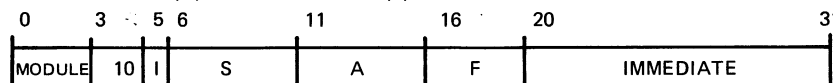
RR XFER (A) F (B) → (S) IF C = 1 AND MSIG = 1 XFER TO NEXT INSTRUCTION OTHERWISE
XFER TO PAGE ADDRESS ON CURRENT PAGE



RR CONTROL (A) F (B) → (S)

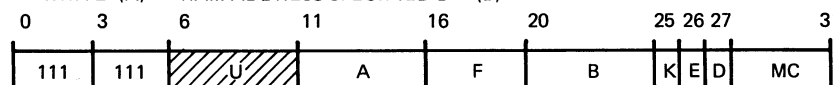


RI IMMEDIATE (A) F IMMEDIATE → (S)



WRITE INSTRUCTION

R WRITE (A) → RAM ADDRESS SPECIFIED BY (B)



SHOULD BE
NULL SELECTED

- A SELECTS REGISTER GATE TO A BUS
- B SELECTS REGISTER GATED TO B BUS
- S SELECTS REGISTER TO RECEIVE S BUS
- F SELECTS FUNCTION OF ADDRESSED MODULE
- E ENABLE SETTING OF CONDITION CODE
- C IF SET TRANSFER IS CONDITIONAL
- I B FIELD IS INDIRECT ADDRESS OF DATA
- D DECODE NEXT INSTRUCTION
- K FSEL EXTENSION
- U UNUSED
- T TESTED F FIELD FOR THE "TRUE" CONDITION
- X EXECUTE

- MC FIELD DESIGNATIONS (MEMORY CONTROL)
- 0000 NO MEMORY ACTION
 - 0001 INCREMENT MLC BY INSTRUC. LENGTH
 - 0010 PRIVILEGED WRITE HALFWORD
 - 0011 DATA WRITE HALFWORD
 - 0100 NOT USED
 - 0101 INCREMENT MAR BY 4; WRITE DATA F W
 - 0110 PRIVILEGED WRITE FULLWORD
 - 0111 DATA WRITE FULLWORD
 - 1000 READ HALFWORD AND SET BIT
 - 1001 INCR. MLR BY INSTR. LENGTH; READ INSTR.
 - 1010 PRIVILEGED READ H W
 - 1011 DATA READ HALFWORD
 - 1100 INSTRUCTION READ
 - 1101 INCREMENT MAR BY 4, READ DATA F W
 - 1110 PRIVILEGED READ F W
 - 1111 DATA READ FULLWORD

Figure 2. Control and Module Instructions

3.5.1 Module 0. Module 0 addresses the control module. As shown in Figure 2, instructions are interpreted differently for Module 0 than the others. In the normal sequence of instructions (e.g., no branches), the hardware of the control module controls the reading of its memories, and gates the registers specified by the instruction. When it is addressed by an instruction, it is for the purpose of a conditional transfer. Module 0 does not manipulate the Condition Code or Processor flag register.

Branch/Execute Instructions. There are two types of transfers recognized by Module 0. The most common is the Branch. The Branch (BR) instruction conditionally transfers control of the CPU to a specified address of control memory and proceeds sequentially from there. The second type of transfer, commonly called an execute, transfers control to a single instruction at a specified address of control memory, then normally returns to the original sequence. Any type of instruction may be executed including additional execute instructions to any depth. However, an execute which results in a branch does not return to the continuing sequence. Bit 04 of the instruction determines whether the instruction is a branch or execute type.

Address Link/Register Return. There are two type of Module 0 instructions; address link and register return. They are selected by the state of Bit 03 of the instruction.

The linked transfer is similar in function to the user level Branch and Link (BAL) instruction, and can be used to transfer to subroutines when they may be entered from more than one location. The location of the next sequential instruction, following the transfer, is deposited in the register specified by the Link field of the instruction (Bits 11:15), and a transfer is conditionally executed to the effective address.

When the address link is selected, the transfer address is specified by Bits 14:25 of the instruction.

The register return is used when the transfer address is contained in a register. In this instruction, a branch is taken to the location contained in the register specified by Bits 20:24.

Conditional Branches. All transfers are conditional upon a state selected by the F field and T field of the instruction. By selective coding of the F field, either the Condition Code of the user level machine or the status of the CPU can be tested. The codes are shown in Table 1.

TABLE 1. FUNCTION CODES FOR CPU INSTRUCTIONS

X	T	F	MNEMONICS	OPERATION
0	0	110	BAL	Branch and Link Unconditional
0	0	111	BALA	Branch and Link and Arm Interrupts
0	1	111	BALD	Branch and Link and Disarm Interrupts
0	0	000	BALZ	Branch and Link on CPU Zero
0	1	000	BALNZ	Branch and Link on Not CPU Zero
0	0	001	BALL	Branch and Link on CPU Less
0	1	001	BALNL	Branch and Link on CPU not Less
0	0	010	BALG	Branch and Link on CPU Greater
0	1	010	BALNG	Branch and Link on CPU not Greater
0	0	101	BALV	Branch and Link on CPU Overflow
0	1	101	BALNV	Branch and Link on No CPU Overflow
0	0	100	BALC	Branch and Link on CPU Carry
0	1	100	BALNC	Branch and Link on no CPU Carry
0	0	011	BALF	Branch and Link if the logical product of user M1 field and User's CC is Zero
0	1	011	BALTF	Branch and Link if the logical product of user M1 field and User's CC is not Zero
0	1	110	BDC	Branch & Mask Console interrupt (no real branch is performed)

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

3.5.2 Non-CPU Instructions. As stated previously, when the module number is not zero, the CPU does not operate on the instruction, and the fields are interpreted differently. The module field (Bits 0:2) and the F field (Bits 16:19) are interrogated individually by the other modules. There are four types of non-CPU instructions selected by Bits 3 and 4 of the instruction. They are:

1. RRX. The RRX is a Register-to-Register and Transfer instruction. It is effectively a four-address instruction in that it gives the register address of the two operands, the register address for the results, and the location for the next sequential instruction.

The two operands are addressed by the A field (Bits 11:15) and the B field (Bits 20:24). The contents of these two registers are gated, respectively, to the A Bus and B Bus of the computer.

The S field (Bits 6:10) selects the destination register to which the results are gated from the S Bus.

The page address field (Bits 26:31) selects the low order address of the next instruction. The high order bits are taken from the current location address. The C field (Bit 25) being true makes the transfer conditional upon a signal returned by the addressed module at the completion of the instruction. (The ALU, for example, returns the Carry flag as its signal.) If the module signal, which is designated MSIG, is true, and Bit 25 of the micro-instruction is true, the branch does not occur, and the next sequential instruction is executed. Any other condition causes the transfer to be effected.

2. RRC. The RRC is a Register-to-Register Control type instruction. The interpretation of the instruction fields is identical to that of the RRX, with the exception of Bits 25:31 which contained the page address within a RRX instruction. Bits 25:31 of the RRC instruction provide the micro-control of the CPU and are described in Section 3.5.3.
3. RIM. The RIM instruction provides an immediate field for ease of generating constants and bit masks. Immediate, is the term generally used to infer that the immediate contents is the actual operand rather than the address where the operand is found. This 12-bit immediate field (RIR 20:31) is converted to a 16-bit operand by extending the sign bit (RIR 20) when gating onto the B Bus. The S field and A field of the instruction are interpreted identically to that of the RRX and RRC instructions.
4. RWT. The RWT is the Store or Write instruction of the repertoire if the CPU is equipped with a writable control store. There are several notable differences pertaining to this instruction.

-Although the module number cannot be zero, it may be any other, as the CPU never communicates with the other modules.

- The S field is not interpreted and should be null selected.

- The F field is not interpreted.

- The B field addresses the register containing the address to be written into.

- The A field addresses the register containing the data to be stored in control store.

Bits 25:31 of the RWT instruction are interpreted as a control field, as in a RRC instruction.

3.5.3 Micro-Control (MC). To facilitate the emulation task of the CPU, certain instructions allow an order of micro-control within the CPU. The instructions possessing this capability are the Module 0 (RRC and RWT) instructions.

MC Field. The MC field is the user memory micro-control which allows various controls over the user memory instruction Location Counter (MLC), the user Memory Address Register (MAR), and the reading and writing of the user memory. The significance of the bits of the MC field are shown in Table 2.

There are certain hardware connotations to the MC operations which are not made apparent by Table 2. They are:

1. The micro-control specified by the MC field is conditional when used within Module 0 instructions. The read memory is only effected if the operation does not result in a transfer. (This conditioning is used to expedite the emulation of the user branch instruction.)
2. All of the micro-control is effected before the STRT occurs with the exception of data read and data write. This control is effected after completion of the instruction, which allows the micro-programmer to use the MAR or MDR as a destination and begin a read/write data immediately. It also allows the execution of the increment and the addressing of the MAR as the destination register simultaneously, which has functional utility.

D Field (Decode Instruction). The D field bit informs the CPU to halt the sequential flow of micro-instructions and begin to emulate the next user instruction. The Operation-Code (op-code) field of the new user instruction is in the UIR and provides a vector to a control store address where the emulation sequence begins. This implies that the micro-programmer must have done an instruction read in the current or a prior instruction using one of the proper MC field designations. The execution of a decode is conditional when used within Module 0 instructions, and, like the instruction fetch, is only performed if the operation does not result in a transfer.

E Field. This field is used to Enable (E) or disable changing of the Condition Code (CC) of the PSW. When changing is enabled, the Condition Code is changed under control of the module addressed until again disabled by this field. (The ALU, for example, jams its C, V, G and L flags into the Condition Code upon completion of its function.) The meaning of the Condition Code is a function of the module addressed. Flags are disabled at the beginning of an emulation sequence.

K Field. The K field of the micro-instruction is an extension of the F field of the instruction. It is available only on the RRC and RWT instructions and constitutes the Control Signal (K SIG) to the modules. Its meaning, just as the F field, is defined by the module addressed by the current RRC instruction. The ALU, for example, reinterprets shifts to be halfword when K SIG is active. It is also used to extend the functions of the I/O module.

TABLE 2. MC FIELD

BITS				MEANING
28	29	30	31	
0	0	0	0	No Action
0	0	0	1	IL Increment LOC by Instruction Length
0	0	1	0	PW2 Privileged Write Halfword (two bytes)
0	0	1	1	DW2 Data Write Halfword
0	1	0	0	No Action
0	1	0	1	I4DW4 Increment MAR by 4, Data Write Fullword
0	1	1	0	PW4 Privileged Write Fullword
0	1	1	1	DW4 Data Write Fullword
1	0	0	0	RAS Read Halfword and Set Sign Bit
1	0	0	1	ILIR Increment LOC by Length and Read Instruction
1	0	1	0	PR2 Privileged Read Halfword
1	0	1	1	DR2 Data Read Halfword
1	1	0	0	IR Instruction Read
1	1	0	1	I4DR4 Increment MAR by 4, Data Read Fullword
1	1	1	0	PR4 Privileged Read Fullword
1	1	1	1	DR4 Data Read Fullword

- IL The Location Counter (LOC) is incremented by the length in bytes of the last user level instruction fetched.
- PW2 The Memory Access Controller (MAC) is disabled and the halfword in MDR (Bits 16:31) is written into the addressed location.
- DW2 The halfword in MDR (Bits 16:31) is written into the addressed location. MAC is not disabled.
- I4DW4 The Memory Address Register (MAR) is incremented by four, then the fullword in MDR (Bits 0:31) is written into the location addressed by MAR.
- PW4 The MAC is disabled and the fullword in MDR (Bits 0:31) is written into the addressed location.
- DW4 The fullword in MDR (Bits 0:31) is written into the addressed location.
- RAS The halfword at the addressed location is read then re-written with Bit 0 of the halfword set. The original value of the halfword replaces MDR Bits 16:31. Bits 0:15 of the MDR are set equal to Bit 16 of MDR (sign extension).
- ILIR LOC is incremented by the length in bytes of the last user instruction fetched, then an Instruction Read is started from the address specified by the new value of LOC.
- PR2 The MAC is disabled and the halfword at the addressed location is read and copied to MDR Bits 16:31. Bits 0:15 of MDR are set equal to MDR Bit 16.
- DR2 The halfword at the addressed location is read and copied to MDR Bits 16:31. Bits 0:15 of MDR are set equal to MDR Bit 16.
- IR An Instruction Read is started from the memory address specified by LOC.
- I4DR4 MAR is incremented by four, then the fullword at the location addressed by the new value of MAR is read and copied to MDR.
- PR4 MAC is disabled, then the fullword at the location addressed by MAR is read and copied to MDR.
- DR4 The fullword at the location addressed by MAR is read and copied to MDR.

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

3.5.4 Control Store Data Storage. Data may be retrieved from the ROM or the Dynamic Control Store (DCS) during execution of RRC, RRX, or RIM instructions when the I field bit (instruction Bit 5) is true. When the I field bit is set, the data addressed onto the B Bus is used as the store address of the ROM or DCS, and causes the CPU to replace this data with the addressed data before signaling the addressed module to begin its task.

3.5.5 Interrupts. The hardware of the computer provides nine priority interrupts. Each interrupt has a unique trap location associated with it. Recognition of an interrupt causes the instruction stored at its respective trap location to be performed. The RLR contents are preserved to allow the address of the interrupted sequence to be saved, if desired, so that control can be returned at the completion of the interrupt routine. Certain interrupts can be disabled by bits of the PSW as designated in Register Peculiarities and Tables 3 and 4. In addition, all interrupts can be enabled/disabled as a group by a micro-instruction. All interrupts not masked by PSW bits are interrogated when a new user level instruction is decoded, regardless of the status of the group enable. The group enable is automatically disabled at the beginning of a user emulation, and must be enabled by instruction if the programmer wishes to recognize interrupts. Tables 3 and 4 list by priority the pertinent information for each interrupt.

TABLE 3. INTERRUPT TRAPS

INTERRUPT	TRAP ADRS (HEX)	MASK	GROUP ENABLE
Memory Access Controller (Instruction)	1FE	PSW21	NO
Memory Access Controller (Data)	207		
Primary Power Fail	206	NONE	YES
Machine Malfunction	205	PSW18	YES
Display Panel	204	NONE	YES
External Interrupt Level 0	203	See Table 3	YES
External Interrupt Level 1	202		
External Interrupt Level 2	201		
External Interrupt Level 3	200		
Illegal Instruction	208	NONE	N/A
Privileged Instruction	208	PSW23	N/A

PSW Bits 17 and 20 define the external Interrupt enable status of the Processor as shown below:

PSW	BITS
17	20
0	0
0	1
1	0
1	1

All Levels Disabled
 Higher Levels Enabled
 All Levels Enabled
 Current and Higher Levels Enabled

where the current level is a function of the currently active register set. See Table 4.

TABLE 4. EXTERNAL INTERRUPT ENABLE

PSW BITS					EXTERNAL INTERRUPT ENABLED			
17	20	25	26	27	LEVEL 0	LEVEL 1	LEVEL 2	LEVEL 3
0	0	X	X	X	NO	NO	NO	NO
0	1	0	0	0	NO	NO	NO	NO
0	1	0	0	1	YES	NO	NO	NO
0	1	0	1	0	YES	YES	NO	NO
0	1	0	1	1	YES	YES	YES	NO
0	1	1	0	0	YES	YES	YES	NO
0	1	1	0	1	YES	YES	YES	NO
0	1	1	1	0	YES	YES	YES	NO
0	1	1	1	1	YES	YES	YES	NO
1	0	X	X	X	YES	YES	YES	YES
1	1	0	0	0	YES	NO	NO	NO
1	1	0	0	1	YES	YES	NO	NO
1	1	0	1	0	YES	YES	YES	NO
1	1	0	1	1	YES	YES	YES	YES
1	1	1	0	0	YES	YES	YES	YES
1	1	1	0	1	YES	YES	YES	YES
1	1	1	1	0	YES	YES	YES	YES
1	1	1	1	1	YES	YES	YES	YES

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

3.5.6 Registers. The basic CPU has up to 8 sets of general registers each containing 16 user registers, 16 floating registers, and 8 general purpose registers for use by the micro-programmer. In addition, the bulk of the remaining CPU registers is also available to the micro-programmer.

A register is available to the micro-programmer if he can address it to one or more of the internal busses. Table 5 tabulates the addressable registers and their respective address on the designated bus. Also listed are register mnemonics, descriptions, and the register peculiarities.

TABLE 5. REGISTER ADDRESSING

BUS ADDRESS (HEX)	S BUS	B BUS	A BUS
00:0F (16 General Registers)	UR _n	UR _n	UR _n
10:17	MR _n	MR _n	MR _n
18	YS	YS	YS
19	YD	YD	YD
1A	MLC	MLC	YX
1B	MDR	MDR	YDP1
1C	MAR	EFFECTIVE ADDRESS	
1D	PSW	YSI YD	PSW
1E	YDD	YDI YSI	
1F	NULL	NULL	NULL

Register Mnemonics and Descriptions.

<u>MNEMONIC</u>	<u>REGISTER</u>	<u>COMMENT</u>
UR	User General Registers	16 registers manipulated by emulated language
MR	Micro-level General Registers	8 additional GP registers available to the micro-program
PSW	Program Status Word	16 bit register containing interrupt enables and flags
MDR	Memory Data Register	
MLC	Memory Location Counter	Location Counter of emulated program
MAR	Memory Address Register	
NULL	No register selected	Gates 0 to A and B Busses, S Bus data is lost
YS	User Source Register	Register selected by Bits 12:15 of emulated instruction (contents of USR)
YSI	User Source Register Immediate	Bits 12:15 of the emulated instruction (USR) gated onto B Bus
YX	User Index Register	Same as YS except NULL gated to A Bus if field is 0 (contents of USR=0)
YD	User Destination Register	Register selected by Bits 8:11 of emulated instruction (contents of UDR)
YDI	User Destination Register Immediate	Bits 8:11 of the emulated instruction (UDR) gated onto the A Bus
YDP1	User Destination Register Plus 1	Register selected by Bit 8:11 of emulated instruction +1 (must be odd)
YDD	User Destination Register Direct	S Bus 12:15 replaces UDR contents

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

Register Peculiarities. The last four bits of the PSW contain the Condition Code of the emulated computer. In general, these bits can be manipulated by any addressed module unless the PSW is the S Bus destination or their change has been inhibited by the micro-instruction. The individual bits of the PSW which have hardware implications are:

PSW 17, 20, 25:27	ATN interrupt enable and level selection
PSW 18	Machine Malfunction enable
PSW 23	Privilege instruction/Memory Protect enable
PSW 25:27	User register set selection
PSW 28	C flag of Condition Code
PSW 29	V flag of Condition Code
PSW 30	G flag of Condition Code
PSW 31	L flag of Condition Code

The following additional registers have the indicated capabilities and connotations.

1. The MDR receives data asynchronously from memory. It is used in the address calculation for RX3 instructions.
2. The MDR, MAR, and MLC being addressed cause the Processor to interlock with memory when they are the source or destinations of the current instruction and the Processor is requesting memory service.
3. The MAR and MLC can be incremented by the micro-control.
4. The MLC is used in the address calculation for RX2 instructions.

3.5.7 CPU Flags. The CPU contains a flag register which is independent of the PSW flags and is manipulated by any module which attempts to affect the PSW Condition Code by activating the SCC control line of the CPU Control Bus. When the SCC control line is active, the state of the VCC, CCC, GCC and LCC are unconditionally jammed into the CPU flag register and conditionally into the PSW Condition Code. The changing of the PSW is controlled by the micro-programmer by the E field of the micro-instruction. The state of the CPU flags can be individually tested by the Module 0 instructions.

3.5.8 Arithmetic Module (ALU/FALU) Programming. The Arithmetic Logic Unit (ALU) is a standard module in the Model 8/32 hardware. It is addressed as Module 1 in the module field of the instruction for fixed point operations, as Module 3 for floating point operations, and is capable of 27 functions. Refer to Table 6. Communication with the ALU is asynchronous. By design, the ALU is never busy and for the majority of ALU functions, response is within 120 nanoseconds. (This allows most ALU referenced instructions to be completed in 240 nanoseconds.)

TABLE 6. MODULE 1 (FIXED POINT) AND MODULE 3 (FLOATING POINT) OPERATION

F FIELD	MODULE 1 (FIXED POINT)	MODULE 3 (FLOATING POINT)
0 0 0 0	Subtract	Not used
0 0 0 1	Add	Load
0 0 1 0	Subtract with Carry	Subtract With Carry
0 0 1 1	Add With Carry	Add With Carry
0 1 0 0	Not used	Not used
0 1 0 1	Logical AND	Compare
0 1 1 0	Logical Exclusive OR	Not used
0 1 1 1	Logical OR	Not used
1 0 0 0	*Logical Shift Right	Subtract
1 0 0 1	*Logical Shift Left	Add
1 0 1 0	Rotate Right	Not used
1 0 1 1	Rotate Left	Not used
1 1 0 0	*Arithmetic Shift Right	Compare and Equalize
1 1 0 1	*Arithmetic Shift Left	Not used
1 1 1 0	Signed Multiply	Multiply
1 1 1 1	Signed Divide	Divide

* When used in conjunction with the K bit of the RRC instruction, shifts are halfword (16 bits).

For functions which require more than one ALU cycle (i.e., shifts, rotates, and multiply/divide), the ALU does not respond with a finish signal until the completed results are on the S Bus.

Multiply/divide can be performed only on the 24 general registers and must address the same register pair on both the A and S Busses. The same restrictions that apply to these operations at the user level must be adhered to at the micro level.

A user emulated multiply/divide instruction is micro-coded by selecting the ALU (Module 1); addressing the UDR on the S Bus, the UDRP1 onto the A Bus, the USR onto the B Bus, and the required function code for the operation. When the ALU signals its completion, the results have already been deposited in the UDR.

To implement a shift or rotate instruction, the register to be manipulated is addressed onto the A Bus, the shift count is put onto the B Bus (27:31), and the S Bus is gated to the destination register.

The ALU generates valid CPU flags for all instructions except multiply/divide. The C flag is gated as MSIG.

3.5.9 I/O Module Programming. The I/O module performs a multiplicity of functions. In general, it is addressed to communicate with the multiplexor channel. It has the additional capability of performing byte manipulations for the CPU both in conjunction with an I/O exchange and without one. Furthermore, the I/O module contains the Machine Control Register (MCR) which stores machine trouble conditions and interrupts the CPU. The contents of the MCR can be sensed, tested and cleared. Module Number 2 has been assigned to the I/O module.

Multiplexor Channel. The Multiplexor Channel, generated on the I/O module, is operationally identical to the standard INTERDATA Multiplexor Bus in all respects. The Multiplexor Bus is a byte or halfword-oriented I/O system which communicates with up to 255 peripheral devices.

A single instruction from the CPU contains the device address, the encoded function, and up to 16 bits of output data when needed. The Multiplexor Bus generator provides single or multi-cycle operation to address the device, transmit the decoded function, send or receive over 16 bi-directional data lines, and synchronize the exchange.

The normal byte or halfword operation consists of an address cycle and a data cycle. However, during a Read/Write block sequence, the address cycle is not used. For halfword functions (RDH/WDH) with a byte oriented device controller, two data cycles are used to transfer the halfword.

Byte Manipulation. The I/O module has the capability of performing byte manipulation both in conjunction with an I/O operation and without one. The byte steering is under control of the least significant memory address bit in the MAR and also the KSIG line. For halfword operations, this manipulation is inoperative but the double data cycle with packing/unpacking results when the Halfword (HW) Test line is inactive.

I/O Module Function Codes. The encoded I/O module functions and the byte manipulations are described in Section 3.7.3.

Machine Control Register (MCR). The Machine Control Register (MCR) consists of four flip-flops, four straps, and the Console Attention (CATN) and SNGL leads from the Display controller. MCR bit assignments are:

<u>BIT</u>	<u>MNEMONIC</u>	<u>MEANING</u>
15	EPF	Early Primary Power Fail
14	IRLMP	Instruction Read Local Memory Parity Fail
13	DMPF	Data Memory Parity Fail
12	IA/STF	Illegal Address and/or Start Timer Fail
11	STF	Start Timer Fail
10	CATN	Console Attention flip-flop (from Display Controller)
09	RSET	Register Sets Available
08	spare	(strap)
07	SNGL	SNGL flip-flop (from Display Controller)
05	BNKB	Bank B (strap)
04	BNKA	Bank A (strap)

The IRLMP and DMPF flip-flops store signals received from the Memory Bank Controller (MBC). Signals to get the EPF and STF bits are generated on the IOU board by the Power Monitor and Start Timer circuits. The composite bit (MCR12) can be strapped to represent IA, STF, or both. MCR11 is always set by STF. The testable straps are wired for logical ONE or ZERO as required. A Machine Malfunction (MMF) interrupt is generated when any of Bits 12, 13, 14 or 15 are true. The SMCR function gates MCR12:15 to the CC Bus, MCR08:15 to S08:15, and MCR04:05 to S04:05. The CMCR operation clears MCR11:15 where there are ONES in B11:15. This permits selective clearing of some bits while the rest of the MCR continues to monitor other machine functions without loss of data.

Start Timer (STRT). A 35 microsecond timer is started by the STRT signal and cleared by the MFIN signal from any selected module. Should the timer run out before the MFIN signal arrives, a malfunction exists; i.e., non-existent module, circuit trouble, or no SYN return from the Multiplexor Channel. If a D Bus function was called for, the False SYNC code (0100/CVGL) is placed on the CC Bus and a pseudo MFIN signal is sent to restart the CPU clock. Also, S16:31 is forced to X'0004' if the function required is of the Sense Status type. For a non-D Bus operation, the Start Time Fail (STF) bit is set in the Machine Control Register (MCR11), and a pseudo MFIN signal is sent to restart the CPU clock. If STF is strapped into MCR12, the MMF interrupt is also generated.

3.6 Processor Block Diagram Analysis

3.6.1 **CPU (Figure 3).** The architecture of the Processor is structured about four busses which provide inter-communication between the CPU and the remaining modules.

The CPU selects the module via the Control Bus (C Bus), specifies the function, and signals that data is available on the A and B Busses. The addressed module signals when it has completed its function and transmits flag type data back over this bus.

The A Bus and B Bus contain the two operands offered simultaneously to the addressed module to be manipulated. Most of the CPU registers can be gated onto the A and B Busses.

Data from an addressed module is returned to the Processor via the S Bus.

The ROM Instruction Register (RIR) is a 32-bit register that latches the current instruction read from the control store memory to provide the control gating for each instruction. Data can also be gated from the RIR onto the B Bus for indirect data fetches. One of the many functions performed by the encoded instruction is register gating. This is performed by the select logic to encode the A, B, and S SEL lines, these lines determine which registers are gated onto the A and B Busses, and which register is the S Bus destination.

The Control Store Address gates (CSA) select the address of the control store to be interrogated next. Inputs to the CSA may be the ROM Location Counter (RLC) to select the next instruction, certain bits of the RIR for branching, the B Bus for indirect addressing or branches, the translated vector for the next user instruction to be emulated (XLTR), or interrupt trap address from the interrupt logic.

The ROM Location Register (RLR) preserves the address of the current instruction. It is loaded with the address of the current instruction every time the RIR is loaded, except when the instruction is an interrupt or execute type which do not alter the ROM Location Register. The RLR is gated to the RLC to perform RLR+1 for the next sequential instruction.

The general user registers and the 8 general micro-registers are contained in the A stack and B stack. This is a redundant pair of registers, implemented as such to allow gating of any two registers of the machine simultaneously onto both the A Bus and B Bus. Gating of these stacks is controlled by the A, B, and S SEL lines, just as the other A, B, and S source and destination registers.

The Program Status Word register (PSW) is an architectural feature of the user level machine. Certain bits of the PSW are used to mask interrupts, control the privileged mode of the Processor, and to contain the Condition Code of the user level program. This register can be gated to the A and from the S Busses. The Condition Code portion can also be loaded independently of the register addressing.

The Memory Location Counter (MLC) aids the emulation capability of the Processor. It is used to contain the memory location of the current user level instruction. In addition to being capable of being gated to/from the B and S Busses, this register may be incremented by the length of the last emulated user instruction under micro-control of the programmer.

The Memory Address Register (MAR) contains part of the address to be used by the main memory for a read/write operation. This register can be loaded from the S Bus, and may also be incremented by micro-control of the programmer. The manipulation of this register is interlocked with the memory operation by hardware to remove timing restraints from the micro-programmer.

The Memory Data Register (MDR) is loaded from the S Bus with data to be written into the main memory, or it is loaded from the Memory Data Bus with the contents of the addressed memory on a memory read operation. This register may be gated onto the B Bus of the Processor, and, like the MAR, hardware interlocks remove the timing considerations of the memory system from the micro-programmer when this register is referenced.

The User Instruction Register (UIR) stores the 8-bit op-code of the next user instruction to be emulated. It is loaded from memory when a Read operation is designated as an instruction fetch by the micro-code. The 8-bit op-code is translated to a vector which designates the beginning address of the emulation sequence for a particular instruction. The 8-bit op-code is also gated to a Privileged/Illegal ROM which is coded to detect these types of instructions and cause an interrupt to the Processor.

The memory User Destination Register (UDR) and memory User Source Register (USR) store the destination and source fields of the next instruction to be emulated when it is read from memory. This data is transferred to the User Destination Register (YDR) and User Source Register (YSR) at the beginning of a new emulation sequence to provide residual control for the instruction.

Memory Control is effected by the micro-control field of the micro-instruction. The ability is provided to cause data read, data write, and instruction read.

3.6.2 ALU. The ALU is a standard module of the Model 8/32 and provides the basic arithmetic/logic capability. It communicates with the CPU over the A, B, S, and C Busses in a manner identical to other modules.

The ALU becomes active when it recognizes its address on the Control Bus (Module Number 1 or 3), and the CPU signals start (STRT). The function to be implemented is determined from the Control Bus.

ALU functions may be of two types. The simple functions (add/subtract and logical) cause the ALU to immediately return a Module Finished signal (MFIN). For these type of instructions, the A and B Busses are gated through the ALU, and the required function is performed and gated onto the S Bus.

For the complex type functions (multiply/divide, shift, and floating point) the ALU clock is enabled and a hardware sequence is entered to perform the required operation. The shift gates are used to shift the A Bus or the Sum Bus right or left back into the A latch and onto the A Bus again as determined by the ALU algorithms. In the case of fixed point multiply/divide the ALU stores half of the completed results before signaling the CPU with MFIN. The other half is dumped onto the A Bus from the MQ register and gated through the ALU onto the S Bus when MFIN is activated.

3.6.3 IOU Board. (Refer to the Block Diagram in IOU section.) The IOU board contains the I/O Control, the Display controller, the TTY controller, the Machine Control Register (MCR), the Power Monitor, Initialize circuits and the Start Timer.

The Display and TTY controllers have access to the CPU via the Multiplexor Channel D Bus and the I/O Control in the same manner as other peripheral device controllers. The Display controller provides a visual display of the contents of all system registers and any main memory location, together with the capability of manually entering data and programs. It shares D Bus drivers/receivers with the TTY controller and signals the CPU directly with the Display (DSPLY) interrupt. The Console Attention (CATN) signal appears as Bit 10 of the MCR.

The TTY controller, which supports the Model 33/35 Teletype, provides serial/parallel conversion and all standard TTY control features. It contains a full character buffer in the receive mode to permit a program service interval of one character time (100 milliseconds). The detailed descriptions of the Display and TTY controllers are covered in Sections 12 and 13 respectively.

I/O Control. The I/O Control performs a multiplicity of functions. The main function is to generate Multiplexor Channel D Bus from the CPU busses whenever it is addressed by Module Number 2 and the proper function selections are made. The control also performs byte manipulation for the CPU both in conjunction with an I/O operation and without. Common function decoders also generate signals to sense/clear the MCR (which stores Machine Malfunction conditions), to set the system Stop flip-flop with a Power Down/Initialize function (FPOW), and to gate 4-bits of the B Bus (12:15) to the front terminal strip of the chassis for external signaling purposes; e.g., multi-CPU operations.

Four function select lines together with the KSIG line pick 1 of 32 possible functions as shown on Table 7.

TABLE 7. I/O CONTROL FUNCTIONS

	FSEL0X				FUNCTION	
	0	1	2	3	KSIG=0	KSIG=1
0	0	0	0	0	RD	RDR
1	0	0	0	1	WD	WDR
2	0	0	1	0	SS	SSR
3	0	0	1	1	OC	OCR
4	0	1	0	0	RDH	*STBR
5	0	1	0	1	WDH	*LBR
6	0	1	1	0	ACK	*LDWAIT
7	0	1	1	1	*SMCR	*CMCR
8	1	0	0	0	RDA	RDR
9	1	0	0	1	WDA	WDR
A	1	0	1	0	SSA	SSR
B	1	0	1	1	OCA	OCR
C	1	1	0	0	RDHA	*STB
D	1	1	0	1	WDHA	*LB
E	1	1	1	0	*THW	*EXB
F	1	1	1	1	*POW	*POUT

* Functions that do not require operation of the Multiplexor Channel D Bus.

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

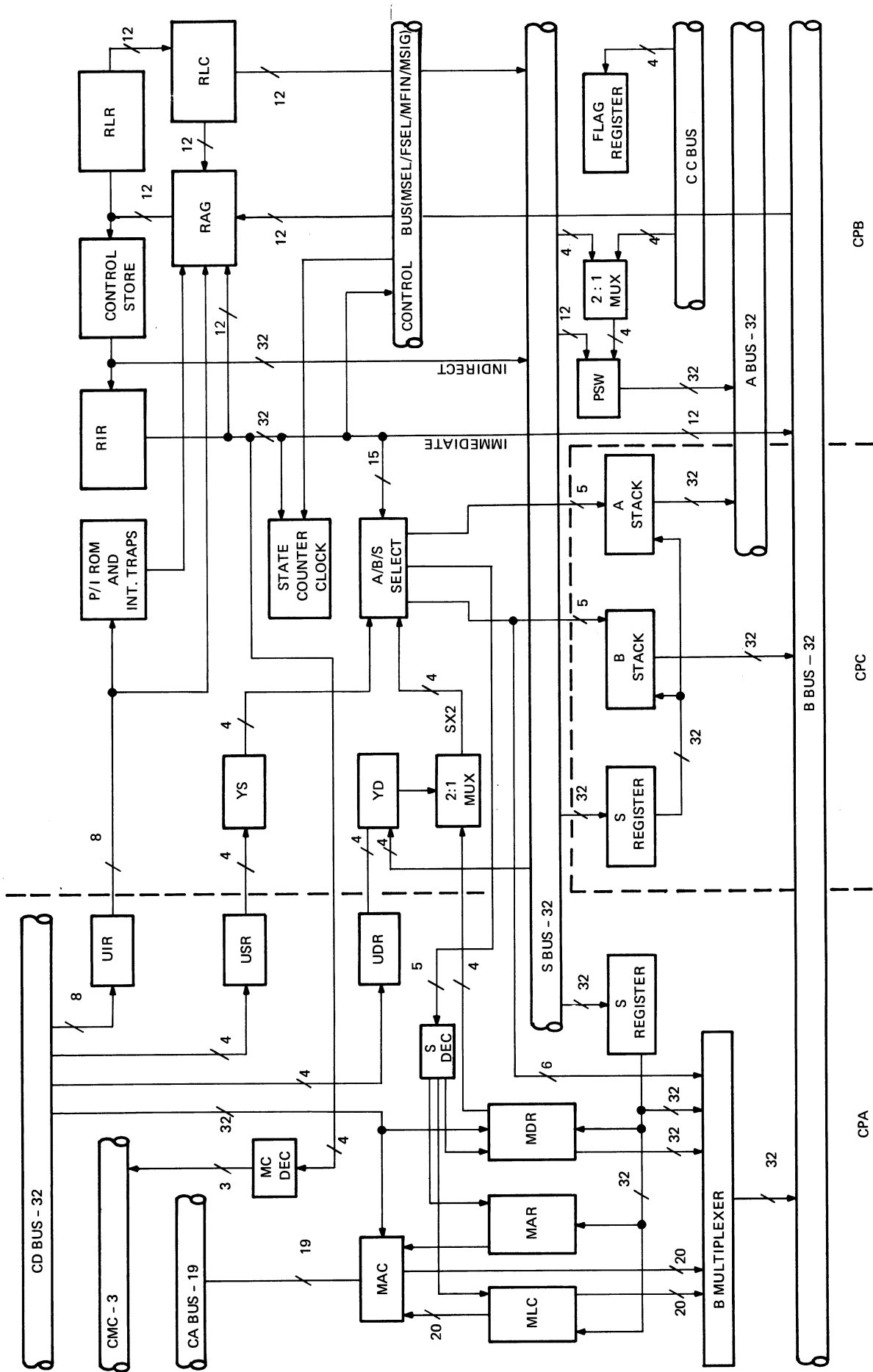


Figure 3. CPU (CPA, CPB, and CPC) Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The D Bus functions may be performed with or without an address cycle depending upon the state of FSEL00. KSIG is used to specify register type operations or to distinguish between the halfword functions (RDH/WDH) and some non-D Bus operations (STB/LB).

The function mnemonics are listed as follows:

MNEMONIC	FUNCTION	COMMENT
RD/RDA WD/WDA OC/OCA SS/SSA	Read Data Write Data Output Command Sense Status	byte/indexed byte/indexed byte/indexed byte/indexed
RDR/RDRA WDR/WDRA OCR/OCRA SSR/SSRA ACK	Read Data Write Data Output Command Sense Status Acknowledge Interrupt	byte/register byte/register byte/register byte/register byte/register
RDH/RDHA	Read Data Halfword	two data cycles for byte controllers
WDH/WDHA	Write Data Halfword	two data cycles for byte controllers
THW	Test HW Control Line	State of HW → MSIG
STB LB STBR LBR	Store Byte Load Byte Store Byte Load Byte	halfword/indexed byte/indexed halfword/register byte/register
SMCR CMCR	Sense Machine Control Register Clear Machine Control Register	
EXB LDWAIT	Exchange B-Bytes Load Wait Flip-Flop	B (16) to FWAIT
POW POUT	Release Initialize Relay Gate Output Pulses	

I/O Control/Operation. When the I/O Control is addressed and given a D Bus function code, it creates a one, two or three cycle Multiplexor Channel operation. The device address on A Bus (22:31) is gated to D Bus (06:15) together with the ADRS Control line whenever address type functions are specified. The returned SYN signal terminates the address cycle and initiates the next control function (DA, DR, CMD or SR). Delay timing within each cycle insures that the relationship of the Control line and D Bus signals meet the Multiplexor Channel Timing requirements.

The halfword functions (RDH/WDH) have a single data cycle when the Halfword (HW) Test line is active and two data cycles when a byte oriented controller is addressed (HW Test line inactive).

Output data is gated from the B Bus to the D Bus. The input data is gated from the D Bus to the input Data Register and then onto the S Bus, via ROM controlled S Bus Multiplexors. In a Sense Status operation, Data Register Bits 12:15 are also gated to the Condition Code (CC) Bus via the CC MUX. On all other D Bus operations, four zeros are placed on the CC Bus together with the CC strobe (SCC0). These zeros clear the CPU flag register but do not enter the PSW unless the CPU Enable bit is set.

At the end of the operation, the I/O Control returns a MFIN signal to restart the CPU clock.

The ACK function generates a single cycle on the D Bus and activates one of four Acknowledge lines, according to B30:31.

Byte handling functions. Selected bytes from the A and B Busses are gated onto the S Bus, via ROM controlled S Bus Multiplexors. (See IOU section for details.)

POW, POUT, LDWAIT, SMCR, CMCR, and THW Functions. These functions are described in detail in the IOU section.

Power Monitor. The Power Monitor contains the Primary Power Fail (PPF) detect circuits, the system Initialize Relay (K1) and the associated delay circuits used to control the relay amplifier. This logic provides an orderly system shutdown whenever the system is initialized or when power is turned off or lost.

The 12VAC input from the CPU logic supply is monitored for low line voltage. When trouble is detected, the Early Power Fail (EPF) bit is set in the MCR to create an MMF interrupt. After approximately 1 millisecond, the PPF interrupt is generated and the relay release operation begins. The POW function also initiates the release operation.

For multi-CPU systems, the various Power Monitors may be interconnected for proper sequencing of the sub-systems.

4. CPA GENERAL DESCRIPTION

4.1 CPA Block Diagram Description (Refer to Figure 4.)

The major Processor elements contained on the CPA are the Memory Address Registers (MAR), Memory Data Registers (MDR), the memory address arithmetic unit, the optional Memory Access Controller (MAC), part of the B Bus Multiplexor (BMUX), and an S-Bus buffer register (SREG).

The Memory Address Registers consist of the Memory Location Counter (MLC) and the Memory Address Register (MAR). The Memory Data registers consist of the Memory Data Register (MDR), the User Instruction Register (UIR), the User Destination Register (UDR), and the User Source Register (USR).

The memory address arithmetic unit consists of a Summer (SUM1X), a Multiplexor (MALX), a second Summer (SUM2X) and a second Multiplexor (MUXB).

The optional Memory Access Controller (MAC) consists of a set of 16 Base Registers (BR), a Status Register unit (STATR), and Base Register arithmetic logic (SUM3X and LIMIT).

The B-Bus interface consists of the B Multiplexor (BMUX) and the B-Bus Addressing circuit (BADR). The S-Bus interface consists of the S Buffer Register (SREG) and the S-Bus Addressing circuit (SADR). Additional logic (MC Decoder) is used to decode the 4-bit MC Bus from the CPA to provide the CPB and the memory system with appropriate commands.

4.1.1 CPA Block Diagram Analysis (Figure 4). The following paragraphs provide brief descriptions of the function and data interfaces of each of the blocks in the CPA.

Memory Location Counter (MLC). The MLC is a 20-bit register which stores the starting address of the most recent instruction. It is loaded by a micro-instruction command (micro-control) from the S-Bus through SREG and an ST Bus, and it can be incremented by micro-control. Its outputs are connected to MALX for instruction fetches; to MUX B for relatively-addressed (RX2) data operations, and to BMUX for connection to the B-Bus.

Memory Address Register (MAR). The MAR is a 20-bit register which stores the absolute address for RX1 data operations, the relative address for RX2 data operations, and the total index (sum of the contents of the first and second index registers) for RX3 data operations. It can be loaded from the ST Bus or incremented by micro-control, and its outputs connected, except for Bit 31, to SUM2X for data address calculation. Bit 31, which is only used in I/O operations, is connected directly to Bit 31 of the Memory Address Bus (CA310).

Memory Data Register (MDR). The MDR contains a 32-bit register and a 32-bit multiplexor. The MDR is used to store data read from memory, to store data to be written into memory, and to store the absolute (non-indexed) address for RX3 data operations. The inputs are connected to the ST Bus for loading from the S-Bus or to the Memory Data Bus (CD00:31) for loading from memory. The outputs are connected to BMUX for connection to the B-Bus and, through a set of CD gates, to the CD Bus for writing into memory. The outputs MDR12:31 are also connected to MUXB for use in RX3 data operations.

User Instruction Register (UIR). The UIR is an 8-bit register which stores the Operation-Code (op-code) of the most recent instruction. It is loaded from CD16:23 on all instruction fetches, and the outputs are connected to CPB through front Connector 5.

User Destination Register (UDR) and User Source Register (USR) are used as follows: The UDR is a 4-bit register which stores the RI field of all user instructions, and the USR is a 4-bit register which stores the R2 field of RR instructions, the N field of SF instructions, and the X2 (or FX2) field for indexed instructions. These registers are loaded from CD24:27 and CD28:31, respectively, on all instruction fetches, and the outputs are connected to CPB through front Connector 5.

Summer 1 (SUM1X). The SUM1X is a 20-bit adder which provides temporary increments to the MLC for RX and RI instruction fetches, and, when properly enabled by PSW 21, adds the relocation field from the optional MAC Base Register (BR), if used, to the program address from MALX.

Multiplexor (MALX). The MALX is a 20-bit multiplexor which selects between MLC and the output of SUM2X, for instruction fetches and data operations respectively.

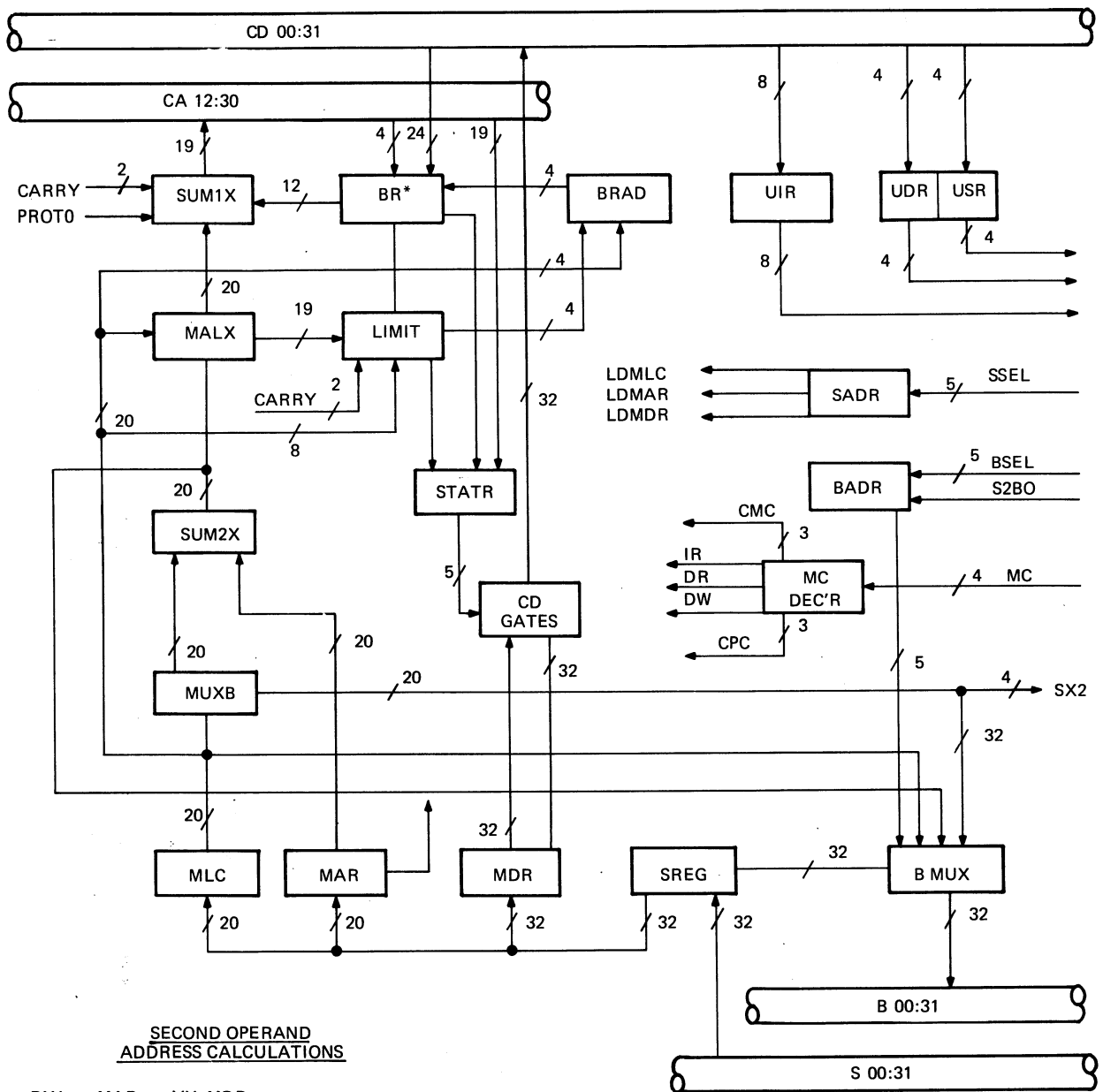
Summer 2 (SUM2X). The SUM2X is a 20-bit adder which sums MAR with MLC (for RX2 instructions), or MDR (for RX3 instructions), or neither (for RX1 instructions).

Multiplexor B (MUXB). The MUXB provides selection between MLC and MDR for inputs to SUM2X.

B Bus Addressing Circuit (BADR). The BADR is a decoder which accepts the BSEL00:04 Bus, instruction format data, and a special command line S2B0 as inputs, and by means of BMUX, connects the appropriate outputs from SR, MDR, MLC, and SUM2X to the B Bus.

S Bus Addressing Circuit (SADR). The SADR is a decoder which accepts the SSEL00:04 Bus as inputs and causes MAR, MLC, or MDR to load from the SR00:31 Bus. SR is a buffer register which loads from the S Bus at each system clock and drives the SR00:31 Bus.

MC Decoder. The MC Decoder accepts the MC00:03 Bus from the CPB and decodes this bus to provide the Memory Command Bus CMC00:02 and all of the internal CPA control signals.



SECOND OPERAND
ADDRESS CALCULATIONS

RX1: MAR ← YX+MDR
CA ← MAR+BR*

RX2: MAR ← YX+MDR
CA ← MAR+MLC+4+BR*

RX3: MAR ← YX+SX2
CA ← MAR+MDR+BR*

INSTRUCTION ADDRESS
CALCULATION

CA ← MLC+BR*

* OPTION

Figure 4. CPA Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

4.1.2 Memory Access Controller (MAC) Block Diagram Analysis (Figure 4). The following paragraphs provide brief descriptions of the function and data interfaces of each of the blocks in the MAC.

MAC Base Registers (BR). The BRs are a set of 16 24-bit registers which store the relocation, limit and interrupt control data for the optional Memory Access Control (MAC). BR receives a 4-bit address from the Base Register Address System (BRAD), is loaded with 24-bits of data from the MDR on command from the micro-program, provides an 8-bit limit field (BR04:11) to the Limit Comparator (LIMIT), a 12-bit relocation field (BR12:23) to SUM1X, and a 4-bit interrupt control field (BR24:27) to STATR.

MAC Base Register Address System (BRAD). The BRAD contains a 4-bit register, two 4-bit multiplexors, a 4-bit adder, and provides addresses to the BR from three sources: the Memory Address Bus CA26:29 for loading, MCL12:15 for instruction fetches, and MALX12:15 (through the 4-bit register) for data operations. The 4-bit adder provides a carry capability whenever MLC is incremented across a memory segment boundary.

MAC Status Register Unit (STATR). The STATR is a 5-bit Interrupt Status register with associated address decoding and interrupt controls. STATR is disabled whenever PSW21 is inactive or whenever a privileged micro-control is effected. (Privileged in this context means that the optional MAC relocation and protection are disabled, exactly as if PSW21 were made inactive.)

When relocation and protection are disabled, a decoder senses CA12:31 and traps the locations assigned to the optional MAC. It is then possible to load BR or to read the five bits of STATR using the same procedures that are used to read or write to memory. When relocation and protection are enabled, references to the trapped locations results in accesses to memory. The Memory Access Interrupt logic is also activated, under control of BR24:27.

MAC Limit Comparator (LIMIT). The Limit Comparator compares each memory address with the 8-bit limit field in BR and, when the protect function is enabled, causes an interrupt to be generated if an attempt is made to access a memory address which is larger than the limit.

MAC Summer 3 (SUM3X). The SUM3X monitors the MLC and anticipates when the incremented MLC (for RX and RI instructions) passes a segment boundary. When this occurs, the BR address is incremented by one, and a delay is initiated to allow time for a BR address change.

4.2 Memory Addressing (Refer to Functional Schematic 35-536D08 for mnemonic location.)

The 8/32 memory address data is derived from MLC, MAR, MDR, the optional MAC BR, and two carry signals. Selection from among these sources, and the computations used to arrive at the final address, are determined by the machine cycle (instruction fetch or data operation), instruction format (RR, SF, RX, or RI), and status of the optional Memory Access Controller (MAC).

During instruction fetches, which are initiated by particular states of the MC00:03 Bus from CPB, the program memory address is taken from MLC. RX and RI instructions require increasing the effective address by two or four bytes to access the second and third halfwords of the instruction. If the MAC option is included and relocation is enabled by PSW21, an additional 12-bit relocation field is added to MLC. The effective address is then:

$$CA = MLC + BR \text{ (optional MAC) } + \text{ carries (2 or 4)}$$

this addition occurs in SUM1X. (Sheet 8)

When memory is accessed for data operations in response to a user instruction, the effective address may be the sum of as many as five parts: an absolute address, a relative displacement, a first index, a second index, and an optional MAC relocation field. For RX1 instructions, the absolute address is contained in MDR18:31 and the index in a user register addressed by the contents of register YX in CPB. For RX2 instructions, the relative displacement is contained in MDR17:31, the reference address in MLC (incremented by four) and the index in a user register addressed by the contents of register YX in CPB.

For RX3 instructions, the absolute address is contained in MDR 24:31, the first index in the user register addressed by the contents of register YX in CPB, and the second index in the user register addressed by the contents of register SX in CPA. Each of these program addresses can then be modified by the optional MAC relocation field from BR. The address calculations are:

RX1:	MAR CA	(YX) + MDR MAR + BR*
RX2:	MAR CA	(YX) + MDR MAR + MLC + 4 + BR*
RX3:	MAR CA	(YX) + (SX) MAR + MDR + BR*

*Optional MAC

In each of these formats, the first addition is performed in the ALU and, the second addition is performed, simultaneously, by hardware in the CPA.

4.2.1 CPA Address Computation Instruction Fetch Address Computation. (Sheets 3, 6 and 8) The computation $CA \leftarrow MLC + \text{carries} + BR$ (optional MAC) is done in SUM1X as shown on Sheet 8. It consists of five 19-067 4-bit ALU ICs and one 19-068 Carry look-ahead IC which is connected across the 16 most significant bits of the ALU. The "A" inputs to the ALU ICs are connected to MALX12:31. For instruction fetches, SIR1 is active, and the outputs of MALX16:31 are MLC12:31 (Sheet 3) and the outputs of MALX12:15 are either MLC12:15 or are open, depending on whether the optional MAC is enabled. If the optional MAC is disabled or not included, PROT1 (8R2) is inactive and MALX12:15 are enabled. If MAC is enabled, PROT1 is active (high) and MALX12:15 are inactive and effectively all zeros.

The "B" inputs to SUM1X28:30 (8D7) are derived from Carry signals C00 and C01 (Sheet 1), and RX2F1 (Sheet 10). For RX1, RX2, and RI1 formats, MLC must be incremented by two bytes to read the second halfword of the instruction from memory. C001 is made active, and both C011 and RX2F1 are inactive. Since the ALUs of SUM1X are operating on low-active data, these conditions cause the B1 input to be active (low) through the NAND gate (8B4), the B2 input to be inactive (high), and a count of two bytes to be added to MLC28:31 to produce CA28:30. If a Carry (Cn+4) is produced by the lowest-order ALU, this is propagated through the remaining bits of CA by means of the 19-068 Carry look-ahead IC device.

For RX3 and RI2 instructions, which require that three halfwords be read from memory, C011 is made active after the second halfword is read from memory. This disables the B1 input to the ALU (8D7) and, through the AND-OR-Invert gate (8B3), causes the B2 input to the ALU to become active, adding four bytes to the memory location from MLC.

When the optional MAC is disabled, PROT1 (8R2) is inactive, PROT0 (8E6) is active (high) and the ALU is in the A only mode, producing $CA \leftarrow MLC + \text{carries}$. When the optional MAC is enabled, PROT1 is active, PROT0 is inactive, and SUM1X is placed in the A+B mode, where the B inputs to SUM1X12:23 are BR120:230 (MAC option) (Sheet 6), producing $CA \leftarrow MLC + \text{carries} + BR$ (optional MAC).

Data Read/Write Address Computation (Sheets 3, 4, 6, 8 and 9)

The address computations required for data operations are determined by the instruction format, as shown previously in Section 4.2.

For RX1 instructions, the computation which must be performed in hardware is:

$$CA \leftarrow MAR + BR \text{ (optional MAC)}$$

The following conditions exist at the inputs to the gates (8A3 and 8A4) which produce ALU inputs B1 and B2; SDRDWO is active, and C011 and RX2F1 are inactive. Both B1 and B2 are thus held inactive, inhibiting any address carries into SUM1X. The Memory Address Bus CA12:30 is then equal to the outputs of MAMLX12:30, if the optional MAC is disabled or not included (shown previously in this section), or the sum of MAMLX12:30 and BR12:23 (optional MAC) if the MAC is enabled.

MAMLX (Sheet 8) is a 20-bit wide, two-input multiplexor which switches the inputs to SUM1X between MLC and SUM2X, depending on the machine cycle as indicated by SIR1. For data operations, SIR1 is inactive, and $MAMLX \leftarrow SUM2X$.

SUM2X (Sheet 9) is a 20-bit ALU similar to SUM1X. The A inputs are MAR12:31, and the B inputs are MUXB12:31. The A inputs to MUXB are MLC12:31, the B inputs to MUXB are MDR12:31 and the select input to MUXB is RX3F1, so that for RX3 instructions $MUXB \leftarrow MDR$ and for other formats $MUXB \leftarrow MLC$. The control inputs to the ALU are RX1D0A and RX1D0B, which are both active low for RX1 instructions. This control condition cause the outputs of SUM2X to be equal to the A inputs, or MAR12:31, and the address calculation for RX1 instructions is correct.

For RX2 instructions, the computation which must be performed in hardware is:

$$CA \leftarrow MAR + MLC + 4 + BR \text{ (optional MAC)}$$

Since RX1D0A and RX1D0B are inactive (high) SUM2X is in the A plus B mode. The A inputs are MAR12:31, and the B inputs are MUXB12:31. The select input to MUXB (Sheet 9) is RX3F1, which is inactive (low) for RX2 instructions. MUXB12:31 are then connected to the A inputs, which are MLC12:31. Thus SUM2X calculates MLC + MAR.

Since the RX2F1 input to the AND-OR-Invert gate (8B4) is active, when the system control state reaches State 0 (CS001 active) a carry is enabled at the 4-bit ALU (8D7). This increments the output of the ALU (Sheet 8) by four. If the optional MAC is present and enabled, BR12:23 are summed into the ALU and into the address. If the optional MAC is not present or not enabled, the address output is MAMLX + 4. Since SIR1 (8C3) is inactive, MAMLX12:31 is SUM2X12:31 and, as shown previously, this is MAR + MLC. Thus, $CA \leftarrow MAR + MLC + 4 + BR$ (optional MAC), as required for RX2 instructions.

For RX3 instructions, the computation which must be performed in hardware is:

$$CA \leftarrow MAR + MDR + BR \text{ (optional MAC)}$$

The calculation of MAR + MDR is done in SUM2X by RX3F1 (9C2) becoming active and switching MUXB12:31 to MDR12:31. The final calculation is done by the ALU (Sheet 8) as for RX1 instructions.

Since MDR is used to provide part of the address for data reads and data writes, it is essential that the address be kept stable during the memory operation. For data reads, MDR is used as a double-rank buffer. The data from memory is loaded into the register (Sheet 4), by making both MCLK0 and the Load inputs on Pin 09 of the 19-070 IC devices simultaneously active (low). The outputs MDR00:31 do not change until MCLK0 becomes inactive, which occurs after memory has been read and the address may change. For data writes, where the data word must be loaded into MDR to write into memory, the calculated address is first loaded into MAR, and then the data is loaded into MDR. This last operation is performed by the micro-program.

4.2.2 Base Register Selection

NOTE

This section is only applicable if the optional Memory Access Controller (MAC) is provided.

The Base Registers (BR) are selected for loading or when MAC is enabled for relocation and protection. When selected for loading, the registers are addressed in the same manner as memory locations and the associated memory locations receive and store the same data as the BR. When the MAC is enabled, the BR is selected using the four most significant bits of the program address: MLC12:15 for instruction fetches and SUM2X12:15 for data operations.

There are 16 Base Registers (BR), selected by the four address bits at Pins 01, 13, 14, and 15 of the 19-075 IC devices (Sheet 6). The 1MB memory is segmented into 16 64KB segments, and the four most significant bits of the 20-bit program address determine which segment, and therefore which Base Register (BR) is in use. For instruction fetches, the 4-bits are taken from MLC12:15. For data reads or writes, the 4-bits are taken from SUM2X12:15.

Base Register Selection for Loading. The Base Registers (BR) and the Status Register (STATR) (see Section 4.1.2) are assigned a group of memory addresses starting with an odd multiple of X'100' from '300' to '900', and ending at address '43' within the group. The MAC is configured to trap all 256 addresses within the group. The particular group used is a function of the system I/O requirements, and selection of the starting location of the group is by means of one or more jumpers (7B8). Table 8 shows the required jumpers for various starting locations.

TABLE 8. STARTING LOCATION JUMPERS

STARTING LOCATION	05J11 TO:	05J13 TO:	05J15 TO:
'300'	05J06	05J03	05J01
'500'	05J05	05J04	05J01
'700'	05J06	05J04	05J01
'900'	05J05	05J03	05J02

Whenever the MAC is disabled, by PSW21 being reset or a Privileged memory command present, PROT0 (7A5) is inactive, and if any address within this trapped interval appears on the CA Bus, a SRTR1 signal (7N6) is made active. This signal (6C8) causes a multiplexor to switch the address inputs of the 19-075 Base Register IC devices (Sheet 6) to CA26:29. The desired register is then selected. If the address is obtained because of a memory write command, a request pulse (CREQ0) is transmitted to memory and, simultaneously, an active pulse appears on RQFF0 (7A9). This causes the flip-flop (7F8) to be set, making BRWR1 active (7N7). This signal is gated with CA300, the halfword select bit of the memory address, and with FWWRT0, a signal (2M7) which is active whenever a fullword write is commanded, to produce SEGWEA0 and/or SEGWEB0. These in turn enable writing from MDX041:271 into the odd or even halfwords of BR, or both.

Base Register Selection for Relocation and Protection (R and P). Whenever PROT0 (7A5) is active, due to PSW21 being set and no privileged memory operation in the Micro-Control (MC) field of the micro-instruction, SRTR1 (7N6) is made inactive. This causes the 19-132 multiplexor at the address inputs to BR (Pins 01, 13, 14, and 15 of the 19-075 Base Register IC devices) to be connected to the output of a 19-133 4-bit adder (6B6). The four address bits to BR then depends on whether an instruction fetch or a data operation is being performed and also whether the carries needed for fetching the second and third halfwords of RX and RI instructions increment MLC beyond a segment boundary.

For instruction fetches, the segment number and therefore the Base Register (BR) address is determined by MLC12:15. Whenever R and P are enabled, and a data operation is not being performed, SRTR1 (6A8) and SDRDW1 (6A4) are inactive, and the address inputs to BR is connected to MLC12:15 through the multiplexors (Sheet 6).

If the program address for an instruction fetch is within one halfword of a segment upper boundary, and the instruction format is RX or RI, the subsequent halfword carry causes the four most significant bits of the program address to increment by one. MLC12:15 does not change, so a simulated carry C3X0 (6A5) is made active by the circuits shown on Sheet 12, which are discussed later. An active C3X0 increments the output of the 19-133 IC Adder (6A6) by one and causes the BR address to increase by one.

If the program address for an instruction fetch is with two halfwords of a segment upper boundary and the instruction format is RX3 or RI2, the second carry into the program address causes the four most significant bits of the program address to increment by one. C3X0 is made active and the BR address is incremented by the adder (6A6).

Sheet 12 shows the circuits which generate the simulated carry signal C3X0. The instruction location within the segment is contained in MLC16:30. For a carry to cause the instruction location to move to the next segment, all of the bits in MLC from MLC29:16 must be active. In addition, if MLC30 is active, the first carry of the instruction fetch sequence propagates up to the Base Register (BR) select bits MLC12:15.

If MLC16:23 are active, the input on Pin 05 of the C3X0 gate is high, whenever SIR1 is active (during an instruction fetch). If, at the same time MLC24:29 are active, the inputs on Pin 01 of the 19-058 gates at locations 12E5 and 12E6 are active. If MLC30 is also active, and C001 becomes active, signals GT0, GT1, and C3X0 all become active.

If MLC30 is inactive, C001 does not generate C3X0. However, if the instruction format is RX3 or RI2, C011A becomes active when the third halfword is read from memory. This causes GT0, GT1, and C3X0 to become active, signalling that the address has incremented beyond the segment upper boundary. GT0 and GT1 are used in limit checking and are discussed later.

When a data operation is commanded by the micro-control bits, a SDRDW1 signal (2S9) is decoded from MC00:03 and latched in a tracking latch (Sheet 2) when DREQ1 becomes active at the start of the memory operation. This causes the multiplexor outputs (6B4) to be connected to the inputs of a quad flip-flop (6B2). These flip-flops store the four most significant bits of the most recent data operation, so that as soon as it is determined that a data operation is to be performed, the Base Register (BR) address lines are switched to what is most likely the correct segment. If, however, the new data address is in a different segment than the most recent data address, a 19-117 4-bit comparator (6C2) is enabled to compare the new segment number on SUM2X 12:15 to the old segment number stored in the 19-131 quad flip-flop. Approximately 100 nanoseconds after the start of the data operation a P100N1 pulse is gated into the quad flip-flop and updates the stored segment number. The output AEQB1 of the comparator is used to cause an 80 nanosecond delay in memory operation whenever the segment number changes, to allow time for the new base register to be accessed and the address calculation to change.

4.2.3 Base Register Write Operation (see MAC note Section 4.2.2). As described in Section 4.2.2, the Base Registers (BR) are addressed from CA26:29 whenever R and P is enabled and one of the trapped memory locations is addressed on the CA Bus. If, at the same time the MC field of the micro instruction calls for a Write operation, a SDW1 signal is decoded from the MC field and appears active (7H8). Subsequently, a request is made to memory and a RQFF0 signal (7A9) becomes active for approximately 50 nanoseconds. The leading edge of RQFF0 sets the flip-flop (7G8) and causes BRWR1 (7N7 and 6H2) to become active.

The Base Registers (BR) are addressed in exactly the same way as memory locations, so that the even halfword, the odd halfword, or the fullword can be written into. If the memory command is Full Word Write, a FWWRT0 signal (6J1) becomes active. This gates BRWR1 to SEGWEA0 and SEGWEB0, the Write Enable inputs to the Base Register (BR).

SEGWEA0 is active whenever FWWRT0 or CA300 are active, and enables writing into the most significant, or even, halfword of the Base Register BR16:27. SEWEB0 is active whenever FWWRT0 is active or CA300 is inactive, and enables writing into the least significant, or odd, halfword of the Base Register, BR02:15.

The data inputs to BR are connected to the outputs of multiplexor MDX04:27. For fullword write operations, MDX04:27 are switched to CD04:27, which in turn are connected to MDR04:27 (Sheet 4 and Sheet 11). For halfword write operations, the data to be written into BR is contained in MDR16:31, whether the location to be written into is odd or even. This data is brought to the odd halfword inputs to BR16:27 from MDR16:27 through tri-state gates (Sheet 11) to CD16:27 and through MDX16:27 to BR. This data is also routed from MDR20:31 through tri-state gates (Sheet 11) to CD20:31 and through gates (11L4 to 11L9) to MDX04:15 and the inputs to BR04:15. Thus, the halfword data is presented simultaneously to the inputs to both the odd and even halfwords of BR. It is written into whichever halfword has an active Write Enable i.e., SEGWEA0 or SEGWEB0.

4.2.4 Status Register Selection, Read and Write Clear (see MAC note Section). As described in Section 4.2.2, memory references to the locations assigned to the Base Registers (BR) and Status Register (STATR) are trapped by logic shown on Sheet 7. Whenever the address is '—40' thru '—43', and a memory reference is started so that the flip-flop at 7F8 is set, a STAT0 signal (7M7) becomes active. If the memory reference is a read reference, the SDR1 signal (7K9) is active, and when memory is read, MDRCLK1 (7L9) also becomes active. This causes RSTR0 (7R9) to become active which connects the outputs of the Status Register flip-flops IR27:31 (sheet 7) to the CD Bus by means of five tri-state gates (Sheet 11). To insure that the CD Bus is not being driven from the memory at the same time that it is being driven from the Status Register, the STAT0 signal causes the CMC Bus (2R9) to change from a Read command to a Write command so that the memory does not drive the CD Bus.

If the memory reference is for a Write operation, the Status Register is cleared. SDW1 (7L7) is decoded from the micro-instruction MC field and, when STAT0 is active, an active condition of RQFF0 caused by the memory reference makes CSTA0 active which clears the Status Register flip-flops IR27:31.

4.2.5 Memory Access Interrupts (see MAC note Section 4.2.2). Whenever the Memory Access Controller (MAC) is enabled, because PSW21 is set and a privileged memory reference is not in process, certain conditions may be detected which causes the CPA to signal the CPB that an interrupt must be taken. This signal occurs in one or more of the following ways:

1. For interrupt conditions which occur during data references to memory, a Memory Access Interrupt (MAI0) signal is made active (7N4).
2. For interrupt conditions which occur during instruction fetches, an ININT0 signal is made active (7N2) and the output of the User Instruction Register UIR24:31 (Sheet 5) is forced to 'FF'.

Memory Access Interrupt During Data Operations. There are four conditions which causes a Memory Access Interrupt (MAI0) signal to become active during a data operation. These are each represented by a bit in the Status Register IR27:30, as shown in Table 9. Also shown are the conditions of BR25:26 which enable the two Write Protect interrupts, which are described as follows:

TABLE 9. MEMORY ACCESS INTERRUPT SIGNALS

BR25:26	IR BIT	MEANING
--	27	INVALID ADDRESS
--	28	NON-PRESENT ADDRESS
X1	29	WRITE PROTECT VIOLATION
10	30	WRITE/INTERRUPT CONDITION

1. Invalid Address Interrupt for Data Operation. An invalid address is an address which exceeds the upper limit of a memory segment as determined by BR04:11 (the Limit field). A pair of 19-117 comparators (Sheet 6) continually compares these Base Register (BR) bits to the eight most significant bits of the un-relocated program address MALX16:23 and, when an attempt is made to address a location beyond the Limit field, LIME1 (6N6) becomes active.

As soon as the memory reference is started, DREQ1 (7A1) becomes active, so that the output of an AND gate (7C1) becomes active if LIME1 is active due to an invalid address. This enables setting of the IR 27 flip-flop (7D1) and, since SIRO is inactive (high) causes NW0 and NW1 (7K2) to become active, which enables setting of the MAI flip flop (7M4).

As soon as an attempt is made to reference memory for a data operation to the invalid address, a RQFF0 signal (7A9) becomes active for 50 nanoseconds. If the optional MAC is enabled, SPROT1 (7A5) is active. Because the optional MAC is enabled, SRTR0 (7A5) is inactive (high), so that the leading edge of RQFF0 sets the IR27 and the MAI flip flops.

If the memory reference is for a data write, SDW1 (7J5) is active, and NW1 causes CDW0 to become active. CDW0 (2K8) causes the Write command to memory on CMC00:02 to be converted to a Read command. When DREQ1 becomes inactive at the end of the memory operation, NW1 becomes inactive. However, since IR27 is set, CDW0 is maintained active as long as both the IR27 and MAI flip flops are set, or until the SDW1 signal is made inactive by another type of memory reference.

As described in Section 4.2.4, the MAI flip flop is cleared whenever a read or write reference to the Status Register (STATR) is made. This removes the CDWO signal, allowing memory Write operations to resume. Also attempting to write into the Status Register clears IR27, making CDWO inactive and permitting Write operations to resume.

2. Non-Present Address Interrupt for Data Operations. When BR27 is reset, and any memory reference is made, the IR28 and MAI flip flops are set similarly to the invalid address interrupt described previously.
3. Write Protect Violation. Whenever the optional MAC is enabled and BR26 is set, the IR29 flip flop (7D4) and the MAI flip flop are set if an attempt is made to write to memory. All attempted Write operations are changed to reads by CDWO until the MAI or IR29 flip flops are reset.
4. Write/Interrupt Condition. Whenever the optional MAC is enabled, BR25 is set and BR26 is reset, the IR30 and MAI flip flops are set, but writes are not changed to reads. This allows the program to continue while the Processor is interrupted.

Memory Access Interrupts During Instruction Fetches from Memory

There are three conditions which cause MAI0 to become active during an instruction fetch. These are shown in Table 10 together with the Status Register bits which represent the condition and the Base Register bit, if any, which enables the interrupt.

TABLE 10. MEMORY ACCESS INTERRUPTS

BR24	IR	MEANING
--	27	INVALID ADDRESS
--	28	NON-PRESENT ADDRESS
1	31	EXECUTE PROTECT VIOLATION

1. Invalid Address Interrupt for Instruction Fetch. The Invalid Address Interrupt for Instructions is similar to that for data operations except that CDWO is not made active, and the MAI flip flop is not set. Instead, a latch (7M2) is set, causing ININT0 to become active. At the same time that ININT0 is made active to the CPB, ININT0 A is made active (5A9), presetting the UIR to 'FF'. This simulated user operation code is used in CPB to vector to a micro-code subroutine to process the interrupt while preventing the data at the invalid address from being executed as an instruction. The active condition of ININT0 is maintained until another instruction fetch is started, at which time ISTB0 (7L2) becomes active and resets the ININT latch. IR27 remains set until the Status Register is cleared.
2. Non-Present Address Interrupt for Instructions. When BR27 is reset and an instruction fetch is attempted, IR28 and ININT0 are set. Subsequent operation are similar to the Invalid Address interrupt (see 1).
3. Execute Protect Violation. When BR24 is set, and an attempt is made to fetch and execute an instruction; IR31, ININT0, and MAI0 are made active. Any subsequent attempts to write into memory are changed to reads until IR31 or MAI are reset. The UIR output to CPB is forced to 'FF' by ININT0 and remains so until the next instruction fetch.

4.3 Memory Reference Operations

All Processor operations which require reference to memory begin with a specification in the MC field of the micro-instruction. Table 11 shows the interpretation of MC00:03, and lists the micro commands in terms of the effect on memory reference operations.

TABLE 11. MICRO COMMANDS

OPERATION	MC 00:03
INSTRUCTION READ	1001 1100
DATA READ	1X1X 1000 1101
DATA WRITE	0X1X 0101

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

4.3.1 MC Field Decoding (Sheet 2). The MC Bus MC00:03 is connected to the CPA via front cable Connector 2. The following commands are decoded directly from the MC field; providing that PASSIA is not active, signifying that a branch is not occurring (for conditional micro-instructions, implementation of the MC field is inhibited whenever a branch is allowed):

CPC001	- Increment MLC by the length of the last command
CPC011	- Increment MAR by four
IR1	- Instruction Read
DRDW1	- Data Read/Write operation
RH0	- Read Halfword

The following commands are decoded and stored in quad flip flops (Sheet 2):

SDRO	- Data Read
SPROT1	- Relocation and Protection enabled
SRH0	- Read Halfword
SDW0	- Data Write
SIR1	- Instruction Read

These commands are stored whenever a new Memory Reference MC field is presented to CPA (as decoded by a 19-058 gate at 2F9) and a system clock CK1A occurs while the Processor is in Control State 0 (2H8). Since a new micro-instruction is read into the RIR of the CPB each time the Processor enters Control State 0, and a system clock occurs at the following transition from Control State 0, these flip flops always have stored the most recently commands to memory. The decoded function DRDW1 from a 19-116 decoder (2L4) is stored differently in a tracking latch (2N9). As long as no memory references are in operation, SDRDW1 tracks the decoded DRDW1. When a memory reference is started, DREQ1A (2L5) becomes active and remains active as long as memory is being interrogated. This signal freezes SDRDW so that the Processor cannot start another memory operation until the first is completed.

The Memory Command Bus CMC00:02 is effectively stored in the quad flip-flop at 2J7. However, these signals are enabled by DREQ1 in gates at 2M7, and 2M8, and are modified by STAT0 and CDW0 from the optional MAC. When STAT0 is active, which can only occur when the Status Register is addressed, the memory is forced to a write condition, which causes the memory to release the CD Bus so it can be driven by the Status Register. CDW0 is active only when an illegal data write memory reference is commanded, which causes all memory Write operations to be converted to Read operations, by modifying Memory Command Bits CMC00:01.

4.3.2 Strobes and Delayed Clock. Each memory reference is started by one or more strobes. A NAND gate (12B1) accepts System Clock (CLK1A), the Instruction Read decoded command (IR1), a CS001 signal which indicates that a new micro-instruction is in the RIR and should be executed, and PASS0A from CPB which indicates that no branch is taking place. This combination makes ISTB0 and DSTB0 active. DSTB1 is the input to two cascaded 100 nanosecond delay lines (Sheet 12). The quiescent levels of the delay line outputs are low, except when a pulse which propagates down the delay line causes the outputs to become high. The 20 nanosecond tap is connected to a gate at 12J1 together with DSTB0. The resulting output STB0 is a 20 nanosecond wide negative pulse occurring after the trailing edge of DSTB0. This pulse is used to reset the MCLK0 latch (4A3).

When DSTB0 becomes active, a flip-flop at 12K2 is preset and a flip flop at 12K3 is cleared. The outputs, both high, of these flip flops are gated in an AND-OR-Invert gate (12M3) to produce DLCK0, which becomes active at the trailing edge of CLK1, when SIR1 becomes active, or at the leading edge of CK1A when SDRDW becomes active.

The width of DLCK0 is determined by whether an instruction or a data operation is in process and whether, in each case, a change in memory segment, and therefore the Base Registers, is required.

During an instruction fetch, SIR1 is active and DLCK0 terminates when the flip-flop at 12K2 is reset. The clock for this flip flop occurs approximately 40 nanoseconds after the trailing edge of DLCK0. If, at this time, the MLC has been incremented and a carry is being propagated to the four most significant bits of MLC (MLC12:15), C7X1 and CK1A (12B1) become active and the two gate latch at 12F3 is set. The output connected to the D input of the flip flop is high, and the flip flop is not reset at the trailing edge of the 40 nanosecond clock.

If MLC12:15 are not going to change, the Base Register (BR) selection does not change, and C3X1 remains low. The latch is reset at the end of every memory operation by P180N0-P200N1, the D input to the flip flop remains low, and DLCK0 is terminated approximately 60 nanoseconds after the trailing edge of CK1A.

During a data operation, ISTB0 is not generated, but DSTB0 and STB0 occur when CK1A, STRT1, and DRDW1 are all active simultaneously, that is, during the system clock which is to initiate a data reference to memory. At this time, SDRDW1 is made active, and DLCK0 terminates when the flip flop at 12K3 is set. The clock to this flip flop occurs approximately 80 nanoseconds after the trailing edge of DSTB0. At this time, a 4-bit comparator (6C2) is comparing the stored segment number with the desired segment number as indicated by SUM2X 12:15, and, if the two are different, AEQB1 at the D input to the flip flop is low. The flip flop remains reset and DLCK0 continues. If the actual and desired segment numbers are the same, AEQB1 becomes active, the flip flop is set and DLCK0 terminates approximately 100 nanoseconds after the trailing edge of CK1A.

If DLCK0 is not terminated at 60 nanoseconds for instructions or at 100 nanoseconds for data references, a pulse generated by P180N0-P200N1 direct clears the first flip flop and direct sets the second flip-flop, terminating DLCK0 at approximately 180 nanoseconds and allowing time for the memory address to settle before a memory request CREQ0 is activated.

If the optional MAC is not present, or if MAC is disabled, SPROT0 (12B4) is in active. This gates P100N1 and terminates DLCK0 approximately 80 nanoseconds for both instruction and data references.

4.3.3 Instruction Read (see Figures 5A, 5B, and 5C, Instruction Read). When an MC field decodes to Instruction Read, IR1 (2N5) becomes active. Since the new MC field is stored in RIR during Control State 0, the conditions required by the four-input 19-060 NAND gate (12B1) are met when the next system clock CK1A becomes active. This causes ISTB0 and DSTB0 to become active. ISTB0 (1L5) causes the IREQ flip flop (1L5) to be direct-set, making IREQ0 active, and DISTB0 (1L6) makes DREQ0 active.

Following the trailing edge of DSTB0, a 20 nanosecond STB0 pulse is made active (12K2).

The negative-going DSTB0 sets two flip-flops on Sheet 12. Since SIR1 (2K7) is latched active at the trailing edge of CK1A, DLCK0 (12R3) becomes active at the trailing edge of DSTB0 (and CK1A) (1K5). DLCK0 and STB0 become active at the same time, but DLCK0 remains active longer than STB0. DREQ1 is made active before DLCK0, so that at the trailing edge of DLCK0 (which is variable depending on addressing conditions) a preset is applied to the CREQ flip flop (1L2). This preset pulse is only 20 nanoseconds wide because it is gated by SHP1 from a 30-019 delay line (1N4), which is pulsed by the leading edge of DLCK0. After 50 nanoseconds, the delay line output is inverted (IR3) and clocks the CREQ flip flop off, so that 50 nanosecond CREQ0 and RQFF0 pulses are generated.

CREQ0 is transmitted to the memory subsystem to request a memory operation in accordance with the Memory Command Bus CMC00:02. At the same time RQFF0 is made active (1A1), causing MDRCLK0 and MDRCLK1 to become active.

Whenever memory is not being referenced, DREQ1 is inactive. This signal holds a two-stage Johnson Counter, consisting of two flip flops (Sheet 1), reset. As soon as DREQ1 becomes active, this counter is free to be clocked.

The leading edge of MDRCLK1 causes the C00 flip flop to be set, making C001 and C000 active. C001 and C010 are gated by CD170 (1G7) in a 19-062 AND-OR-Invert gate. If CD170 is inactive (high) for RR and SF formats, the D input to the DREQ flip flop (1M7) is low.

When the memory has completed the operation requested, CRDY0 (1A2) becomes active for 50 nanoseconds. This causes MDRCLK1 and MDRCLK0 to become inactive. The trailing edge of MDRCLK0 resets DREQ0 if the D input is low, which occurs if CD170 is high. This condition results when the instruction format that is being read from memory is SF or RR. The trailing edge of MDRCLK0 also always resets IREQ0. Thus, the memory operation terminates after one halfword if the instruction is SF or RR.

If CD170 is low at the trailing edge of MDRCLK0, DREQ1 is not reset. A delayed MDRCLK propagates through the delay network (1D2) and produces a positive pulse at the output of the AND gate at 1K3. Since DREQ1 and SHP1 are both active, the CREQ flip flop is direct set and another request to memory is started. The second RQFF0 makes MDRCLK0 active, and the leading edge of MDRCLK1 sets the C011 flip flop (1D6). When memory places the second halfword of the instruction on the CD Bus, Bits CD16 and 17 and the operation code in UIR are examined (Sheet 10) to determine if the instruction format is RX3 or RI2. If it is, the RXILO signal (1E7) is made active. When CRDY0 is returned by memory, MDRCLK0 becomes inactive. If RXILO is low, the D input to the DREQ flip flop is high, and DREQ0 remains active. If the instruction format is RX1, RX2, or RI1, requiring only two halfwords from memory, RXILO is inactive and DREQ0 is terminated at the trailing edge of MDRCLK0.

If DREQ0 remains active after the second CRDY0 is received from memory, the second MDRCLK1 propagates through a delay network at 1D4 producing a delayed pulse DD1 (1J3) which causes another CREQ0, and causes the memory to read another halfword – the third halfword of RX3 and RI2 instructions – onto the CD Bus. The leading edge of the third MDRCLK1 causes the C001 flip-flop to reset. The AND-OR-Invert gate (1G8) decodes C000-DREQ1 and causes the D input to the DREQ flip-flop to become low. The trailing edge of the third MDRCLK0 then terminates DREQ0.

In addition to sequencing out the correct number of halfwords for instruction fetches, the signals developed on Sheet 1 also control loading of the data into the UIR, USR, UDR, and MDR. Whenever an instruction is fetched, the first halfword, on CD16:31, is always loaded into UIR (CD16:23), UDR (CD24:27), and USR (CD28:31) (Sheet 5). The load pulse LDUIR0 (5A3) is obtained from a gate at 4M2 and consists of MDRCLK0 gated by IREQ1, which is active only during the first halfword out of memory.

The second halfword is loaded into both the most and least significant halfwords of the MDR (Sheet 4). The 19-070 devices used for the MDR require a simultaneous low at the Clock (Pin 2) and Load (Pin 9) inputs. Loading occurs on the trailing edge of the Load input. The Clock input MCLK0 is pulsed low at the beginning of each instruction fetch by STB0 setting a two-gate latch at 4B3, and remains low until DREQ0A is terminated and at least one system clock CK0A has occurred. This never occurs until the second and, if required, third halfwords have been loaded into MDR, since DREQ0A is active during the entire instruction fetch.

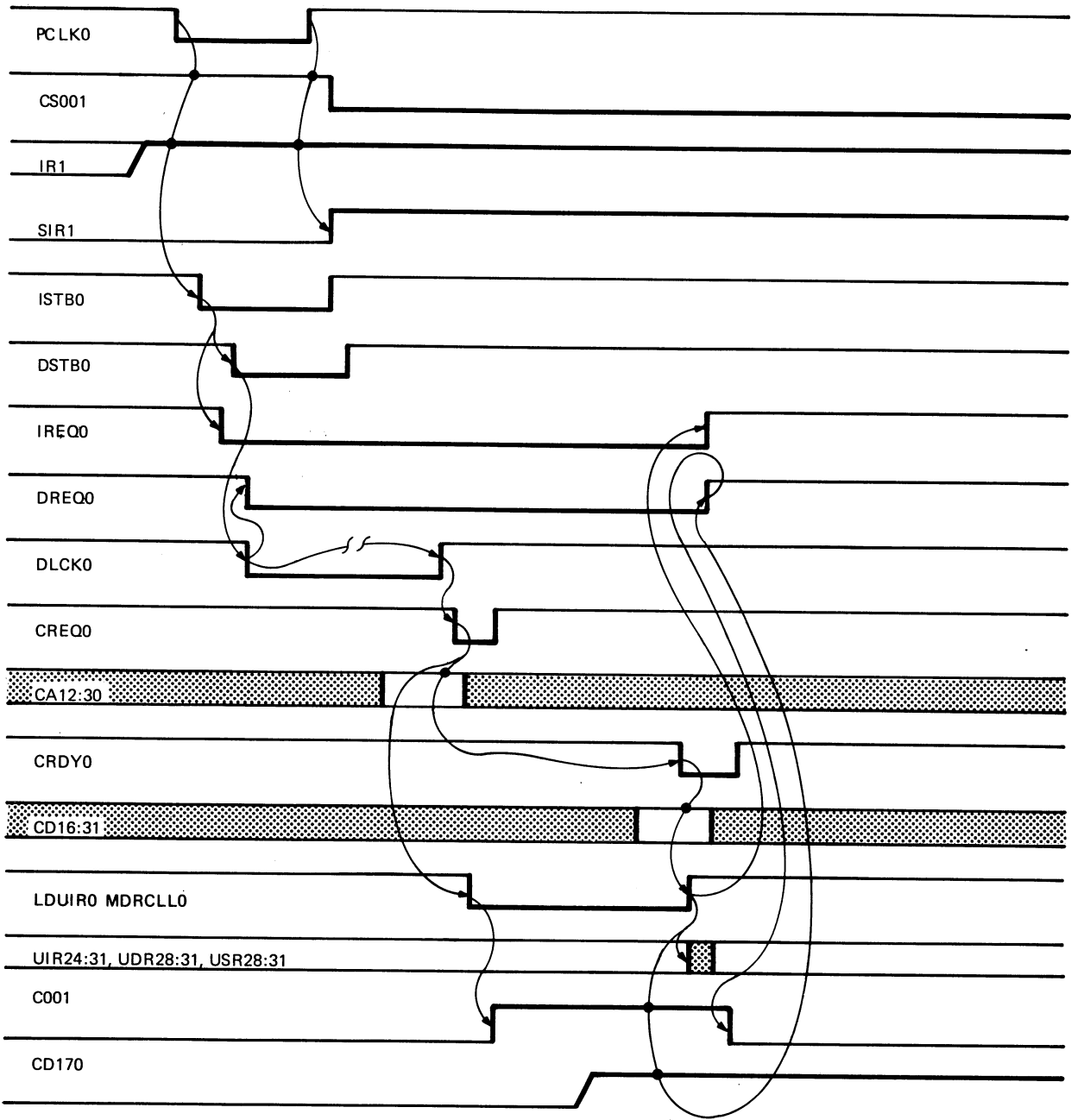


Figure 5A. Instruction Read, RR or SF Formats

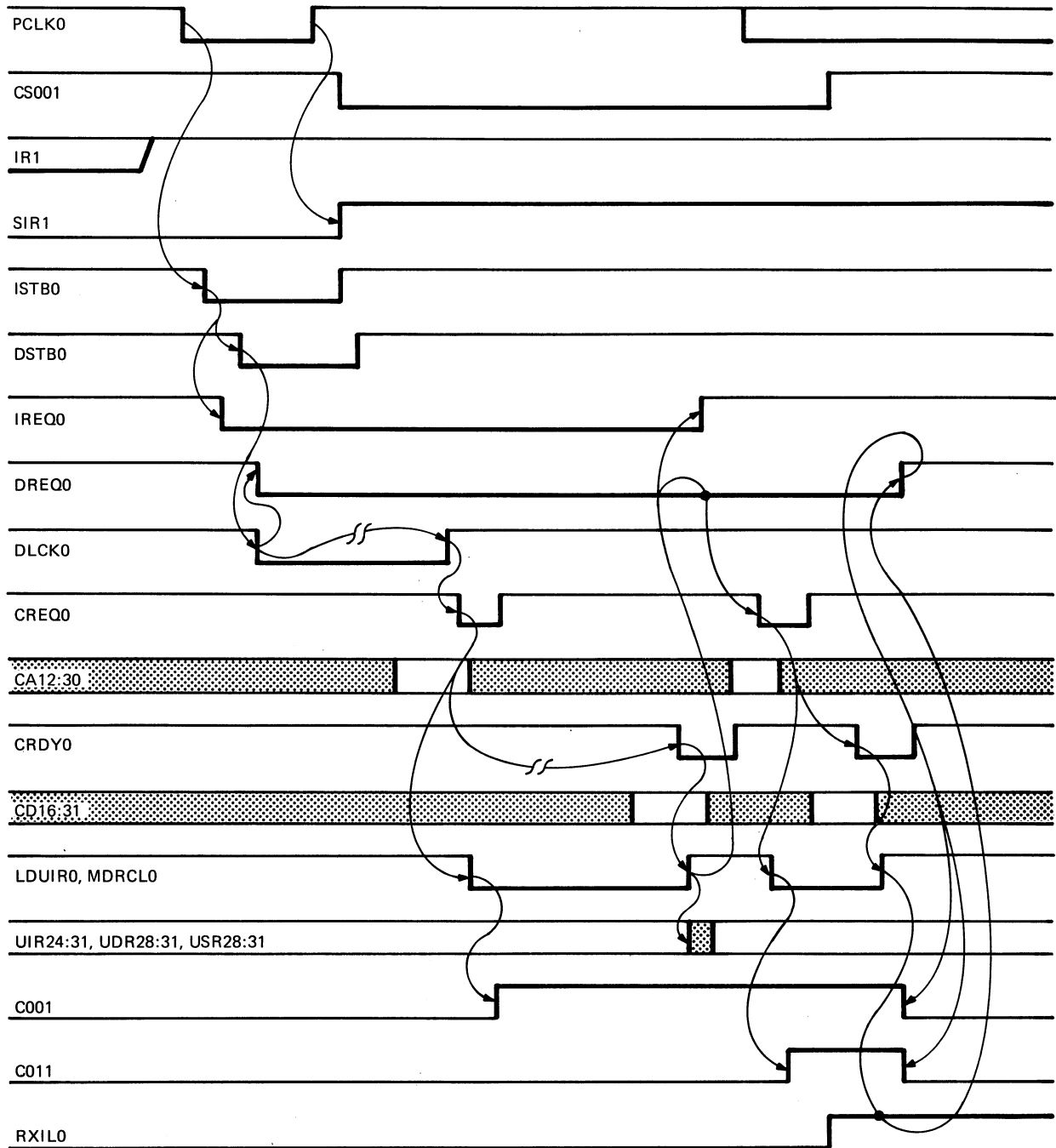


Figure 5B. Instruction Read, RX1, RX2 or R11 Formats

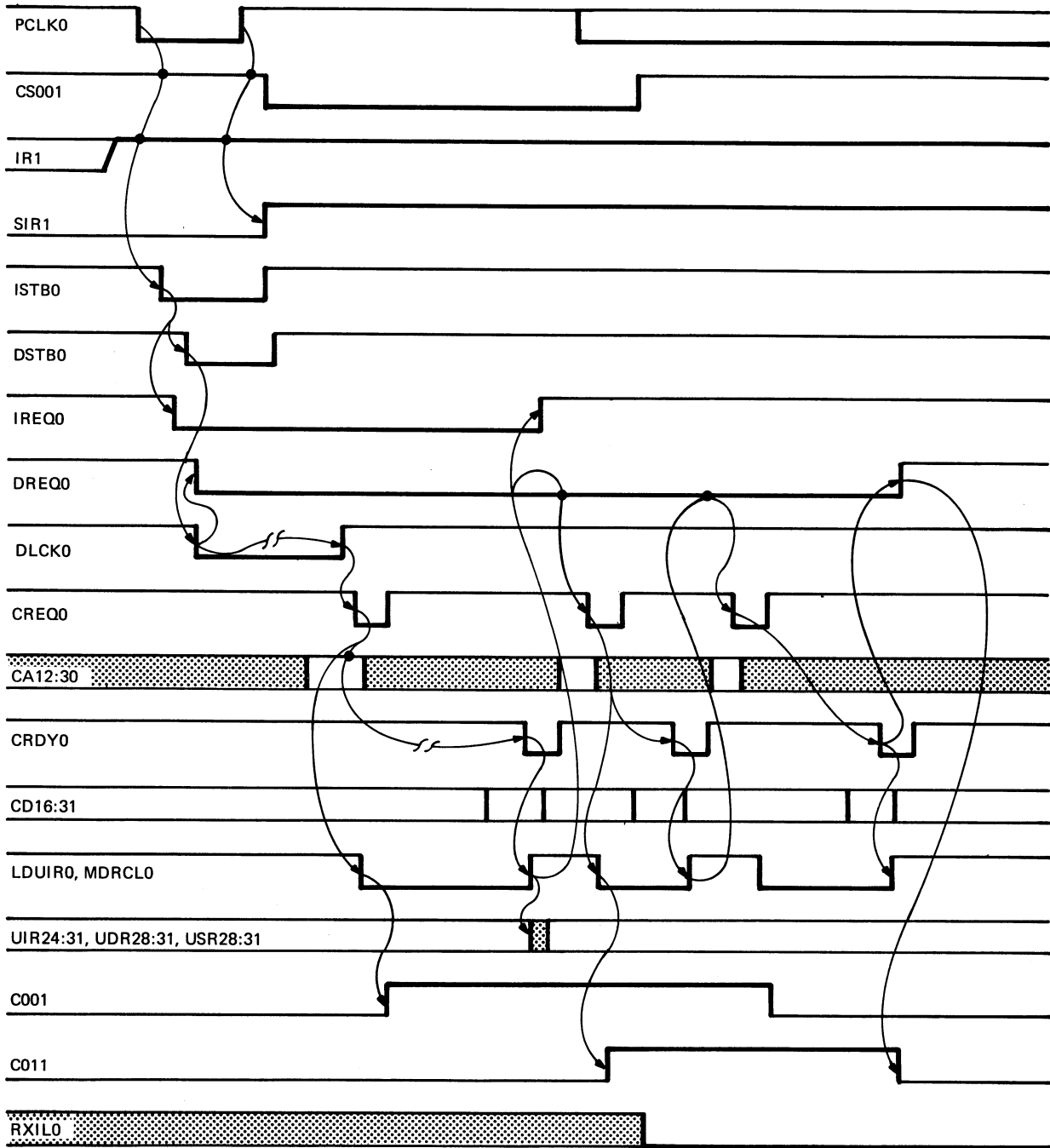


Figure 5C. Instruction Read, RX3 or RI2 Format

While MCLC0 is low, the second MDRCLK0 is gated through AND-OR-Invert gates (Sheet 4) since IREQ1A (4D1) and CC0 (4J2) are both low, causing the data at the inputs to MDR00:31 to be accepted. The data inputs to MDR16:31 are received from CD16:31 through MDX16:31 (Sheet 4) since LDMDR0 is high at this time. However, the 19-066 multiplexors to MDX00:15 (Sheet 4) are disabled because the control input CDXN0 is inactive. CDXN0 comes from an AND-OR-Invert gate (11C3), is inverted at 4J4, and is inactive when C011 is active, which occurs during the second and third halfwords. At this time, tri-state gates (Sheet 11) connect MDX00:15 to CD16:31, so that the data on CD16:31 (the second halfword) is written into both MDR00:15 and MDR16:31.

When the third halfword is available on CD16:31, CC0 is inactive and no load pulse appears at MDR00:15. However, a second load pulse appears at MDR16:31, and the new data on CD16:31 is written into MDR16:31, overriding the previously written second halfword.

When DREQ0A becomes inactive and a system clock has been generated, MCLK0 terminates, and the data which had been loaded into MDR00:31 appears in the outputs of the 19-070 devices.

Instruction Format Decoding and Storage. Since a possible micro-command could be "Increment MLC by the length of the last instruction" it is necessary to store information about instruction format. In addition, information is required to make MDR output appear as proper fullwords, although for RX1, RX2, and RI1 instructions, only the least significant halfword of MDR has meaningful data.

Part of format decoding is accomplished by PROM (10B3). UIR outputs are used as address inputs to PROM, and the four outputs are RX001, signifying RX type instruction; RI021, signifying RI2 type instruction; RRSF1 signifying SF or RR format; and RI1X1, signifying RI1 type instruction.

A CC0 signal produces a clock signal when the second halfword of a two or three halfword instruction is available on the CD Bus. Figure 6 shows the relationship of CC0 to MDRCLK0. At the trailing edge of CC0, a flip flop at 10J6 is set whenever an RR, SF, RX3, or RI2 instruction is decoded from the PROM (10C3) and the states of CD16 and 17. Whenever this flip-flop is set, and CPC001 is active, an INCR1 signal is made active (10M6). CPC001 is a decode from the MC field which commands that the MLC be incremented by the length of the last instruction, so that INCR1 commands that the MLC be incremented by one or three halfwords (two or four bytes).

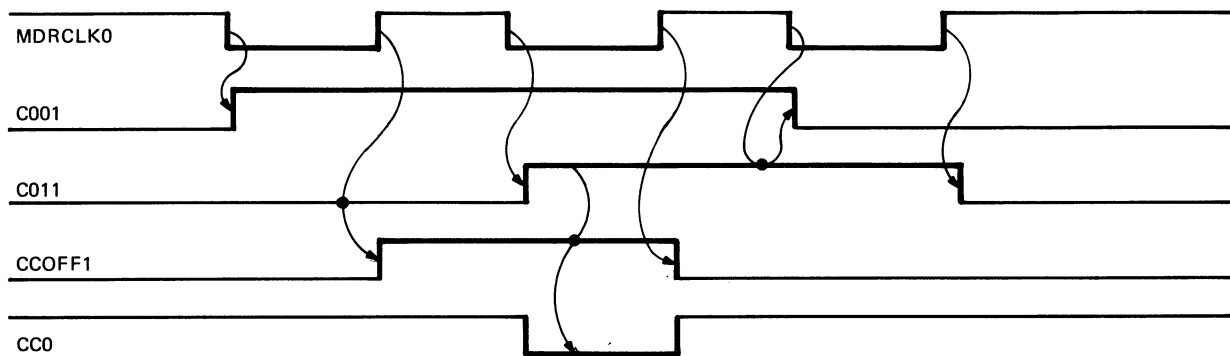


Figure 6. Second Halfword Clock Timing

Whenever an instruction is fetched and RRSF1 (10C3) is inactive, the instruction must have two or three halfwords – a gate at 10L7 decodes this condition and makes INCR021 active, commanding that the MLC be incremented by at least two halfwords (four bytes). Taken together, INCR1 and INCR021 indicate that MLC be incremented by one, two, or three halfwords.

The RX1L0 signal (10M5) is used (1E7) to cause the third halfword of RX3 and RI2 instructions to be read from memory. This signal comes from the RI021 output of the PROM and the output of a gate at 10G2 which decodes RX001·CD171·CD160·SDR0, which is the definition of an RX3 instruction format. The RX1L0 signal is not stored because it is only needed at the second halfword readout from memory.

Whenever the RX3 decode shown previously is active at the D input to the RX3 flip flop (10J1), a LDH0 signal which is used to load the MDR's most significant halfword is also used to set the RX3F1 signal.

A similar decode RX001·CD161·SDR0 is used to detect an RX2 instruction and set the RX2 flip flop (10J3). When the RX3 flip flop and the RX2 flip flop are set, two signals, RX1D0A and RX1D0B, are made active to control the arithmetic function of SM2X (Sheet 9). The RX1D0 signal (10M3) is gated by SDR0 and is only active during the instruction fetch. This signal (12H7) is used to control the effective MDR output as read to the B Bus. For RX1 instructions, B00:15 are made equal to MDR16 whenever MDR is gated to the B Bus. The RX2 output (10J4) is also used (12H7) to extend the least significant halfword on the B Bus, but for RX2 instructions it is MDR17 which is used. Another circuit (12H9) switches the signal BIT161 to MDR171 for RX2 instructions.

A RX3D0 signal (10M5) is used in CPB to control double indexing. This signal is enabled by RX3EN1 which is made active at the start of each memory reference and inactive whenever MAR is loaded. This occurs when $MAR \leftarrow YX + SX$.

A MSIG0 signal (10N4) is used to indicate to CPB that an RX2 instruction is being executed. This signal is tested by the micro-code for certain instructions. When an instruction other than RX2 is being executed, MSIG0 can become active from other sources. JUTY1 is made active for instructions other than RX2 to indicate to CPB that MSIG0 should be ignored.

4.3.4 Data Read Operation. (see Figure 7 Data Read Operation). When the MC field of the RIR in the CPB indicates a memory Data Read operation (see Section 4.3) the following signals are made active: SDR0, SDR1 (2N2), DRDW1 (2M4), and SDRDW1 and SDRDW0 (2R9). In addition, if the command is for halfword read, SRH0 (2N2) is made active, and if the command is a privileged read, SPROT1 (2N2) is also made active.

When DRDW1 is active during Control State 1, and if no interrupts are being processed, STRT1 (12A2) is active. When CK1A becomes active, DSTB0, STB0 and DLCK0 are initiated.

DSTB0 causes DREQ0 (1R7) to become active, and the trailing edge of DLCK0 (after 100 nanoseconds if there is no change in Base Register or after 190 nanoseconds if the Base Register address changes) causes CREQ0 to be initiated. This makes MDRCLK1 and MDRCLK0 active. When data is available on the CD Bus, memory generates CRDY0, which terminates MDRCLK0 and MDRCLK1.

Since a Data Read operation is proceeding, SIR1 (1G4) is inactive, and SIR0 (1G7) at the input to the AND-OR-Invert gate is high. At the leading edge of MDRCLK1, the C001 flip-flop is set, and the output of the AND-OR-Invert gate connected to the D input to the DREQ flip-flop becomes low. At the trailing edge of MDRCLK0, the DREQ flip-flop is reset, terminating memory Data Read operation.

Data is loaded into the 32-bit MDR from the 32-bit CD Bus on the trailing edge of MDRCLK0. At this time SIR1 (4A3) is low, enabling DREQQA to pass through two gates and become MCLK0.

Since SDR1 (4A2) is active, MDRCLK1 is gated to Load inputs (Pin 9) from all stages to the MDR, and data is loaded at the trailing edge of MDRCLK0. However, DREQQA is terminated at the trailing edge of MDRCLK0, so that the new data appears at the outputs from MDR at the trailing edge of MDRCLK0.

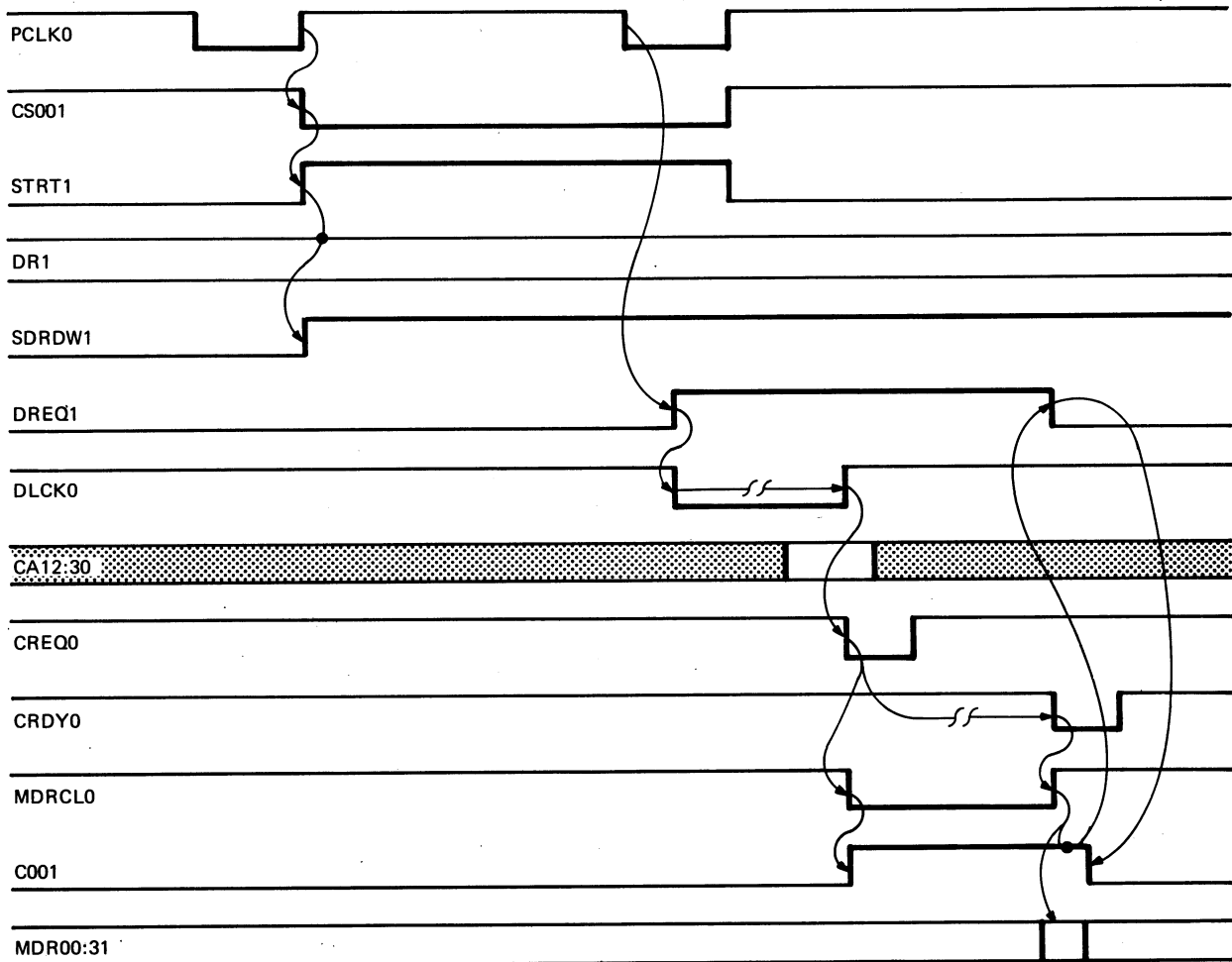


Figure 7. Data Read Operation

The Read operation does not distinguish between fullword and halfword read. However, when halfword read is commanded, SRH0 (2N2) becomes active. This causes the B-Bus multiplexor select signal BMXSLA (12N7) to be set on the following system clock CK0A, and the B-Bus multiplexor is switched from MDR00:31 driving B00:31, to MDR16:31 driving B16:31, and MDR16 driving B00:15. The Halfword Sign Bit 16 is propagated through the most significant bits of the B Bus.

The Read operation does not distinguish between normal read and privileged read. However, when a privileged read is decoded from the MC field, SPROT1 is reset, and all operations are accomplished with program addresses, without relocation and protection.

4.3.5 Data Write Operation (see Figure 8 Data Write Operations). When the MC field of the RIR in the CPB indicates a memory Data Write operation (see Section 4.3) the following signals are made active: SDW1, SDW0 (2N3), DRDW1 (2M4), and SDRDW1 and SDRDW0 (2R9). If the command is a privileged write, SPROT1 (2N2) is also made active.

All of the operations described previously for data read occurs for data write, except for loading MDR from the CD Bus. Before the Write operation is initiated, the required data loaded into MDR by the micro-program, as described in Section 4.4.4. MDR is gated to the CD Bus (Sheet 11). When SDW1 is active (11A3), the control inputs to the tri-state gates (Pins 1 and 15 of the 19-134 devices) become low. This places the output of MDR00:31 on CD00:31 for writing into memory.

When SDW0 (4C3) is active, the gates producing MCLK0 are inhibited, and, since the clock input to the 19-070 devices of MDR are not enabled, the data in MDR is not disturbed during the data Write operation.

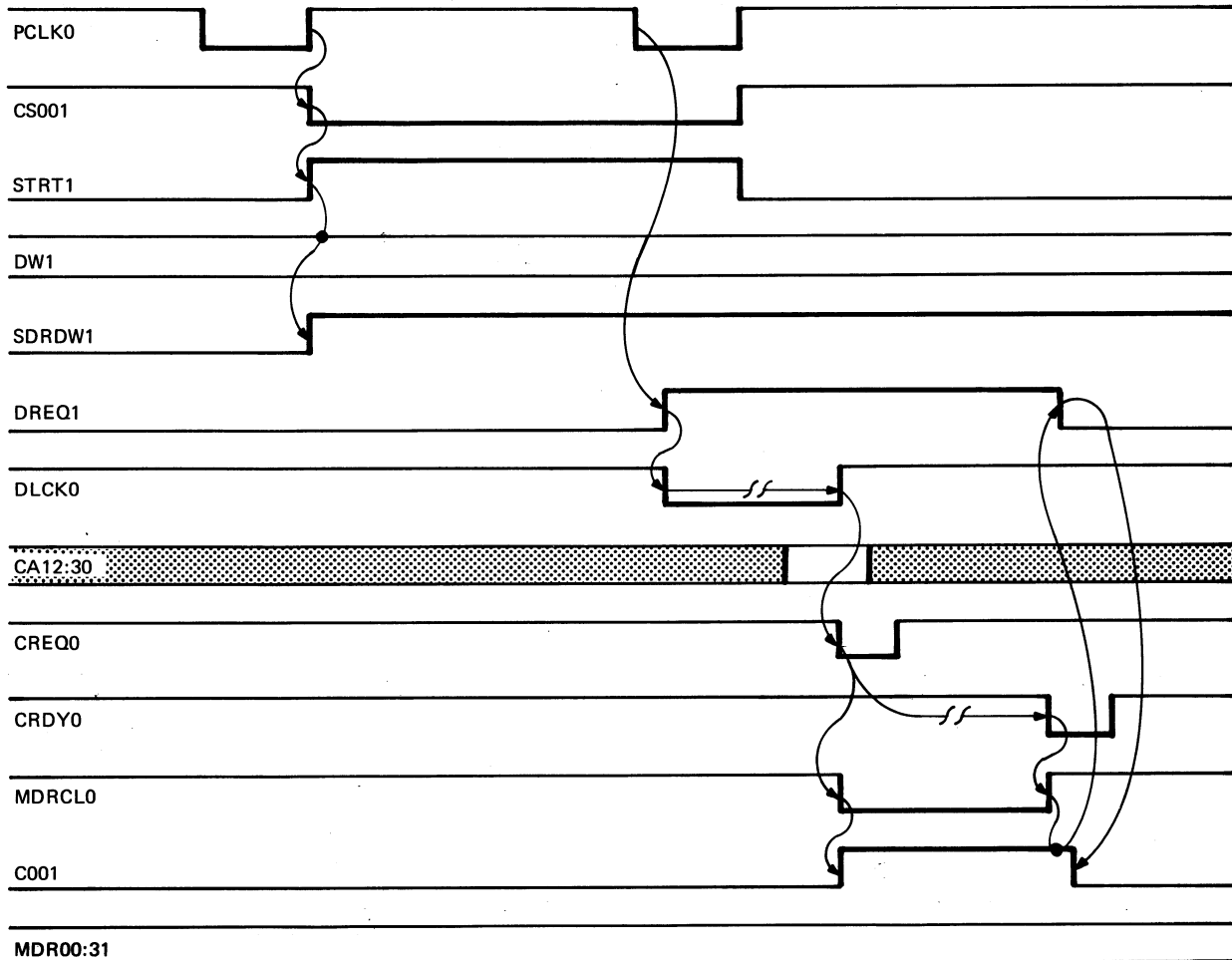


Figure 8. Data Write Operations

4.3.6 **CD Bus Interface.** The CD Bus interface is a bi-directional interface with various parts of CPA, including multiplexing between the even and odd halfwords CD00:15 and CD16:31. Figure 9 shows the various data paths which are involved in this interface.

For Fullword Data operations, CD00:31 are connected to MDX00:31, a tri-state multiplexor, and hence to the inputs of MDR00:31 for loading from memory. The outputs of MDR00:31 are connected through CD gates to CD00:31 for writing into memory.

For Halfword Write operations, MDR00:31 are connected to CD00:31 although the data on CD00:15 is meaningless. Memory insures, based on the Memory Command Bus CMC00:02, that only CD16:31 are written for Halfword Write operations.

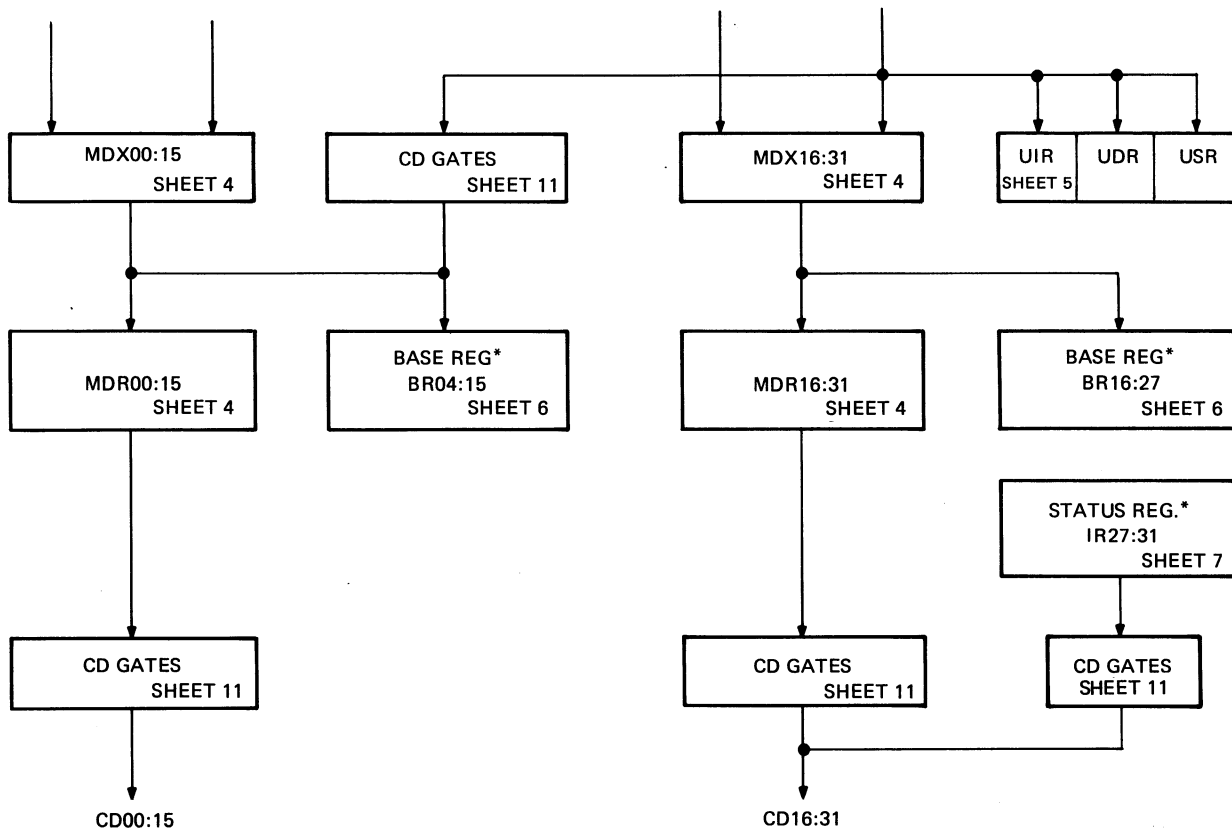
For Halfword Read operations, CD00:15 are multiplexed into MDR00:15, and CD16:31 into MDR16:31, although the data on CD00:15 is meaningless. When the MDR is multiplexed to the B Bus for Halfword Read operations, B00:15 is connected to MDR16:1, propagating the Halfword Sign Bit MDR16 through the most significant bits (B00:15).

During instruction fetches, for RX1, RX2, and RI1 formats, the second halfword of the instruction appears on CD16:31. Through the upper CD gates in Figure 9, and through MDX16:31, this halfword is written into both halfwords of MDR. For these formats, MDR00:15 is replaced on the B Bus with MDR16:1 for RX1 and RI1, and MDR17:1 for RX2 formats.

For RX3 and RI2 formats, the second halfword is also loaded into both MDR halfwords. The third halfword also appears on CD16:31 and is loaded into MDR16:31 through MDX16:31, so that MDR00:31 ends up with the second instruction halfword in MDR00:15 and the third halfword in MDR16:31.

When conditions are correct for writing into the optional MAC Base Registers (BR), the data in MDR is placed on the CD Bus as if to write into memory (the data is actually written into memory) and the memory command (fullword or halfword, even or odd halfword) determine the connections from CD to BR. For fullword write the inputs to BR04:15 are connected to CD04:15 through MDX00:15, and the inputs to BR16:27 are connected to CD16:27 through MDX16:31. For writing the most significant (even) halfword BR04:15 only, the data, which appear on CD16:31 is connected to the inputs of BR04:15 through the CD gates on Sheet 11. For writing a halfword into BR16:27, the data is connected through MDX16:31.

When IR27:31 is to be read out, MDR is disconnected from CD by disabling the lower CD gates, and a set of four tri-state gates at 11A1 are enabled, connecting IR27:31 to CD27:31.



* OPTIONAL MEMORY ACCESS CONTROLLER (MAC)

Figure 9. CD Bus Interface

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

4.3.7 Incrementing MAR and MLC. Certain codes of the MC field indicate that the MLC or the MAR must be incremented before the next instruction is executed. Table 12 lists these micro-commands with the appropriate MC fields.

TABLE 12. INCREMENT MICRO-COMMANDS

INCREMENT MAR BY 4	MC00:03
	X101
INCREMENT MLC BY LENGTH OF LAST INSTRUCTION	X001

The increment commands are decoded (2E5) into CPC001, to increment MLC; and CPC011, to increment MAR.

Incrementing MLC. Whenever CPC001 (2E5) is active, and the Processor is in Control State 0 (CS001 at 10J8 is active) one or both of the increment signals INCR1 and/or INCR021 are active, depending on the length of the previously executed instruction (see Section 4.3.3).

On Sheet 3, there is a four-bit address loop which is used to increment MLC by two, four, or six depending on the state of INCR1 and INCR021. The 19-131 quad flip-flop (3D4) stores the four least significant bits of MLC. The inputs to these flip-flops come from a 19-066 multiplexor (3D1), and the inputs to this multiplexor are either SR28:31, for loading MLC from the S Bus, or ADA28:31, the outputs of a 19-133 four-bit adder (3D8). As long as LDMLC0 (3C2) is inactive (high) the multiplexor, flip flops, and adder form a loop. At each system clock CK0B, the contents of the quad flip flop are updated with the outputs of the four-bit adder. As long as INCR1 and INCR021 are both inactive, no change occurs in MLC. When INCR1 becomes active, the output of the adder increases by two, and this incremented data is loaded into MLC28:31 at the next system clock. When INCR021 becomes active, the output of the adder increases by four, and this incremented data is loaded into MLC28:31 at the next system clock. If both INCR1 and INCR021 are active, the output of the adder increases by six, and this incremental data is stored in MLC28:31 at the next system clock.

Whenever the incremented output of the adder overflows ADA28, a carry is propagated MLC12:27 from the C4 terminal of the 19-133 adder (3E8). The carry input to MLC15 is brought to 6A5 to cause an increment in the optional MAC Base Register (BR) address.

Incrementing MAR. Whenever the "Increment MAR by four" command results in CPC011 becoming active, MAR12:29 (Sheet 3) is incremented by one. However, an increment of one at MAR29 represents an increment of four bytes, since MAR30:31 are stored in two flip-flops at 3A7 and 3B7, and these two bits are not incremented. CS001 is used (3F3) to enable counting of MAR, so that only one increment occurs for each increment command.

4.4 S Bus Operations

All data and addresses from the Processor to memory communicate via the S Bus and CPA. The data are stored in a register on CPA (SR00:31), the outputs of which are connected to the inputs to MDR, MLC, MAR, and the B multiplexor.

The S Register shown on Sheet 5 is loaded from the S Bus on the leading edge of CPX (5F8 and 4K3) which is synchronous with the leading edge of the system clock.

4.4.1 S Address Decoding. Information about the destination of the data on S00:31 is obtained on SSEL00:04 from the CPB. Whenever data is available on the S Bus for loading into a Processor register, a STRT1 signal is made active by CPB, and STRT0 (2B7) enables a 19-129 3-to-8 decoder (2B7). Table 13 shows the coding of SSEL00:04 for loading MDR, MAR, and MLC; the 19-129 decoder implements the decoding in accordance with this table.

4.4.2 MAR Load. MAR is a 20-bit register which consists of two flip-flops at 3A7 and 3B7, and five 19-135 4-bit counters. The two flip-flops store the two least significant bits of MAR (MAR30:31) which do not require incrementing. The 4-bit counter stores MAR12:29, since MAR is incremented on command by four bytes only.

Loading of MAR occurs whenever SSEL00:04 decodes to LDMAR0 active (low). During this condition the trailing edge of the system clock CK1A (3A2) causes the data on SR12:31 to be loaded into MAR12:31 (Sheet 3).

4.4.3 MLC Load. MLC is a 20-bit register which consists of a 19-131 quad flip flop (3D4) and four 19-135 4-bit counters. The quad flip flop, which stores MLC28:31, is loaded through a 4-bit multiplexor (4D2) from SR28:31 when LDMLC0 (from SSEL00:04) is active (low) and MLC12:27 is loaded directly from SR12:27 on the trailing edge of the system clock CK0B.

4.4.4 MDR Load. MDR00:31 (Sheet 4) is loaded from SR00:31 through a multiplexor MDX00:31 whenever LDMDR0 (decoded from SSEL00:04) is active. Loading occurs whenever the Load input to the 19-070 devices (Pin 9) and the Clock input (Pin 1) are simultaneously low.

LDMDR0 gates CK1B (Sheet 4) into the Load inputs. LDMDR1 gates CK1B into the MCLK0 inputs (Sheet 4). Loading effectively occurs at the trailing edge of the system block.

TABLE 13. BUS SELECT DECODING

BUS ADDRS (HEX)		S-BUS (06:10)	B-BUS (20:24)	A-BUS (11:15)
00:0F and M37 00:0F and M37 10:17		UR FR MR	UR FR MR	UR FR MR
18	11000	YS	YS	YS
19	11001	YD	YD	YD
1A	11010	MLC	MLC	YX
1B	11011	MDR	MDR	YDP1
1C	11100	MAR	MAR	---
1D	11101	PSW	YSI	PSW
1E	11110	YDD	YDI	---
1F	11111	NULL	NULL	NULL

- A – Select: YDP1 – Even YD converted to Odd Number by forcing LSB to ONE. Also manipulated by RWCO line for 64 bit Read operations
- B – Select: On RX3 operations and B-Field = X'1B' (MDR),
 1. SX2 = 0, Convert X'1B' to X'19' (4D)
 2. SX2 = 0, Convert X'1B' to X'1F' (NULL)
- S – Select: On 64 bit Write operations, Even S-SEL converted to Odd number by forcing LSB to ONE

4.5 B-Bus Operations

All data from MLC, MAR, and MDR (except for the 4-bit SX2 field) communicate from CPA to the Processor on the B Bus (B00:31). A 32-bit multiplexor shown on Sheet 13 selects from among the sources.

4.5.1 B-Bus Source Selection. B-Bus source addresses are received by CPA on a 5-bit BSEL00:04 Bus and a single S2B0 line. When S2B0 is active, it overrides the BSEL00:04 address and causes the B Multiplexor on Sheet 8 to connect SR00:31 to B00:31. When S2B0 is active it disables the decoder at 12K5 which produces select signals for the B Multiplexor and enables BMXNA1, which enables the eight 4-bit multiplexors (Sheet 13) to connect the S Register to the B Bus. When S2B0 is inactive, BMXNA1 is inactive, and the SR-to-B multiplexors are disabled. If BSEL001 is active, a RGEN0 signal (12N6) is active, enabling a 19-129 3:8 decoder (12K4). The select signals BSEL011:041 are decoded into BMXNB1, BMXNC1, and BMXSLB1. When BMXSLB1 is low, the MDR is multiplexed to the B Bus. When BMXNC1 is low, and BMXSLB1 is high, MLC12:31 is connected to B12:31, and B00:11 is made inactive. When BMXNC1 is low and BMXSLB1 is low, the outputs SUM2X12:31 of the program address ALU are multiplexed to B12:31.

4.5.2 MDR Halfword Operation for Halfword Data Read. When a halfword is read from memory into MDR16:31, it is necessary to convert this to fullword format by extending Bit 16 across B00:15. This is accomplished at 12M7 by latching up BMXSLA1 in a flip-flop whenever the SRH0 signal (12H7) is active. Since at this time, during a Data Read operation, RX2F0 (12G9) is high, BIT161A and BIT161B (12N9) are both the same as MDR161. When BMXSLA1 is active (high) (Sheet 13) and BMXNB1 is low (selecting the MDR to drive the B Bus) each of the B-Bus signals B00:15 are connected to BIT161A or BIT161B.

4.5.3 MDR Halfword Operation for Instruction Fetches. When an RX1 or RI1 instruction is read from memory into CPA, the second halfword is loaded into MDR16:31. When MDR is gated to the B Bus, MDR161 must be propagated through B00:15 in the same manner as when a Halfword Read operation is performed, and the same circuits, using the RX1D0 or RI1X0 signals in place of SRH0, cause this result to be obtained. When an RX2 instruction is fetched from memory, the RX2F0 and RX2F1 signals (Sheet 12) are made active. This causes BMXSLA1 to become active, propagating BIT161A and BIT161B on B00:15, and connecting BIT161A and BIT161B to MDR171, so that MDR171 is propagated through B00:15 in place of MDR161.

5. CPB FUNCTIONAL DESCRIPTION (Refer to Functional Schematic 35-537D08.)

As stated previously, the logic of the control module is divided among the three CPU boards – The CPA, CPB, and the CPC. The 35-555 CPC board contains the register stacks and optional Dynamic Control Store, and the 35-536 CPA board contains the memory control and optional base register circuits. The major portion of the CPU's control logic, i.e., the control store, state counters, clock, register gating, interrupts, Module 0 gating, and their associated logic is contained on the 35-537 CPB board. The following information refers exclusively to the 35-537 CPB board and its associated schematics.

5.1 State Counter and Control

The CPU control logic is governed basically by two flip-flops which determine the current state of the Processor. The flip-flops are referred to as Control State A (CSA) and Control State B (CSB). The CPU can be in one of the four possible binary combinations of these two elements, which are decoded as Control State 0:3.

The four states of the CPU are represented by the transition diagram in Figure 10.

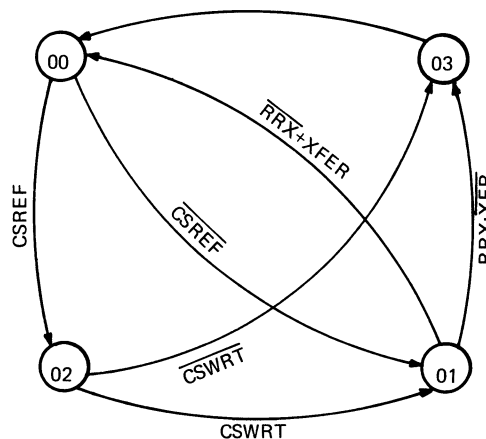


Figure 10. CPU State Diagram

The basic functions of the CPU can be described by correlating the significant logic events of this diagram.

5.1.1 Control State 0 (CS00). An instruction sequence begins in CS00. This state constitutes instruction decoding time. Module number and function selects are gated to the Control Bus. The registers to be selected are decoded, and gated onto the A and B Busses. Logical decisions are made as to the next state transition, micro control of memory, and interrupt recognition. All micro control of memory with the exception of data read and data write is accomplished on the transition from CS00.

When the CPU makes a transition from CS00, it enters Control State 1 (CS01) or Control State 2 (CS02) depending on whether the micro-instruction is a Control Store Reference (CSREF). CSREF is selected by the I Bit (see Figure 10) of the micro-instruction.

5.1.2 Control State 2 (CS02). CS02 is only entered for the purpose of reading data from or writing data into the control store memory. When it is entered for reading, the intent is to replace the data on the B Bus with the data stored at the address specified by the B Bus before signaling a module to Start (STRT). This is accomplished on the transition from CS02 to Control State 1 (CS01). When the state is entered for Control Store Writing (CSWRT), there is no further module communications, and the next transition is to Control State 3 (CS03) to gate the next sequential instruction from control store.

5.1.3 Control State 1 (CS01). This state is entered from CS00 if there was no Control Store Reference (CSREF), or from CS02 when the instruction specified an indirect reference from control store, and is the state where communications to the other modules is performed. At this point, data is valid on the A and B Busses and if the module selected is other than Module 0, STRT is active and the CPU halts until the referenced module returns the Module Finished signal (MFIN).

The transition from CS01 can be to CS00 or CS03.

Control State CS01 to CS03. This transition only occurs on an RRX instruction if the transfer is conditioned and logically should not occur (see RRX instruction description). This transition is necessary because the instruction at the transfer location has been read from the control store during CS01 and the next sequential instruction must then be selected during CS03.

Control State CS01 to CS00. This transition is the default transition when the transition to CS03 is not made.

The ROM instruction Register (RIR) is always loaded on the transition from CS01.

5.1.4 Control State 3 (CS03). This state is entered from CS02 after a Control Store Write (CSWRT) or on a conditional RRX transfer if the transfer is not taken. The only logical function performed in CS03 is addressing the next sequential micro-instruction. The RIR is always loaded on the transition from CS03.

5.1.5 Control State logic. The control state flip-flops (14H2) are initialized to CS03 by XCLR0 on system clear or whenever an address is jammed into the ROM Location Register (RLR). These flip-flops are implemented in D type logic optimized to implement the state diagram of Figure 10. The logic implementation is shown in Table 14.

TABLE 14. CONTROL STATE LOGIC IMPLEMENTATION

TRANSITION	CSA ACTIVE LOGIC	CSB ACTIVE LOGIC	COMMENTS
CS00 to CS02	NONE	CS001·RIR051·MOD000	CONTROL STATE 0 AND INDIRECT BIT SET AND NOT MODULE 0
CS00 to CS01	CS001·(RIR051+MOD000)	NONE	CONTROL STATE 0 AND MODULE 0 OR INDIRECT BIT NOT SET
CS02 to CS01	CS021	NONE	ANY TRANSITION FROM CS02 SETS CSA
CS02 to CS03	CS021	CS021·RIR031·RIR021	CONTROL STATE 2 AND CSWRT INSTRUCTION
CS01 to CS00	NONE	NONE	
CS01 to CS03	RRX1·CS011·RIR251·MSIG1·JUTY0	RRX1·CS011·RIR251·MSIG1·JUTY0	RRX INSTRUCTION AND CS01 AND CONDITIONAL AND MSIG1 AND MSIG NOT INHIBITED BY CPA (JUTY0)*
CS03 to CS00	NONE	NONE	

* For an explanation of JUTY1, see Section 4.3.3

The control state flip-flops are encoded as CS00:03 by gates located at 14M. Note that STRT is encoded as CS011. It is inhibited by INT1 or TRAP121 which are interrupt signals and effectively neutralize the instruction to cause a null cycle to fetch the interrupt instruction.

5.2 Clock and Control (Sheet 13)

The clock for the CPU is a gated oscillator which uses a dynamically selected tap of a delay line to establish the clock frequency. Figure 11 shows a simplified illustration of the clock logic.

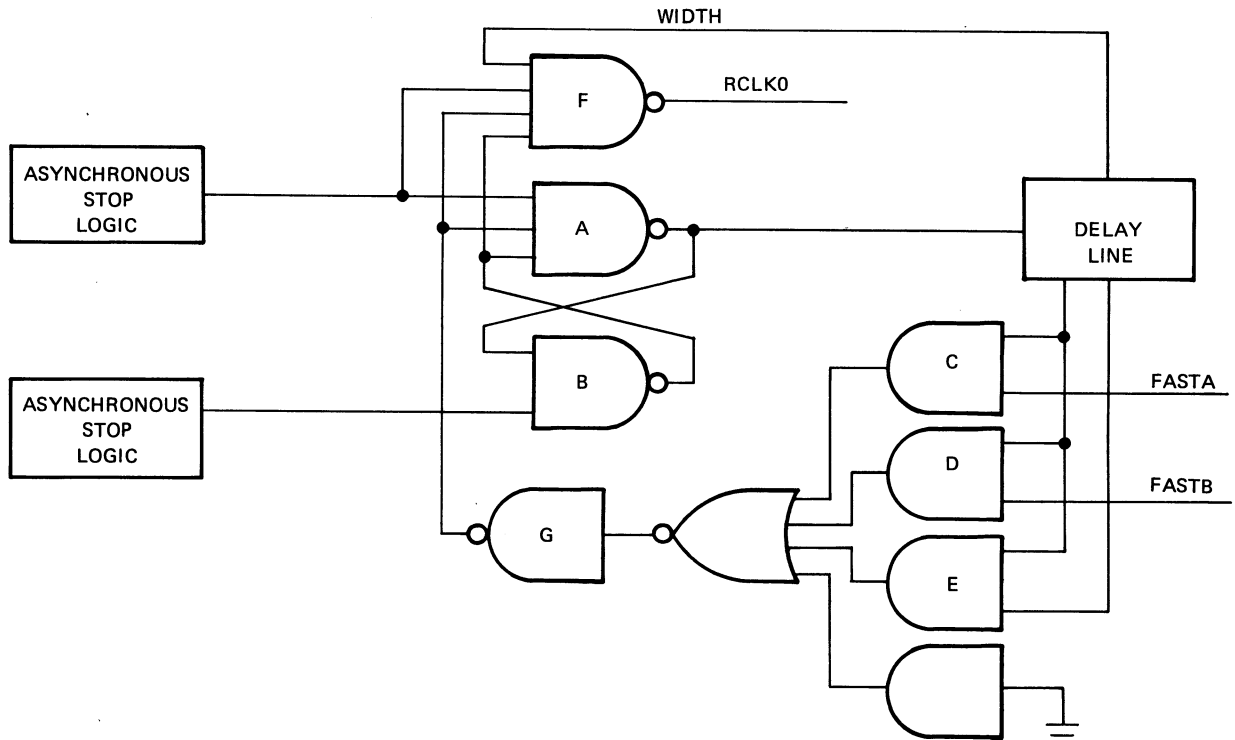


Figure 11. Simplified Clock Circuit

The basic oscillator is comprised of the delay line driver (Gate A), the delay line itself, three gates which are used to dynamically select the desired delay (Gates C, D, and E), and Gate G which provides the necessary inversion for oscillations.

The oscillator can be gated by additional inputs to Gate A. One input is provided by synchronous logical conditions which would cause the clock to stop (i.e., MFIN, or memory interlocks). The second input is from asynchronous conditions (i.e., external stops, manual clock control) and is enabled and latched by Gates A and B to provide proper synchronism to the oscillator. Gate F provides the primary clock for the CPU (PCLK0) and is enabled by the oscillator and the two stop functions. An additional input is a width control from the delay line to modify the duty cycle of the primary clock.

As mentioned, the taps of the delay line are dynamically selected by Gates C, D, and E. The purpose of this is to provide two basic clock frequencies to the CPU. The necessity of this is caused by decoding and access time of the control state and address gating. Since this time is normally longer than one normal clock period, the address gating is decoded so that if the normal transition of the CPU state for the instruction is from CS00 to CS01, both states can be used to access the next micro-instruction. However, if any other state is entered, only one state is provided to access the next instruction and the clock is stretched out. The logic is designed such that if either Gate C or D is enabled, that state is provided the minimum time period; if neither is enabled, the longer period is established.

The actual logic for the clock circuits is on Sheet 13 of the CPB schematics. There are, however, two delay lines cascaded to provide the necessary delay. It should also be noticed in the actual circuit that an eight position switch allows selection of four different taps for each delay for marginal, nominal, and slow clock adjustments. (Note that only one switch of each set can be closed at one time.)

The logic represented by FASTA and FASTB on Figure 11 can be observed as $CS01 \text{ and } CSB0 \overline{(MOD000 \cdot RIR051 + MOD001 \cdot RIR031)}$, which indicates that CS01 is always a fast clock unless there is a control store reference (MOD000·RIR051) or the instruction is a Register Link (MOD001·RIR031). CS02 and CS03 are always afforded a slower clock.

The asynchronous stop logic is provided by the 19-062 gate (13F3). The following logical conditions can be observed at this gate.

CS011·D1·IREQ1. This logic is used to stop the Processor in CS011 when the decode bit is set (use the op-code of the instruction for a control store vector) and the memory has not completed the reading of the op-code (IREQ1). The LC delay network on IREQ1 is used to delay the response to IREQ to provide adequate access time of the control store when IREQ is removed.

STRT1·CS011·MOD000·MFIN0. This logic stops the Processor clock when communicating with the other modules until MFIN is returned.

INT0·DREQ1·CS001 (MC001 + MC021 + MC031). This gate stops the Processor clock in CS00 if it is a valid instruction (INT0), if any memory control is selected, and memory is busy (DREQ1).

The remaining gate stops the Processor clock at CS00 if it is not an immediate and the MDR is selected on the B Bus or at CS01 if the MDR, MAR, or MLC are selected as S Bus destinations and the memory is busy (DREQ1).

5.3 Control Store

The CPU is directed through its paces by instructions fetched from control store and loaded into the ROM Instruction Register (RIR). The sequential and non sequential (branch) flow of instructions is controlled by the CPU logic by the control store address selection which determines where the next instruction is coming from.

A simplified block diagram of this control is shown in Figure 12.

RIR. The ROM Instruction Register is loaded at the beginning of each instruction from the control store and holds the instruction for interrogation by the control logic while data or the next instruction is accessed. The RIR and MC field logic is located on Sheet 11 of the schematics.

Control Store. The control store holds the micro-program of the computer. Data may also be retrieved from the control store. The maximum addressable range of control store is 4K words. The 8/32 micro-program is contained in 1,280 words of control store (Sheet 10).

RLR. The ROM Location Register stores the address of the current micro-instruction. It is loaded at the beginning of each instruction from the control store address gates unless it is an interrupt or an execute type instruction. It is not changed on an interrupt so that the address from the interrupted sequence can be preserved if desired (Sheet 9). (Execute instructions are described in the Micro-programming specification, 05-058A15.)

RLC. The ROM Location Counter is used to add one to the RLR on sequential instructions. It is the RLR + 1 loaded back into the RLR that causes the RLR to increment. The switch input from the test aid is a second input to the RLC. When it is desired to JAM an address into the RLR, the test aid logic clears the RLR and gates the desired logic to the RLC and through the address gates to select the desired starting address (Sheet 9).

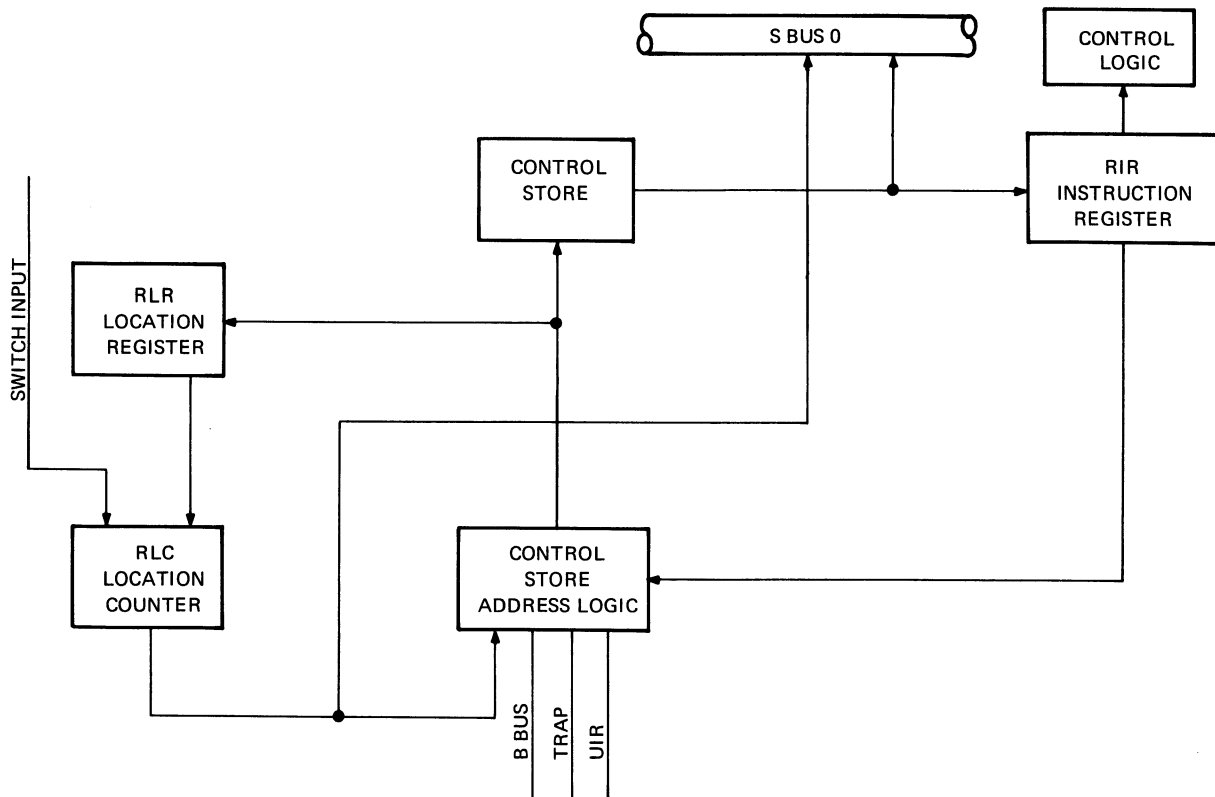


Figure 12. Simplified Control Store Diagram

Control Store Address Gates. The Control Store Address gates select one of five possible inputs for control store addresses. The simplified diagrams (Figures 13A and 13B) show the multiplexor selection for the various address paths. By correlating the diagrams to the actual schematics on Sheets 8 and 9 it is seen that when NEXT is gated (next sequential address) the actual data is RLC4:15. In the same manner for REGLNK or CSREF it is B20:31, and for ADRS LNK it is RIR14:25.

5.4 Bus Selection

5.4.1 **ASEL.** The effective register to be gated into the A Bus is selected by ASEL00:03 (3H2). The actual bus address can be derived from the A Field of the instruction, the YS register, or the YD register. Signal AYDS0 (3A3) decodes from the A field of the instruction whether the A field of the instruction directly or the YS/YD register is used for the ASEL and selects the appropriate side of the multiplexor (3H2). When the YS/YD selection is made, that selection is made by RIR151 at the multiplexor (3C2). The AENO signal is used to null select the field if the YX (user index register) is selected and index register zero (no selection) is made to gate zeros to the A Bus.

5.4.2 **BSEL.** The effective register to be gated to the B Bus is selected by BSEL0:3 (3M4). The effective bus address can be derived from the B field of the instruction, the YS register, the YD register, or the second level index register in the case of an RX3 instruction. The BYDS0 signal (3J8) is used to select either the B field of the instruction or the YS/YD/SX2 register at the multiplex (3K6).

The BYDS0 signal is inactive unless the YD or YS register is selected by the micro-instruction and RIR20:24 (the B field of the instruction) is gated at BSEL0:4. When the B field selects the YS/YD register, the BYSD0 signal becomes active and the YS or YD register is gated to the BSEL from the multiplexor (3C7).

An exception is created when the MDR is selected during an RX3 instruction when the signal RX3D1 from CPA is active. In this case, the BYDS0 signal is made active again and the second level index register field of the MDR, which is sent from CPA as SX28:31 is selected through the multiplexors (2K2 and 3C7) becomes the address of the B register. The SX2NZ1 signal (3E8) determines if the register selected is zero and causes a null selection on the B Bus to be disabling (BEN0) to the BSEL multiplexor. The intent of this is to cause the first and second level indexing to be performed in the ALU as the CPA hardware adds the MDR to MAR to allow RX3 type indexing to be performed transparent to the micro-programmer. (See the Micro-programming Specification, 05-058A15.)

5.4.3 **SSEL.** The effective address to be gated to SSEL0:4 is determined by the SEL field of the instruction. The multiplexor (4E1) selects either the S field itself (RIR06:10) or the YS/YD register for the effective address. When the effective address is YS/YD, the multiplexor (4C2) selects the proper register as per RIR101 from the S field. The effective S address is then latched at CS001 and held available during CS001 of the next instruction when the CPC is storing the results back into the register stack.

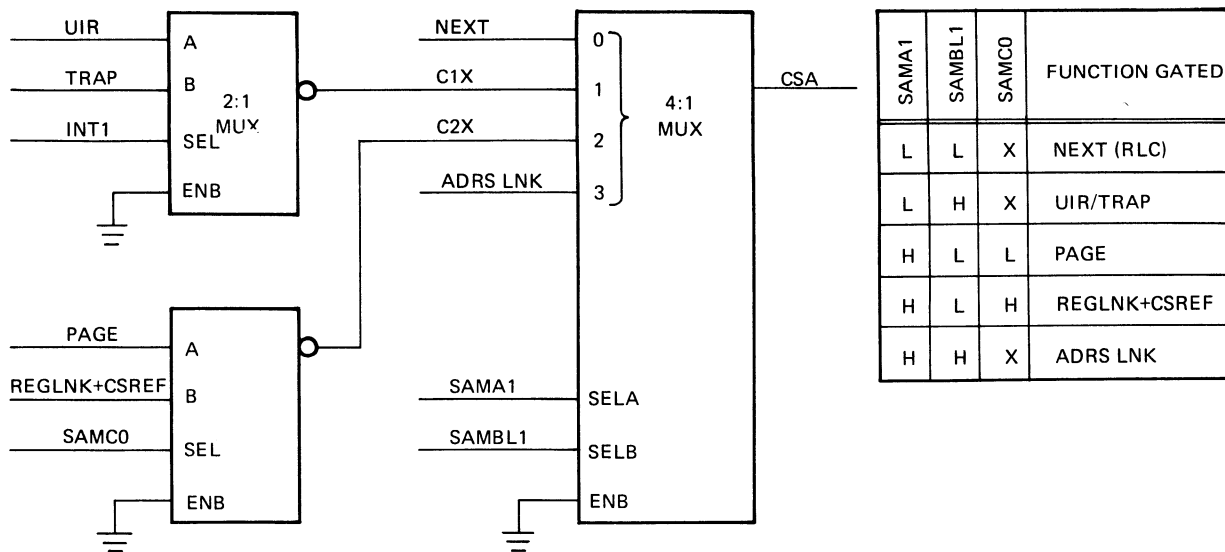


Figure 13A. Control Store Address Gating Low (CSA10:15)

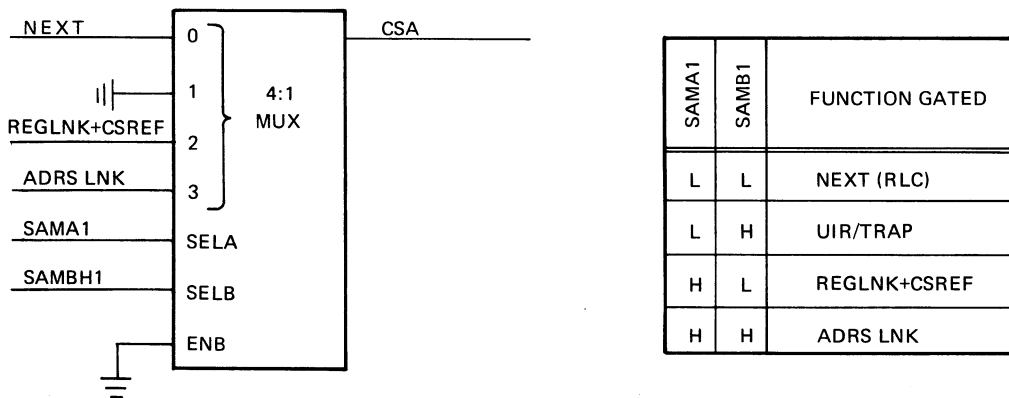


Figure 13B. Control Store Address Gating High (CSA4:9)

5.5 Interrupts

The Model 8/32 employs a hardware priority interrupt scheme which has nine vector traps. See Micro-programming Specification, 05-058A15.

There are three different interrupt mechanisms.

1. **MAI interrupt.** The MAI interrupt is an immediate interrupt and cannot be disabled. It is caused by a data read/write violation in the Memory Access Controller (MAC) on CPA. When this interrupt is sensed, the next instruction is interrupted and a trap to 207 is taken. It is the highest priority interrupt. In addition, all registers are inhibited from changing until the interrupt is acknowledged. This is accomplished by the TRAP120 term in the STRT logic (14J5).

A MAI interrupt is acknowledged when MAI0 (12B8) becomes active, setting the flip-flop at 12D8. This is synchronized by the flip-flop at 12G8 which causes the flip-flop at 12M3 to set at the transition from CS01 or CS03. This flip-flop (12M3) being set activates the INT1 signal which tells the CPU logic that an interrupt is in process (INT1 causes the CPB logic to in effect, "throw away" the current instruction without operating on it while it fetches the next instruction from the interrupt trap). INT1 stays active until the next transition from CS01 or CS03 when the instruction at the trap address is loaded into the RIR. The flip-flop at 12G8 is also input to the priority encoder (12H5) where it forces the trap address "7" for the interrupt vector.

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

2. Group Interrupts. The second class of interrupts are second priority of interrupts and are enabled/disabled as a group by the flip-flop at 12M3. This flip-flop (12M3) may be set/reset by Module 0 instructions, and is automatically enabled at the end of an emulation sequence (D1·PASS0-CSA0) to field all active interrupts and disabled at the entry to a new emulation sequence (D1·PASS0-CSA1). Interrupts in this group are also enabled by PSW bits as described in Section 5.7.

Interrupts in this group are interrogated and latched in registers CS01 (12D3 and 12D6). These registers are prevented from changing while an interrupt trap is being processed to keep the address stable to the control store. These interrupts are then conditioned by their respective PSW bits (note that ATN0:3 interrupts are conditioned by a PROM addressed by several PSW bits at 12D2), and activate the priority encoder 12H4. If this device is enabled by TENO the next CS01 or CS03 sets the flip-flop at 12M6 causing the Processor to activate INT1 and begin an interrupt cycle.

3. Privileged/Illegal Interrupts. These interrupts are the lowest priority interrupts and are activated only at the end of an emulation sequence if there are no other interrupts pending. All possible instructions are programmed as privileged, illegal, or neither in a PROM addressed by the UIR (Users Instruction Register). The output of this PROM is then interrogated at the end of each emulation sequence (D1·PASS0-CSA1) for privileged or illegal instructions. An exception is the MAC instruction interrupt which disables the illegal instruction interrupt and sends an op-code of 3FF as UIR24:31, and sets the flip flop at 12M7 thus activating INT1 and causing the CPU to begin the interrupt sequence.

5.6 PSW Register

The PSW register is an architectural feature of the user level machine and is used to enable interrupts, select register stacks, and contains the Condition Code for user level branch tests.

The PSW (Sheet 5) is loaded when it is addressed as the S Bus destination. A peculiarity of the PSW is that the Condition Code (PSW28:31) can be modified by the current instruction if it is enabled by the microcode (see Section 3.5.3) and the module selected is manipulating the CPU flag register (SCC0 active). See Section 3.5.7. This logic is accomplished by multiplexing the S BUS which is latched in SR28:31 (4C5) at the clock's leading edge with the data to be jammed into the Processor flag register (4E7 and 4E8) from CC0, VCC0, GCC0, and LCC0 at the multiplexor (4E5). A Composite Clock CCCLK0 (4K7) is generated for the PSW28:31 latch.

5.7 Branch Control

The PSW Condition Code (PSW28:31) (4K5) and the Processor flag register (4E7 and 4E8) can be tested for conditional branches by Module 0 instructions.

The PSW Condition Code is compared to the YD register (14B7) for the BTC and BFC tests of the user level architecture and, input to the 8:1 multiplexor (14F5) along with the Processor flag register where the appropriate input is selected by the F field of the instruction as an input to the PASS gate (14K8). PASS0 and PASS1 is tested throughout the CPB logic where pertinent for branch decisions.

5.8 A, B, and S Gating

Various registers are gated to the A, B, and S Busses by the CPB.

YSI and YDI. These registers are gated to the B Bus by the multiplexor at 2H4.

PSW. This register is gated to the A Bus by the tri-state buffers (4M4, 5M3, 5M6, and 5M7).

RIR20:31. These bits are gated to the B Bus for immediate operands (with Bit 20 extended as the sign bit) (Sheet 6).

CSD00:31. The Control Store Data Bus is gated to the S Bus for control store data references (Sheet 7).

RLC04:15. The RLC is gated to the S Bus for link addressing (Sheet 7).

5.9 Test Aids (Sheet 1).

The following test aids are incorporated on the CPB board.

5.9.1 Address Match/Stop/JAM. A control store address can be selected on the three hexadecimal rotary switches located on the front edge of the CPB board (1B2, 1B4, and 1B7) which can be used to sync on, stop on, or JAM to the RLR for a starting address. These switches are compared to the Control Store Addresses (CSA04:15) at the comparators (1G2, 1G4, and 1G7) and produce an output which is ANDed with Clock (CLK1B) and CS000 (1J6) to provide a scope SYNC at TP101-7 (1M6). The output of these switches is also routed to the RLC to provide a way to force a predetermined address. (See Section 5.3.)

5.9.2 MATCH/JAM/SNGL. A rotary switch in conjunction with a push button switch (1M7, and 1M9) is used to control three flip-flops for the purpose of stopping at a selected control store address (MATCH), JAMMING the RLR to a desired address, and single stepping the CPB clock.

When the switch is in the SNGL mode (SNGL LED bit) (1N7), every Processor clock resets the RUN flip-flop (1L8) which generates KLCLK0, an input to the CPU clock circuit. A subsequent toggle of the momentary push button sets the RUN flip-flop and allows the clock to run and generate another clock, resetting the RUN flip-flop.

When the switch is in the JAM mode, a toggle of the push button sets the JAM flip-flop (1L7) which enables JB041:051 to the RLC and causes SETRLC0 (1L5) to clear the RLR.

When the switch is in the MATCH mode, an address comparison sets the MTCH flip-flop (1L6) which activates KLCLK0 in the same manner as SNGL and halts the Processor clock. Subsequently toggling the push button resets the MTCH flip-flop causing the Processor to run until it once again gets an address comparison.

6. CPC GENERAL DESCRIPTION (see CPC Block Diagram Figure 14).

Processor board CPC consist of the A Stack (ASTK), B Stack (BSTK), A Address Logic (AAD), B Address Logic (BAD), Write Clock Timing Logic (WCLK) S Buffer Register (SBUFF), and Register Set Select Logic (RSSL).

ASTK is shown on Functional Schematic 35-555 Sheet 1, BSTK on Sheet 2, AAD and BAD on Sheet 4, WCLK on Sheet 5, SBUFF on Sheet 3, and RSSL on Sheet 3 and Sheet 5.

ASTK is a 256X32 Read/Write Register array. The 32 inputs are connected to the outputs of SBUF (SB001:311) and the 32 outputs (A000:310) are connected to the A Bus which runs along the lower back panel. There are eight address inputs (AAD000:040, AAD051, PSW260A, and PSW270A), a Bus Enable control (ASTKNO), and a Write Clock (WCLK0A).

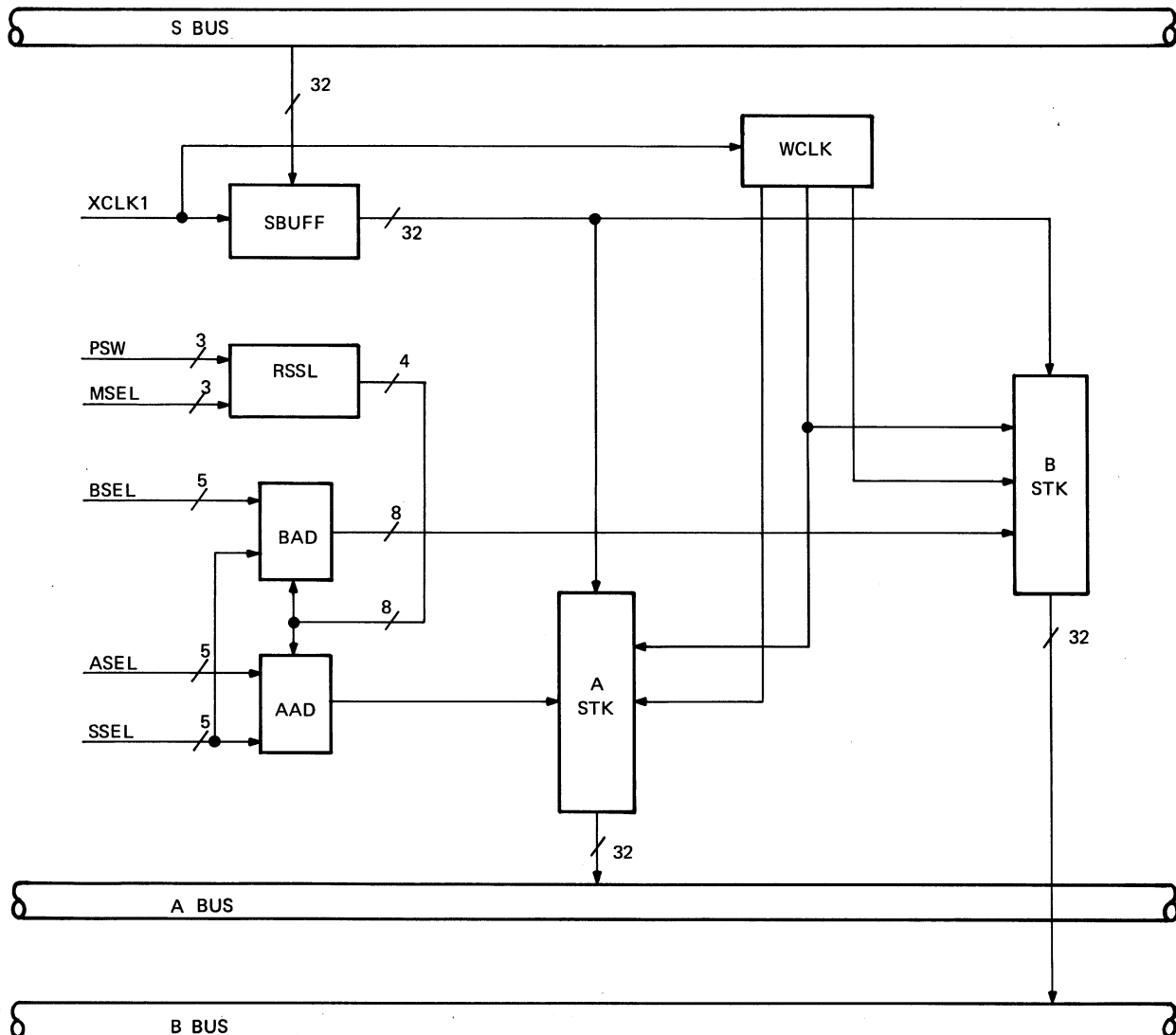


Figure 14. CPC Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

BSTK is identical to ASTK, except that it is connected to the B Bus (B000:310) and receives B Address inputs (BAD000:040, BAD051, PSW260B, and PSW270B) and B Bus Enable (BSTKN0).

ASTK and BSTK can be individually addressed to the A Bus and B Bus, but are always simultaneously selected for loading from SBUFF.

RSSL receives three PSW bits (PSW260, 270 and 251), three Module Select bits (MSEL000:020), and three internal signals (AKLO, BKLO, and SKLO). ADKO is active whenever a Micro-program Register (MRO:7) or a Floating-Points Register (FRO:F) is selected for the A Bus, independent of the state of the PSW bits. BKLO is similarly active for the B Bus. SKLO is active whenever MR or FR are selected to receive data from SBUFF. The four outputs of RSSL (PSW260A, 260B, 270A, and 270B) are used as stack address bits, together with AAD051 and BAD051, to select one of eight sets, from 16 registers each, in ASTK and BSTK for the 16 General Registers (GR0:F).

AAD receives a 5-bit A Select Bus (ASEL001:041), a 5-bit S Select Bus (SSEL001:041), PSW Bit 251 from the CPC, and control signals from WCLK and RSSL. ASTK address bits AAD000:040 and AAD051 are decoded from these inputs.

BAD receives a 5-bit B Select Bus (BSEL001:041), an S2B0 signal, the other inputs to AAD (except ASEL), and produces BSTK Address Bits BAD000:040 and BAD051.

SBUFF is a 32-bit register having inputs connected to the S Bus (S000:310) and having outputs (SB001:311) connected to the data inputs of ASTK and BSTK. The clock input to SBUFF is SCLKI from WCLK.

WCLK produces the timing signals for CPC. Inputs from CPB are PCLK0, the System Clock; STRT0, the Start command for most operations; and SCLR0, the System Clear signal. An input from the ALU is RWC0, which initiates and times 64-bit register Read and Write operations. Outputs from WCLK are WSEL1, WSELIA, WSEL0, WSEL0A, WCLK0, WCLK0A, SODD041, and DWC0. The latter two signals are also used in 64-bit Read and Write operations.

6.1 A and B Stacks (ASTK and BSTK).

Refer to Functional Schematic 35-555, Sheets 1 and 2. The A and B Stacks are functionally identical, so only the operation of the A Stack is described.

The Stack (ASTK) consists of 32 19-077 256x1 Read/Write memories. The data input terminal of each bit cell is connected to the appropriate output of SBUFF (SB001:311). Whenever the enable signal ASTKN0 is active (low) and a Write Clock (WCLK0) is made active (low), the data levels "1" or "0", present at each data input terminal is stored in the bit location for that cell determined by the state of the eight address lines (AAD000:040, AAD051, PSW260A, and PSW270A) which are common to all 32 cells. Thus, after a simultaneous active state of ASTKN0 and WCLK0, lasting at least 40 nanoseconds, the 32-bit output of SBUFF is stored in a specific location in ASTK.

When WCLK0 is made inactive (high) and ASTKN0 remains active, the stored data from the selected location is presented at the ASTK outputs which are connected to the A Bus (A000:310).

When ASTKN0 is made inactive (high) the ASTK outputs are placed in a high-impedance state, removing ASTK from the A Bus.

Figure 15 shows the timing of the ASTK address signals, control signals, and outputs for a Write operation at the nominal 120 nanoseconds clock period. These are described in more detail in Section 6.4.

6.2 S Buffer (SBUFF)

Refer to Functional Schematic 35-555, Sheet 3. SBUFF consists of eight 4-bit 19-131 registers. The data input to each cell of each register is connected to the appropriate S Bus signal (S000:310) and the data output of each cell is connected to the appropriate data input of ASTK and BSTK (SB001:311).

The clock inputs (SCLK1) are derived from a NAND gate at 5C5. For 32-bit operations, SCLK1 is PCLK0 inverted. For 64-bit operations, two SCLK1 active states are generated; the first during the first WCLK0 derived from RWC0 and the second SCLK1 during the subsequent PCLK0. SBUFF is loaded on the leading edge of SCLK1.

Figure 16A shows SBUFF timing for a 32-bit write, and Figure 16B shows the timing for a 64-bit write.

6.3 Stack Addressing

(Refer to Functional Schematic 35-555 Sheet 4 and Figure 17).

The stack addressing scheme is described in terms of ASTK. To explain BSTK, substitute B for A.

6.3.1 Read Addressing. ASTK is addressed for reading to the A Bus with ASEL001:041, MSEL000:020, PSW260, PSW270, and PSW251. There are 10 sets of 16 registers each of which are addressed as Fixed-Point General Register (GR) Sets 0:7, Floating-Point General Registers (FR), and Micro-programming Register (MR). There is also an expansion set of registers for future use as Double-Precision Floating-Point Registers.

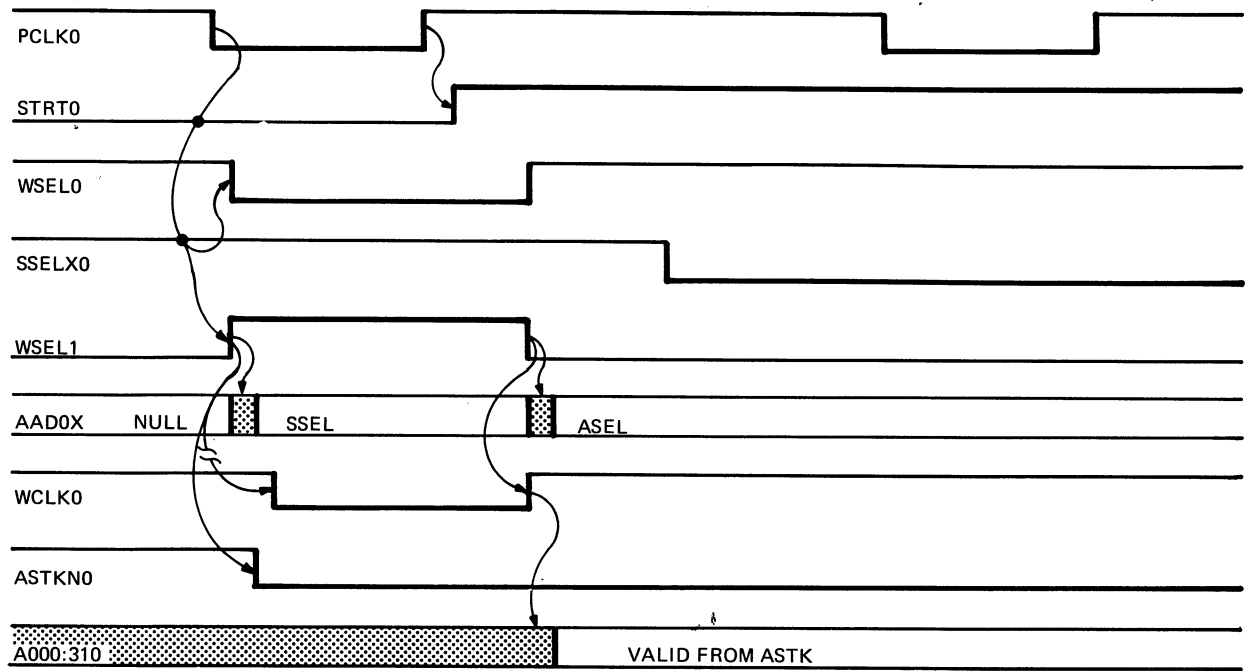


Figure 15. A Stack Timing Diagram, 32-Bit Write

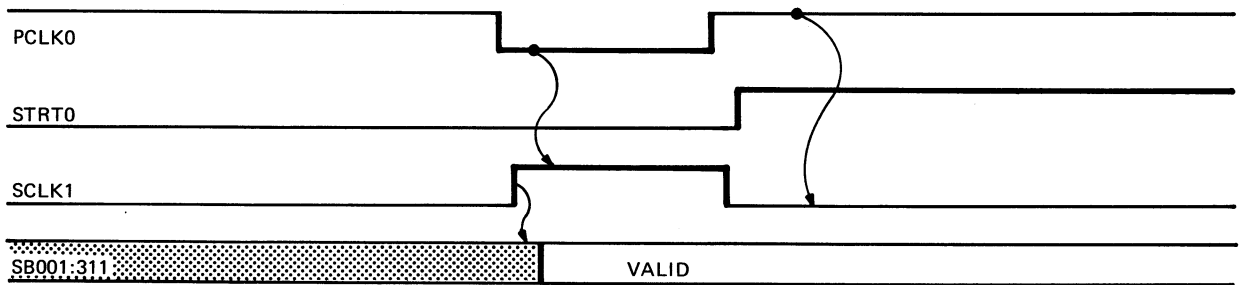


Figure 16A. SBUFF Timing, 32 Bit Write

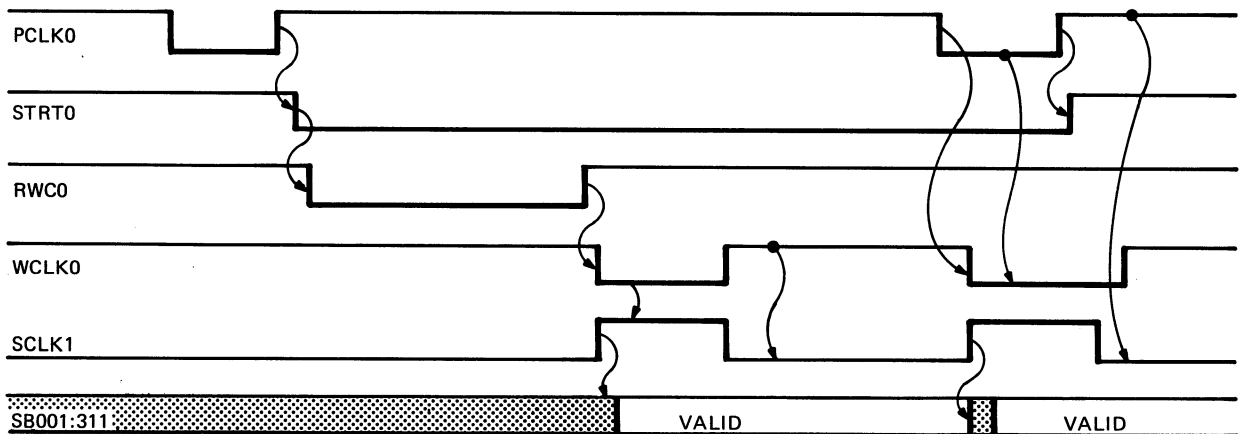


Figure 16B. SBUFF Timing, 64 Bit Write

The register sets are addressed by means of Select Bit ASEL001, the Module Select Bits MSEL000:020, and PSW Bits 25:27. When Module 1 is selected (MSEL000-MSEL010-MSEL021) ASEL001 inactive (low) selects GR, and ASEL001 active (high) selects MR. When Module 3 (MSEL000-MSEL011-MSEL021) is selected, FR is selected, and, when Double Precision Floating-Point is available (DPFR) Module 7 (MSEL001-MSEL001-MSEL021) selects this set.

One of the eight sets of GR is selected by decoding PSW25:27 in accordance with the coding shown on Figure 17. When FR, MR, and DPFR are selected, the PSW bits are ignored.

REGISTER SET		ADDRESS INPUTS						IC PIN											
		ASEL*			PSW			AAD*			PSW								
		0	1	2	3	4	MOD	25	26	27	2	1	15	14	11	10	9	7	
FIXED POINT GENERAL REGISTER	S E T	0						0	0	0	1	X	X	X	X	0	0	0	
		1						0	1	0							1	0	
		2						0	0	1							0	1	
		3	0	X	X	X	X	1	0	1							1	1	1
		4							1	0	0	0	X	X	X	X	0	0	0
		5							1	1	0							1	0
		6							1	0	1							0	1
		7							1	1	1							1	1
FLOATING-POINT GENERAL REGISTERS		0	X	X	X	0	3	D	D	D	1	X	X	X	X	1	0	0	
DOUBLE-FLOATING-POINT GEN. REG. (FUTURE)		0	X	X	X	0	7	D	D	D	1	X	X	X	X	1	0	0	
MICRO PROGRAMMING REGISTERS (MR0:7)		1	X	X	X	X	X	D	D	D	0	X	X	X	X	1	0	0	

X = "1" or "0"
D = Don't Care

*: For BSTK Use: BSEL, BAD, PSW260B, and PSW270B

Figure 17. Stack Addressing Scheme

The address inputs to the memory elements are developed on Sheet 4 in accordance with Figure 17. When WSEL0 is active and WSEL1 inactive, the following logical relations are obtained:

$$\begin{aligned}
 \text{AAD001} &= \text{ASEL001} + \text{M37X1} \cdot \text{PSW251} \\
 \text{AAD010:040} &= \text{ASEL010:040} \\
 \text{AAD050} &= \text{M37X0} \cdot \text{ASEL000} \\
 \text{PSW260A} &= \text{M7X1} + \text{PSW260} \cdot \text{ASEL000} \cdot \text{M37X0} \\
 \text{PSW270A} &= \text{PSW270} \cdot \text{ASEL000} \cdot \text{M37X0}
 \end{aligned}$$

The enable input ASTKN0 to ASTK is developed on Sheet 4 as follows:

$$\text{ASTKN0} = \text{ASEL001} \cdot \text{ASEL011}$$

64-Bit Read from ASTK. When pairs of registers are to be read for Double-Precision instructions (Fixed-Point divide and Double-Precision Floating Point Arithmetic) the addressed register location is always odd. When the first 32-bit word is read, RWC0 is active (low) causing DWC0 (4G5) to be low, forcing AAD040 high, and converting the address to the next lower even register. RWC0 is then made inactive and the second 32-bit word is read from the addressed odd location. The 64-bit read is only implemented for ASTK. BSTK does not respond to RWC0.

6.3.2 Write Addressing. ASTK is addressed for writing from SBUFF with SSEL001:041, MSEL000:020, PSW251, PSW260, and PSW270.

Selection at the register set for writing is similar to selection for reading, except that SSEL001:041 are used in place of ASEL001:041, and S37X1 and S7X1 are used in place of M37X1 and M7X1. S37X1 is latched up with M37X1 on the leading edge of CS001-CLK1, and S7X1 is similarly latched up with M7X1.

For 32-bit Write operations, when WSEL1 is active and WSEL0 is inactive, the address inputs to the memory elements are developed on Sheet 4 in accordance with the following logical relations:

AAD001	=	SSEL001 + S37X1·PSW251
AAD010:040	=	SSEL010:040
AAD050	=	S37X0·SSEL000
PSW260A	=	S7X1 + PSW260·SSEL000·S37X0
PSW270A	=	PSW270·SSEL000·S37X0

The enable input ASTKN0 to ASTK is developed on Sheet 4 as follows:

$ASTKN0 = SSEL001 \cdot SSEL011$

Write addressing of BSTK is similar to ASTK, with the substitution of B for A.

64-bit write into ASTK (Sheet 5). When pairs of registers are to be written into for Double-Precision instructions (Fixed-Point Multiply and Divide, and Double-Precision Floating-Point Arithmetic) the addressed register is always even. During the read part of the read/write sequence, RWC0 is active (low), and if STRT0 is also active, the flip-flop at 5D3 is set. This enables the preset inputs to the WSEL flip-flops (5D3), and when RWC0 becomes inactive, these flip-flops are set and WSEL1 and WSEL1A are made active. Through the mechanism described in Section 4, a WCLK0 is generated which sets the flip-flop in 5E1 and resets the flip-flop in 5D3.

The preset signal which sets WSEL1 from RWC0 becoming inactive, is OR'd with PCLK0 at 5C5, and produces SCLK1, to load SBUFF from the S Bus with the first 32-bit word of the 64-bit result.

When the flip-flop in 5E1 is set by the trailing edge of the first WCLK0, S0DD041 is made active although SSEL042 is inactive because of an even register selection. The next time PCLK0 becomes active, a second WCLK0 is generated. The second 32-bit word is then loaded into the register location which is one higher than the addressed location. At the trailing edge of the second WCLK0, the latter flip-flop is reset.

Figure 16B shows the timing for the 64-bit Read/Write operation.

6.4 Read/Write Control. (refer to Functional Schematic 35-555, Sheet 5.)

ASTK and BSTK are normally in the Read mode, since WSEL1 and WSEL1A are inactive, and WSEL0 and WSEL0A are active. Referring to Section 6.3, this causes ASTK to drive the A Bus and BSTK to drive the B Bus from register locations determined by the address select lines ASEL, BSEL; the PSW Bits 25, 26, and 27; and MSEL00:02.

If the CPU micro-program requires writing data into the register stacks, STRT0 becomes active and either SSEL001 (for MR selection) or SSEL011 (for GR or FR selection) but not both, becomes active. SSELX0 remains inactive (high) and the J inputs to the WSEL flip-flops at 5G3 are enabled. At the leading edge of the following system clock (PCLK0) from CPB, the flip-flops are set, causing WSEL0 and WSEL0A to become low and WSEL1 and WSEL1A to become high. (This changes the stack addressing inputs to the SSEL Bus.) The high state of WSEL1 is propagated down a 100 nanosecond delay line at 5E5, and after the first 10 nanoseconds, WCLK0 and WCLK0A are made active for 50 nanoseconds. At the end of WCLK0, the WSEL flip-flops are reset through the direct clear input, terminating the Write operation and returning ASTK and BSTK to the Read mode.

The 64-bit Read and Write operations are modification to the basic read/write cycle, and have been described previously.

7. ALU (refer to Functional Schematic 35-538D08)

The ALU is a standard module of the Model 8/32 System which implements both fixed point and floating point arithmetic/logical functions. The ALU communicates with the CPU over the A, B, S, and C Busses with all communications being completely asynchronous. The ALU becomes active when it recognizes its address on the Control Bus. For fixed point operations the ALU is addressed as Module Number 1 and for floating point operations as Module Number 3. The CPU signals a Start (STRT) and the ALU performs the function as determined from the Control Bus (FSEL). Refer to Table 15. ALU functions may be of two types. The simple functions (fixed point Add, Subtract, and logicals; and floating point Load, Add With Carry, Subtract With Carry, and Compare) causes the ALU to immediately return a finish signal (MFIN) as these functions are completed within 120 nanoseconds. For these instructions (refer to Figure 18), the A and B Busses are gated to the ALU chips where the function is performed and the result is gated to the S Bus. The ALU does not generate a clock for any of these functions and all gating is performed asynchronously.

For the complex functions (Multiply, Divide, and Shifts; and floating point Add, Subtract, and Compare and Equalize) the ALU clock is enabled and the hardware implementation of these instructions is sequential. For these instructions the Multiplier/Quotient (MQ) shift register, the A Latch (AL) register, and the four shift multiplexors are enabled to perform the iterative operation determined by the instruction. The shift multiplexors are used to shift A or S right or left into the A latch as outlined in the ALU algorithms. For these functions, the ALU does not return MFIN until the operation is completed and the result is on the S Bus.

TABLE 15. ALU FUNCTION CODES

FSEL(HEX)	OPERATION	
	MSEL=X'1'	MSEL=X'3'
0	SUBTRACT	UNUSED
1	ADD	FLOATING POINT LOAD
2	SUBTRACT WITH CARRY	FLOATING POINT SUBTRACT WITH CARRY
3	ADD WITH CARRY	FLOATING POINT ADD WITH CARRY
4	UNUSED	UNUSED
5	LOGICAL AND	FLOATING POINT COMPARE
6	EXCLUSIVE OR	UNUSED
7	LOGICAL OR	UNUSED
8	*LOGICAL SHIFT RIGHT	FLOATING POINT SUBTRACT
9	*LOGICAL SHIFT LEFT	FLOATING POINT ADD
A	ROTATE RIGHT	UNUSED
B	ROTATE LEFT	UNUSED
C	*ARITHMETIC SHIFT RIGHT	COMPARE AND EQUALIZE
D	*ARITHMETIC SHIFT LEFT	UNUSED
E	MULTIPLY	FLOATING POINT MULTIPLY
F	DIVIDE	FLOATING POINT DIVIDE

*KSIG is an extension of the FSEL field and is used to signal halfword shifts.

KSIG is only valid for shift instructions and should not be set for any other instruction type.

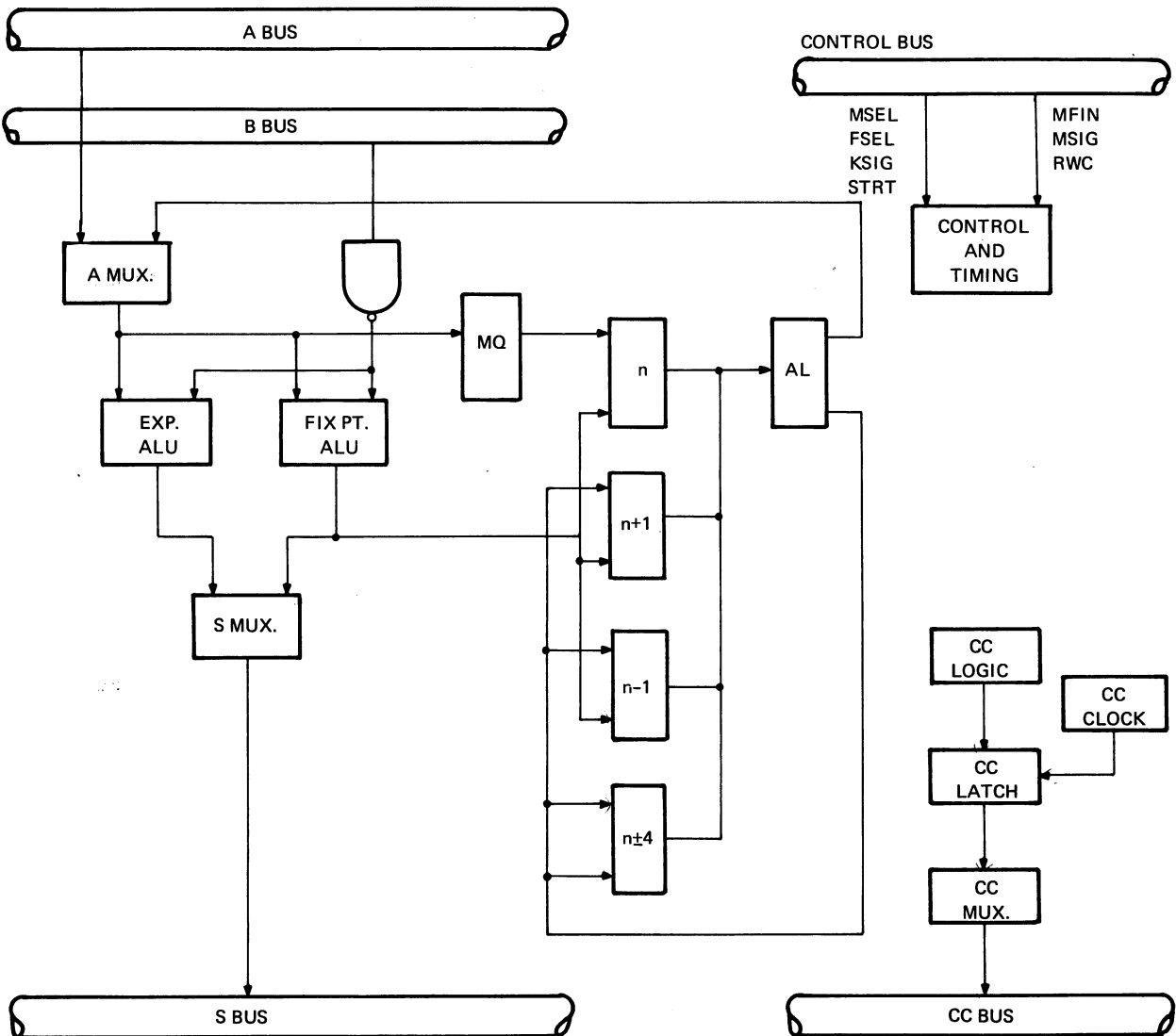


Figure 18. ALU Functional Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

7.1 Arithmetic State Register (AS) (Sheet 5).

The ALU can be in one of five arithmetic states as designated by the conditions of the State Registers ASA, ASB, and ASC. These are further encoded as:

$$\begin{aligned} \text{AS001} &= \text{ASA0} \cdot \text{ASB0} \cdot \text{ASC0}, \\ \text{AS011} &= \text{ASA1} \cdot \text{ASB0} \cdot \text{ASC0}, \\ \text{AS021} &= \text{ASA0} \cdot \text{ASB1} \cdot \text{ASC0}, \\ \text{AS031} &= \text{ASA1} \cdot \text{ASB1} \cdot \text{ASC0}, \\ \text{AS061} &= \text{ASA0} \cdot \text{ASB1} \cdot \text{ASC1}. \end{aligned}$$

The State Register is direct cleared by STRT1, therefore AS001 is the quiescent state of the ALU. Furthermore, the ALU remains in AS001 for the simple functions (FSEL000) previously described and only makes state transitions for the complex functions (FSEL001) when a clock is generated. The various transitions which are possible are described in the ALU Algorithms section. The State Register is implemented in J-K type logic which is tempered with a clock (activated only for FSEL001). The transition diagram is shown in Figure 19 and the logic determining each transition is listed in Table 16.

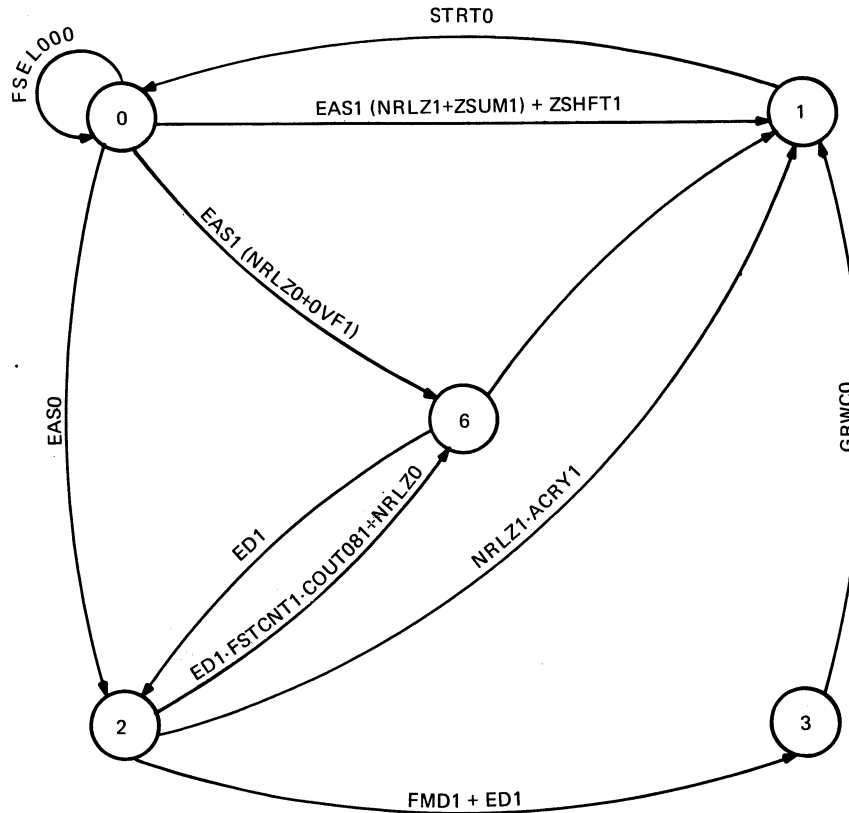


Figure 19. ALU State Transitions

7.2 ALU Algorithms

1. Simple Functions (FSEL000)

These arithmetic/logical functions do not require that the ALU clock be generated; they employ combinational logic to perform the required function. When one of these functions is to be performed, ALSTRT1 is used to gate MFINO back to the CPU, relying on the basic CPU clock frequency to allow the operation to be performed by the basic arithmetic/logical elements and the result to appear on the S Bus before being strobed into the destination register. Shown in Figure 20 is a timing diagram for the immediate response functions with respect to the CPU clock. Note that STRT0 is precisely the width of one CPU clock cycle or approximately 120 nanoseconds. Upon receipt of STRT0, the ALU immediately (≈ 15 nanoseconds) returns MFINO to prevent inhibiting the CPU clock. As the CPU latches the S Bus on the leading edge of the active clock, it is necessary for the ALU to complete its function and present the results to the S Bus within 70 nanoseconds. On the trailing edge of the active clock, the CPU switches control states and releases STRT0. It is during this next control state that the CPU writes the result into the destination register.

TABLE 16. STATE REGISTER LOGIC

TRANSITION	ASA LOGIC	ASB LOGIC	ASC LOGIC	COMMENT
AS001 TO AS011	$J = ZSHFT1 + EAS1 \cdot COUT080 \cdot (NRLZ1 + ZSUM1)$			ABORT SHIFT IF SHIFT COUNT IS ZERO. FLT. PT. ADD/SUB. COMPLETE IF NO MANTISSA OVERFLOW AND RESULT IS NORMALIZED OR ZERO.
AS001 TO AS021		$J = EAS0 \cdot AS001$		UNCONDITIONAL TRANSFER IF NOT FLT. PT. ADD/SUB.
AS001 TO AS061		$J = EAS1 \cdot (COUT081 + NRLZ0 \cdot ZSUM0)$	$J = EAS1 \cdot (COU081 + NRLZ0 \cdot ZSUM0)$	FLT. PT. ADD/SUB. RESULTS IN MANTISSA OVERFLOW OR UN-NORMALIZED MANTISSA
AS021 TO AS011	$J = ACRY1 \cdot NRLZ1$	$K = ACRY1 \cdot NRLZ1$	$J = ACRY1 \cdot NRLZ0$ $K = NRLZ1$	SHIFT COMPLETE OR FLT. PT. MULT. COMPLETE AND NORMALIZED.
AS021 TO AS031	$J = ACRY1 \cdot NRLZ1$	$K = FMD0 + ED0$		FIX PT. MULT./DIV. COMPLETE. FLT. DIVIDE COMPLETE.
AS021 TO AS061		$K = ACRY1 \cdot NRLZ1$	$J = NRLZ0 \cdot ACRY1 + ED1 \cdot COUT081 \cdot FSTCNT1$ $K = NRLZ1$	FLT. MULT. RESULTS IN UN-NORMALIZED MANTISSA. FIRST ITERATION OF FLT. DIVIDE REVEALS DIVIDEND LESS THAN DIVISOR.
AS031 TO AS011		$K = GRWC0 \cdot AS031$		FIX MULT/DIVIDE—FIRST HALF OF RESULT WRITTEN INTO DESTINATION REGISTER. FLT. DIVIDE – MQ TRANSFERRED TO AL.
AS061 TO AS021			$J = ED1 \cdot FSTCNT1$	FLT. DIVIDE – DIVIDEND HAS BEEN MADE SMALLER THAN DIVISOR. CONTINUE DIVIDE.
AS061 TO AS011	$J = AS061 \cdot NRLZ1$		$K = NRLZ1$	FLT. PT. RESULT HAS BEEN NORMALIZED.
AS011 TO AS001	RESET = STRT1	RESET = STRT1	RESET = STRT1	RESET TO AS001 WHEN CPU REMOVES STRT1.

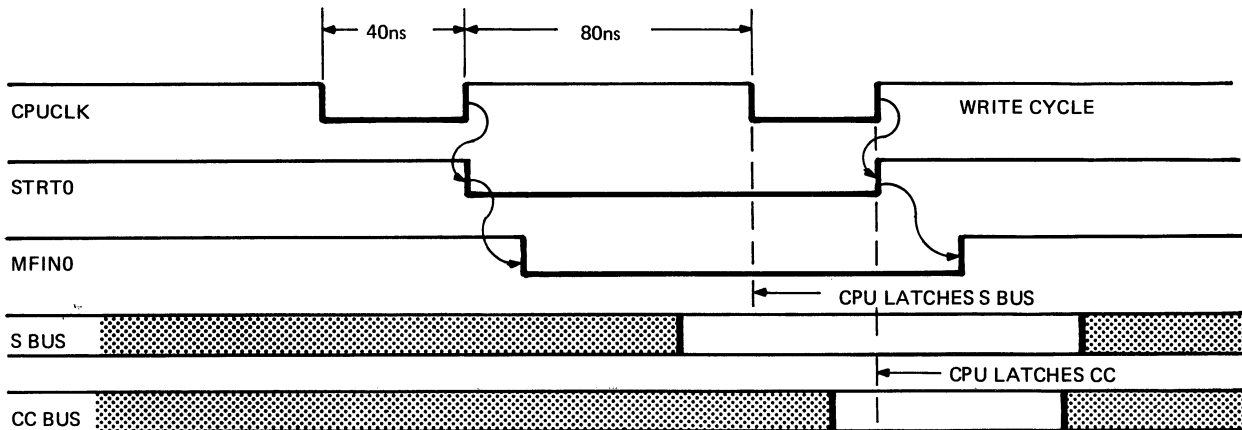


Figure 20. ALU Bus Timing – Immediate Response Functions (FSEL000)

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The fixed point simple functions are Subtract, Add, Subtract With Carry (SWC), Add With Carry (AWC), logical AND, logical Exclusive OR, and logical OR. When the instruction to be performed is an Add/Subtract, the carry state into the arithmetic element must be generated. This is done by the gate (9B8) whose output is labeled CIN310. Since the 19-067 device performs a subtraction by internal 1's complement addition, a carry must be generated for Subtract. Similarly, a carry in is generated for Add With Carry (A+B+1) and suppressed for Subtract With Carry (A-B-1).

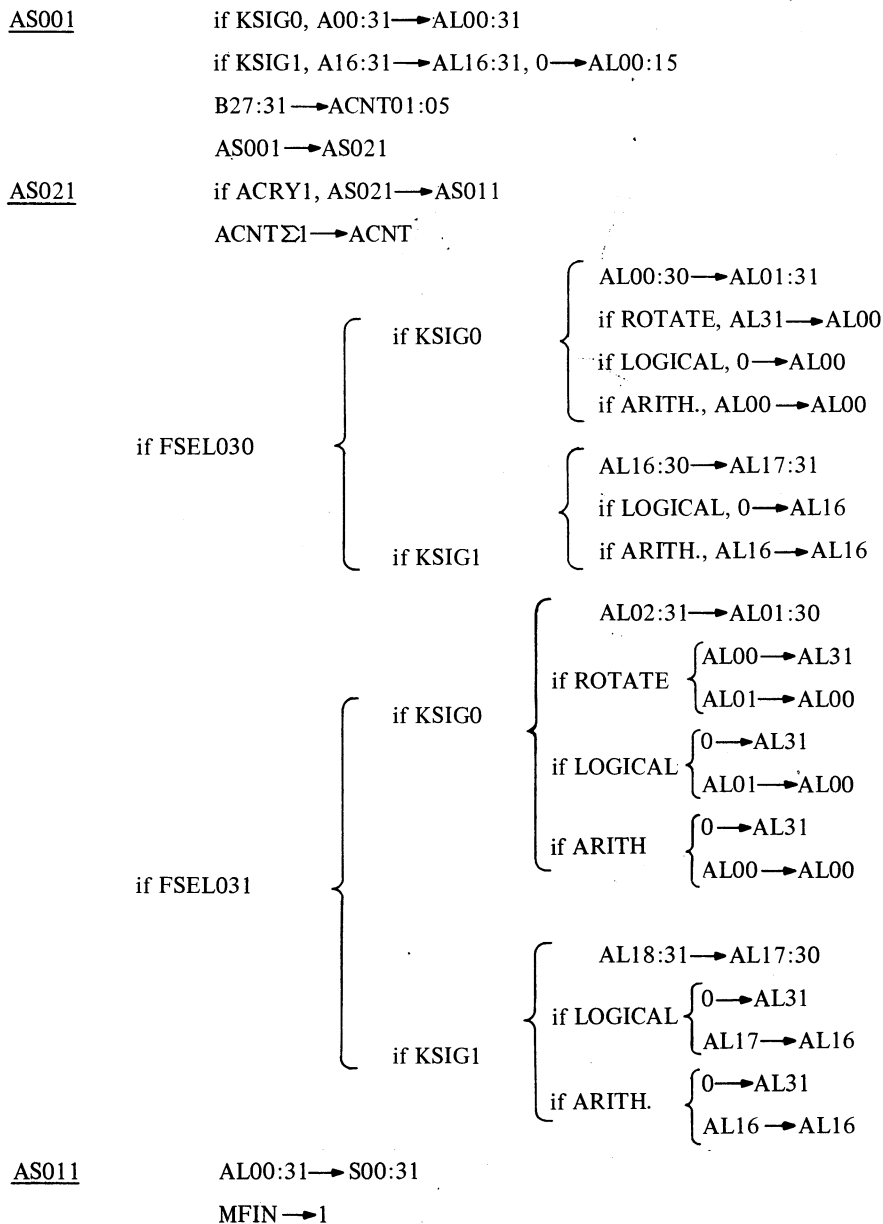
The floating point simple functions are Load (ELD), Subtract With Carry (ESWC), Add With Carry (EAWC), and Compare (EC). These instructions are floating point instructions only in the sense that they manipulate floating point data. The hardware implementation is identical for that of the fixed point instructions and none of the exponent hardware is used.

2. Complex Functions (FSEL001)

Shift Instructions

The ALU can perform both halfword (16-bits) and fullword (32-bit) shifts. When a shift is to be performed, the word to be shifted is taken from the A Bus and the shift count is taken from the B Bus (B27:31). In AS001, the A Bus is transferred to the AL register. If a halfword shift is to be performed, KSIG1 is set, and the most significant 16-bits of the AL register are inhibited by killing the clock to those devices. The shift occurs in AS021, AL being continuously loaded from the proper shift multiplexor. When the correct number of shifts have occurred, ACRY1 forces the transition to AS011. During AS011, the contents of the AL register are transferred to the S Bus, the flags are generated onto the CC Bus, and MFIN is returned to the CPU.

The Shift Algorithms are:



Fixed Point Multiply

The ALU performs signed multiplication on two 32-bit operands. The multiplier is transferred from the A Bus to the MQ register during AS001 and the multiplicand remains on the B Bus throughout the operation.

If the multiplier is positive, a product is formed by adding the multiplicand to the shifted product conditional upon the multiplier. If the multiplier is negative, the product is formed by subtracting the multiplicand from the shifted partial product conditional upon the 2's complement of the multiplier.

Multiplication is accomplished by examining each successive bit of the multiplier as it is shifted out of the MQ register. A 32-bit product is formed by shifting either AL (the partial product) or S (=ALΣB) back into AL and into the MQ conditional upon M1, the multiplier bit. The logic gate for M1 is located at 7N6. If SUM1 is set (multiplier positive), M1 is simply MQ31. If SUM0 is set (multiplier negative) M1 becomes the 2's complement of the multiplier. The 2's complement of the multiplier is taken by detecting the first MQ bit which is set and thereafter complementing the remaining MQ bits. This is accomplished by the complement flip-flop (7J7) and the M1 logic gate.

Inherent in the Multiply algorithm is a look ahead feature which permits deciding in advance whether to add and shift, or just shift the partial product. If M1 is set, AL is added to B and the sum is shifted into the AL and MQ registers. If M1 is reset, the AL is shifted back into the AL and MQ registers. Since more time is required to perform both an add and shift (M1) the ALU clock is divided by two when M1 is set and permitted to run at its basic speed when not set (see Section 7.3 ALU Clock).

When the Multiply is completed, the most significant portion of the result is written into the destination register in AS031 and MQ is transferred to the AL. In AS011, the least significant portion is written into the destination register.

The algorithm for Multiply is:

```

AS001      A00:31 → MQ00:31
              0 → AL00:31
              0 → ACNT01:05
              if A000, 1 → SUM1
              if A001, 0 → SUM0
              AS001 → AS021

AS021      if ACRY1, AS021 → AS031
              GRWC → 0
              ACNT Σ1 → ACNT
              if SUM1, ALΣB
              if SUM0, ALΔB
              if M1 {
                  S00:S30 → AL01:31
                  (A00 ⊕ B00) · ZSUM0 → AL00
                  MQ00:30 → MQ01:31
                  S31 → MQ00
              }
              if M0 {
                  AL00:30 → AL01:31
                  AL00 → AL00
                  MQ00:30 → MQ01:31
                  AL31 → MQ00
              }

AS031      AL00:31 → S00:31
              MQ00:31 → AL00:31
              if GRWC0, AS031 → AS011

AS011      AL00:31 → S00:31
              MFIN → 1
    
```

Fixed Point Divide

The Fixed Point Divide algorithm is implemented by subtracting the divisor from the shifted dividend to determine if it is greater or not. If the dividend is determined to be smaller than the divisor, the quotient digit for that test is made to be a zero, and the dividend is shifted left again to repeat the process. If the dividend is determined to be larger, the quotient digit for that test is made to be a one and the difference, shifted left, is stored as the new dividend. In the implementation of signed divide, if the two operands are of unlike signs, the subtraction is performed by the addition of the unlike operands and the 1's complement of the quotient is accumulated. When the complemented quotient is formed, it is corrected to the 2's complement in AS011.

An obstacle in performing signed division using complementary arithmetic arises when the intermediate dividend is a negative number and both the intermediate quantities (the absolute value of the dividend - divisor) and the remainder equal zero because the logic does not detect the quotient digit of 'one'. When this case arises, the computed result = true quotient -1, with the remainder equal to the divisor. To detect this case, a flip-flop (RZR0) (7F1) monitors this condition and causes a correction cycle in AS011.

Because of the difference in scaling of the divisor (2^{63}) and the dividend (2^{31}) and the fact that both the quotient and remainder must be scaled (2^{31}), an extra division cycle is performed in AS031 to compute Q31. To properly scale the remainder, the last summation is inhibited from shifting. Moreover, if the absolute value of the Q31 digit is '1', the correct remainder is on the S Bus during the first cycle of AS031 and remains there throughout AS031. Should the absolute value of the Q31 digit be '0', the correct remainder is in the AL register, and ALU control is modified to force the transfer of AL to the S Bus.

The least significant portion (remainder) of the result is written into the destination register in AS031, and the most significant portion (quotient) is written into its destination register in AS011.

The algorithm for Fixed Point Divide is:

```

AS001      A00:31→MQ00:31
           GRWC→0
           if A00⊖B00, SUM0→1
           if A00⊕B00, SUM1→1
           0→ACNT01:05
           0→RZR01
           A00:31→AL00:31
           AS001→AS021
AS021      if (FSTCNT1 and |Q|=1), DFLT→1
           if ACRY1, AS021→AS031
           ACNT Σ1→ACNT
           if SUM0, (ALΔB)
           if SUM1, (ALΣB)
           if COUT000⊖BG001 {
               1→MQ31
               MQ01:31→MQ00:30
               MQ00→AL31
               AL01:31→AL00:30
           }
           if COUT000⊕BG001 {
               0→MQ31
               MQ01:31→MQ00:30
               MQ00→AL31
               S01:31→AL00:30
           }

```

AS031 if SUM0, AL Δ B
if SUM1, AL Σ B

if COUT000 $\bar{\oplus}$ BG001 $\left\{ \begin{array}{l} 1 \rightarrow \text{MQ31} \\ \text{MQ01:31} \rightarrow \text{MQ00:30} \end{array} \right.$
if COUT000 \oplus BG001 $\left\{ \begin{array}{l} 0 \rightarrow \text{MQ31} \\ \text{MQ01:31} \rightarrow \text{MQ00:30} \end{array} \right.$

if (MQ31 \oplus AXB1)·SUM0, (AL Δ B) \rightarrow S00:31
if (MQ31 \oplus AXB1)·SUM1, (AL Σ B) \rightarrow S00:31
if MQ31 $\bar{\oplus}$ AXB1, AL00:31 \rightarrow S00:31
MQ00:31 \rightarrow AL00:31
if GRWC0, AS031 \rightarrow AS011

AS011 AL00:31 \rightarrow S00:31
MFIN \rightarrow 1

Compare and Equalize

The Compare and Equalize instruction is always performed prior to a floating point Add/Subtract. The instruction effectively aligns the exponents of the two operands by shifting the mantissa of the smaller operand.

To simplify the logic for determining the larger operand, B00 is inhibited (forced to a one) during AS001. The difference of the two operands is taken (A-B), and the BGTR flip-flop (6C2) is loaded with the information (SBGTR1) determining the larger operand. The logic for this determination (3D3) is:

$$\text{SBGTR1} = \text{A001} \oplus \text{B001} \oplus \text{S001}$$

but since B001 = 1,

$$\text{SBGTR1} = \text{A000} \oplus \text{S001}$$

If the BGTR flip-flop is set, B is the larger operand and A is shifted, or if BGTR is reset, A is the larger operand and B is shifted. The exponential difference is computed simultaneously and this result becomes the hexadecimal shift count. However, if this shift count exceeds 510, the operation is abandoned as significance is shifted out of the mantissa, the result being zero. The four bit magnitude comparator (8J2) compares the exponent difference to 510 and X0VF1 (8N4) determines if the shift count is less than 510. One additional problem occurs if the exponent of B is greater than the exponent of A. The difference results in a 2's complement number and does not reflect a true shift count. Should this occur, OCMPI (8D7) is active and complements the difference and inhibits ACNT for one shift cycle in AS021, yielding the correct number of hexadecimal shifts.

During AS021, either the A or B Bus is inhibited (forced to all ones) and a subtraction is performed. The net result is to transfer the operand which is to be shifted into the AL register. Thereafter, the operand is shifted hexadecimally to the right according to the shift count. When the shift is complete (ACRY1), the transition is made to AS011 where the result is gated to the S Bus with the sign and exponent field zero filled. When the Add/Subtract instruction follows, the CPU always gates the larger operand onto the A Bus and the shifted operand to the B Bus.

The algorithm for Floating Point Compare and Equalize:

AS001 SUM0 \rightarrow 1
BG001 \rightarrow 1
A Δ B
A01:07 Δ B01:07
FXS05:07 \rightarrow ACNT05:07
if X0VF0, AS001 \rightarrow AS021
if X0VF1, AS001 \rightarrow AS011

```

AS021      if ACRY1, AS021→AS011
           if FSTCNT1 {
                   if OCMP1, ACNT→ACNT
                   if OCMP0, ACNT Σ1→ACNT
                   if BGTR1, A00:31→AL00:31
                   if BGTR0, B00:31→AL00:31
           }
           if FSTCNT0 {
                   ACNT Σ1→ACNT
                   0→AL08:11
                   AL08:27→AL12:31
           }

AS011      if XOVF1, 0→S00:31
           if XOVF0, 0→S00:07, AL08:31→S08:31
           MFIN→1

```

Floating Point Add/Floating Point Subtract

One additional characteristic of floating point arithmetic beyond that discussed in the Compare and Equalize algorithm arises from floating point notation. The mantissa is represented by sign and magnitude. Positive numbers have a Sign bit equal to zero and negative numbers have a Sign bit equal to one. However, unlike fixed point notation, negative numbers are not represented in 2's complement format. Therefore, when performing an addition with unlike signs, a subtraction must be performed to obtain the true sum. Similarly, when performing a subtraction with unlike signs, to obtain a true difference an addition must be performed. This is accomplished by the FAXB1 address bit to the ALU ROM. The FAXB1 flip-flop (9N6) is set during AS021 of the Compare and Equalize instruction and the logic for this bit is $A001 \oplus B001$.

In AS001, the mantissa's of A and B are added/subtracted and the exponent of A is presented to the exponent up/down counters. If adding (SUM1), it is possible to overflow the resultant mantissa (COUT081) and a correction cycle is executed in AS061. If subtracting (SUM0), it is possible that the result may not be normalized and a normalize shift is executed in AS061. Should neither of these conditions arise, the transition to AS011 is direct and the result is gated to the CPU.

The algorithm for Floating Point Add is:

```

AS001      if FAXB0, SUM1→1, AΣB
           if FAXB1, SUM0→1, AΔB
           A01:07→XS01:07
           S00:31→AL00:31
           if NRLZ0·ZSUM0 + COUT081·SUM1, AS001→AS061
           if (NRLZ1+ZSUM1)·SUM1·COUT080, AS001→AS011

AS061      if SUM0 {
           0→AL28:31
           AL12:31→AL08:27
           }
           if SUM1 {
           0→AL08:10
           1→AL11
           AL08:27→AL12:28
           }
           if NRLZ1, AS061→AS011

AS011      if ZSUM1, 0→S00:31
           if ZSUM0 {
           AL08:31→S08:31
           XS01:07→S01:07
           A00→S00
           }
           MFIN→1

```

The algorithm for Floating Point Subtract is:

```

AS001          if FAXB0, SUM0→1, AΔB
                  if FAXB1, SUM1→1, AΣB

                  S00:31→AL00:31
                  A01:07→XS01:07
                  if NRLZ0·ZSUM0+COUT081·SUM1, AS001→AS061
                  if (NRLZ1+ZSUM1)·SUM1·COUT080, AS001→AS011

AS061          if SUM0 { 0→AL28:31
                          AL12:31→AL08:27
                          }
                  if SUM1 { 0→AL08:10
                          1→AL11
                          AL08:27→AL12:31
                          }
                  if NRLZ1, AS061→AS011

AS011          if ZSUM1, 0→S00:31

                  if ZSUM0 { A00→500
                          XS01:07→S01:07
                          AL08:31→S08:31
                          }
                  MFIN→1
  
```

Floating Point Multiply

The Floating Point Multiply, with minor differences, is implemented the same way as the fixed point multiply. The 24-bit mantissas are multiplied by adding and shifting to obtain a resultant 24-bit fraction (the least significant part of the result is discarded). The exponents are added to obtain the resultant exponent and the logic for the sign bit is A001 + B001.

At the completion of the iterative cycles, it is possible for the resultant mantissa to be unnormalized. If this is the case, a transition is made to AS061 where a normalize shift occurs. The results are gated back to the CPU in AS011.

Exponent underflow/overflow may occur as a result of adding the exponents or as a result of the normalize shift (the exponent is decremented for each hexadecimal digit shifted). Should exponent Overflow occur (OFL1), the V flag is set and if Underflow occurs (UFL1) the V flag is set and G and L flags reset.

The algorithm for Floating Point Multiply is:

```

AS001          SUM1→1
                  A00:31→MQ00:31
                  0→AL00:31
                  710→ACNT01:05
                  A01:07ΣB01:07→XS01:07
                  AS001→AS021

AS021          if ACRY1·NRLZ1, AS021→AS011
                  if ACRY1·NRLZ0, AS021→AS061
                  ACNTΣ1→ACNT
                  ALΣB
  
```

		{	S07:30→AL08:31
	if M1	{	(A00 + B00)·ZSUM0→AL00
		{	MQ00:30→MQ01:31
		{	S31→MQ00
		{	AL08:30→AL09:31
	if M0	{	0→AL08
		{	MQ00:30→MQ01:31
		{	AL31→MQ00
<u>AS061</u>			AL12:31→AL08:27
			0→AL28:31
			(XSΔ1)→XS
			if NRLZ1, AS061→AS011
<u>AS011</u>		{	AL08:31→S08:31
	if UFLO	{	XS01:07→S01:07
		{	A00⊕B00→S00
	if UFL1, 0→S00:31		
	1→MFIN		

Floating Point Divide

Floating Point Divide is implemented by continuously subtracting the mantissa of B from the mantissa of the shifted partial remainder (AL) to ascertain which is the larger. If the partial remainder proves to be the larger, the quotient digit (Q31) is set to a one and the left shifted difference is taken as the new partial remainder. If the partial remainder is less than the divisor, the partial remainder is shifted left and the cycle is repeated.

Since the mantissas are true magnitude, the larger mantissa is readily detected by COUT081. However, if significance is shifted out of the nth partial remainder, the n+1 partial remainder is, by definition, larger than the divisor. Therefore, the true logic for the quotient digit is:

$$Q311 = COUT081 + AL071$$

where AL071 detects a one being shifted out of the partial remainder.

The exponent result is obtained by subtracting the divisor exponent (B01:07) from the dividend exponent (A01:07) and the sign bit is derived from A001 ⊕ B001.

On the first divide cycle, if COUT081 is detected, it is necessary to execute a correction cycle to be able to represent the results in 24-bits plus the sign. The mantissa (of the partial remainder) is shifted right one hexadecimal digit and the exponent is incremented. On the next clock, the ALU returns to AS021 and continues the divide iterations.

When the divide has been completed (ACRY1), the ALU goes to AS031, and gates MQ to the AL register and then goes to AS011.

The algorithm for Floating Point Divide is:

<u>AS001</u>	SUM0→1
	A00:31→AL00:31
	A01:07ΔB01:07→XS01:07
	7 ₁₀ →ACNT
	AS001→AS021
<u>AS021</u>	if ACRY1, AS021→AS011
	if FSTCNT1·Q311, AS021→AS061
	ACNTΣ1→ACNT
	ALΔB

if Q311	{	1→MQ31 MQ01:31→MQ00:30 0→AL31 AL01:31→AL00:30
if Q310	{	0→MQ31 MQ01:31→MQ00:30 0→AL31 AL01:31→AL00:30

<u>AS061</u>	XS01:07Σ1→XS01:07 0→AL08:10 AL07→AL11 AL08:27→AL12:31 AS061→AS021
<u>AS031</u>	MQ00:31→AL00:31 AS031→AS011
<u>AS011</u>	AL08:31→S08:31 XS01:07→S01:07 A00 BB 00→S00 MFIN→1

7.3 ALU Clock (Sheet 6)

The ALU clock is a gated oscillator whose basic frequency is determined by a tapped delay line. The basic clock period is factory adjusted for 60 nanoseconds. The clock is enabled for all complex operations (FSEL001). It may be inhibited for test purposes by removing the ground strap from Connector 7. With the ground strap removed, the ALU may be single stepped through an instruction using the push button switch located at 16R. The enable logic at gate 13A06 is true when the conditions ALSTRT1·FSEL001·AS010 are met. This results in a clock output at MQCLK0, ACLK1, ACLK0, ALCLKA0, and ALCLKB0. The clock is inhibited during AS011 and the ALU becomes static until the CPU removes **STRT**.

For shifts, the basic 60 nanosecond clock provides the ALU with enough time to perform the required operations. However, for Multiply/Divide and other special cases, 60 nanoseconds is insufficient time to perform the required operation. For example, when dividing, both a subtraction and a shift must be performed in each iterative cycle. For these cases, the ALU clock period is doubled. The disable logic at gate 13B08 determines if the 120 nanosecond clock is to be generated. The cases when this is true are as follows:

1. Always in AS031 to allow the CPU to write the first word of a double precision result (Multiply/Divide) into the destination register.
2. Always when dividing.
3. In CAE on the first count of AS021 to permit the transfer of either operand (A or B) through the ALU chips to the AL. (see CAE algorithm).
4. When multiplying if the M1 bit is set. If M1 is set, both an add and shift are performed. If M1 is reset, only a shift need be performed.

7.4 Arithmetic Iterative Counter (ACNT01:05) (Sheet 4)

The iterative shifting of the ALU is controlled by a modulo 32 counter which is enabled during AS021. It is in AS021 that the iterative operations of shift and Multiply/Divide occur and this counter is used to determine completion of the operation and, therefore, the time at which transition to the next arithmetic state should occur. The counter is used in four different modes as described below.

1. Shifts.

For these operations, the counter is loaded with the 1's complement of the shift count which is taken from the B Bus and loaded at the transition from AS001 to AS021. The most significant bit is inhibited for halfword shifts if KSIG is set. The transition from AS021 is enabled when the counter reaches a count of 30, as determined by the logic signal ACRY1 (gate 09D06). Should a shift count of zero occur, this is detected in the Arithmetic State Register and the operation is aborted by inhibiting the transition to AS021.

2. Fixed Point Multiply/Divide

For these two instructions, the counter is initialized to a count of zero and is enabled to count during AS021. When the counter reaches a count of 31₁₀, the transition from AS021 to AS031 is enabled by ACRY1.

3. Floating Point Multiply/Divide

For these instructions, the counter is loaded with a count of 7₁₀ to allow for seven fewer iterations (the size of the exponent field). For floating point Multiply, when the counter reaches a count of 30₁₀ the transition is made from AS021 to AS011 if the result is normalized, or to AS061 if the result is not normalized. For floating point Divide, when the counter reaches a count of 31₁₀, the transition from AS021 to AS031 is made.

4. Compare and Equalize

This instruction is always performed prior to execution of a floating point Add/Subtract instruction. It is used to align the exponents of the two operands by shifting the mantissa of the lesser operand. The smaller operand is shifted right hexadecimally an amount determined by the difference of the two exponents. The shift count, therefore, is loaded from FXS051:071 which is the difference of the exponents. It is possible that this difference may result in a 2's complement number. Should this occur, the 1's complement of this result is loaded into the counter, and the counter is inhibited on the first count of AS021. This is accomplished by the logic at gate 09C03. If this difference should result in a shift count greater than 5₁₀, the operation is aborted since this would result in shifting significance out of the mantissa. Should this occur, a signal called XOVF1 forces a shift count of zero and the operation is aborted. When the shift is complete, the counter reaches a count of 31₁₀, and ACRY1 forces the transition from AS021 to AS011.

7.5 Arithmetic Condition Code (Sheets 2 and 3)

The ALU gates appropriate Condition Code flags to the CPU for all ALU functions. When the ALU senses its address and receives a start (STRT) it signals the CPU with SCC0 that a new Condition Code is available. Figure 18 (ALU Functional Block Diagram) shows that the ALU Condition Code circuits consist of combinational logic which determines the resultant condition of each instruction. These are latched in a register. The clock which latches the Condition Code is gated in one of two ways. For the simple functions (FSEL000), the clock results from STRT, delayed an appropriate amount of time to allow the ALU to complete its function. For complex functions (FSEL001), the clock is generated in AS011 at the conclusion of an instruction. The Condition Code is then gated onto the bus through a tri-state multiplexor. The representation of each flag is as follows.

1. VCC0 (Arithmetic Overflow).

The logic for this flag is shown on Sheet 3 (SVCC0). It is enabled for fixed point Add, Subtract, and Divide; and floating point Add, Subtract, Compare and Equalize, Multiply, and Divide. The flag is active for fixed-point Add/Subtract instructions when an overflow is determined by the logic:

$$ASIGN0 \cdot S001 \cdot (BG001 + SUM1) \oplus ASIGN1 \cdot S000 \cdot (BG001 \oplus SUM1).$$

The V flag is active for fixed point Divide on the first iteration of the Divide if the quotient bit is determined to be a one. This condition is called a Divide Fault (DFLT) and indicates that the result cannot be contained in 31 bits plus sign. The V flag also sets for fixed point Divide at the end of the divide algorithm if the calculated sign of the quotient is incorrect. For floating point instructions, all mantissa overflow is correctable by shifting the mantissa and adjusting the exponent. Therefore, floating point overflow/underflow is a function of exponent arithmetic alone. The V flag is set for the following conditions:

$$\begin{aligned} OFL1 &= AM011 \cdot (BG011 \oplus EM1) \cdot FXS011 \oplus SOFL1 \\ OFL1 &= CAE1 \cdot XOVF1 + AM010 \cdot (BG011 \oplus EM1) \cdot FXS010 \oplus SUFL1 \end{aligned}$$

2. CCCO (Carry).

The logic for this flag is shown on Sheet 2. It is enabled for fixed point Add/Subtract, Shifts, and Divide; and floating point Compare, and Compare and Equalize. For fixed point Add/Subtract the logic is $SUM1\ COUT001 + SUM0\ COUT000$ respectively. For floating point Compare, the logic is essentially, SLCCO, and is used to signal the larger of the two numbers. The C flag is also active for fixed point Divide to signal a divide fault and for floating point Compare and Equalize to signal B as the greater operand (BGTR0). For Shift type instructions, the C flag is the state of the last bit to be shifted. This is selected by the eight to one (8/1) multiplexor whose select control lines are encoded to yield the proper bit for every type shift. The selected bit is then latched by the flip-flop shown at 2E2.

It should also be noted that the Module Signal (MSIG0) from the ALU is identical to CCCO.

3. LCC (Less Than Zero)

The L flag represents the sign of any arithmetic operation. For fullword fixed point operations it is the sign of the result and for halfword shifts it is the sign of Bit 16 (S161). For floating point operations, it is the sign of the floating point result except where exponent underflow occurs or if the floating point result is zero. For these cases the L flag is forced to the inactive state. For floating point compare the logic $[(A001 + B001) \oplus COUT000] \cdot ZSUM0$.

4. GCCO (Greater Than Zero)

This flag logically represents the occurrence of not less than zero and results not equal to zero and not exponential underflow. This can be logically represented as follows:

$$GCCO = LCCO \cdot ZSUM0 \cdot UFLO$$

7.6 Arithmetic Elements and ROM Control (Sheets 9:17)

The heart of the ALU is built from the four bit arithmetic/logical elements (INTERDATA Part Number 19-067) and a format ROM used to control them. Also used in conjunction with the ALU chips are a two level carry-look ahead scheme (INTERDATA Part Number 19-068).

As previously stated, the ALU is essentially controlled by a 256X4 bit ROM. FSEL001:031 and MSEL011 address the ROM and determine the required control for the given instruction. ASIGN1, BSIGN1, and FAXB1 provide needed additional information to insure correct control for fixed point Multiply/Divide and for floating point Add/Subtract. Shown in Table 17 is a listing of ALU control and the respective operations as a function of the address bits. One additional control bit (ALOG1) is required to correctly specify logical operations from arithmetic operations. The logic for this gate (12L2) is:

$$MSEL010 \cdot FSEL000 \cdot FSEL011$$

and essentially decodes the logical operations as per the FSEL field.

There are two levels of gating beyond the ROM outputs on AMOD00:03. These are to provide two basic override functions. The first is included for the Compare and Equalize instruction. The ROM is coded for a Subtract to obtain the difference of A and B. However, once we determine which is smaller, we wish to load the smaller mantissa into the A latch where it can be shifted. The bus which is not to be shifted is inhibited (forced to all one's) and the transfer into the A latch is accomplished by forcing a carry in and modifying AMOD00:03 to perform an addition. The gate which provides this over-ride to ROM control is located at 9H4.

The second override function provided is to transfer A to S. This is accomplished by XFR0 (12M5). The cases for which this is necessary are as follows.

1. In AS001 for shifts and Multiply/Divide (fixed and float) to transfer operand from the A Bus to the A latch.
2. In AS011 to transfer contents of A latch to the S Bus.
3. In AS031 in fixed point multiply to transfer contents of A latch to the S Bus.
4. In AS031 of divide (fixed and float) to transfer contents of A latch to S Bus under certain conditions (see divide algorithms).

7.7 MQ Register (Sheets 10:17)

The Multiplier Quotient Register is used exclusively in Multiply/Divide instructions. It is comprised of eight MSI four bit shift registers which are capable of shifting left or right.

Control for the MQ registers is located at 7N4. The A Bus is always loaded into MQ in AS001 by forcing both SR1 and SL1 high. This is accomplished by clearing the Control flip-flop (7K4) with STRT1. For multiply, SR1 is active to perform right shifts and SL1 is inactive. The opposite is true for divide when shifts left are performed.

TABLE 17. ALU ROM CONTROL

FAXB1	BSIGN1	ASIGN1	MSEL011	FSEL001	FSEL011	FSEL021	FSEL031	AMOD031	AMOD021	AMOD011	AMOD001	FUNCTION	COMMENTS
X	X	X	0	0	0	0	0	0	1	1	0	SUM0	SUBTRACT
X	X	X	0	0	0	0	1	1	0	0	1	SUM1	ADD
X	X	X	0	0	0	1	0	0	1	1	0	SUM0	SUB. WITH CARRY
X	X	X	0	0	0	1	1	1	0	0	1	SUM1	ADD WITH CARRY
X	X	X	0	0	1	0	1	1	1	0	1	A·B	LOGICAL AND
X	X	X	0	0	1	1	0	0	1	1	0	$A \oplus B$	LOGICAL EX. OR
X	X	X	0	0	1	1	1	0	1	1	1	A + B	LOGICAL OR
X	X	X	0	1	0	0	0	0	0	0	0	A	LOG. SHIFT RIGHT
X	X	X	0	1	0	0	1	0	0	0	0	A	LOG. SHIFT LEFT
X	X	X	0	1	0	1	0	0	0	0	0	A	ROTATE RIGHT
X	X	X	0	1	0	1	1	0	0	0	0	A	ROTATE LEFT
X	X	X	0	1	1	0	0	0	0	0	0	A	ARITH. SHIFT RIGHT
X	X	X	0	1	1	0	1	0	0	0	0	A	ARITH. SHIFT LEFT
X	X	0	0	1	1	1	0	1	0	0	1	SUM1	MULT.: A POS.
X	X	1	0	1	1	1	0	0	1	1	0	SUM0	MULT.: A NEG.
X	0	0	0	1	1	1	1	0	1	1	0	SUM0	DIV.: SIGNS ALIKE
X	0	1	0	1	1	1	1	1	0	0	1	SUM1	DIV.: SIGNS DIFFER
X	1	0	0	1	1	1	1	1	0	0	1	SUM1	DIV.: SIGNS DIFFER
X	1	1	0	1	1	1	1	0	1	1	0	SUM0	DIV.: SIGNS ALIKE
X	X	X	1	0	0	0	1	1	0	0	1	SUM1	FLT. PT. LOAD
X	X	X	1	0	0	1	0	0	1	1	0	SUM0	FLT. PT. SUB. WITH CARRY
X	X	X	1	0	0	1	1	1	0	0	1	SUM1	FLT. PT. ADD WITH CARRY
X	X	X	1	0	1	0	1	0	1	1	0	SUM0	COMPARE
0	X	X	1	1	0	0	0	0	1	1	0	SUM0	FLT. PT. SUB. – SIGNS ALIKE
1	X	X	1	1	0	0	0	1	0	0	1	SUM1	FLT. PT. SUB. – SIGNS DIFFER
0	X	X	1	1	0	0	1	1	0	0	1	SUM1	FLT. PT. ADD – SIGNS ALIKE
1	X	X	1	1	0	0	1	0	1	1	0	SUM0	FLT. PT. ADD – SIGNS DIFFER
X	X	X	1	1	1	0	0	0	1	1	0	SUM0	COMPARE AND EQUALIZE
X	X	X	1	1	1	1	0	1	0	0	1	SUM1	FLT. PT. MULTIPLY
X	X	X	1	1	1	1	1	0	1	1	0	SUM0	FLT. PT. DIVIDE

NOTE: SUM1 = SUM, SUM0 = DIFFERENCE

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

7.8 AL Register and Shift Multiplexors (Sheets 10:17)

The AL registers are comprised of eight MSI quad D type flip-flops with double rail output. They are used in all complex functions (FSEL001) as a holding register for shift type operations.

Shifts are performed by enabling one of four multiplexors depending on the type of shift to be performed. The multiplexor outputs are OR - tied together and perform the following types of shifts.

1. n: has S001:311 and MQ001:311 as inputs. Does not shift; used for transferring MQ or A Bus to AL register.
2. n+1: has S011:311 and AL021:311 as inputs. Performs left one shifts for Shift instructions and Divide instructions. End points are determined by AGL001 and AGL311 (Sheet 8).
3. n-1: has S001:301 and AL001:301 as inputs. Performs right one shifts for Shift instructions and Multiply instructions. End points are determined by AGR001 and SGR001 (Sheet 8).
4. n+4: has AL121:311 and AL081:271 as inputs. Shifts left hexadecimally (n+4) to normalize and shift right hexadecimally (n-4) to correct overflow conditions or for Compare and Equalize instruction.

7.9 Exponent Arithmetic (Sheet 8)

Exponential arithmetic is accomplished through the use of two 19-067 4-bit arithmetic/logic elements. The exponent fields (Bits 01:07) of A and B are either added to or subtracted from each other, depending upon the instruction. The result is loaded into an up-down counter, where the exponent may be incremented or decremented as required by post-normalization or overflow correction.

As previously mentioned (Section 3.2.4), in the Compare and Equalize instruction, the exponent difference may result in a 2's complement number and it was necessary to take the 1's complement of this for use as the shift count. This is accomplished by the Exclusive-OR gates connected to the ALU chips and the control signal OCMP1. The 4-bit magnitude comparator is used to determine if the magnitude of the exponent difference is greater than 5₁₀. Should this be the case, the Compare and Equalize instruction is aborted since significance would be shifted out of the mantissa. XOVF1 detects this case.

INTERDATA uses excess 64 notation to express floating point numbers. As a result of an exponent addition or subtraction, the result becomes unbiased (i.e., the excess 64 is lost). To restore excess 64 notation to the exponent field in floating point Multiply/Divide, the most significant bit of the exponent field is complemented. This is accomplished by the Exclusive-OR gate whose logic is FXS011⊕EMD1.

8. I/O GENERAL DESCRIPTION

The 8/32 Input/Output Unit (IOU) performs a multiplicity of functions. Its main function is to communicate, via the Multiplexor Channel, with up to 1,023 peripheral devices. It has an additional capability of performing byte manipulations for the CPU both in conjunction with and without an I/O exchange. In addition, the I/O module contains the Machine Control Register (MCR), the Display Controller, the Teletype Controller, the Power Monitor and System Initialize circuit, and the Start Timer. Module Number 2 is assigned to the IOU module.

9. FUNCTIONAL DESCRIPTION

9.1 I/O Control Functions

IOU communicates with both the CPU (via the A, B, S, and CPU Control Busses) and peripheral controllers (via the Multiplexor Bus) using request/response signaling. Timing is completely asynchronous (rather than quantized) to achieve maximum speeds.

IOU becomes active when it recognizes its address (Module Number 2) on the Memory Select (MSEL) lines from CPU and STRT0 goes active. The function decode logic then directs the control and gating logic to perform a required function, encoded on Function Select lines (FSEL00:03) and KSIG. (Function Code Extension line). At the completion of the operation, a MFIN signal is sent back to the CPU.

IOU control is capable of performing 32 functions shown in Table 18.

Functions in Table 18 are divided into three distinct categories.

1. Multiplexor Channel Operations (D Bus Operations).
2. Byte Manipulations.
3. Auxiliary Functions.

TABLE 18. FUNCTION MNEMONICS

MNEMONIC	FUNCTION	COMMENT
RD/RDA WD/WDA OC/OCA SS/SSA	READ DATA WRITE DATA OUTPUT COMMAND SENSE STATUS	BYTE/INDEXED BYTE/INDEXED BYTE/INDEXED BYTE/INDEXED
RDR/RDRA WDR/WDRA OCR/OCRA SSR/SSRA ACK	READ DATA WRITE DATA OUTPUT COMMAND SENSE STATUS ACKNOWLEDGE INTERRUPT	BYTE/REGISTER BYTE/REGISTER BYTE/REGISTER BYTE/REGISTER BYTE/REGISTER
RDH/RDHA	READ DATA HALFWORD	TWO DATA CYCLES FOR BYTE CONTROLLERS
WDH/WDHA	WRITE DATA HALFWORD	TWO DATA CYCLES FOR BYTE CONTROLLERS
THW	TEST HALFWORD	MSIG GENERATION
STB LB STBR LBR	STORE BYTE LOAD BYTE STORE BYTE LOAD BYTE	HALFWORD/INDEXED BYTE/INDEXED HALFWORD/REGISTER BYTE/REGISTER
SMCR CMCR	SENSE MACHINE CONTROL REGISTER CLEAR MACHINE CONTROL REGISTER	
EXB LDWAIT	EXCHANGE B-BYTES LOAD WAIT FLIP-FLOP	B 16 to FWAIT
POW POUT	RELEASE INITIALIZE RELAY GATE OUTPUT PULSES	

9.1.1 Multiplexor Channel Operations. D Bus functions include: RD, RDR, WD, WDR, SS, SSR, OC, OCR, RDH, WDH, ACK, RDA, RDRA, WDA, WDRA, SSA, SSRA, OCA, OCRA, RDHA, and WDHA (see Table 18).

When the I/O control is addressed and given a D Bus function code, it creates a one, two, or three cycle Multiplexor Channel operation.

The device address on A Bus (22:31) (refer to 8/32 IOU Block Diagram Figure 21) is gated to D Bus (06:15) together with the activating ADRS Control line whenever address type functions are specified. The addressed controller responds by returning a SYN signal which terminates the address cycle and starts a data cycle.

Delay timing within each cycle insures that the relationship of the MUX Bus Control lines and D Bus signals meet the Multiplexor Channel timing requirements.

The halfword functions (RDH/WDH) have a single data cycle when the Halfword (HW) Test line is active (communicating with halfword-oriented controller). Two data cycles are required when HW is inactive (byte-oriented controller is being addressed).

Output data is gated from the B Bus to the D Bus via the D Bus tri-state multiplexors controlled by ROM.

Input data coming from the D Bus is latched in the D Bus receivers and then gated onto the S Bus through S Bus Multiplexors which are also ROM controlled. In a Sense Status operation, D Bus receiver Bits 12:15 are also gated to the Condition Code (CC) Bus via the CC Mux. During all other D Bus operations, four zeros are placed on the CC Bus together with CC Strobe (SSCO).

At the end of the operation, the I/O control returns a Module Finished (MFIN) signal to restart the CPU clock.

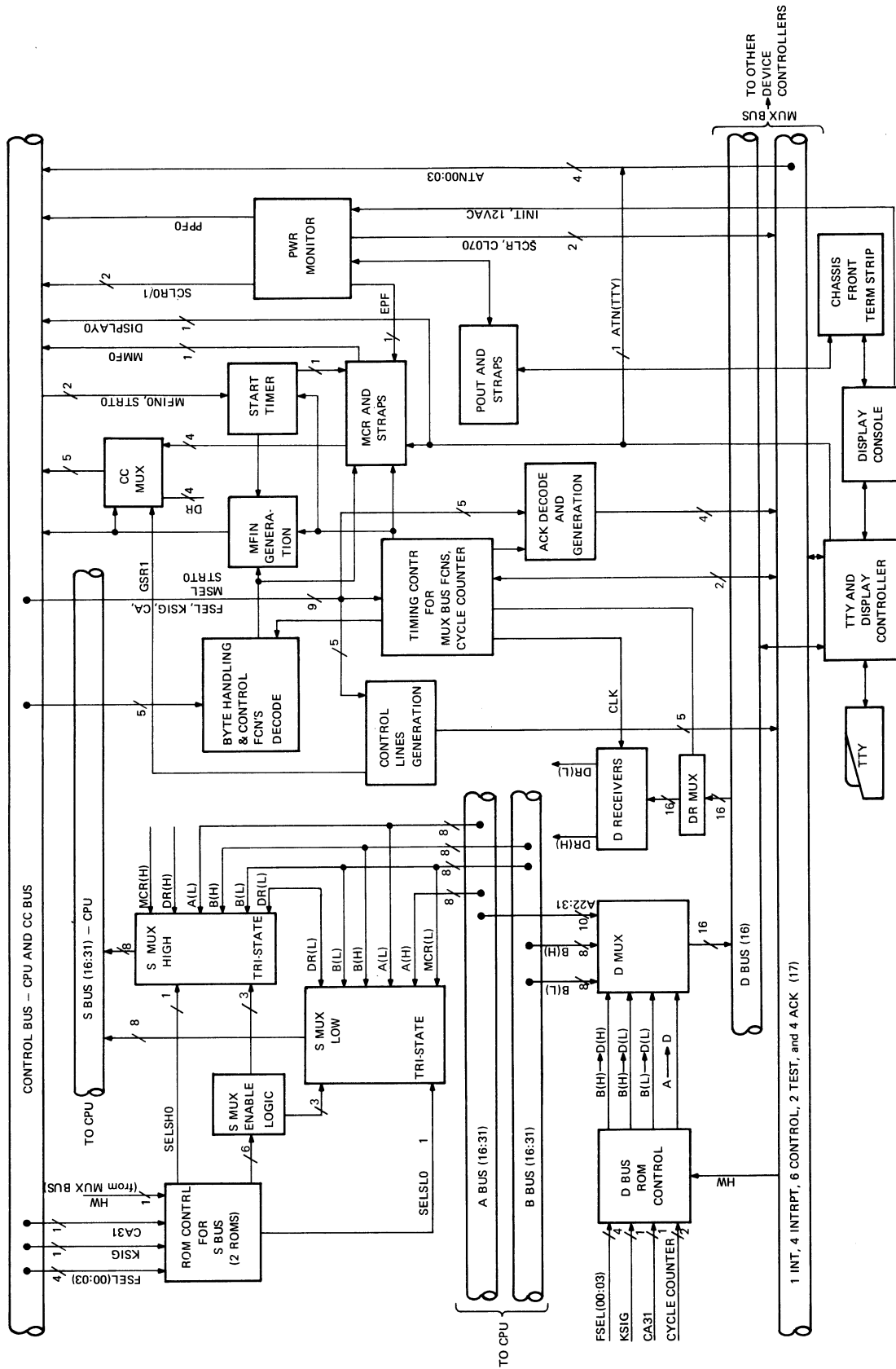


Figure 21. IOU Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The ACK function generates a single cycle on the D Bus during which it activates one of four ACK lines and gates 10-bits of interrupting device address from the D Bus onto the S Bus.

All Multiplexor Channel functions are covered in detail in Section 10.

9.1.2 Byte Manipulations Functions. These include: STBR, LBR, STB, LB, and EXB (see Table 18). These functions do not activate the D Bus since only selected bytes are gated from the CPU's A and B Buses back onto the CPU's S Bus. This is achieved by ROM controlled S Bus Multiplexors. See Section 11.1 for more detailed description of these functions.

9.1.3 Auxiliary Functions. These include: SMCR, CMCR, LDWAIT, THW, POW, and POUT (see Table 18).

The SMCR and CMCR functions provide a means for Sensing and Clearing the Machine Control Register (MCR).

The LDWAIT function controls the ON/OFF indicator light on the Display Console.

The THW function generates MSIG according to the state of the HW (Halfword) Test line.

The POW function releases the System Clear relay.

The POUT function gates 4-bits on the B Bus (27:31) to a set of board stakes for external signaling purposes.

All auxiliary functions are covered in more detail in Section 11.2.

9.2 Machine Malfunction and Power Fail Hardware

Space is also provided on the 8/32 IOU board for a Machine Control Register (MCR), which stores machine trouble conditions; a Power Monitor and System Initialize circuit, and a Start Timer. This additional IOU hardware is covered in Section 10.11.

10. MULTIPLEXOR CHANNEL (MUX) BUS

10.1 Multiplexor Channel IOU

The main function of the IOU board is to provide a means for communicating with up to 1,023 peripheral device controllers and interfaces, including Display Console and Teletypewriter. IOU accomplishes this by generating Multiplexor Channel D Bus from the CPU busses whenever it is addressed by Module Number 2 and D Bus operation is requested by the CPU.

This byte of halfword oriented input/output system consists of 33 lines:

- 16 Bi-directional Data lines (also used for address).
- 6 Control lines (to identify the contents of the data line).
- 2 Test lines.
- 4 Interrupt lines.
- 4 Acknowledge lines (daisy-chains).
- 1 System Initialize line.

The 4 Interrupt lines terminate on the CPU-B (CPB) board and the 4 Acknowledge lines originate on the IOU module. The Initialize line is available to all system modules, controllers, and the local memory. Only one Interrupt line and one acknowledge daisy-chain is provided on a given I/O back panel. A single instruction from the CPU contains the 10 bit device address, the encoded function and up to 16 bits of output data when needed. The MUX Bus generator provides single or multicycle operation to address the device controller, transmit the decoded function and send or receive over the 16 Bi-directional Data lines and synchronize the exchange. The normal byte or halfword operation consists of address cycle, followed by a data cycle. During a Read/Write Block sequence, the address cycle is not used. For halfword functions (e.g., RDH or WDH) with a byte oriented controller, two data cycles are used to transfer the halfword.

The following definitions apply to the lines in the MUX Bus:

16 Data Lines (D00:15)

The 16 Bi-directional Data lines are used to transfer one 8-bit byte or one 16-bit halfword between the CPU and the device controller. Data Lines D08:15 are used for byte transfer. The 10-bit address sent from the CPU (or returned on an Acknowledge operation) uses Data Lines D06:15.

Control Lines (Manipulated by the Processor)

- SR** Status Request. The Processor signals the last addressed device controller to send the device status to Data Lines D08:15, followed by a SYN.
- DR** Data Request. The Processor signals the last addressed device controller to present data to the Data lines, followed by a SYN. (One byte or halfword of data is sent depending on whether the device is halfword or byte oriented.)
- DA** Data Available. The Processor signals the last addressed device controller that the data on the Data lines is valid. The device controller accepts the low byte or the entire halfword and responds with a SYN.
- CMD** Command. The Processor signals the last addressed device controller that the command byte on Data Lines D08:15 is valid. The device controller accepts the command byte and responds with a SYN.
- ADRS** Address. The Processor signals that it presented 10-bits of address on Data Lines D06:D15. The device controller that recognizes its address responds with a SYN.
- If no device controller recognizes its address in approximately 30 micro-seconds, the IOU generates a False SYN (FSYN).

CL070 This line is activated by the IOU whenever any of the following occur:

1. The Initialize key on the Display Console is depressed.
2. The key-operated ON/OFF/LOCK Power Switch on the Display Console is turned OFF.
3. The primary power input falls below minimum operating level.
4. Auxiliary initialize inputs are activated (e.g., from LSU).

Test Lines (Manipulated by device controllers).

- HW** Halfword. The Halfword line is activated by a halfword oriented device controller whenever it is communicating normally with the Processor (when its address flip-flop is set).
- SYN** Synchronize. This signal is generated by the device to inform the Processor that it has properly responded to a Control line.

Interrupt and Acknowledge Lines.

- ATN 00:03** Attention. Any device controller desiring to interrupt the CPU, activates one of the four ATN lines and holds that line until the corresponding ACK signal is received.
- ACK 00:03** Acknowledge. The CPU acknowledges one of the four interrupts by asking the IOU to perform an ACK function. The IOU in response activates one of the four ACK lines, selected by 2-bits of the B Bus (30:31), each of which can feed a daisy-chain priority wiring pattern. The responding device controller presents its address on Data Line D06:15, followed by a SYN signal.

Initialize Line

- SCLR** System Clear. This is a metallic contact to ground that occurs during Power Fail, Power Up or Initialize. The current carrying capability of the contact is limited. External circuits should not be connected directly to it. Refer to the bus buffer or buffered I/O channel for these applications.

NOTE:

All Control lines, except ACK are connected in parallel to all devices. These lines are activated by the Processor in response to an external interrupt. The ACK line is connected in series with all devices. If no interrupt is pending in the first controller when the ACK signal arrives, the signal is passed on daisy-chain fashion to the next controller, and so on until it is captured by the interrupting controller. See definition of ACK.

All busses are the false type (i.e., a low voltage level is active and a high voltage level is inactive.)

Each device controller is permitted one TTL load on any of the Data lines, Control lines, Acknowledge line, or the Initialize line. Furthermore, each device controller is permitted one OR tie onto a Data line, Test line, or Interrupt line. The controller bus driver must be either a high-power open-collector TTL gate or the tri-state equivalent.

A maximum of 16 I/O device loads can be driven from the IOU's unbuffered MUX Bus including Selector Channels, Bus Buffers, Bus switches, and Sub-Channel Controllers.

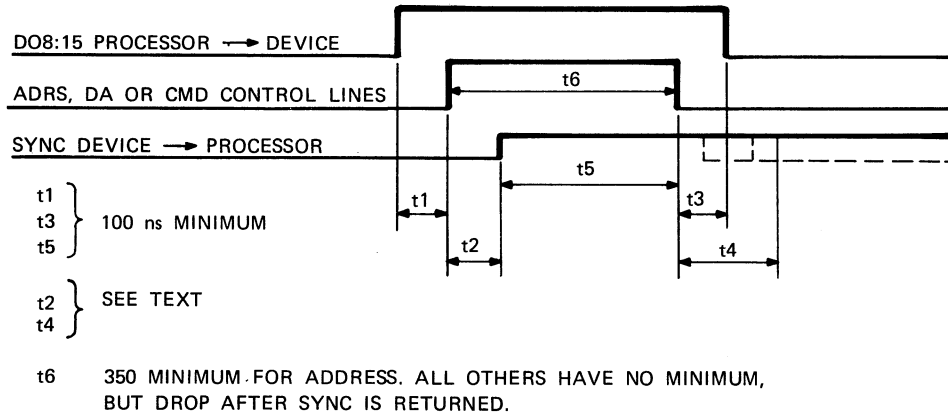
10.2 Multiplexor Channel Timing

Input and output operations on the MUX Bus use request/response signaling. This allows the system to run at its maximum speed. Timing for typical input/output operations are shown on Figure 22. Detailed timing is shown on Figures 23 and 24.

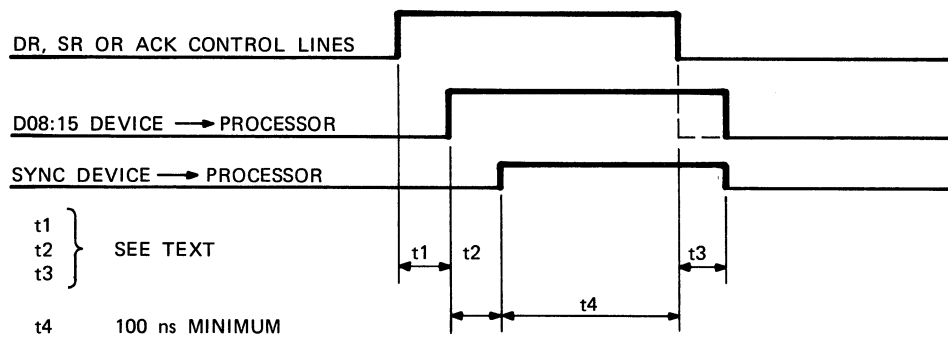
On output, the CPU places signals on Data Lines D08:15, followed by an appropriate Control line signal. This Time delay (t_1) varies but it is guaranteed to be at least 100 nanoseconds. When the device controller has received the Data line information, it sends the SYN signal to the CPU which terminates the Control line signal. The SYN Time delay (t_2) should be only long enough to guarantee proper reception of the output data. The Control line/Data line removal time (t_3) is guaranteed to be at least 100 nanoseconds. The SYN removal time (t_4) should be minimized since the CPU does not proceed until the SYN signal is removed.

It should be noted that the times shown are defined for signals on the MUX Bus. Within a given controller, one signal may pass through more gates than another signal and these additional delays must be considered.

For the input operation, the CPU activates one of the input type Control lines and the currently addressed device controller gates onto Data Lines D08:15—keeping Time delay (t_1) at a minimum. The SYN Time delay (t_2) must guarantee that all the returned data is on the Data lines, considering the slowest data gates and the fastest SYN gates. The CPU removes the Control line signal when SYN is received, with a minimum Time delay (t_4) of 100 nanoseconds. The SYN removal time (t_3) should be minimized since the CPU does not consider the operation complete until the SYN signal is removed. When the Control signal is ACK, Time delay t_1 includes the cumulative contention circuit delays for all the controllers, between the responding controller and the CPU.



(A.) OUTPUT



(B.) INPUT

Figure 22. Multiplexor Channel Timing

NOTE

With a SYN delay of 50 nanoseconds, device controllers must be designed to accept a minimum width of 170 nanoseconds on all Control lines and Acknowledge signals except ADRS, which is guaranteed to be 350 nanoseconds minimum. The SYN delay in the device controller may be increased to effectively lengthen the Control line signals if it is absolutely necessary. It is essential to realize that the CPU does not proceed until the SYN signal is returned and removed. While the slower data transfer rates may not affect a particular controller, the overall system performance is degraded. Furthermore, if a device controller fails to respond with a SYN signal within 25 to 35 microseconds, the CPU aborts the I/O operation.

10.3 Multiplexor Channel and Multiplexor Operations (MUX)

Operational and circuit description also refer to Section 9.1.1. This section covers the circuits which implement the D Bus operations.

10.3.1 MUX Channel Operation. MUX Channel is a byte or halfword oriented Input/Output system which communicates with up to 1,023 peripheral device controllers or interfaces. When I/O control is addressed and given a D Bus function code, it creates one, two, or three MUX Channel operations. The halfword functions (RDH/WDH) have a single data cycle when the HW (Halfword) Test line is active and two data cycles when a HW is inactive (communicating with byte-oriented controller).

10.3.2 Typical Output Case. (All cycles: Processor → Devices). A device controller receives 10-bits of address (over Data Lines D06:15) during the address cycle (T1). In the following first data cycle (T2), an 8-bit Command byte, or one byte of data (over Data Lines D08:15), or a Halfword of data (over Data Lines D00:15) is sent to the device.

For Halfword (HW) functions the cycle counter generates a second data cycle (T3) if necessary, in which case an additional byte of the Processor data is sent to the device over Data Lines D08:15 (see Section 10.3.1).

10.3.3 Typical Input Case. (Address Cycle: Processor → Device, Data Cycles: Device → Processor). A device controller receives 10 bits of address (over Data Lines D06:15) during the address cycle. In the following first data cycle, an 8-bit device Status byte, or one byte of device data (over Data Lines D08:15), or a halfword of device data (over Data Lines D00:15) is gated on the Processor's S Bus.

For Halfword (HW) functions, the cycle counter generates a second data cycle if necessary, in which case an additional byte of the device data is sent to the Processor (see Section 10.3.1).

10.3.4 I/O Function Gating. Table 19 shows IOU function gating specifications.

A single instruction from the CPU contains the device address, the encoded function, and up to 16 output data when needed.

D Bus functions may be performed with or without address cycle depending on the state of FSEL00.

KSIG is used to specify register type operations or to distinguish between Halfword (HW) functions (RDH/WDH) and some non D-Bus operations (STB/LB).

For byte designation used in Table 19 refer to the following information.

B(H) refers to Bits 16:23 from the corresponding B Bus.
B(L) refers to Bits 24:31 from the corresponding B Bus.
A(H) refers to Bits 16:23 from the corresponding A Bus.
A(L) refers to Bits 24:31 from the corresponding A Bus.
S(H) refers to Bits 16:23 from the corresponding S Bus.
S(L) refers to Bits 24:31 from the corresponding S Bus.
D(H) refers to Bits 00:07 from the corresponding D Bus.
D(L) refers to Bits 08:15 from the corresponding D Bus.
MCR(L) = MCR08:15
MCR(H) = MCR00:07

Note that Bits 00:15 of the Processor's A, B, and S Busses do not have an appearance on the IOU module (see block diagram Figure 21). The contents of S00:15 is zero when bytes are gated to S(H) and/or S(L).

10.3.5 MUX Bus Generation. (Circuit Description and Internal Timing.) The circuits which generate the D Bus and the companion Control Test lines are described in this section. Also a detailed description of one output function (WDA), and one input function (RDHA) are used to explain the operation of the MUX generation circuits.

1. MUX Generation Circuits. As seen on Figure 25, there are five general circuit groups for MUX generation:

- a. Input circuits.
- b. Cycle counter.
- c. Control line logic and bus drivers.
- d. D Bus gating and receiving logic.
- e. Internal Timing Differentiation and control logic.

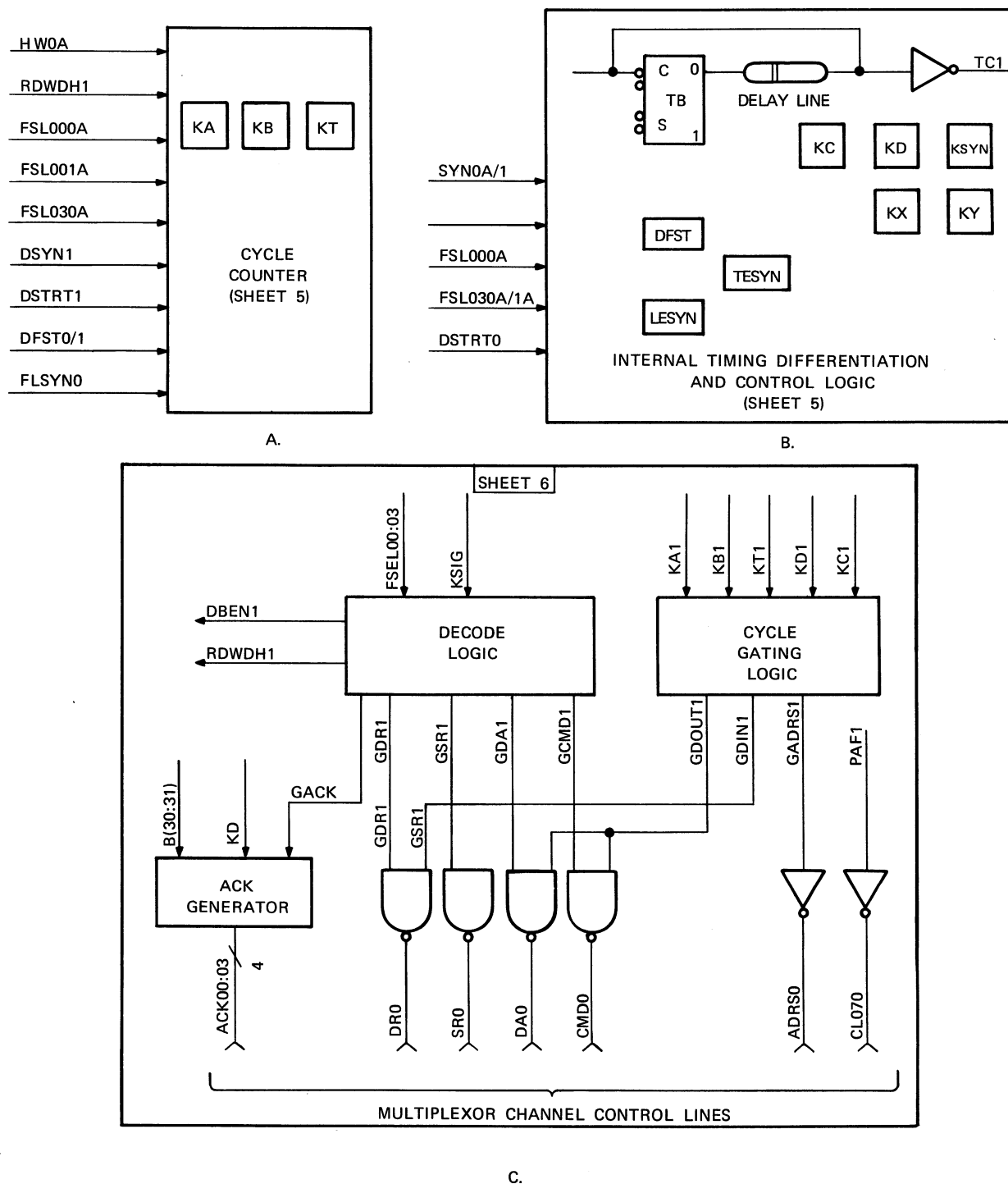


Figure 25. Multiplexor Circuit Generation Description

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

INPUT CIRCUITS

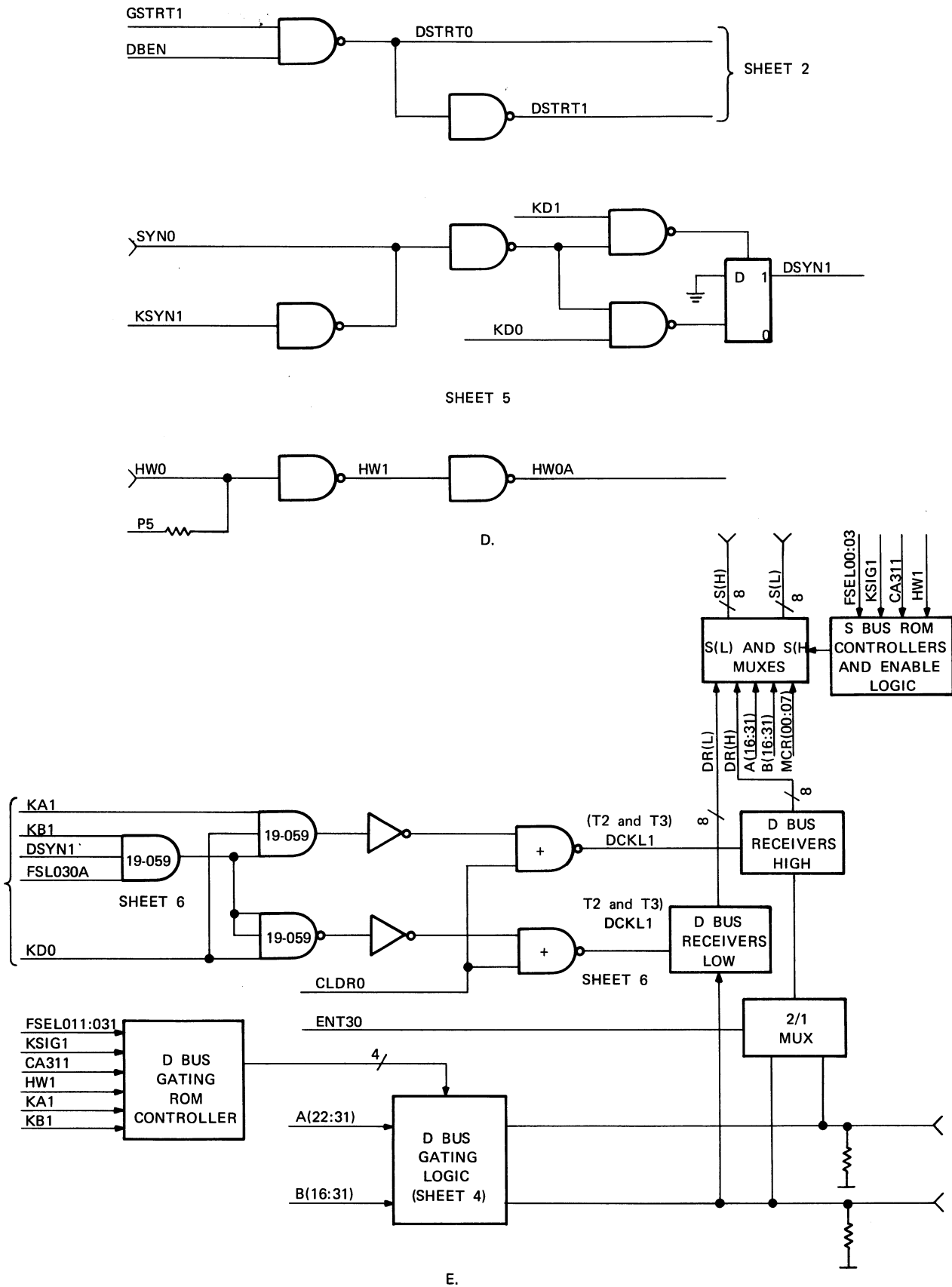


Figure 25. (Continued) Multiplexor Circuit Generation Description

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The two edge triggered D type flip-flops KA and KB, connected as a Johnson Counter, together with the KT flip-flop (5N5) make up the cycle counter circuit. The sequence starts with DFST0 setting the KA flip-flop (5J6). When there is no address cycle, DFST1 and FSL000A (5K5) also set the KB flip-flop (5L6). The counter advances on the trailing edge of the DSYN1 (5E7) signal which is stretched (if necessary) on the ADRS, CMD, and DA cycles. This insures that any byte gated to the D Bus remains for at least 100 nanoseconds after the associated Control line signal is removed. The sequence stops when the Terminate flip-flop (KT) (5N5) is set and the MFIN line to the CPU is activated. Operation of the KSYN flip-flop (5G8) which provides the SYN stretch, is described later.

A timing chart for the cycle counter is shown on Figure 26. From the idle time period T0, the counter is preset into period T1 or T2 depending upon the presence or absence of an address cycle. The sequence normally terminates at the end of T2 unless a halfword operation with a byte oriented device requires a second data cycle, T3 (ENT30) (5N7).

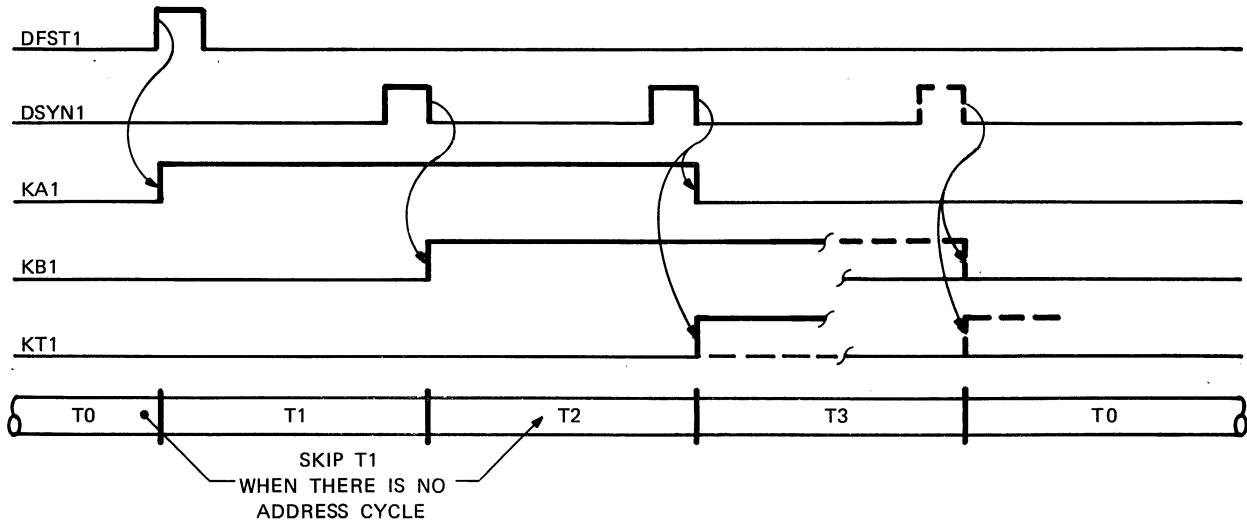


Figure 26. Cycle Counter

All D Bus operations begin when DBENI is gated by the GSTRT1 signal (1G8) to produce the DSTRTO and DSTRTI signals (5G9). These signals in turn are used by the STRT timer (14H8), the cycle counter, and the timing control circuit.

The SYN0 Test line is the main source of timing in the request/response signalling system used on the Multiplexor Channel. It must be carefully terminated and deglitched before being presented to the cycle counter and timing control. The leading edge of the Test line is gated by KD1 (5G6) to direct set the DSYN flip-flop (5E7). When the Control line signal to the bus is terminated, the KD flip-flop is cleared and the Test line is now connected to the clock input of the D type, edge triggered, DSYN flip-flop where the trailing edge clears the flip-flop. Once the DSYN flip-flop has been set or cleared, ringing or noise near the edges of the SYN0 signal is ignored. The trailing edge of SYN0 (5A6) is extended on output operations by the high KSYN1 signal (5A6). This is described in detail later.

The Halfword Test line (HW0) (5J2) produces the signals HW1 and HW0A which control data gating on the D Bus and indicate to the Multiplexor Channel circuits whether a byte oriented or halfword controller is in use.

The 10-bus drivers (Figure 25C), are the source of the Multiplexor Channel Control lines. Inputs are cycle gating and function decoder signals. GDOUT1 (6M2) is active for output functions. GDINI (6K3) is active for input functions.

Both contain the intra-cycle timing signals KC1 (6N2) or KD1 (6L2). When data is placed on the D Bus for ADRS/CMD, or DA operations, the KC1 pulse provides the Control line timing, delayed 100 nanoseconds from the beginning of the cycle. For ACK, SR, or DR operations, the non-delayed KD1 pulse gates the Control lines. Both KC1 and KD1 are removed 100 nanoseconds after SYN0 is received, in accordance with Multiplexor Channel timing requirements. Only one out of four ACK lines is made active during any given ACK functions. The 19-129 3:8 decoder (6D2) selects the active ACK line according to the state of Bits 30 and 31. The signal CL070 (14K7) goes low active when Primary Power Failure (PPF1) is detected high, or INIT, or POFF go high.

The D Bus consists of 16 Bi-directional Data Lines terminated on the IOU board.

The 19-134 tri-state D Bus drivers multiplex 10-bits of A Bus and 16-bits of B Bus onto the D Bus. Their tri-state outputs are tied together in two groups to form D Bus high and D Bus low. Only one output for each D Bus line is enabled at a time, the others are in the high impedance state. This is achieved with the help of the D Bus ROM controller (4D3). Unlike the static selection of the S MUX controller, the D Bus ROM controller address selection (hence enabling of D Bus drivers) changes on every cycle of the sequence (KA1 and KB1 are used as address select outputs).

The 19-071 edge triggered D latches, used as D Bus receivers, load on the low to high transition of the clock leads DCKL1 and DCKH1 (6A5). The high byte register normally receives data from D00:07 during period T2 (KA1 and KB1) gated through the two-to-one MUX by ENT30 (6H6) in the high state. For the double data cycle, ENT30 is low active so that the first byte on D08:15 enters both DR00:07 and DR08:15 during T2. The second byte on D08:15, during T3, is registered in DR08:15 only to overwrite the first byte. Clock logic for DCKL1 and DCKH1 (6A2) uses the common term KB1-DSYN1. FSL030A which is active for all input functions, including ACK during periods T2 and T3. Final gating with KD0 loads the registers at the moment the selected Control line signal is removed; i.e., about 100 nanoseconds after the beginning of the SYN signal when the Data Lines have settled. The Data Register outputs feed the S MUX and the CC MUX.

The timing and control circuits provide the intra-cycle timing and SYN stretching features mentioned in earlier sections. These circuits consist of six edge triggered J/K flip-flops, a two stage counter, an R/S flip-flop, a 100 nanosecond tapped delay line, and the interconnecting logic. Three of the flip-flops, DFST (5E5), LESYN (5G2), and TESYN (5E6) detect the transitions of the DSTRT0 and DSYN1 signals respectively, and feed the delay line R/S flip-flop combination.

The TB flip-flop (5B4) is set by a low signal on Terminal 1, 2 or 4. A low signal on Terminal 9, 10, or 12 clears the flip-flop. When a momentary set pulse is applied, the high-to-low transition at TB0 travels down the delay line emerging after X nanoseconds as TC0 (5C2) to clear the flip-flop. This produces pulses TC0 and TC1 which are X nanoseconds wide and start X nanoseconds after the set TB pulse (where X is the tap delay plus the flip-flop transition times). Using the 50 nanosecond tap (Terminal 13 of the line), X is approximately 50 nanoseconds and the trailing edge of TC1 occurs 100 nanoseconds after the set pulse. When the set pulse is long enough to still be present after the end of TC0, TB0 again goes low to generate another pulse; i.e., the circuit acts like a gated oscillator. As seen on the timing charts which follow, both the single and multi-pulse modes are used.

The timing chart on Figure 26 shows a data output operation (CMD or DA) with an address cycle. The sequence starts with period T1 when the KA flip-flop is set. At the end of the first SYN pulse, the KB flip-flop is set and period T2 starts. The end of the second SYN pulse clears the KA flip-flop and sets the KT flip-flop. With KT0 low, gating to the D Bus/Control lines is suppressed (6L1) and the pulse generator is killed (5C3).

On both address and data cycles, the data bytes must be on the D Bus at least 100 nanoseconds before the Control line signal starts and must remain active for 100 nanoseconds after the Control line signal is removed. In addition, the Control line must remain active for 100 nanoseconds after SYN arrives. The width of the ADRS0 Control line pulse must be at least 350 nanoseconds. This insures that the Address flip-flop on a controller, separated from the CPU by one or more bus buffers, can be reliably cleared even with a fast SYN response from a local controller.

The DFST1 lead sets the Delay Line flip-flop (TB) (5B4) if it is either an output data cycle (FSL030A low) (5A4) or an address cycle (FSL000A low) (5A4). The KD flip-flop (5G6) is always set by DFST. Flip-flop KC (5G5) toggles set at the end of TC1 since its J input DSYN0 and direct clear (KD1) are both high. KC1 sets the KSYN flip-flop (5H8) and gates the Control lines as described earlier. The LESYN flip-flop (2G3) sets on the leading edge of DSYN and sets the TB flip-flop again. The KD flip-flop toggles clear on the first TC1 pulse after its K input goes high and in turn direct clears the KC flip-flop. For non-address cycles, the K input to the KD flip-flop (KDK1) goes high as soon as SYN is received, the next TC1 pulse clears the KD flip-flop after 100 nanoseconds. On the address cycle (period T1 with the KB flip-flop cleared), the TC1 pulses are fed to a two stage Johnson Counter, flip-flops KX and KY (5M8). The KDK1 input to the KD flip-flop is held low until after three TC1 pulses have been registered on the KX and KY flip-flops. The next TC1 pulse clears the KD flip-flop. In this manner the minimum width of KC1 and the ADRS0 signal are equal to 300 nanoseconds plus the SYN return delay (KDK1 = KXO-DSYN1). The KSYN flip-flop is cleared 100 nanoseconds after the KD flip-flop is cleared since its K input (KD0) is high when the next TC1 pulse arrives. KSYN (5A5) forces SYN1 and DSYN1 high as long as the KSYN flip-flop is set. This insures that DSYN1 and SYN0 remain active for at least 100 nanoseconds after KC1 and the Control line signals, gated by KC1, are ended. A fast SYN response from a device controller is not able to terminate the cycle prematurely and violate the timing rules for the D Bus.

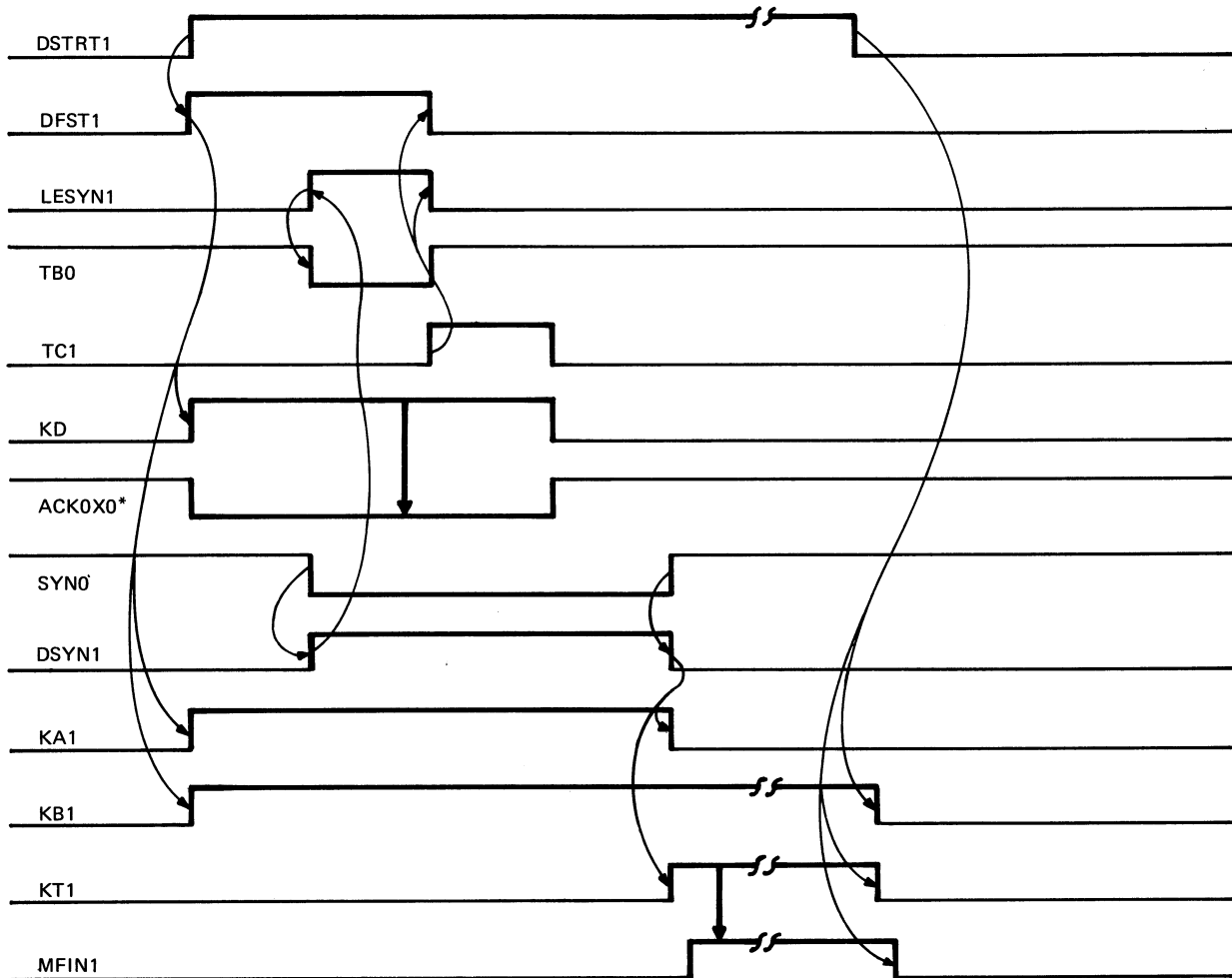
Note that the LESYN flip-flop remains set until the first TC1 pulse after the KC flip-flop is cleared. This produces the multi-pulse mode of the delay line; i.e., a group of TC1 pulses at 100 nanosecond intervals. Also note that while the KD flip-flop is not used directly for Control line timing, it is part of the logic for the KC and KSYN flip-flops.

The TESYN flip-flop (5E6) sets on the trailing edge of the DSYN1 signal and sets the TB flip-flop again if an output data cycle is required (FSL031A) (5B5). The KD flip-flop is direct set by TESYN. Timing for the output data cycle is similar to the ADRS cycle with two exceptions. First, the KX and KY flip-flops are not used to stretch the Control line signal and second, a double data cycle may be generated for the WDH operation to a byte oriented device. During the T2 SYN pulse, the logic that sets the KT flip-flop (5N6) also produces a low level on SKT0 (5N3). This causes the TESYN flip-flop to ignore the end of SYN1 (since both the J and K inputs are low) and the TB flip-flop is not set. When a double data cycle is needed, the set KT logic does not become active until period T3. The TESYN flip-flop sets on the end of the T2 SYN signal and thus pulses the TB/delay line circuit for timing control during period T3. It also applies to the ADRS and DR operations.

Figure 23 shows timing for a data input operation (SR or DR) with an address cycle. The TB flip-flop is set with DFST since KC1 is needed for Control line gating during the address cycle. It also applies to the ADRS and DR operations. The KX and KY counter insure the minimum width of the ADRS0 signal. TESYN sets the KD flip-flop at the end of period T1 but does not set the TB flip-flop. KD1 gates the Control lines for the input bytes. The KC and KSYN flip-flops are not used on the data cycles. A double data cycle is produced for the RDH operation to a byte oriented device.

For non-address functions, the sequences start with period T2 since the KA and KB flip-flops are both set with DFST. The data cycles on Figures 23 and 24 are essentially the same.

Timing for the ACK function is shown on Figure 27. It consists of a single data cycle (T2). The KC and KSYN flip-flops are not required. The TB flip-flop and the delay are pulsed only by LESYN to time the removal of the Control line signals.



* One of ACK00:03 selected by B30:31

Figure 27. Multiplexor Channel Timing, ACK

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

On all timing charts, the KT flip-flop generates S Bus and CC Bus gating and eventually the MFIN0 signal. This restarts the CPU clock causing the removal of STRT0, which in turn removes GSTRTI, DSTRT, and the MFIN signals.

A group of clear signals insures proper circuit states for initialize and other operations. CLRA0 (5N6) is low for SCLROB low or any non-D Bus operation (DSTRT1 low). It clears the cycle counter and kills the pulse generator. CLRBO (5C8), used by DFST and TESYN, cannot use DSTRT1 for clearing due to a possible race condition when the DFST flip-flop toggles set. It combines SCLROB, TC1/KD1 (as per timing charts), and MFIN0A a copy of the MFIN0 generated by the IOU board.

CLRC0 (5H8) is low whenever the IOU cycle timer is inactive (period T0). CLRD0 (2C3) uses CLRC0 or TC1/KC0 (as per timing charts) to clear the LESYN flip-flop.

2. Output Operations (WDA). (See Section 10.3.2 and Figures 25E and 24.) A detailed description of the WDA (Address, Write Data) operation is used as an example of output operations.

The cycle timer must generate two cycles to execute the WDA instruction. The address cycle (T1) starts when DSTRT0 sets the DFST flip-flop (see Section 10.3.5). The 10-bits of address must be placed on the D Bus. This is accomplished by the D Bus controller outputs Y1, Y2, Y3, and Y4 equal to HHHH whenever address cycle (T1) is detected, see Figure 28, which shows all possible D Bus gating situations for WD and WDA instructions. The ADRS Control line is activated approximately 100 nanoseconds after the beginning of the address cycle to inform the interrupting device controller that the D Bus contains 10-bits of address. The trailing edge of KSIGI (SYN stretching signal) (5A5) starts the data output cycle (see Figure 24), in which either:

$$\left\{ \begin{array}{l} 0 \rightarrow D(H) \\ B(H) \rightarrow D(L) \end{array} \right\} \text{ for CA311=L} \quad \text{or} \quad \left\{ \begin{array}{l} B(H) \rightarrow D(H) \\ B(L) \rightarrow D(L) \end{array} \right\} \text{ for CA31=H}$$

has to be gated onto D00:15. See Figure 28 and Table 19.

The DA Control line is activated about 100 nanoseconds after the beginning of the output data cycle (T2) to inform the interrupting device controller that the output data is settled on the D Bus. Interrupting device controller latches the data and responds with a SYN. The trailing edge of the KSYN (SYN stretching signal) (5A5) sets the KT flip-flop which generates MFIN (Module Finished) signal.

3. Input Operations (RDHA). (see Section 10.3.3 and Figures 25E and 23.) A detailed description of the RDHA (Address, Read Data Halfword) operation is used as an example of output operations. The cycle timer must generate three cycles to execute the WDA instruction. See Figure 23. The address cycle is identical to the one described in Section 2. Output Operations (WDA). The trailing edge of KSIG1 starts the first data input cycle (T2), in which D00:15 is latched in the D Bus Receivers DR00:15. The Data Request (DR) Control line is activated at the start of the data cycle (T2) to signal the controller to put the data on D00:15 and activates the SYN line. If HW0 is active (Halfword oriented controller) only one input data cycle is necessary. If HW0 is high (byte-oriented controller) then it sends the most significant byte in T2 and the least significant byte (always via D08:15) in T3, the second data cycle. The latching of the D Bus in the D Receivers occurs approximately 100 nanoseconds after SYN is received (to insure settling the data).

The contents of the Data Receivers must be placed on S16:31. This is accomplished by two ROM controllers which control the multiplexing on the S Bus. See Figures 25E and 29.

11. BYTE MANIPULATION AND AUXILIARY FUNCTIONS

11.1 Byte Manipulation Functions

(STBR, LBR, STR, LB, and EXB – see Tables 19 and 20.)

These functions do not activate the MUX Bus since there is only need to gate selected bytes from A and B Busses onto the S Bus. Page 2 of the 35-539D08 schematics show the gating onto the S Bus High. The 19-132 tri-state S Bus 2:1 Multiplexors are tied in two pairs of three each, to allow for effective 6:1 Multiplexing (see Figure 21 block diagram). Only one output of the three tri-states is enabled at a time. Enable and select inputs are generated from Y1 to /4 outputs of S(H) ROM controller and from STC1 line active for non-D Bus operations. The same concept is used for generating S Bus low. (Page 3 of the 35-539D08 schematics.) ROM controller's S Bus-generation for STB instruction is shown on Table 20.

Enable inputs of the S MUX low are used for MFIN generation for the case of byte handling operations (SMFIN is on Page 7 of the 35-539D08 schematics). These lines also become active for D Bus and SMCR operation so that the additional signals (SMCRO and STC1) (7G7) are needed to insure that SMFIN0 (7H8) goes active only for five byte handling operations.

TABLE 19. I/O MODULE FUNCTION GATING

HEX. EQUIV. OF FSEL	FSEL			K SIG	CA31	HW	CA31=0 OR HW=0		CA31=1 OR HW=1		OTHER		
	0	1	2				3	DESTINATION(H)	DESTINATION(L)	DESTINATION(H)		DESTINATION(L)	
0	RD	0	0	0	0	0/1	X	D(L)→S(H)	B(L)→S(L)	B(H)→S(H)	D(L)→S(L)	ZERO→CC	
	RDR	0	0	0	1	X	X	ZERO→S(H)	D(L)→S(L)	ZERO→S(H)	D(L)→S(L)		
1	WD	0	0	0	1	0	0/1	X	ZERO→D(H)	B(L)→D(L)	B(H)→D(H)	B(L)→D(L)	ZERO→CC
	WDS	0	0	0	1	1	X	X	B(H)→D(H)	B(L)→D(L)	B(H)→D(H)	B(L)→D(L)	
2	SS	0	0	1	0	0	0/1	X	D(L)→S(H)	B(L)→S(L)	B(H)→S(H)	D(L)→S(L)	D(12:15)→CC
	SSR	0	0	1	0	1	X	X	ZERO→S(H)	D(L)→S(L)	ZERO→S(H)	D(L)→S(L)	
3	OC	0	0	1	1	0	0/1	X	ZERO→D(H)	B(L)→D(L)	B(H)→D(H)	B(L)→D(L)	ZERO→CC
	OCR	0	0	1	1	1	X	X	B(H)→D(H)	B(L)→D(L)	B(H)→D(H)	B(L)→D(L)	
4	*RDH	0	1	0	0	0	X	0/1	D(L)1→S(H)	D(L)2→S(L)	D(H)→S(H)	D(L)→S(L)	ZERO→CC
	STBR	0	1	0	0	1	X	X	B(H)→S(H)	A(L)→S(L)	B(H)→S(H)	A(L)→S(L)	
5	*WDH	0	1	0	1	0	X	0/1	B(H)→D(L)1	B(L)→D(L)2	B(H)→D(H)	B(L)→D(L)	ZERO→CC
	LBR	0	1	0	1	1	X	X	ZERO→S(H)	B(L)→S(L)	ZERO→S(H)	B(L)→S(L)	
6	ACK	0	1	1	0	0	X	X	D(H)→S(H)	D(L)→S(L)	D(H)→S(H)	D(L)→S(L)	ZERO→CC
	LDWAIT	0	1	1	0	1	X	X	NA	NA	NA	NA	
7	SMCR	0	1	1	1	0	X	X	MCR(H)→S(H)	MCR(L)→S(L)	MCR(H)→S(H)	MCR(L)→S(L)	MCR(12:15)→CC
	CMCR	0	1	1	1	1	X	X	NA	NA	NA	NA	
8	RDA	1	0	0	0	0	0/1	X	SAME AS RD AND RDR BUT PRECEDED BY ADDRESS CYCLE [A (22:31)→D (06:15)]				
	RDRA	1	0	0	0	1	X	X					
9	WDA	1	0	0	1	0	0/1	X	SAME AS WD AND WDR BUT WITH ADDRESS CYCLE				
	WDRA	1	0	0	1	1	X	X					
A	SSA	1	0	1	0	0	0/1	X	SAME AS SS AND SSR BUT WITH ADDRESS CYCLE				
	SSRA	1	0	1	0	1	X	X					
B	OCA	1	0	1	1	0	0/1	X	SAME AS OC AND OCR BUT WITH ADDRESS CYCLE				
	OCRA	1	0	1	1	1	X	X					
C	*RDHA	1	1	0	0	0	X	0/1	SAME AS RDH BUT WITH ADDRESS CYCLE				
	STB	1	1	0	0	1	0/1	X	A(L)→S(H)	B(L)→S(L)	B(H)→S(H)	A(L)→S(L)	
D	*WDHA	1	1	0	1	0	X	0/1	SAME AS WDH BUT WITH ADDRESS CYCLE				
	LB	1	1	0	1	1	0/1	X	ZERO→S(H)	A(H)→S(L)	ZERO→S(H)	B(L)→S(L)	
E	THW	1	1	1	0	0	X	X	NA		NA		HW→MSIG
	EXB	1	1	1	0	1	X	X	B(L)→S(H)	B(H)→S(L)	B(L)→S(H)	B(H)→S(L)	
F	POW	1	1	1	1	0	X	X	NA		NA		RELEASE SCLR RELAY
	POUT	1	1	1	1	1	X	X	NA		NA		B(27:31)→CABLE

TABLE 20. STB INSTRUCTIONS

ROM ADDRESS SELECT IDENTICAL FOR BOTH S BUS CONTROLLERS								DATA TO BE GATED ON S(H) = S(16:23)	S(H) ROM'S OUTPUTS	DATA TO BE GATED ON S(L) = S(24:31)	S(L) ROM'S OUTPUTS							
FUNCTION	1	2	3	4	7	6	5											
	FSEL001	FSEL011	FSEL021	FSEL031	KSIG1	CA311	HW1											
	12	11	10	9	12	11	10	9										
	SEL Y1	A Y2	B Y3	C Y4	SEL Y1	A Y2	B Y3	C Y4										
STB	H	H	L	L	H	L	L	L	A(L)→S(H)	H	L	L	H	B(L)→S(L)	L	L	H	L
	H	H	L	L	H	L	L	H	A(L)→S(H)	H	L	L	H	B(L)→S(L)	L	L	H	L
	H	H	L	L	H	H	L	L	B(H)→S(H)	H	L	H	L	A(L)→S(L)	H	L	L	H
	H	H	L	L	H	H	H	H	B(H)→S(H)	H	L	H	L	A(L)→S(L)	H	L	L	H

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

FUNCTION ADDRESS SELECT	FSEL011	FSEL021	FSEL031	KSIG1	CA311	HW1	KA1	KB1	CYCLE COUNTER STATE	DATA TO BE GATED ON D(00:15)	Y1	Y2	Y3	Y4
	15	1	2	3	4	7	6	5			12	11	10	9
	D BUS CONTROLLER INPUTS										D BUS CONTROLLER OUTPUTS			
	L	L	H	L	L	L	L	L	T0	DRIVERS IN H-Z	H	H	H	H
	L	L	H	L	L	L	L	H	T3	WILL NEVER OCCUR	X	X	X	X
	L	L	H	L	L	L	H	L	T1	A(22:31)→D(06:15) 0→D(00:05)	H	H	H	L
	L	L	H	L	L	L	H	H	T2	0→D(H) B(H)→D(L)	H	L	H	H
WD AND WDA	L	L	H	L	L	H	L	L	T0	DRIVERS IN H-Z	H	H	H	H
	L	L	H	L	L	H	L	H	T3	WILL NEVER OCCUR	X	X	X	X
	L	L	H	L	L	H	H	L	T1	A(22:31)→D(06:15) 0→D(00:05)	H	H	H	L
	L	L	H	L	L	H	H	H	T2	0→D(H) B(H)→D(L)	H	L	H	H
	L	L	H	L	H	L	L	L	T0	DRIVERS IN H-Z	H	H	H	H
	L	L	H	L	H	L	L	H	T3	WILL NEVER OCCUR	X	X	X	X
	L	L	H	L	H	L	H	L	T1	A(22:31)→D(06:15) 0→D(00:05)	H	H	H	L
	L	L	H	L	H	L	H	H	T2	B(H)→D(H) B(L)→D(L)	L	H	L	H
	L	L	H	L	H	H	L	L	T0	DRIVERS IN H-Z	H	H	H	H
	L	L	H	L	H	H	L	H	T3	WILL NEVER OCCUR	X	X	X	X
	L	L	H	L	H	H	H	L	T1	A→D	H	H	H	L
	L	L	H	L	H	H	H	H	T2	B(H)→D(H) B(L)→D(L)	L	H	L	H

Figure 28. D Bus ROM Controller Data Gating for WD and WDA

ROM ADDRESS SELECT IDENTICAL FOR BOTH S BUS CONTROLLERS								DATA TO BE GATED ON S(H) = S(16:23)	S(H) ROM CONTROLLER OUTPUTS				DATA TO BE GATED ON S(H) = S(24:31)	S(L) ROM CONTROLLER OUTPUTS			
FUNCTION	FSEL001	FSEL011	FSEL021	FSEL031	KSIG1	CA311	HW1		Y1	A	B	C		SEL	A	B	C
	1	2	3	4	7	6	5		Y1	Y2	Y3	Y4		Y1	Y2	Y3	Y4
RDH	L	H	L	L	L	L	L	DR(H)→S(H)	L	H	L	L	DR(L)→S(L)	L	L	L	H
	L	H	L	L	L	L	H		L	H	L	L		L	L	L	H
	L	H	L	L	L	H	L		L	H	L	L		L	L	L	H
	L	H	L	L	L	H	H		L	H	L	L		L	L	L	H
RDHA	H	H	L	L	L	L	L	DR(H)→S(H)	L	H	L	L	DR(L)→S(L)	L	L	L	H
	H	H	L	L	L	L	H		L	H	L	L		L	L	L	H
	H	H	L	L	L	H	L		L	H	L	L		L	L	L	H
	H	H	L	L	L	H	H		L	H	L	L		L	L	L	H

Figure 29. ROM Controller Data Gating for RDH and RDHA

11.2 Auxiliary Functions

(SMCR, CMCR, LDWAIT, THW, POW, and POUT – see Table 19).

These functions are described in this section. The SMCR function provides a means for sensing 16-bits of MCR, (Machine Control Register) see Section

The SMCR function is decoded by a 3:8 decoder (7B7), MCR11:15 is placed on the CC Bus by 2:1 CC Bus MUX (7E2). The SMCRO line also generates MFIN and the strobe for Condition Code. The contents of MCR00:15 is gated onto the S00:15 by the 19-132 S Bus Multiplexors (ROM controlled) shown on Pages 2 and 3 of the 35-539D08 schematics.

The DMCR function is decoded by the 19-129 3:8 decoder, whose CMCR0 output enables the four least significant bits of the B Bus (7B2) to clear selectively four MCR registers. (Ones in B27:31 clear the corresponding MCR registers.)

The LDWAIT function is decoded by the 19-129 3:8 decoder, it controls the indicator light on the Display Console (ON or OFF) according to the state of B16.

The THW function is decoded by the 19-129 3:8 decoder (7B7) it generates MSIG according to the state of the HW (Halfword) Test line.

The POUT function is decoded by the 19-129 3:8 decoder it gates four bits (B27:31) to a set of board stakes for external signalling purposes. These signals may be wired to the front chassis terminal strip by adding optional wires to the Display Console connector at the IOU board. The MFIN signal to the CPU is delayed by a timer to set the output pulse width at 1.0±0.3 microseconds.

The POW functions releases the System Clear relay, (see Section 10.11).

12. DISPLAY CONTROLLER

The display controller has access to the CPU via the Multiplexor Channel D Bus and the I/O control in the same manner as other peripheral device controllers. The display controller provides a means for reading the contents of all the system registers or any main memory location and transferring the data to the Display Console.

Data and programs can also be manually entered from the Display Console to the controller and then to the CPU. The display controller signals the CPU directly via the Display (DSPLY) interrupt.

12.1 Addressing Logic

The Display Console device address is wired as (X'01'). The D Bus lines D08:15 are buffered and inverted to create double rail Data Lines (Sheet 8). Two more bits of the D Bus (06:07) are used directly in address decoding logic on Page 9 of the 35-539D08 schematics.

The decoded Display Console address activates B1 line (9H4), which sets the ADB flip-flop (9M4) at the trailing edge of the ADRS0 Control line signal. B1 also generates ADSYNB0 (10K4) which generates the SYN signal and clears the CATN flip-flop (10F6).

12.2 Data Output

The byte of data transferred between the display controller and the Display Console Makes use of 8-bidirectional lines SD00:07 (8H4). Data is placed on this SD Bus when the DAGB0 line is active and is gated to one of the four display registers in the Display Console, by one of the load signals, LA or LB (13J5). LA and LB generation logic is shown on Page 13 of 35-534D08 schematic. Two one-shot timers (13E2 and 13J2) insure that the loading signals conform to Display Console specifications. The XA flip-flop is reset by RST0=ADRSI-INCR0·B1.

12.3 Data Input

XC flip-flop (13E8) controls the SHI0 and SL00 signals which gate the contents of the two least significant bytes of the Console Switch Register to the Processor via SD00:07. The RST0 signal clears the XC flip flop in the same manner as it cleared XA.

12.4 Status Input

The Status byte encoding is shown in Chapter 6, Control Console; in the User's Manual, Publication Number 29-261.

12.5 Control Logic

Complimentary pulsed ESN00 and ESN01 signal from the console are fed into a deglitching R-S flip-flop (10C5). ESN00 and ESN01 are activated by depressing various keys of the Display Console keyboard (see Section 12.1). This results in setting a CATN flip-flop and generating display controllers private interrupt to the CPU-DSPLY (10J8). DSPLY interrupt is also generated by depressing the SNGL key on the Console keyboard, which sets the SNGL flip-flop. The SNGLO flip-flop (10F8) can be sensed by CPU as MCR07. The INCR flip-flop (10C2) which determines either incremental or normal mode sets on the trailing edge of the CMGB0 control line. All Control lines for the display controller (6N5-9) are derived from the MUX Bus control lines by gating them with the output of the controllers Address flip-flop ADB (9M4). These Control lines are also used for the Display Consoles SYN generation (7C8). The D Bus drivers and receivers, SYN generation logic, and part of the address decode logic is shared with the Teletypewriter controller.

13. TELETYPE CONTROLLER

The built-in Teletype (TTY) device controller interfaces an ASR;KSR 33 or 35 TTY to the Processor. It provides the serial/parallel conversion required for data transfer between the parallel D Bus and the serial, eight level, start/stop ASCII code signal used by the TTY (see Figure 30).

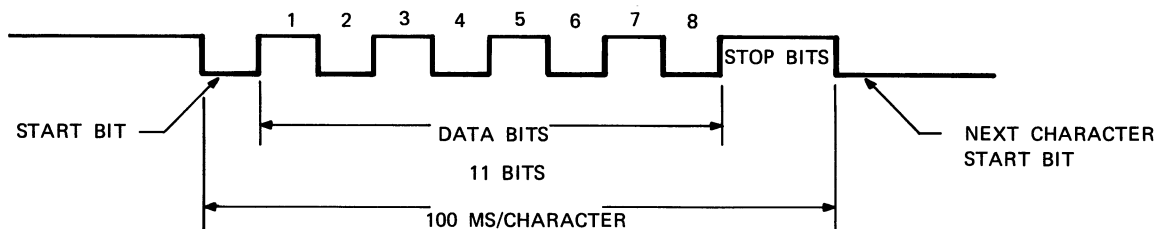


Figure 30. Serial ASCII Code U (Even Parity)

13.1 Block Diagram Analysis

Figure 31 is a block diagram of the TTY controller. The control circuits consist of the Command flip-flops (read or write, etc.) which direct the flow of information, circuits to control ATN/ACK functions, and logic to generate the status bits and control the timer.

The serial information received from the TTY is sampled by the timer and strobed into the Shift Register. When all the data has been shifted in, the data in the Shift Register is transferred to the Buffer Register. It is then gated through D Bus tri-state drivers on D08:15), by the Data Request signal (DRG), Status Request Signal (SR), and Address (ADRS) Control lines. A bit-by-bit copy of the received data may also be sent to the TTY printer/tape punch when the Block flip-flop (BLK) is cleared. In the Write or Send mode, the data byte is placed directly (parallel) into the Shift Register and then shifted out (serially) to the TTY.

13.2 Bus Communications and Address Circuits

Communications between the Processor and the TTY controller is via the Control lines, Test lines, and the low order eight bits of the D Bus. The bus receivers (Sheet 8) are shared with the display controller. The Data Lines D08:15 are buffered to form the DL00:07 lines. When the wired address X'02' is detected, Line A0 is active and the TTY address flip-flop (ADA) (9M3) is toggled set on the trailing edge of the ADRS1 signal (9J2). This enables the other Control lines for the TTY controller (Sheet 10). While the ADRS1 signal is active, the ASYNA0 line goes low and generates the return SYNO signal (11G9).

The D Bus sent logic consists of 19-136 tri-state bus drivers (Sheets 8, 9, and 11) controlled directly by DRGA0, SRGA0 and ATSYNO TTY Control lines, which are derived from the corresponding MUX Bus control and TTY Address flip-flop (Sheet 10).

NOTE:

For systems where X'02' has been assigned to another device, the TTY controller may be strapped for X'82'. (see Sheet 7).

13.3 Status and Commands

The bit assignments for TTY status and command bytes is shown in Table 21.

TABLE 21. TELETYPE STATUS AND COMMAND BYTE

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE	ERR	*	BRK	*	BSY	EX	*	DU
COMMAND BYTE	DISABLE	ENABLE	UNBLOCK	BLOCK	WRITE	READ		

DISARM

* Unassigned status (will return zero).

STATUS BYTE

- ERR The Error bit is set when a character is not taken from the controller buffer before another character is assembled.
- BRK The Break bit is set at the end of one character time when the line is held in the space condition for a period greater than a character period.
- BSY Read Mode. The Busy bit is normally set and is reset when data is available for transfer to the Processor. Write Mode. The Busy bit is normally reset and is set when data is being transferred to the terminal.
- EX The Examine bit is set when BRK or ERR is set.
- DU The Device Unavailable bit is set when the terminal is powered down or in Local mode.

COMMAND BYTE

DISABLE	Disables device interrupts; allows queuing of interrupts.
ENABLE	Enables device interrupts. Note that a command byte with both Bits 0 and 1 set, DISARMS the interface, no interrupt queuing.
UNBLOCK	Allows the Printer to print data entered via the keyboard or tape reader.
BLOCK	Disables the Unblock feature.
WRITE	The interface is placed in the Write mode.
READ	The interface is placed in the Read mode.

The command flip-flops EBL, ARM, BLK, and WT (9J7, 9L7) are loaded with the trailing edge of the CMGA0 signal (9D7). The contents of the flip-flops remain unchanged if the D input is low. The Write Storage flip-flop (WT) (9L7) unconditionally accepts the Read/Write signal from the Processor, however, the Write Execution flip-flop (WRT) (12D5) can only be updated when the timer has stopped; i.e., when TMG0 (12A4) is high.

The EBL and ARM flip-flops (12J7) are loaded from DL00 and DL01 as described in Table 21. They control the action of the Interrupt flip-flop (INTR) (12E8) and the interrupt line ATN0 (12G7).

The Block flip-flop (BLK) controls the serial feedback of data from the TTY receiver to the TTY driver. When reading a non-ASCII tape, it is inconvenient and undesirable to permit the received data to reach the printer/stunt box and operate the bell, line feed, form feed, etc., functions. This feedback is broken when the BLK flip-flop is set. Sending data to the TTY from the Shift Register is not affected by the BLK flip-flop.

The Busy (BSY) status bit is controlled by the Write Execution flip-flop not the WT flip-flop. The Break bit remains set as long as the Break key is depressed at the TTY. The Error bit (overflow) is cleared by either a Data Request, any command, or the system initialize signal SCLR0.

13.4 Timer Circuits

The timer consists of the control flip-flop (TMG) (12D2), a 440 HZ multi-vibrator MTA (12H3) and MTB (12K3), a two-stage clock counter MTC (12G4) and MTD (12H4), and a character counter (TA, TB, TC, and TD) (12L6). In the idle or reset state with the TMG flip-flop cleared, TMG1 (12D2) is low to disable MTA and MTB, to clear MTC and MTD, and to preset the character counter to the count of five.

When the TMG flip-flop is toggled set at the end of DAGA0 (12A2) in the Write mode; TMG1, TMG1A and DTMG1 all go high to enable the timer. The 440 Hz pulse train (MTB1) (12L3) drives the two-stage counter (MTC and MTD) and a decoder gate to generate the 110Hz train of clock pulses (CLK0 and CLK1) (12K4) and the shift pulses (SHFT1) (12N4). After the end of the ninth clock pulse, TB1, TC1, and TD1 are all high, thus forcing FSTP0 (12M5) low to terminate the train of shift pulses. During the eleventh clock pulse, EOC0 (12A2) goes low, and the TMG flip-flop is toggled clear on its trailing edge. This produces a train of eleven clock pulses and nine shift pulses having a period of 9.09 milliseconds (110Hz) with the trailing edge of the first pulse occurring 9.09 milliseconds (one bit period) after TMG is set. The pulse width is approximately 1.15 milliseconds (one-eighth of a bit period).

The idle timer is also started (by the direct set pulse ST0) (12E3) when the received Start bit arrives from the keyboard or tape reader or due to depression of the Break key. This is not dependent on the Read/Write mode since the BRK condition must be detected in both modes. The width of the ST0 pulse is determined by delay Capacitor 02HC1 (8G1) which generates the delayed TMG0 signal DTMG0 (12C4). Since the MTD flip-flop is direct set ST0, the first CLK/SHFT1 pulse occurs 4.545 milliseconds (half of a bit period) after the TMG flip-flop is set; the period of the pulses is still 9.09 milliseconds. Received data is sampled/shifted at the center of each bit. The TMG flip-flop is toggled clear at the end of the EOC and TTMG pulses as before.

13.5 Data Output

The TTY controller is in the Write mode when both the WT and WRT flip-flops are set. To send data to the TTY, the DAGA0 line (11A5) goes low to load DL00:07 into the Shift Register, clears the Start bit flip-flop (DRN) (11N6) and toggles set the Timing Gate flip-flop (TMG) (12D2). Note that if the timer was already running when the Data Available Control signal is received, the DAGA0 signal (11A5) would be blocked by TMG0 (12D2) low, no return SYN would be generated, and the false sync condition would be detected after 35 microseconds. For this reason the WDH instruction must not be used with the TTY controller.

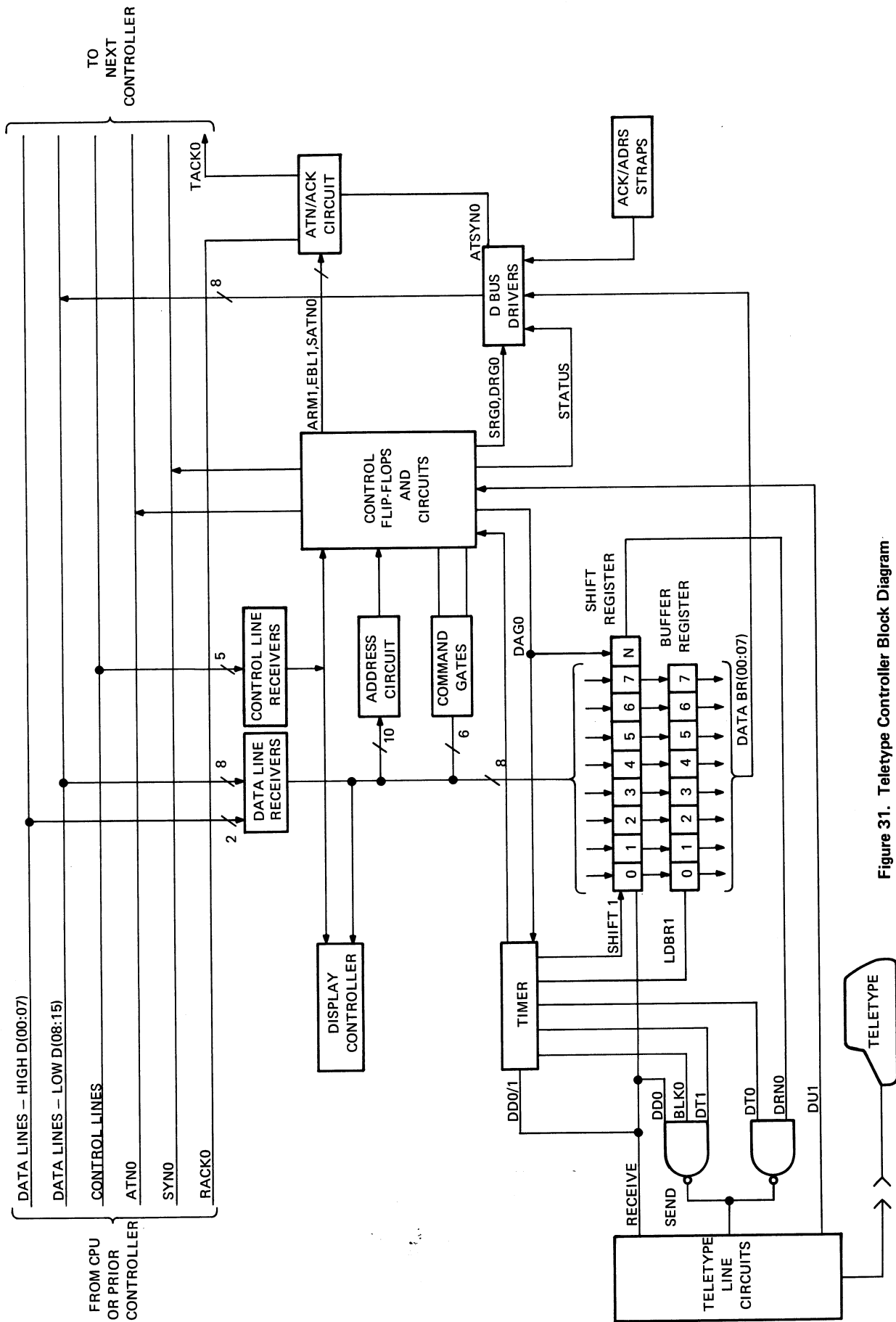


Figure 31. Teletype Controller Block Diagram

When the timer starts, shift/clock pulses are generated as described earlier and shown on Figure 32. The bit stored in the DRN flip-flop is connected to the transmit line (TNSB1) (16D2) by the high states on the device transmitting (DT0) and the TMG1 lines. Since the DRN flip-flop is initially cleared by DAGA0, TNSB1 goes low, and the gate driving TNS0 turns off to send the open-loop Start bit condition. At the end of each shift pulse, as the eight data bits are sequentially transferred into the DRN flip-flop, a high state at the serial input of the Shift Register (DX1) (11B2) gradually loads the register with all ones (including the DRN flip-flop).

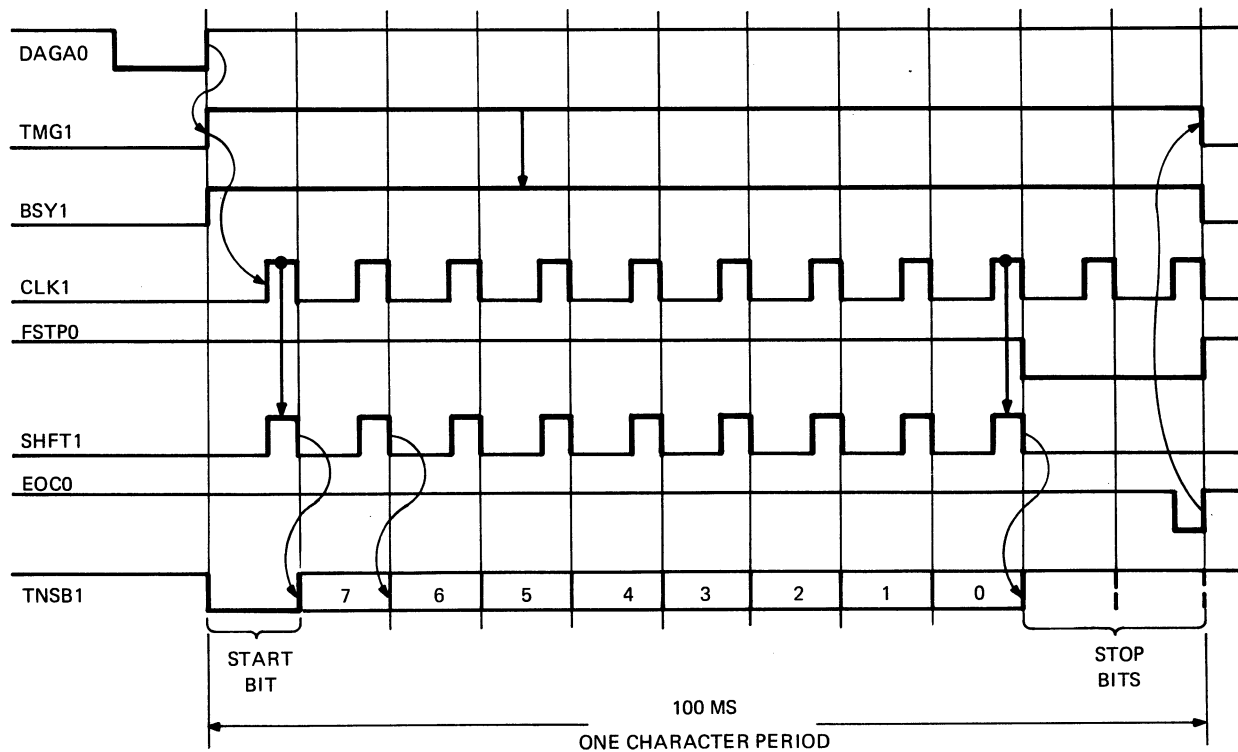


Figure 32. Write Mode (Output) Timing, Teletype

During the last two clock periods, after shifting has stopped, the ONE Level stored in the DRN flip-flop is sent out as the closed-loop Stop bit condition. The EOC pulse clears the TMG flip-flop to generate the closed-loop idle condition.

With the WRT flip-flop set, the status bit BSY1 (12D6) is active when TMG1 is active. Should a command which clears the WT flip-flop (Read mode) be received while the timer is running, the WRT flip-flop (and the definition of BSY status) does not change until the TMG flip-flop is cleared and TMG0 (8F8) gates WT1 into the WRT flip-flop.

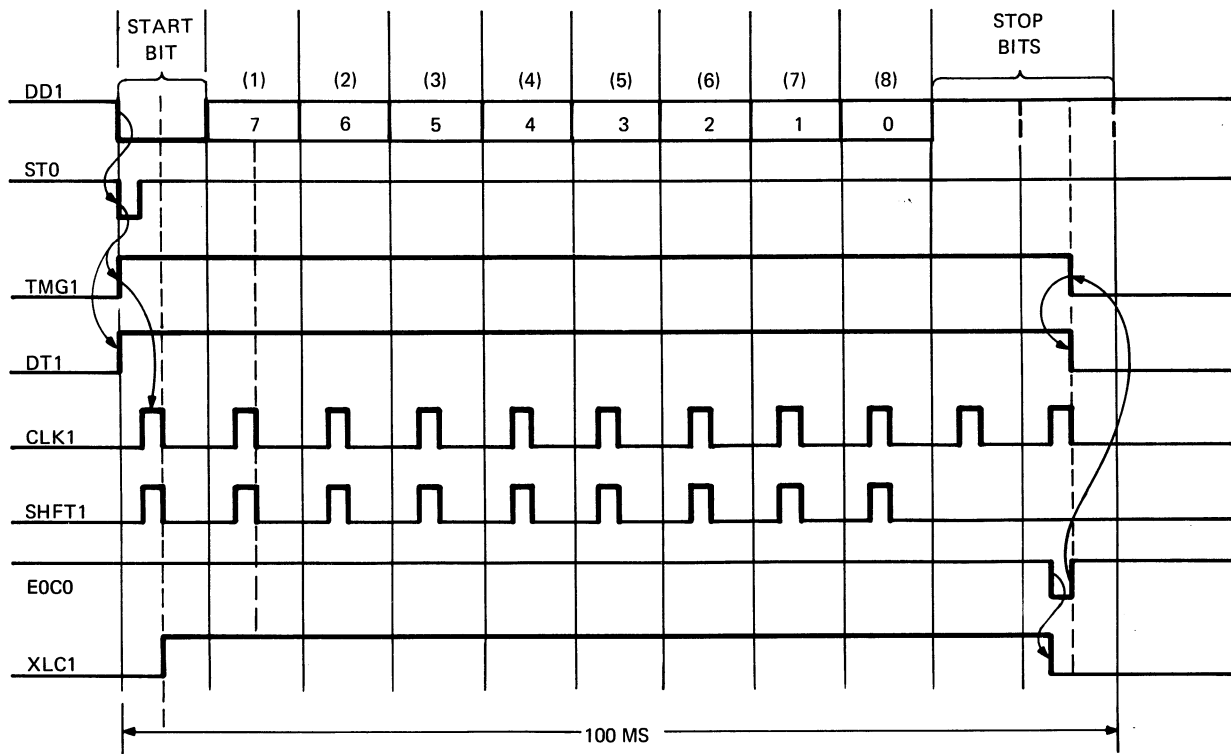
13.6 Data Input

The timer circuit can be started from the TTY receive loop in either the Read or Write mode as described in Section 13.4. This insures that the Break condition is always detected. However, serial data cannot enter the Shift Register (DX1) (11B2), unless the TTY controller is in the Read mode; i.e., the WRT flip-flop is cleared and WRT0 high. The Load Buffer Register pulses (LDBR1) (8G2) are generated only in the Read mode.

The Device Data line (DD1) (16G7) is high active when there is current flowing in the receive loop. This represents the logic ONE level and also the idle loop condition. The signal from the receive loop is filtered by an RC network (180 ohms/2.2 mfd) (16J8) and then reshaped by the Schmidt Trigger circuit (composed of a pair of inverters and two resistors) (1GE7) to generate the DD0 and DD1 signals.

When DD0 and DD1 first become active, the timer is started by the ST0 pulse (as described in Section 13.4) and the Device Transmitting flip-flop (DT) (11J6) is set. This flip-flop forces the TNSB1 line high and partially selects the TNSA1 gate, subject to a high level on the BLK0 and DD0 lines; i.e., the serial feedback circuit to the TTY Printer/Punch. The DT flip-flop also arms the Line Check flip-flop (XLC) (11M7) by placing a high level on the D input.

As seen on Figure 32, the XLC flip-flop is toggled set at the end of the first SHFT pulse. During the first SHFT pulse, the receive loop is checked to insure that the loop is still open; i.e., a legitimate Start bit has started the timer. If the loop is closed, DD1 is high and the Start Glitch pulse (GLTCH0) (12B2), is generated to clear the TMG flip-flop at the end of the SHFT pulse. The timer is reset, there are no EOC or LDBR pulses, the Buffer Active flip-flop (BA) (8E8) and the BSY status are unchanged.



Note: Bit Designations (X) are Paper Tape Channel Numbers.

Figure 33. Read Mode (Input) Timing, Teletype

The serial data at the Shift Register input (DX1) (11B2) is active when the DD1 line is active. The nine SHFT1 pulses move the received data into and along the Shift Register until the Start bit and the eight data bits occupy DRN and SR00:07. Shifting occurs at the end of each SHFT pulse; i.e., the center of each bit.

The TMG flip-flop toggles clear at the end of the EOC pulse and clears the DT flip-flop. The XLC flip flop is cleared by EOC if the loop is closed due to a Stop bit, DD1 high (16E7). In the case of a missing Stop bit (or Break condition), the XLC flip-flop remains set after the EOC pulse has cleared the TMG flip flop. The function TMG0·XLC1 causes BRK0 (8K7) to go low, and lines BRK1, EX1, and EX0 to become active. The timer cannot restart on the open loop condition since $ST0 = DTMG0 \cdot DDO \cdot XLC0$.

The BRK condition continues until the receive loop is closed. The $DD1 \cdot TMG0$ function then clears the XLC flip-flop.

In the Read mode, BSY1 (12D6) is low whenever the Buffer Active flip-flop (BA) (8E8) is set. The EOC1 pulse generates the LDBR1 pulse to load the Buffer Register and toggle set the BA flip-flop. The DRGA0 signal (8A9) clears the BA flip-flop when the buffer is gated to the D Bus. An overflow or error state exists if the LDBR1 pulse finds the BA flip-flop still set, the Overflow flip-flop (OV) (8G8) is then set. The OV and BA flip-flops are cleared by the DRG pulse, any CMG pulse, or the initialize signal SCLR0B.

13.7 Interrupt Circuit

The TTY controller generates an interrupt for a negative transition on BSY1. This transition toggle sets the DFBSY0 (12E7) flip-flop which in turn direct sets the INTR flip-flop (12E8). This forces ATN1 high and ATN0 low.

The Processor responds by executing an Acknowledge interrupt. When the TTY controller has first priority, the RACK0 lead goes low forcing RACK1 and DRACK1 high. With GATN1 high, the TACK0 gate is blocked and the ATSYN0 line goes low. This gates the controller address X'02' to the D Bus, generates the return SYN0, direct clears the DFBSY flip-flop, and clears the INTR flip-flop at the end of ATSYN0.

When the system uses the Memory Protect and/or the Real Time Clock controllers, the RACK0/RACK0 daisy-chain is wired to the higher priority controllers before it reaches the TTY over the back panel.

As noted, in Section 13.3, the Disable command clears the EBL flip-flop forcing the EBL1 (9H7) and GATN1 lines low. Interrupts may be queued by setting the INTR flip-flop. The Disarm command forces the ARM 1 lead (9H8) low to clear the INTR flip-flops and hold them clear; interrupts are not queued.

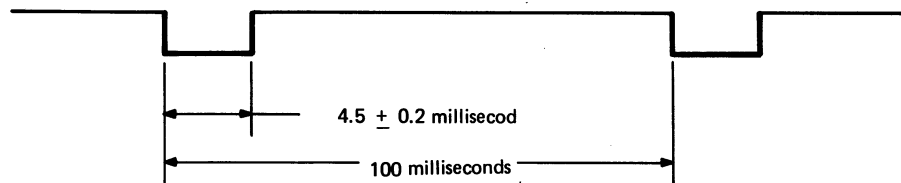
13.8 Initialization

The system initialize signal SCLR0 (15K2) conditions the TTY controller by setting the BLK flip-flop and clearing all other control flip-flops. This presets the controller in the Read mode with interrupts disabled.

13.9 TTY Timer Adjustment

The only adjustment on the TTY controller controls the frequency of the 440Hz timing multivibrator. The adjustment is made in the following manner:

1. Initialize the system.
2. Connect an oscilloscope to TP-TMG1A (located at the stake near Connector 2).
Vertical scale: 2 volts/centimeter
Horizontal scale: 1 milliseconds/centimeter
Sync: internal, negative
3. Generate a continuous stream of data from the TTY by reading a tape or by the Repeat function of the keyboard.
4. Adjust Potentiometer at location 14R (next to the test point TMG1A) for the waveform shown below.



13.10 Machine Control Register (MCR) (Sheet 7)

A Machine Malfunction (MMF) interrupt is generated when Bit 11, 12, 13, 14, or 15 of the Machine Control Register (MCR) is set (7G4). The MCR bits are assigned and gated (with the SMCR function) as indicated in Table 22.

The CMCR function clears MCR11:15 where there are ONES in B27:31. The system Initialize (SCLR0) clears MCR10:15 – the straps are not affected. The SMCR function is described in Section 11.2.

13.11 Power Monitor and System Initialize

All circuits for the Power Monitor are on Sheet 15. The master reset signal SCLR0 (15F2) is active when the Initialize Relay K1(15B9) is de-energized. During normal operating conditions, all voltages are present and the POWDN0 line (15G7) is high. This allows the delay transistor and the Darlington circuit (15A8) to conduct. As long as these transistors conduct, the Initialize Relay K1 remains energized and the SCLR0 line is held high to +5 volts by a resistor (15E2).

If any of the four items listed in Section 10.1 (CL070) occur, the STPF1 line (14J2) goes high and starts the one millisecond EPF timer (14K2). The leading edge of EPF0 (14L2) sets Bit 15 in the MCR (7G2), generating a Machine Malfunction (MMF) interrupt. In response to MMF, the user has an opportunity to do any necessary system resetting and data storage.

TABLE 22. MCR BIT ASSIGNMENT

BIT	MNEMONIC	MEANING	S-BUS	CONDITION CODES
MCR15	EPF	EARLY MF	S31 and	LFC0
MCR14	IRMP	INSTRUCTION PARITY FAIL	S30 and	GFC0
MCR13	DMPFO	DATA PARITY FAIL	S29 and	VFC0
MCR12	APF	AUTO DRIVER PARITY FAIL	S28 and	CFC0
MCR11	STF	STRT TIME OUT FAIL	S27	
MCR10	CATN	CONSOLE ATTENTION	S26	
MCR09	RSTS	REGISTER (STRAP)	S25	
MCR08	SPARE	(STRAP)	S24	
MCR07	SNGLI	DISPLAY CONTROLLER SNGL F-F	S23	
MCR05	BNK B	BANK B (STRAP)	S21	
MCR04	BNK A	BANK A (STRAP)	S20	

At the end of the one millisecond EPF delay, the trailing edge of EPF0 (14G5) toggle sets the Primary Power Fail flip-flop (PPF) (14H5) causing the PPF interrupt (14K6) to be sent to the CPU and a low active signal on CL070 (14K6). PPF1 also starts another one millisecond timer XPF (14K5). When the PPF interrupt is detected, the micro-program stores the PSW and register stack in the main memory and sends the POW function to the IOU. The Stop flip-flop (STP) (14N5) is either toggled set by the trailing edge of XPF1 or direct set by FPOW0 (14N4), whichever occurs first. When STP1 goes high, POWDN0 goes low to turn off the transistors of the Darlington circuit and de-energize the Initialize Relay K1. The GSTP1 lead (15J7) is normally high. It is unused except in some multi-CPU systems.

Loss of AC or DC power also de-energizes the relay. POWDN0 goes low when the -15 volt input (N15) (15D8) to the inverter is lost. The Darlington circuit cannot operate the relay if either the +5 volt collector supply (P5) or the +15 volt base supply (P15) (15B6) is missing. Should the AC input (AC1 and AC3) (15B2) be too low or missing, the Power Fail Detector circuit removes the base drive to the Darlington circuit.

If the AC input is lost (or fluctuates enough) the potential at the base of 02BQ3 becomes more negative, 02BQ3 conducts and supplies base drive to 02BQ1. The 4.7K resistor (15K4) provides positive feedback from 02BQ1 to 02BQ3 causing these transistors to turn on. The emitter voltage of 02BQ3 drops, 02BQ4 turns off, 02BQ2 turns on and commences to discharge the delay capacitors (15B6). With 02BQ1 conducting, its collector voltage approaches ground and generates the low active signal PFDT0 (15K4). As described earlier, this starts the sequence which puts a low level on POWDN0 and completes the capacitor discharge. The Darlington circuit has no base drive so the relay is deenergized.

The Initialize Relay K1 is a dry reed unit with Single Pole Double Throw contacts. The normally closed contact of the de-energized relay (K1) provides a metallic ground on the system Initialize line (SCLR0 (15K2).

For a sequence due to POFF, LSU, INITO, EXA0, or EXBO low (14A2) clearing STP allows the POWDN0 lead to go high and the delay capacitors (15B6) to charge slowly through the base resistors of Transistor 03AQ1. When the threshold of the Darlington circuit is reached, the circuit conducts and the Initialize Relay K1 is energized thereby removing the ground from the SCLR0 line.

In the case where Initialize is caused by a failure to P5, N15, P15 or the AC supply, the Initialize Relay K1 de-energizes and remains in that state until the fault is corrected.

13.12 Primary Power Fail Check

The Primary Power Fail Detector is located on the IOU board. The circuit is checked and adjusted as follows:

1. Connect the Primary Power Cord of the CPU power supply into a variable voltage source (Variac or equivalent).
2. With the line voltage set at the nominal value of 115.0 VAC, turn the Power on.
3. Adjust Potentiometer at location 00R to generate the Power Fail condition of CL070 (back panel Terminal 122-0) when the AC line voltage is set for 103.5V (i.e., 10% low). System Initialize line (SCLR0) Terminal 105-0 should become low active in less than 2 milliseconds after STPIA (Test Point 110-4) goes active.
4. With nominal line voltage, load the Model 8/32 Test Program and depress the RUN Key. While the program is running, remove the AC line cord from the primary power source.

NOTE:

The TTY will run-open if connected into a different power source.

5. Connect the AC line cord back into the power source. The TTY should stop cycling. Depress the EXECUTE switch and the test program should continue to run.
6. Repeat Step 4, but turn the Console Power switch OFF instead of removing the AC line cord.

13.13 Start Timer

The Start Timer circuit is shown at location 15E6. With the timer-kill (KSTM1) (15C7) in its normally low state, the ungated STRT0 signal (15B8) enables the 30 microsecond timer STMA (15E6). When the selected module generates a MFIN0 (15H8) signal, then it clears the timer and disables the timer flip-flop (STMB) (6F8).

The STRT Timer (30 microseconds) is activated whenever the CPU sends the STRT signal to the various system modules (ALU, FAU, IOU, etc.) and is cleared by the MFIN signal from the module addressed by the MSEL (00:02) lines. Should the time out occur before the MFIN signal arrives, one of the two things happen.

1. On non-MUX Bus operations, Bit 11 of the MCR is set, a pseudo MFIN signal restarts the CPU clock, and the MMF interrupt is generated.
2. In the case time out occurs during a MUX Bus operation, the MCR is unchanged, the False Sync code (0100/CVGL) is placed on both Condition Code Busses and a pseudo MFIN restart the CPU clock. If the MUX Bus operation happens to be of the sense type, X'004' is also gated onto S(16:31).

14. S AND D BUS ROM CONTROLLERS

14.1 S Bus High ROM Controller (19-142F45)

000-007	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
008-015	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
016-023	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
024-031	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
032-039	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
040-047	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
048-055	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
056-063	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
064-071	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
072-079	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
080-087	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
088-095	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
096-103	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
104-111	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
112-119	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
120-127	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
128-135	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
136-143	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
144-151	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
152-159	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
160-167	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
168-175	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
176-183	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
184-191	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
192-199	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
200-207	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
208-215	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
216-223	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
224-231	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
232-239	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
240-247	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78
248-255	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F78

14.2 S Bus Low ROM Controller (19-142F46)

000-007	LALL LALL HLLL HLLL HLLL HLLL HLLL HLLL	19-084R00F79
008-015	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
016-023	LALL LALL HLLL HLLL HLLL HLLL HLLL HLLL	19-084R00F79
024-031	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
032-039	HLLL HLLL HLLL HLLL HLLH HLLH HLLH HLLH	19-084R00F79
040-047	LLLL LLLL LLLL LLLL LALL LALL LALL LALL	19-084R00F79
048-055	HLLL HLLL HLLL HLLL LLLL LLLL LLLL LLLL	19-084R00F79
056-063	LLHH LLHH LLHH LLHH LLLL LLLL LLLL LLLL	19-084R00F79
064-071	LALL LALL HLLL HLLL HLLL HLLL HLLL HLLL	19-084R00F79
072-079	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
080-087	LALL LALL HLLL HLLL HLLL HLLL HLLL HLLL	19-084R00F79
088-095	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
096-103	HLLL HLLL HLLL HLLL LALL LALL HLLH HLLH	19-084R00F79
104-111	LLLL LLLL LLLL LLLL LALL LALL LALL LALL	19-084R00F79
112-119	LLLL LLLL LLLL LLLL LALL LALL LALL LALL	19-084R00F79
120-127	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
128-135	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
136-143	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
144-151	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
152-159	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
160-167	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
168-175	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
176-183	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
184-191	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
192-199	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
200-207	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
208-215	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
216-223	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
224-231	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
232-239	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
240-247	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79
248-255	LLLL LLLL LLLL LLLL LLLL LLLL LLLL LLLL	19-084R00F79

14.3 D Bus ROM Controller (19-142F47)

000-007	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
008-015	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
016-023	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
024-031	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
032-039	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
040-047	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
048-055	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
056-063	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
064-071	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
072-079	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
080-087	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
088-095	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
096-103	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
104-111	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
112-119	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
120-127	HHHH HHHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
128-135	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
136-143	HHHH HHHH LHHH HHHH HHHH HHHH LHHH HHHH	19-084R00F80
144-151	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
152-159	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
160-167	HHHH HLHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
168-175	HHHH HLHH LHHH HLHL HHHH HHHH LHHH HLHL	19-084R00F80
176-183	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
184-191	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
192-199	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
200-207	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
208-215	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
216-223	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
224-231	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
232-239	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
240-247	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80
248-255	HHHH HHHH HHHH HHHH HHHH HHHH HHHH HHHH	19-084R00F80

15. EXTENDER BOARD OPERATION

The following steps must be taken to insure proper Extender Board operation when troubleshooting any of the Processor boards on the Extender Board.

1. Remove the 35-537 CPB (lower Slot 7) to modify the CPU Clock speed. An octal switch is located in IC position 15E for this purpose. Switch positions 4 and 8 are to be ON for Extender Board operation. No other switch positions may be placed in the ON position.
2. Place the 28-015 Extender Board in the card file slot of the board to be tested. Note that there are two sets of backplane pins on the Extender Board. Plug the board to be tested into the upper set of pins and plug the Extender Board terminator card (see Step 3) into the lower set of pins.
3. One of two terminator cards must be used for Processor Extender Board operation. The 35-598 Terminator is to be used with the CPA board on the extender, and the 35-599 Terminator is to be used with any of the following: CPB, CPC, ALU, and IOU. The terminator is to be installed as described in Step 2.
4. For Extender Board operation of the CPA, CPB, or CPC, 24 inch extender cables (17-362 and 17-363) are required.

16. MNEMONICS

The following lists provide a brief description of each mnemonic found in the Model 8/32 CPA Schematic Drawing 35-536D08, CPB Schematic Drawing 35-537D08, CPC Schematic Drawing 35-555D08, ALU Schematic Drawing 35-538D08, and IOU Schematic Drawing 35-539. The source of each signal on the respective schematic drawing is also provided.

16.1 CPA Mnemonics, Schematic Drawing 35-536D08

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
ADA281:311	Add-one-loop outputs	Sheet 3
AEQB1	Segment number equality	6E1
B000:310	B Bus	Sheet 13
BDLY0	Base selection delay	6J5
BIT161A } BIT161B }	Bit 16 propagate signal	12N9
BMXNA1	B Mux Enable A – S Bus	12N7
BMXNB1	B Mux Enable B – MDR	12L5
BMXNC1	B Mux Enable C, MLC/CA	12N5
BMXSLA1	B Mux Select Line A – Halfword MDR	12J8
BMXSLB1	B Mux Select B – MLC	12L5
BR040:270	Base Register outputs	Sheet 6
BRWR1	Base Register Write Command	7N7
BSEL001:041	B Bus Source Address	12J4
C001 } C010 } C011 }	Carry Commands	Sheet 1
C3X0	Carry past segment boundary	12G8
CA310	Address Bit 31	3B8
CA120:300	Memory Address Bus	Sheet 8
CACLR0	Buffered Clear	1N8
CC0	Second HW Clock	1F9
CD000:310	Memory Data Bus	Sheet 11
CDW0	Write Conversion Command	7M5
CK1A	System Clock	4M3
CLINT0	Clear Interrupt flip flop	7R8
CLOCK0	System Clock	4A1
CMC000:020	Memory Command Bus	Sheet 2
CPC011/001	Increment Commands	2F5
CRDY0	Memory Ready (response)	1A2
CREQ0	Memory Request	1R2
CS000	Control State 0	2G8
CSTA0	Clear Status Register	7R8
DREQ0	Data Request	1R7
FSR0	Status Register flip flop	7G8
GT0/1	Greater Than segment limit	12G7
INCR021	Increment MLC by Two HW	10M7

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
INCR1	Increment MLC by One HW	10M6
IR1	Instruction Read decode	2K7
IR271:311	Status Register outputs	Sheet 7
IREQ0	Instruction Request	1R5
LDMAR0	Load MAR	2C7
LDMCLO	Load MLC	2B6
LDMDR0	Load MDR	2D8
LDUIR0	Load UIR Clock	4M2
LIME1	Limit violation	6N6
MAI0	Memory Access Interrupt	7R4
MALX120:230	Address Multiplexor outputs	Sheet 8
MAR121:311	Memory Address Register outputs	Sheet 3
MARPU1	Pull up resistor	Sheet 3
MC000:030	Microcontrol field	Sheet 2
MCLK0	Memory Data Clock	4F3
MDR001:311	Memory Data Register outputs	Sheet 4
MDRCL0	Memory Cycle Clock	1B2
MDX001:151	MDR input Multiplexor	Sheet 11
MLC121:311	Memory Location Counter outputs	Sheet 3
MSIG0	Module Signal	10R5
NW1	Write Inhibit	7M2
PROT1	Enable Protect/Relocate	2G4
PSW210	Program Status Word Bit 21	2D4
RGEN0	B Bus Register Enable	12N6
RH0	Read Halfword Command	2F8
RI02Q	RI2 format	10E3
RQFF0	Request flip flop	1N2
RRSF1	RR or SF format	10D3
RSTR/0	Read Status Register	7R9
RX001	RX format	Sheet 10
RX2F0/1	RX2 flip flop	10K4
RX3D0	RX3 Format decode	10R5
RX3EN1	RX3 decode enable	10K5
RX3F1/0	RX3 flip flop	10K5
RXID0	RX1 decode	10M3
RXIL0	RX3/RI2 format	10M5
S000:150	Part of S Bus	Sheet 5
S160:310	Part of S Bus	Sheet 5
S2B0	S Bus to B Bus Override Command	12G6
SCLR0	System Clear	1J8

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
SDR1	Data Read Command	2N2
SDRDW1	Data Read/Write Command	2S5
SDW1	Data Write Command	2N2
SEGWEA/B	Base Register HW Write	Sheet 6
SIR1/0	Instruction Read Command	2J6
SMC000:030	Buffered MC field	Sheet 2
SR000:310	S Register outputs	Sheet 5
SR001:311	S Register outputs	Sheet 5
SRCK0	Status Register Clock	7C5
SRTR0	Status Register Trap	7N6
SSEL001:041	S Bus Address Select Bus	2A7
STB0	Strobe	12M2
SM2X121:311	Summer two outputs	Sheet 9
SX280:310	Second Index Register Address	Sheet 11
UDR280:310	User Destination Register Address	Sheet 5
UIR240:310	User Instruction Register	Sheet 5
USR280:310	User Source Register Address	Sheet 5
XPU1	Pull up resistor	6E5

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
A140:270	A Bus Bits 14:27	Sheet 5
A280:A310	A Bus Bits 28:31	Sheet 4
AENO	Enables ASEL Multiplexor	3F5
ASEL001:041	Selects Register containing A Operand	Sheet 3
ATN000:030	Interrupt Attention lines	Sheet 12
AYDS1/0	ASEL Multiplexor select line	3E3
B000:310	B Bus Bits 00:31	Sheet 6
B280:310	B Bus Bits 28:31	Sheet 2
BALAO	Branch and Link, ARM Interrupts	14J9
BDC0	Branch and Disarm Console Interrupt	14J8
BENO	Enables BSEL multiplexor	3J7
BSEL001:041	Selects Register containing B operand	Sheet 3
BYDS0	BSEL Multiplexor Select line	3K8
C1X071:141	ROM Address: Traps or Op-Code pointers	Sheet 8
C2X101:151	ROM Address: B Bus or ROM Instruction Register indirect field	Sheet 8
CCC0	Carry Condition Code	4D6
CCCLK0	Condition Code Clock loads PSW	4M7
CFLG1	Carry flag	4F6
CLK0	CPU Clock	13N4
CLK1A	CPU Clock	13N4
CLK1B	CPU Clock	13N3
CS000:030	CPU Control States	Sheet 14
CSA041:091	ROM Address	Sheet 9
CSA1/0	Counter State A	14M2
CSA100:150	ROM Address	Sheet 8
CSB1/0	Control State B	14M3
CSD001:311	ROM data	Sheet 11
CSREF0	Denotes control store reference	5N3
CSWRT0	Control Store Write	11G4
D1	Decode Bit	11N6
DREQ0	Data Request	13A3
DSPLY0	Display Interrupt	12A4
E1	Execute Bit	11N5
ENFLG1	Enable PSW flags	11R3
ENPG00:40	Page enable for ROM	Sheet 10
ENSMX0	Enables S Bus multiplexor	7F9
ENYSDX0	Enables YSI/YDI to B Bus	3S5
EXEC1/0	Execute Bit of micro-code	5N1
FLGCLK0	Flag Clock latches Condition Code	4K8

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
FSEL000:030	Function Select lines	Sheet 11
FYX0	User index enable	3D5
GBIM0	Gate Immediate field to B Bus	6D9
GCC0	Greater Than Condition Code	4D8
GEN0	Generate interrupt	12K4
GFLG1	Greater Than flag	4F8
GPSWA0	Enables PSW selection	3M3
ILEGA	Illegal Instruction	12K1
ILEGB	Illegal Instruction	12K2
ILEGC	Illegal Instruction	12K2
INCLK0	Clock Inhibit	13H1
IREQ0	Instruction Request	13A1
INT1/0	Interrupt present	12N6
JB041:151	Jam address Bits 04:15	Sheet 1
JUTY1	Discriminates between RX1, RX2, or RX3 Instructions	14A1
KLCLK0	Disables CPU clock for manual testing	1R8
KSIG0	Extension of FSEL field	11N5
LCC0	Less Than Condition Code	4D9
LFLG1	Less Than flag	4F9
MAI	Memory Access Controller	12B8
MB041:151	Match Address Bits 04:15	Sheet 1
MC000:030	Memory Control field	Sheet 11
MMF0	Machine Malfunction interrupt	12B5
MNCLK1/0	Manual Clock (P.B. switch)	Sheet 1
MOD000/001	Module zero	5M1
MPEN0	Memory Protect enable	12H9
MSEL000:020	Module Select lines	Sheet 11
MSIG0	Module signal	14A1
MTCH	LED indicator signals address match	1S6
MTCH1	Stored XMATCH1	1M6
PASS1/0	Do not take branch	14K8
PCLK0	CPU Clock	13N3
PPF0	Primary Power Fail	12B6
PRIV	Privileged instruction	12K1
PSW141:271	PSW Bits 14:27	Sheet 5
PSW281:311	PSW Bits 28:31	Sheet 4
PSWCLK1	PSW Clock	4K9
RIR000:310	ROM Instruction Register	Sheet 11
RIR201A	Bit 20; ROM Instruction Register	6D9

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
RIRCLK0	ROM Instruction Register Clock	11H8
RLC041:151	ROM Location Counter	Sheet 9
RLR041:151	ROM Location Register	Sheet 9
RRXINH0	Inhibits transfer in RRX micro-instructions	14H1
RUN 1/0	Run mode	Sheet 1
RX3D0	RX3 instruction	2H3
S00:310	S Bus Bits 00:31	Sheet 7
S2B1	Gates S Bus data to B Bus	14G4
SAMA1	ROM Address Select line	5R4
SAMBL1	ROM Address Select line	5N5
SAMBM1	ROM Address Select line	5N4
SAMC0	ROM Address Select line	5N2
SCC0	Signals new Condition Code available	4G8
SCLRI/0	System Clear	14A5
SETRLC0	Sets CPU to CS031 as a result of JAM	1M5
SIN0	Single Step Clock switch:normally open contacts	1J8
SINC	Single Step Clock switch:normally closed contacts	1J9
SLMDR1	Select MDR	3E8
SLYDD1	Select YDD	4M2
SPSW1	PSW Select line	4M3
SR280:310	Status Register Bits 28:31	Sheet 4
SRCLK1	Status Register Clock	4H3
SSEL011:041	Destination Register Select lines	Sheet 4
STRT1/0	Module Start Signal	14M5
SX280:310	Second Index field	Sheet 2
SX2NZ1	Secondary Index field is non-zero	3E8
SYNC-TP	Test Point:Match Address	1N6
TENO	Trap Address enable	12J4
TKILL0	External TP for inhibiting clock	13M1
TRAP121	Interrupt Trap Bit 12	12H8
TRAP130:150	Interrupt Trap Bits 13:15	Sheet 12
UIR240:310	User op-code	Sheet 8
USR280:310	User Source Register Select lines	Sheet 2
VCC0	Overflow Condition Code	4D7
VFLG1	Overflow flag	4F7
XMTCH1	ROM Address compares to Match Address	1H6
XS010:040	Destination Register Address	Sheet 4
YDCLK0	User Destination Register Clock	2E2
YDP1F0	YDP1 enable	3E3

MNEMONICS

MEANING

SCHEMATIC
LOCATION

YDX281:311

User Index field

Sheet 2

YS280:310

User Destination Register

Sheet 2

YSIX0

Selects YSI/YDI to B Bus

3S5

16.3 CPC Mnemonics, Schematic Drawing 35-555D08

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
A000:310	A Bus	Sheet 1
AAD000:040	A Stack Address Bus	Sheet 4
AAD051	A Stack Address Bit 5	Sheet 4
AKL0	A Stack PSW suppress	4G1
ASEL001:041	A Bus Select Bus	Sheet 4
ASTKNO	A Stack enable	4K4
B000:310	B Bus	Sheet 2
BAD000:040	B Stack Address Bus	Sheet 4
BAD051	B Stack Address Bit 5	4L2
BKL0	B Stack PSW suppress	4G2
BSEL001:041	B Bus Select Bus	Sheet 4
BSTKNO	B Stack enable	4K5
M37X0	Floating-Point Module Select	4A1
PSW260	PSW Bit 26	5H6
PSW270	PSW Bit 27	5H7
RWC0	Read/Write Control	5A2
S000:310	S Bus	Sheet 3
S2B0	S Buffer to B Bus Over-ride command	4A4
S37X0	Stored floating point Module Select	4A3
SB001:311	S Buffer outputs	Sheet 3
SODD0	S Bus Odd Register command	5F2
SSEL001:041	S Bus Select Bus	Sheet 4
SSELX0	Stack Load Select	4G4
STWRT1	Start Write command	5A3
WSEL1	Write Select	5K4
WSEL1B	Write Select buffered	5N9
WCLK0	Write Clock	5H5
XCLK0	Buffered Clock	5C4

16.4 ALU Mnemonics, Schematic Drawing 35-538D08

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
A000:310	A Bus	Sheet 10
ACLK1/0	ALU Clock	Sheet 6
ACLKA/B0	ALU Clock	Sheet 6
ACNT051	Bit 5 of iteration counter	4N4
ACRY1	Carry from iteration counter	4R4
AGL001	Bit 0 A input to shift left multiplexor	8R7
AGL311	Bit 31 A input to shift left multiplexor	8H8
AGR001	Bit 0 A input to shift right multiplexor	Sheet 8
AGR081	Bit 8 A input to shift right multiplexor	Sheet 8
AGR161	Bit 16 A input to shift right multiplexor	Sheet 8
AL001:311	A latch outputs	Sheet 10
ALOG1/0	Logic mode	9M8
ALS080:110	Shift multiplexor outputs Bit 8:11	Sheet 12
AM001:311	A multiplexor outputs	10C5
AMOD001:031	Function select control for ALU	Sheet 9
ARITH1/0	Arithmetic shift	1L6
AS000:030	Arithmetic State	Sheet 5
AS001	Arithmetic State	5F8
ASA1	Arithmetic State register A	5N5
ASB0	Arithmetic State register B	5N5
ASIGN1/0	Stored sign of A Bus operand	3G4
AWC1	Add with Carry instruction	1L2
AXB1	Stored Exclusive-OR of Sign bits of A and B operands	6E3
B000:310	B Bus	Sheet 10
BG001:311	B gate outputs	Sheet 10
BGTR1/0	B Operand is Greater in CAE instruction	6D2
BSIGN1/0	Stored Sign of B Bus operand	6E3
CAE1/0	Floating Point Compare and Equalize instruction	1L1
CCC0	CC Bus – C bit (carry)	2N8
CCCLK0	Condition Code Clock	2E3
270	} Carry in Bit 27	} Sheet 9
230		
190		
CIN 030		
150		
110		
310	} Carry in Bit 31	

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
COUT000	Carry out Bit 0	9H9
COUT080	Carry out Bit 8	9E8
DFLT0	Divide Fault	3E2
DV1/0	Divide instruction (fixed or floating point)	1L7
EA0	Floating Point Add instruction	1F2
EAS1	Floating Point Add6Subtract instruction	1L2
ECOUT0	Exponent carry	8D1
ED1/0	Floating-Point Divide instruction	1L1
EC1	Floating Point Compare instruction	1L3
EM1/0	Floating Point Multiply instruction	1L4
EMD1/0	Floating Point Multiply/Divide instruction	1L4
ES0	Floating Point Subtract instruction	1F2
FAXB1	Stored Exclusive-OR of A and B Sign bits	9N6
FD1/0	Fixed Point Divide instruction	1L8
FM1	Fixed Point Multiply instruction	1L8
FMD1/0	Fixed Point Multiply/Divide instruction	1L8
FSEL000:020	Function Code from CPU	Sheet 1
FSTCNT1/0	First Count of arithmetic state 2	6F8
FXS011	Bit 1, exponent sum	8E6
FXS021:071	Exponent ALU outputs	Sheet 8
G001	Carry generate Bit 0	10F2
G041	Carry generate Bit 4	11F2
G081	Carry generate Bit 8	12F2
G121	Carry generate Bit 12	13F2
G161	Carry generate Bit 16	14F2
G201	Carry generate Bit 20	15F2
G241	Carry generate Bit 24	16F2
G281	Carry generate Bit 28	17F2
GATECC1	Gate Condition Code	3G3
GATEEC1	Gate Floating point Condition Code	3R7
GCC0	CC Bus - G bit (greater than)	2R1
GLOW1	Carry generate Bits 16 to 31	9G3
GNP0	Shift Multiplexor Output control	Sheet 7
GNM0		
GN0		
GRWC0	Generate Read Write control	6K8
GX0	Shift Multiplexor output control	7H8
GXLSB0	Shift Multiplexor output control	7H8
INHA1	Inhibit A Bus	7E6

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
INHBO	Inhibit B Bus	7E6
INHX1/0	Inhibit exponent	7H5
KSIG0	Function Code Extension bit from CPU	1A7
LCC0	CC Bus – L bit (less than)	2R1
LOG1	Logical shift	1L5
M1/0	Multiply look-ahead bit	7R6
MDO	Multiply/Divide instruction (fixed or floating point)	1L7
MFINO	Module Finish signal	2J4
MQ001:311	MQ register outputs	Sheet 10
MQCLK0	MQ register Clock	6R7
MQG001	Bit zero input of MQ Shift register	7S8
MQG311	Bit 31 input of MQ shift register	7S9
MP0	Multiply instruction (fixed or floating point)	1L7
MSEL000:020	Module Select code from CPU	Sheet 1
MSIG0	Module Signal (ALU=carry flag)	2R2
NLRZ0	Normalize	5C4
OCMP1	Control signal (one's complement) for CAE instruction	8E2
OCMPL1	Stored Control Signal – one's complement	6E3
OFL1/0	Exponent Overflow	4E2
P001	Carry propagate Bit 0	10F2
P041	Carry propagate Bit 4	11F2
P081	Carry propagate Bit 8	12F2
P121	Carry propagate Bit 12	13F2
P161	Carry propagate Bit 16	14F2
P201	Carry propagate Bit 20	15F2
P211	Carry propagate Bit 24	16F2
P281	Carry propagate Bit 28	17F2
PLOW1	Carry propagate Bits 16 to 31	9G2
ROT1/0	Rotate shift	1L6
ROTR0	Rotate Right shift	1L5
RWC0	Read Write Control signal to CPU	6M8
RWCA1/0	Read Write Control	6F7
RZR01	Remainder Zero flip flop	7G1
S001:311	ALU sum	Sheet 11
S000:310	Sum Bus	Sheet 10
SAP1/0	Shift multiplexor select control	7H4
SBGTR1/0	Set B Greater flip flop	3H3
SC010:050	Arithmetic Shift Count	Sheet 4
SELA0A/B	A multiplexor select control	8R5
SEP1/0	Shift Multiplexor select control	7E5
SETZA0	Inhibits Bits 0:8 in detection of zero sum (ZSUM1)	1OH2

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
SGAETO	S Bus enable	7M2
SGCC0	Set Greater than Condition Code	3N7
SGR001	Bit 0 S input to shift right multiplexor	8G9
SHFT1/0	Shift instruction	1L9
SL1	Shift Left control	7N4
SLCC0	Set Less than Condition Code	3N7
SOFLO	Set exponent overflow	8J8
SR1	Shift Right control	7N4
STRTO	Start signal from CPU	1A9
SUFLO	Set exponent underflow	8J8
SUM1/0	Add/Subtract mode	9K4
SVCC0	Set Overflow Condition Code	3L5
SWC0	Subtract with Carry instruction	1L2
TDFLT1	Toggle Divide Fault	6G9
UFL1/0	Exponent Underflow	4G6
VCC0	CC Bus – V bit (overflow)	2R2
XFR0	Forces S=A on ALU function control	9M5
XLOAD0	Load pulse for exponent up/down counters	8G7
XOVF1	CAE Instruction; exponential difference is greater than five	8R4
XRPA/B/C/D	Pullup resistor for unused logic inputs on IC's (1k ohm to P5)	7G1
SX011:071	Exponent result (stored)	Sheet 8
XSIGN1	Sign of floating point result	3H6
ZSUM1/0	Sum is zero	10J4

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
A1	Output of address decoder for TTY	9H3
A160:310	A Bus low	Sheet 3
AC1/3	12 VAC inputs to Primary Power Fail Detector	15B2
ACK000:030	Acknowledge interrupt MPX Channel	Sheet 6
ADA1	Address flip-flop for TTY controller	9M3
ADB1	Address flip-flop for Display	9M4
ADRS0	Address Control line, MPX channel	6N4
ADSYNA0	Address SYNC for TTY controller	9L2
ADSYNB0	Address SYNC for Display controller	9L4
ARM1	Interrupt Arm flip-flop for TTY	9J7
ATN0	Interrupt Attention for TTY	12G7
ATSYN0	Attention SYNC pulse for ACK address	12N9
B1	Output of address decoder for Display	9H4
B160:310	B Bus low	Sheet 3
BA1/0	Buffer Active flip-flop (sets when buffer is loaded, cleared when buffer unloaded)	8E8
BLK1/0	Serial feedback block flip-flop	9N6
BR001:071	Buffer Register-eight stages, (active only in Read mode)	Sheet 11
BRK0	Break detect signal status Bit L	8H7
BSY1/0	Busy signal (Status Bit 4)	
CA310	Least Significant Bit of address from CPU (byte steering bit)	1B5
CATN1/0	Console Attention flip-flop	10G5
CCC0	CC Bus - C Bit	7R4
CL070	Primary Power Failure Control line	14K7
CLDR0	Clear line for D Bus receivers	5K5
CLK0/1	Timer clock pulses (11 for character)	12L4
CLRA0	Clear line for cycle counter	5R7
CLRC0	Clear line for timing, flip-flop	5H8
CLRST0	Clear ST flip-flop	12E4
CMCR0	Clear MCR11:15	7C7
CMCR110	Clear lines for Machine Control Register	Sheet 7
CMCR130:150		
CMD0	Command Control line, MPX channel	6N4
CMGA0	Command line for TTY controller	10N4
CMGB0	Command line for Display controller	10N5
D000:150	D Bus	Sheet 4
DA0	Data Available Control line, MPX channel	6N4
DAGA0	DA line for TTY controller	10N1

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
DAGB0	DA line for Display controller	10N9
DBEN1	D Bus Enable	6G4
DCKH1	Clock for D Bus receivers high	6A5
DCKL1	Clock for D Bus receiver low	6B5
DD1/0	Device data signals from Schmidt Trigger receiving circuit	16E7
DFST1/0	Timing Control flip-flop, Detects DSTRT	5E5
DL00:070	Buffered D Bus	Sheet 8
DL00X	Strap to TTY address decoder	9E4
DMPF0	Data Memory Parity Fail (from MBC)	7E4
DR0	Data Request Control line, MPX channel	6H4
DR001:151	D Bus receivers	Sheet 6
DRGA0	DR line for TTY controller	10N2
DRGB0	DR line for Display controller	10N8
DRN1/0	Start Bit stage of Shift Register (controls transmit line in Write mode)	11M6
DSPLY0	Display controller interrupt line to CPU	10J8
DSTRTO	Start D Bus operations	5H9
DSYN1		5D6
DT1/0	Device Transmitting flip-flop (set when RCV loop starts the timer)	11G7
DTMG0	Delayed TMG signal	12C4
DU1	Output of Device Unavailable detector-Active for TTY in DEF/local modes	16J5
DX1	Serial data input to Shift Register (line data in Read mode/all ones in Write mode)	11B2
EBL1	Interrupt Enable flip-flop for TTY	9J6
EOC1/0	End of character (output of character counter)	12N6
ENSHA0	Enable signals for S Bus high	Sheet 2
ENSHB0		
ENSHC0		
ENSLA0	Enable signals for S Bus low	Sheet 3
ENSLB0		
ENSLC0		
ENT30	Enter Time period T3	5N2
EPF1/0	Early PPF Timer (1 millisecond)	14L2
ESNC0	Complimentary pulsed signals from Display Console	10A5
ESN00		
EX1	Examine bit of TTY status	8J8
EXA0	Auxiliary initialize inputs	14A3
EXB0		
FLSYNO	False SYNC signal (D Bus operation)	14K8
FPOW0	Decoded Power Down function	7C7

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
FSEL 00:03	Function Select lines from CPU	Sheet 1
FSTPO	Stop function-terminates SHFT pulses	12M5
FTX1/0 } FTY1/0 }	Displays controller SYNC generation flip-flops	Sheet 13
FWAIT1/0	Flip-flop for load Wait function	7E8
GACK0	Gate Acknowledge function	6G2
GADRS1	Gate ADRS line	6R3
GCC0	CC Bus - G Bit	7R2
GCMD1	Gate Command	6M3
GDA1	Gate Data Available line	6M3
GDHBH0 } GDLBH0 } GDLBC0 } GDA0 }	Gating on D Bus signals	Sheet 4
GDIN1	Gate Data In	6K3
GDOU1	Gate Data Out	6M3
GDR1	Gate Data Request	6H3
GLAB1	Gate LA and LB signals	13G4
GLITCH0	Start Glitch signal - clears TMG1	12B2
GP1/0	Gate POUT function	1K6
GPX0	POUT function finished signal	IN6
GSR1	Gate Status Request	6J3
GSTP1	Gate STP (test point)	15K8
GSTR1	STRTO gated with IOU decoded address	IN8
HW1/0	Halfword test line - MPX channel	5J2, 5K2
INCR1/0	Increment/Normal flip-flop	10D2
INIT0	INT key line from Console	14A3
INTRI	TTY interrupt flip-flop	12E8
IRLMP0	Instruction Read Local Memory Parity Fail (from MBC)	7E3
KA } KB }	Cycle counter flip-flops	Sheet 5
KC0/1	Timing Control flip-flop, Control line timing	5F3, 5H4
KD1/0	Timing Control flip-flop Control line timing	5M6
KSIG0	Function code line from CPU	1B4
KSYN1/0	Timing Control flip-flop, SYN stretch	5H8
KT1/0	Cycle counter-Terminate flip-flop	5N6
KTM	Test point. Ground to kill Start Timer	14B5
KY } KX }	Johnson Counter flip-flops for address cycle	5L8 5M8

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
LBO } LA0 }	Signals controlling the loading of display registers	13J5 13J6
LCC0	CC Bus - L Bit	7R2
LDBR0/1	Load Buffer Register pulse (active in Read mode only)	8C7
LDWAIT0	Decoded Load Wait Indicator function	7C8
LESYN1/0	Timing Control flip-flop Detects leading edge of SYNC	5H2
MCR001:091	MCR straps	Sheet 2
MCR110:150	Machine Control Registers	Sheet 7
MFIN0	Module Finish line to CPU	7N8
MMF0	Machine Malfunction interrupt line to CPU	7G6
MSEL000:020	Module Select lines from CPU	1C8
MSIG0	Module Finish line to CPU - Tests the state of HW line	7N9
MSYN1	SYNC from Display or TTY controllers	11D8
MTA0/1 } MTB0/1 }	Master TTY Timer (440HZ Output)	12J3
MTC1 } MTD1 }	Timer clock counter (110HZ Output)	12H4
OV1/0	Overflow error flip-flop	8G8
PA0 } PBO } PCO } PDO }	Pulse output functions (test points)	Sheet 1
PFDT0	Power Fail Detector output	14D3
PF1/0	Primary Power Fail flip-flop	14H5
POFF0	Power Off line from Console switch	14A3
POUTO	Pulse Out function	7C7
PPF0	Primary Power Fail interrupt line	14K6
RACK0	Receive Acknowledge interrupt signal	12F9
RDWDH1	Read-Write Data Halfword	6G4
RN	Negative side of RECEIVE loop	16H8
RP	Positive side of RECEIVE loop	16H6
RST0	Reset line for Display controller	10G3
S160:230	S Bus high	Sheet 2
S240:310	S Bus low	Sheet 3
SCC0	CC Bus - Strobe line	7R6
SCLR0/1	System initialize line MPX channel	15K2
SD001:071	Bi-Directional byte bus to Display Panel	Sheet 8
SELSH0/1	Select signal for S Bus high	2N1

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
SELSL1	Select signal for S Bus low	3F2
SH10 } SL00 }	Signals for sensing Display Console's Switch Register	13J7
		13J8
SHFT1/0	Shift Register pulses, nine per character	12N4
SKT1	Set KT flip-flop	5N3
SMCR0	Sense MCR 00:15	7C7
SMFIN0	B Bus operation finished signal	IL9
SNGL0/1	Single mode flip-flop	10G8
SR0	Status Request Control line, MPX channel	6H4
SR001:071	Shift Register-eight data stages	Sheet 11
SRG0	SR for Display Panel	10N6
SRGA0	SR line for TTY controller	10N3
SRGB0	SR line for Display controller	10N7
SSGL1/0	SNGL key line from Display Console	10A8
ST0	Start idle Timer flip-flop	12E4
STC1	Start gating on S Bus (non-D Bus operation)	2H4
STCLK1	Clock for ST flip-flop	12C3
STD1	Start gating on S Bus (D Bus operation)	2H5
STES1	Set TESYN flip-flop	
STMA0	Start Timer	14F5
STMB0/1	Start Timer flip-flop	14G8
STP1	System Stop flip-flop	14N5
STP1A	Buffered STP1 (test point)	14N5
STPF1/0	Start Power Fail Timer latch	14H3
STPFRO	Start Power Fail routine	14F2
STRTO	Module Start line from CPU	1C7
STT1	Start display controller timer	13E3
SYNO	SYNC test line - MPX channel	5A5
SYNO	SYNC test line MPX channel	11G9
TACK0	Transmit Acknowledge interrupt signal	12N8
TB0	Delay Control line, flip-flop	5B4
TC1/0	Timing Control delay pulses	59B
TDU	Device Unavailable line from TTY	16H4
TERM1	Timing Control flip-flop, Detect trailing edge of SYNC	5E6
THW0	Decoded Test Halfword function	7C7
TMG0/1	Timing gate control flip-flop	12D1
TMGIA	Timing gate test point	12N1
TN	Negative side of SEND loop	16H1
TP	Positive side of SEND loop (TTY)	16H3

<u>MNEMONICS</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
TRNS0	Transmit signal to SEND loop	16F1
TX0/1 } TY0/1 }	Display controller timer	13G2 13J3
VCC0	CC Bus - V Bit	7R4
WAIT1	WAIT light control	13J9
WRT1/0	Write mode execute flip-flop for TTY	12D3
WT1/0	Write mode storage flip-flop for TTY	9N7
XA1/0	Flip-flop for gating LA and LB	13F5
XC1/0	Flip-flop for gating SH, SL	13F7
XLC1/0	Line check flip-flop (checks for START glitches and break conditions)	11M7
XPF1/0	Power Fail stop timer	14L5
XRPA	Pull-Up resistor	13F5
XRPB	Pull-up resistor	5L7
XRPD	Pull-up resistor	9M3
XRPE	Pull-up resistor	7E8
XRPF	Pull-Up resistor	7F1

MODEL 8/32

WRITABLE CONTROL STORE

INSTALLATION SPECIFICATION

INTRODUCTION

The Writable Control Store (WCS) is an option which extends the flexibility of the user level Processor to that of the micro-machine. The WCS has the capability of storing and retrieving data within the control store, plus the capability of dynamically altering control store instructions. In effect, the micro-programmer has the full capability of the user level machine at micro-processor speeds.

The WCS offers 2KB of control store, sufficient to contain 512 instructions or some combination of instructions and data. It is contained on the 8/32 CPC Processor board and requires typically 7 Amperes of 5VDC for power.

This specification provides the necessary information for the installation of the 8/32 Writable Control Store (WCS) option.

PHYSICAL CHARACTERISTICS (35-555 Board also includes 8/32 CPC)

Dimensions – Board 15" x 15"

Weight – 6 lbs. (approximately)

Power – 5VDC at 10 Amperes maximum

Hardware

- one additional power supply regardless of expansion.
- one 35-555F01 Board
- one 17-360 front edge ribbon cable
- one illegal instruction ROM on 8/32 CPB ROM.

UNPACKING

When the WCS option is shipped with a system, it is installed at the factory. All cables and printed circuit boards should be inspected to ensure proper seating.

INSTALLATION

Chassis

Slot 6 of the Basic Processor lower chassis is used for the 8/32 WCS option. The WCS is mounted on the 35-555F01 board (the 8/32 Processor CPC board).

Power

An additional power supply, regardless of configuration, has to be used to provide an extra 5VDC (P5) source for WCS. See Figure 1.

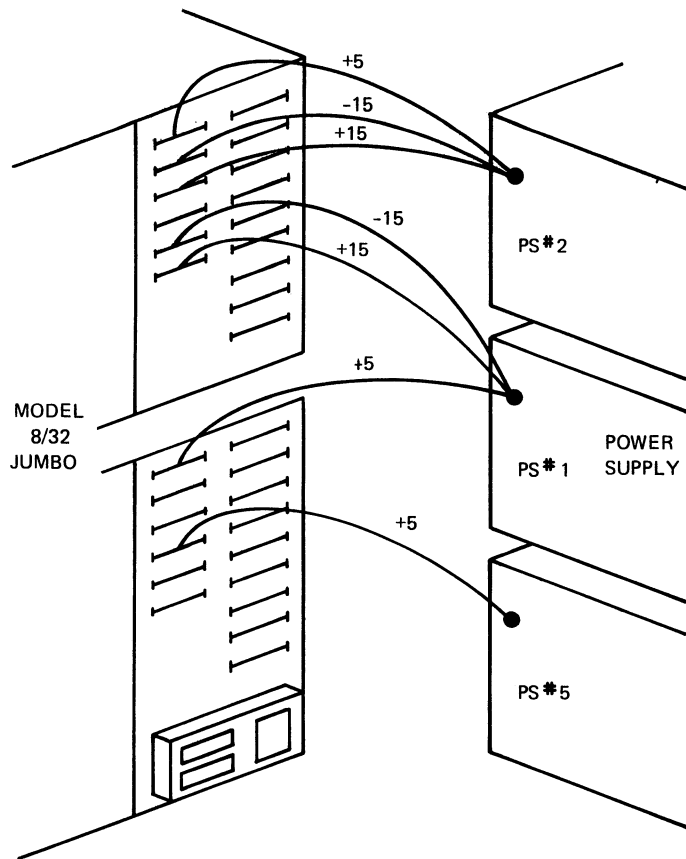


Figure 1. 8/32 Basic Processor With WCS and/or DFU Power Wiring

Cabling

The 17-360 cable connects Connector 4 on the CPB and CPC Processor boards.

Strapping

Refer to the Writable Control Store Maintenance Specification, 35-555F01A21 for strapping details.

Testing

Upon completion of the installation, and before power is applied, all voltages should be checked for shorts between each other and ground. Proper operation of the WCS is tested by the execution of the WCS Test Program, 06-192.

Other

Illegal instruction ROM, 19-084F43, on the 8/32 CPB board (35-537) at Location 00C must be replaced by 19-084F48.

WRITABLE CONTROL STORE MAINTENANCE SPECIFICATION

INTRODUCTION

The Writable Control Store (WCS) is an option which extends the flexibility of the user level Processor to that of the micro-machine. The WCS has the capability of storing and retrieving data within the control store, plus the capability of dynamically altering control store instructions. In effect, the micro-programmer has the full capability of the user level machine at micro-processor speeds.

The WCS offers 2KB of control store, sufficient to contain 512 instructions or some combination of instructions and data. It is contained on the unused half of the 8/32 CPC Processor board and requires typically 7 Amperes of 5VDC for power.

This specification describes the functional operation of the Model 8/32 Writable Control Store and provides information necessary for its maintenance. This specification references CPC Functional Schematic 35-555D08. INTERDATA schematic title, drawing number, and sheet number are located in the lower right corner of each sheet. Each sheet is zoned alphabetically across the top and bottom margins and numerically down the side margins. These schematics are referenced throughout the block diagram and functional analysis text to correlate specific locations on the schematics to the text. When a specific location is referenced by the text, a number-letter-number is used to designate schematic sheet number, and zone location within the sheet. For example, schematic reference (3B5) is found on Sheet 3, at the intersection of Zone B and 5.

BLOCK DIAGRAM ANALYSIS

Data is stored in a 512 x 32 bit array subdivided into two pages, i.e., A and B. Each page stores 256 fullwords of data. Each page is further delineated as a high half which stores Data Bits CSD001:151, and a low half which stores Data Bits CSD161:311. Data to be written into the Writable Control Store is derived from the backpanel A Bus, buffered, and fanned out to Pages A and B. See Figure 1.

The address to be read (or written) is derived from the Control Store Address lines (CSA) 04:15 originating on the 8/32 CPB board. The four most significant address lines (CSA 04:07) are strapped to enable a selectable address range for the WCS. The eight least significant address lines are buffered directly as A0:A7, and select one out of 256 addresses within each page.

FUNCTIONAL SCHEMATIC ANALYSIS

Storage Device

The basic storage element used in the WCS is the 19-077 static bi-polar Random Access Memory (RAM) employing tri-state output, organized 256 words by one bit. It is intended for high speed memory applications where low input loading on chip address decoding, and high capacitive drive capability are required. See Table 1.

The three state output has the characteristic TTL totem pole output with active elements driving both the ONE and ZERO output voltage levels, plus the capability to disable both driving elements to a high impedance state when the device is not selected. The data output can then be tied to a common output bus which can be driven by only one active output or a passive pull-up.

The memory device (19-077) is addressed with the A0-A7 inputs which select one of 256 words. The chip is enabled by making all Memory Enables, Pins 3, 4, and 5 low. If any of the Memory Enables are high, the chip is in the high impedance state. If the Write Enable Pin 12 is high and the chip is enabled, the stored data (complement of data applied at input during write cycle) is read on the output pin. If the Write Enable Pin 12 is low and the chip is enabled, the data on the input pin is written into the addressed word.

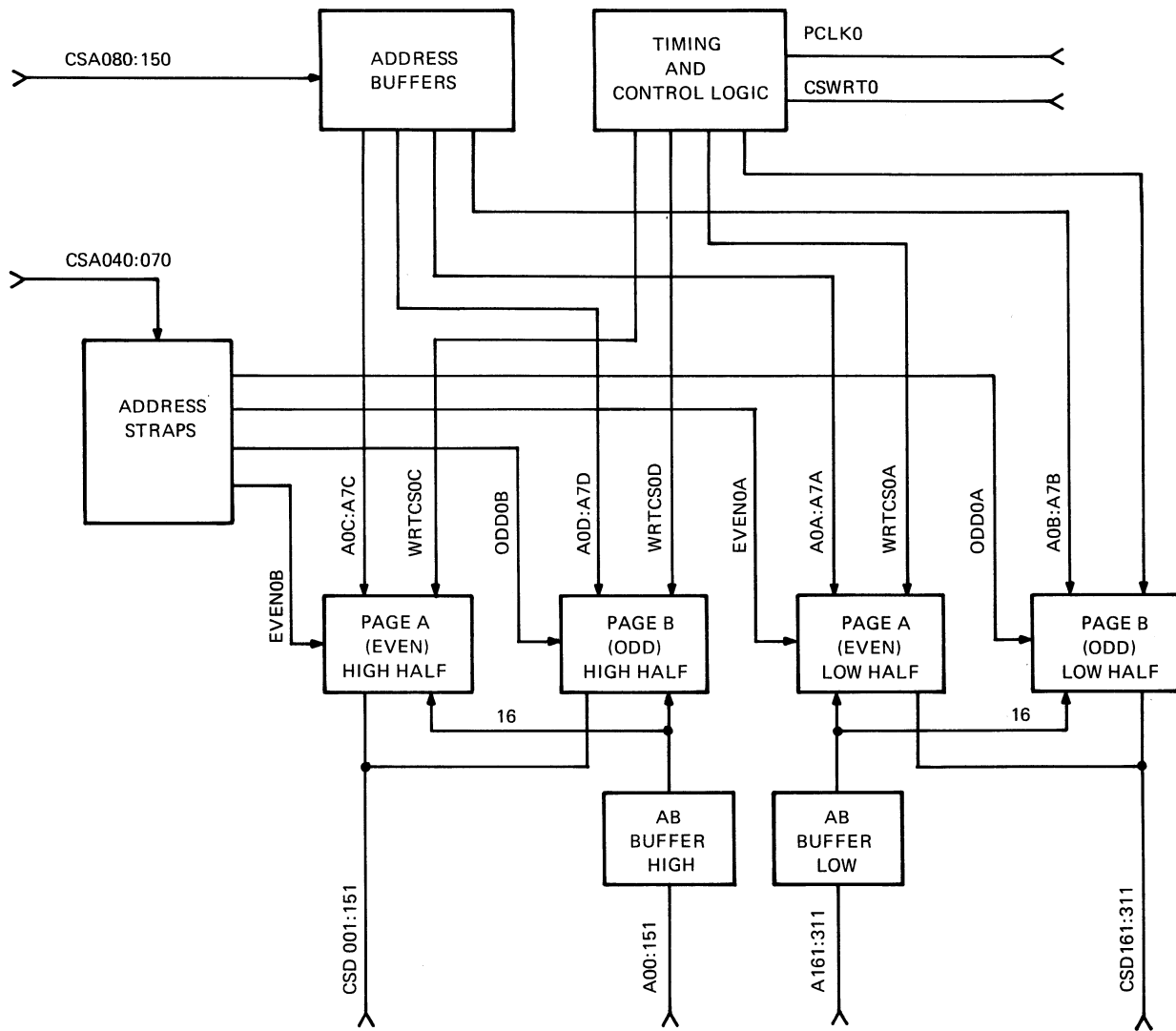


Figure 1. 8/32 WCS Block Diagram

TABLE 1. 19-077 MEMORY DEVICE TRUTH TABLE

CHIP SELECT	WRITE ENABLE	OPERATION	OUTPUT
ALL LOW	LOW	WRITE	UNDEFINED
ALL LOW	HIGH	READ	COMPLIMENT OF WRITTEN DATA
ONE OR MORE HIGH	DON'T CARE	HOLD	HIGH IMPEDENCE STATE

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

Writing into WCS

Refer to Figure 2, Timing Diagram. During Control Store Write operation, the Processor traverses Processor Control States 0, 2, and 3. The Write instruction is decoded in Control State 0. The CPB deposits the WCS address on CSA 04:15 lines at least 10 nanoseconds before Control State 2 is entered. The CSA 070 determines which page is written into by activating either ODD0 or EVEN0 chip enable lines.

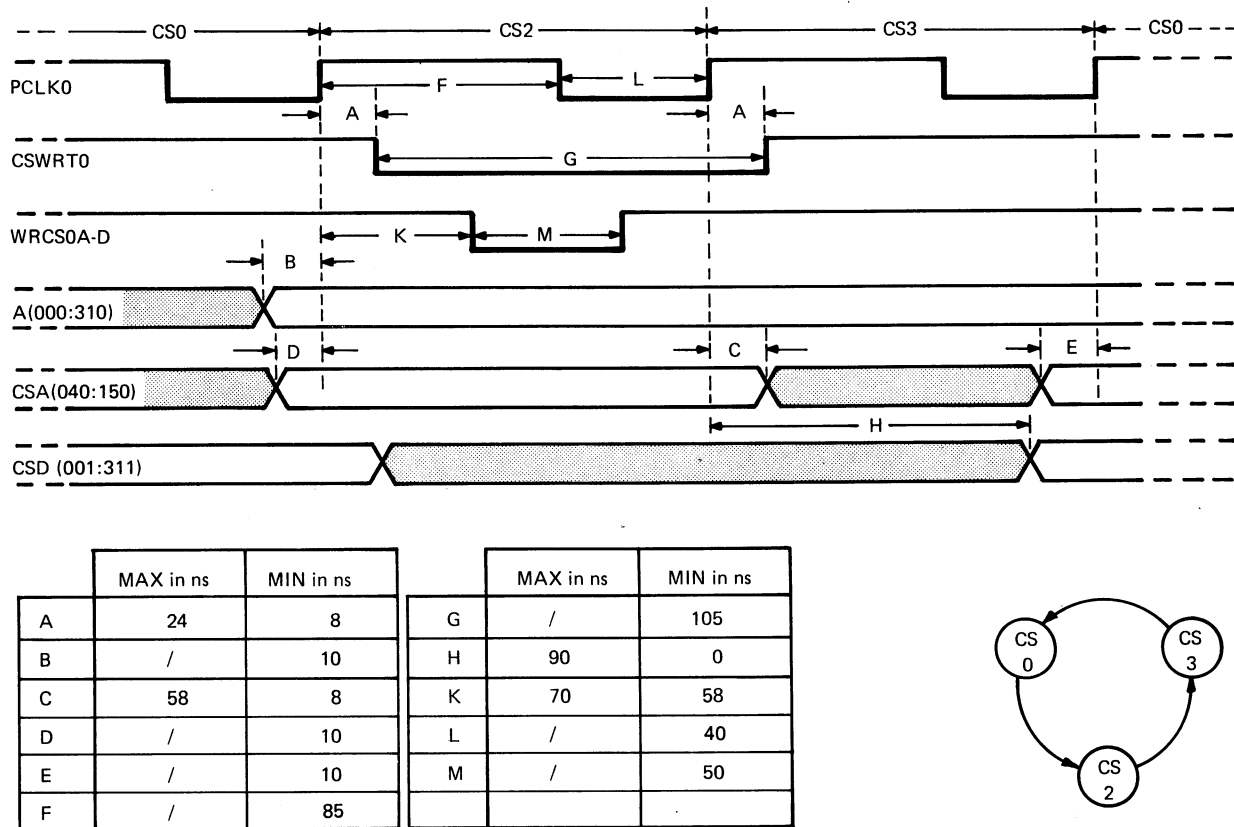


Figure 2. Control Store Write Operations

The CSA 04:06 are factory strapped to inhibit the chip enable lines for any address outside the X'800'-'9FF' range. The CSA 08:15 are buffered directly as A0:A7; and select one out of 256 addresses within each page. The CPB holds address lines static at least 10 nanoseconds after Control State 2 to 3 transition. The data to be written into WCS is deposited on the backpanel A Bus at least 10 nanoseconds before CS0→CS2 transition.

This data is buffered on the WCS board and applied simultaneously to both pages of the WCS. Data is held static throughout the CS2 state.

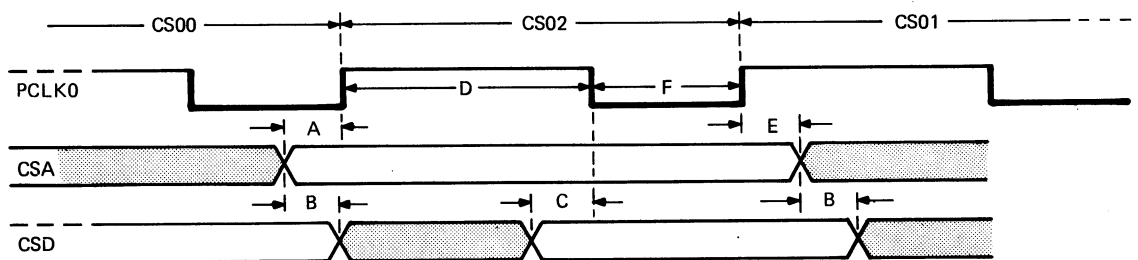
Writing of one 32-bit word is accomplished in Control State 2. Write operation is distinguished from a Read operation by CSWRT going active. This signal is used to derive Write enable pulses WRCS0A-D. The width of WRCS0 is guaranteed to be 50 nanoseconds minimum.

The 40 nanosecond delayed Processor Clock (PCLK0) is used to build the leading edge of WRCS0. This ensures that the switching noise on the leading edge of CSWRT signal is screened off. The trailing edge of WRCS0 is generated by the DPCLK0A signal (Processor Clock Delayed 20 nanoseconds).

Since PCLK0 is high (inactive) for 85 nanoseconds minimum, the width of WRCS0 is guaranteed to be 50 nanoseconds minimum. The DPCLK0A input also ensures that a sufficient data and address hold time is allowed after the trailing edge of WRCS0.

Control Store Read

Refer to Figure 3.



NOTE: CSWRT0 WILL STAY HIGH THRUOUT THE ENTIRE CS READ OPERATION

A = 10 ns MAX

D = 85 ns MIN

B = 10 ns MIN

E = 10 ns MIN

C = 10 ns MIN

F = 40 ns MIN

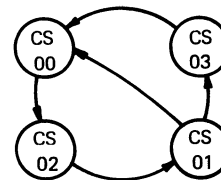


Figure 3. CPU/WCS Interface Timing Control Store Read Operations

The Read instruction is decoded in Control State 0. The CPB gates the address on the busses prior to entering Control State 2 as described in the section on writing into WCS. The CSWRT0 is held high throughout the Read instruction. During CS2, WCS deposits one 32-bit word on Control State Data bus (CSD) 001:311. The data on the bus must be settled at least 10 nanoseconds before the negative going edge of PCLK0 (see parameter C in Figure 3) which latches the data read from the WCS in the Processor Register Stacks.

WCS Strapping

Address strapping.

The 8/32 WCS responds to the Control Store addresses within the X'800'9FF' range.

ROUTINE MAINTENANCE

Routine maintenance consists of running the WCS Test Program 06-192.

MODEL 8/32 MAIN MEMORY SYSTEM MAINTENANCE SPECIFICATION

1. INTRODUCTION

This specification applies to the INTERDATA Model 8/32 Main Memory System components listed in Table 1.

TABLE 1. MODEL 8/32 MAIN MEMORY COMPONENTS

BOARD PART NO.	MNEMONIC	FUNCTION
35-534	LMI	LOCAL MEMORY INTERFACE
35-535	MBC	MEMORY BUS CONTROLLER

2. SCOPE

This specification describes the overall operation of the Model 8/32 Main Memory System consisting of one MBC and two LMI boards used in conjunction with the 35-198 32 KB Memory Module or the 35-552 64 KB Memory Module. The Main Memory System interconnection to the CPA Bus and the Extended Direct Memory Access (EDMA) port are covered by this specification. The specification also provides a block diagram description, timing information, troubleshooting and maintenance, and a mnemonic list for the MBC and LMI.

3. SYSTEM BLOCK DIAGRAM

The basic organization of the Main Memory System is shown in Figure 1. Access to the memory system is made through two ports: The CPU port which interfaces with the CPA board, and the EDMA port which interfaces with the Extended Direct Memory Access Bus.

The Memory Bus Controller (MBC) board controls the two ports into the local memory and provides a third data path which enables the CPA to directly access remote memory located on the EDMA Bus. The MBC also maintains a cache memory of eight halfwords which is used to store instructions in a look-ahead fashion, and is equipped with circuits to resolve contention between the CPU port, EDMA port, and look-ahead cache for access to Local Memory. The MBC maintains the control logic for the EDMA Bus (QUE0, RPC0/TPCO, and SOT0 as described in Section 7).

The Local Memory Interface (LMI) board provides all signals necessary to control the Local Memory Modules (LMMs) and steers halfword data to the appropriate Memory Module bank for halfword operations. In addition, the LMI generates and checks parity, on systems so equipped.

3.1 Memory System Organization

The Model 8/32 Memory System is organized into a pyramid structure which places four Memory Module banks in parallel (see Figure 1). Note that each LMI controls two banks of Memory Modules, each 16 bits wide, which are accessed simultaneously to give a 32 bit wide fullword of data on each memory access cycle of an LMI. The MBC is connected via the 32-bit Local Memory Bus to two LMI boards, each of which controls half of the available memory.

The addressing structure of the memory system is such that the two halves of memory controlled by separate LMI boards are interleaved; i.e., successive fullwords are controlled by alternate LMIs. Looking at the address format (Figure 2), only 19 of 20 address bits are used by the memory system. Bit 17, the fullword bit, determines which of the two LMIs controls the requested address; Bit 18, the halfword bit, determines which of the two banks controlled by the LMI contains the requested halfword (used only for halfword operations). These bits are used only within each LMI to select the appropriate memory bank and Bits 0:16 are then sent to the memory bank to address the desired location within the selected bank. Thus, addresses in which Bits 17 and 18 are both zero, are found in the bank with interleaving address 00 (Figure 1), and similarly for the other three combinations of the two least significant bits.

The two important features of the Main Memory System organization are: 1. Fullword data paths, providing true fullword access capability; 2. Fullword interleaving, reducing the effective memory cycle time for accesses to subsequent fullword addresses.

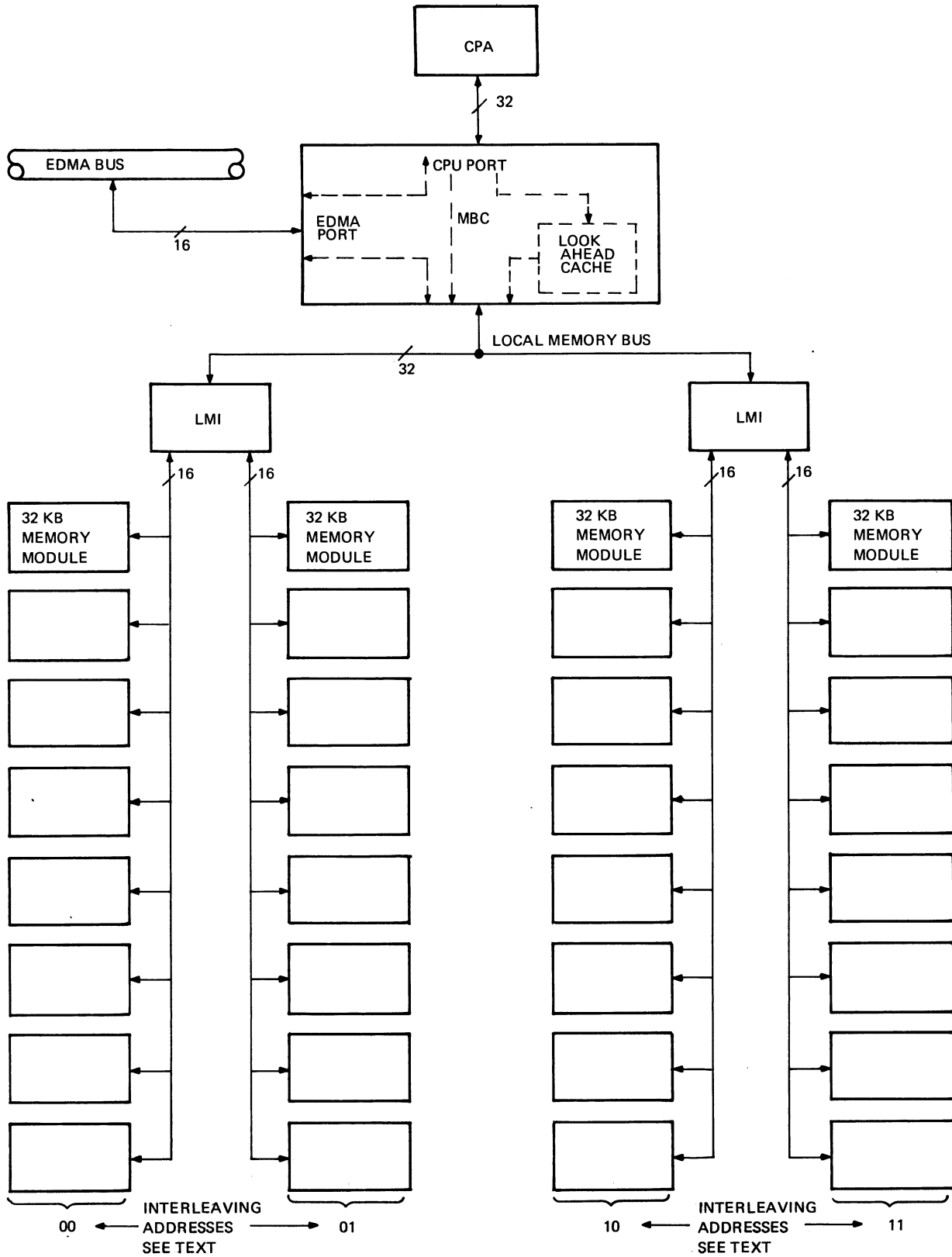


Figure 1. Model 8/32 Main Memory System Block Diagram

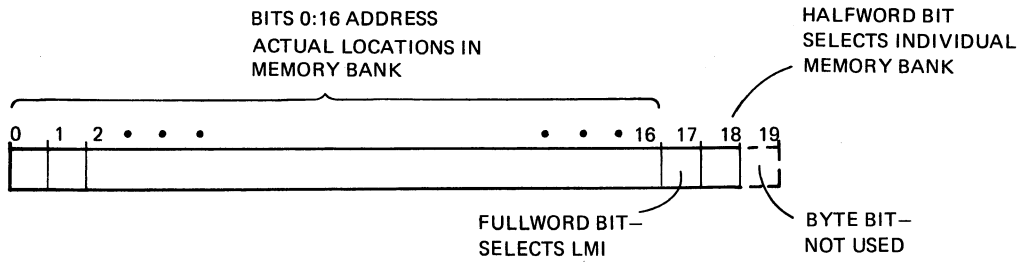


Figure 2. Address Format

4. MEMORY SYSTEM PHYSICAL DESCRIPTION AND INTERCONNECTION

The basic configuration for the Main Memory System consists of one MBC, two LMI boards, and four 35-198 32 KB 750 nanosecond cycle time Core Memory Modules with or without parity. These are arranged as shown in Figure 3. Expansion of memory capacity beyond the basic 128 KB requires that Local Memory Modules in the basic chassis be removed and all Memory Modules be located in the Memory Expansion chassis with interconnection via back panel cables. Refer to Model 8/32 Installation Specification, 01-078A20, for details of memory expansion.

4.1 Memory Bus Controller

The MBC consists of one INTERDATA standard size mother board which occupies upper card file Slot 1. The MBC plugs into back panel Connectors 0 and 1 for power, ground, and CPU, EDMA and LMB interfaces. A backpanel waterfall cable from Connector 1 of the MBC slot to Connector 1 of the lower card file Slot 2 provides the interconnection of the EDMA Bus interface to the I/O slots.

4.2 Local Memory Interface

Each LMI board is one INTERDATA standard size mother board which occupies upper card file Slots 3 and 6. The LMI boards should not be interchanged, as each board is strapped differently. The LMI plugs into Connectors 0 and 1 for power, ground, and LMB and LMM interfaces.

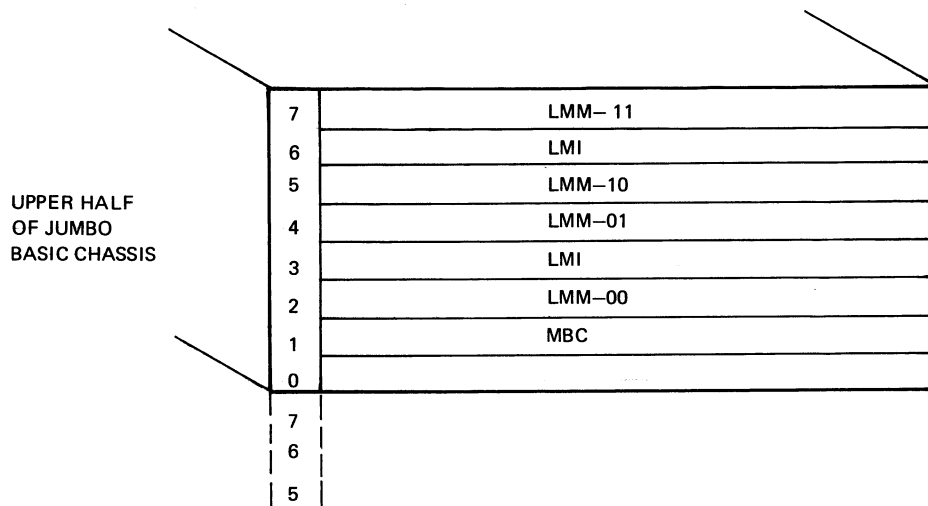


Figure 3. Basic Configuration - Main Memory System

5. COMPATIBLE MEMORY MODULES

NOTE

The descriptions in this section are not meant to limit future memory expansion to the three types of Memory Modules specifically mentioned. Future development may result in additional compatible Memory Modules (for example, semiconductor memories) which meet the same interface specifications and therefore can be accommodated.

The Main Memory System is designed to accommodate three different standard INTERDATA modules within Local Memory as follows:

- 35-198 32 KB 750 nanosecond cycle time Module
- 35-198 32 KB 1 microsecond cycle time Module
- 35-552 64 KB 1 microsecond cycle time Module

The standard configuration rules for the system are such that only one type of Memory Module may be used within Local Memory; i.e., mixing memory types is not permitted except by special configuration. Provisions have been made in the memory system design for special configurations where up to three pairs of LMIs can be connected to the LMB with each pair accommodating a different type of standard INTERDATA Memory Module. That is, each pair of LMIs is strapped to respond only to separate, contiguous blocks of memory address. Figure 4 shows a special configuration.

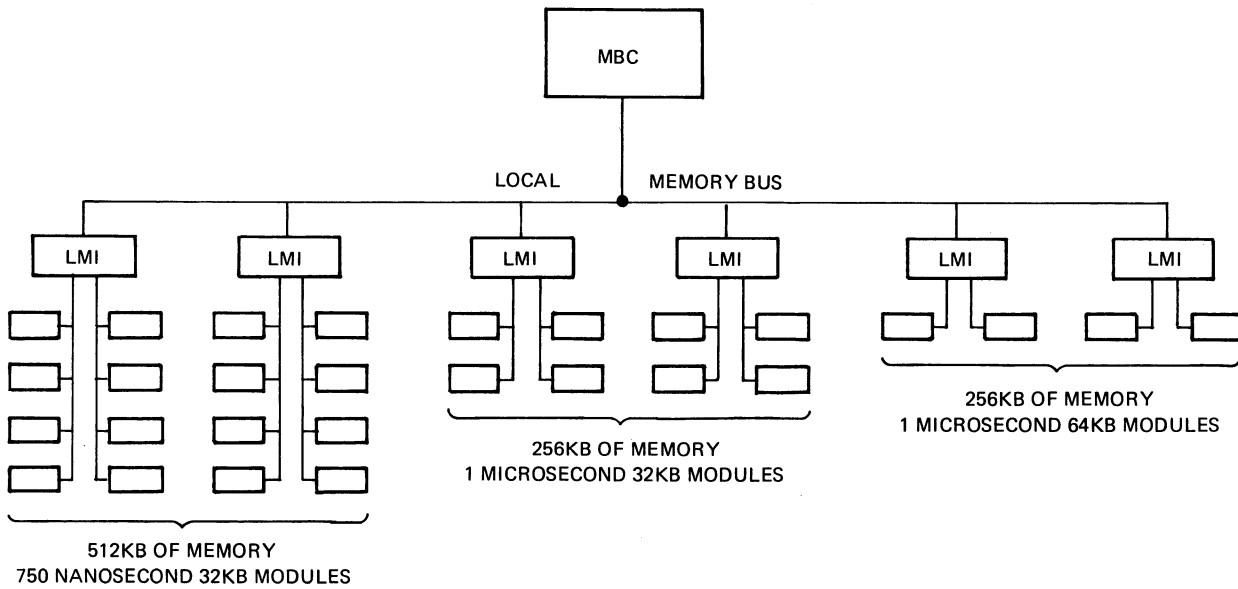


Figure 4. Example of a Memory System Special Configuration Using the Maximum Number of LMI Boards.

6. CPU/MBC INTERFACE

The CPU/MBC interface between the MBC and CPA boards consist of the following lines:

<u>MNEMONIC</u>	<u>NAME</u>	<u>DIRECTION</u>
CA120:300	Control Address	CPA → MBC
CD000:310	Control Data	CPA ↔ MBC
CMC000:020	Control Memory Control	CPA → MBC
CREQ0	Control Request	CPA → MBC
CRDY0	Control Ready	CPA ← MBC

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

Data and Code Lines

Address Lines (Control Address—CA120:300)

These lines transfer the 19-bit address from the CPA to the MBC for all CPU memory accesses.

Data Lines (Control Data—CD000:310)

The bi-direction Data Lines transfer data between CPA and MBC for all CPU memory accesses. Halfword data is always transferred on CD160:310.

Control Code Lines (Control Memory Control—CMC000:020)

The CMC lines carry a code from the CPA to the MBC indicating the type of memory access CPU is requesting. The CMC code is shown in Table 2.

TABLE 2. CMC CODE

CMC BIT			FUNCTION
000	010	020	
0	1	0	READ FULLWORD
0	0	0	INSTRUCTION READ
0	X	1	READ HALFWORD
1	0	0	WRITE FULLWORD
1	1	0	NO ACTION—NOT USED
1	0	1	WRITE HALFWORD
1	1	1	NO ACTION—NOT USED

Control Lines

CREQ0 The Control Request line is pulsed low-active by CPA whenever the CPU is requesting a memory access.

CRDY0 Control Ready is pulsed low-active by the MBC when data is valid during a read cycle, or when a write cycle no longer requires valid CMC code and data from the CPU.

One other control line, DMFP0, is transmitted from the MBC to the IOU board and is maintained low active whenever a memory error is detected from Local Memory or remote memory (on the EDMA Bus).

6.1 Types of CPU Memory Operations

The CPU can be serviced with five different types of memory operations as defined by the CMC code. Read fullword and write fullword are 32 bit data operations; read halfword and write halfword are 16 bit data operations. Instruction read appears at the CPU interface to be a halfword read. However, the MBC responds differently to the instruction read code as described in Section 11.3.

7. MBC/EDMA INTERFACE

The interface of the MBC with the Extended Direct Memory Access (EDMA) Bus provides high speed interconnection between the CPU and up to three remote memory systems, as well as between Local Memory and up to four EDMA Channel controllers (DMAC). These DMACs can include Extended Selector Channels (ESELCH) and custom EDMA controllers. In addition, the MBC interface supports the EDMA Bus control circuits.

The EDMA Bus control circuit maintained by the MBC consists of the following control lines:

- QUE0** The Queue pulse resolves contention for the bus by freezing the request status at its leading edge.
- RPC0/TPC0** Receive Priority Chain (RPC) and Transmit Priority Chain (TPC) are the low-active daisy-chain priority signal which select the highest priority requesting device captured by QUE0.
- SOT0** The negative going leading edge of Start of Transmit (SOT) enables the selected device to start transmission over the bus.

The remaining lines which comprise the MBC/EDMA interface are:

DMA Data Bus Lines

DMX120:150 The DMA Data Bus contains 22 lines that carry multiplexed address and command code, write data, and read data. Data Bus formats are shown in Figure 5.
 DMA000:170

Control Lines

- XREQ0 Common Request line pulled low by any memory or DMAC requesting service.
- LMRQ0 Local Memory Request Queued is pulled low by a DMAC when it is selected by RPC0, if it is requesting Local Memory.
- LOAD0 The negative going edge of Load gates the contents of the DMA Data Bus into the appropriate receiving register of a memory interface on DMAC to memory transfers.
- ANS0 The negative going edge of Answer Sync. loads the contents of the EDMA Data Bus into the appropriate receiving registers on memory to DMAC transfers.
- EOT0 End of Transmit is a high speed timing signal generated by the DMAC or memory at the end of its last data bus pulse. It signals the bus control circuits on the MBC that one device is finished with the bus and it may issue SOT0 to the next user if a request is queued and ready.
- MOBZ0:M3BZ0 There is one MxBZO (Memory Busy) line associated with each of the four memory systems connected to the bus (Local Memory plus three Remote Memories). MxBZO is pulled low by the active DMAC as soon as it receives the SOT0 pulse. It is released by the DMAC at the end of the address pulse but is held down by the memory until the memory is capable of receiving the next command.
- BHO The Bus Hold line is pulled low by an immediate response memory interface and is held low until the answer is returned. This signals the bus control circuit on the MBC that no SOT0 signal can be generated to another device until the immediate response memory is finished with its cycle.

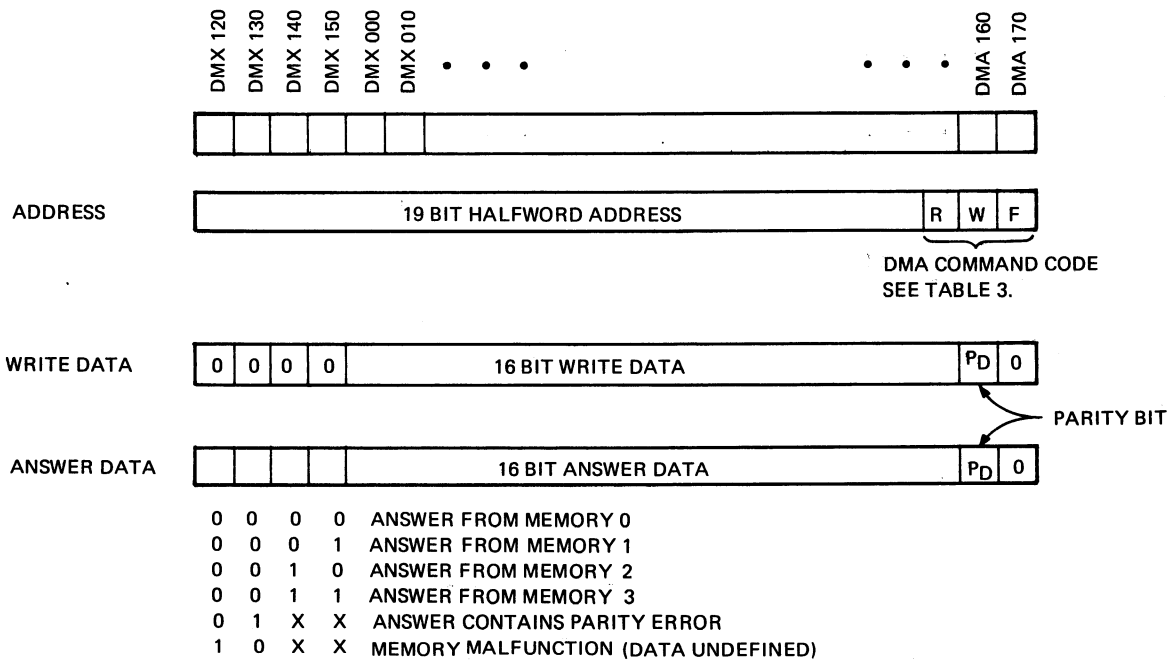


Figure 5. EDMA Data Bus Formats

7.1 EDMA Bus Control Logic

The EDMA Bus control logic is located on the MBC and is used to resolve contention of devices requesting service on the EDMA Bus, queue the selected device (QUE0), establish an order of priority of requesting devices (RPC0/TPC0), and issue a start command (SOT0), indicating that one device has the use of the bus and may begin transmission.

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

7.2 Types of EDMA Operations

Seven types of data transfer operations can occur over the EDMA Bus. Table 3 lists the operations and their identifying command codes which are transmitted on the R, W, and F bits of the EDMA Data Bus address format (DMA150, 160, and 170 respectively) in low active polarity (see Figure 5).

Read halfword and Write halfword are 16-bit data transfers. Read and Set halfword is a remote memory Read operation which, in addition to reading the data, causes the most significant bit of the data halfword in the remote memory to be set after the read is accomplished. Read fullword and Write fullword are 32-bit data transfers in which the data is sent over the EDMA Data Bus in separate 16-bit halfwords.

Burst read and Burst write are block transfer operations in which a single EDMA request initiates a block of continuous memory accesses to sequential fullword addresses.

TABLE 3. EDMA COMMAND CODE

R	W	F	FUNCTION
1	0	0	READ HALFWORD
0	1	0	WRITE HALFWORD
1	1	0	READ AND SET HALFWORD
1	0	1	READ FULLWORD
0	1	1	WRITE FULLWORD
1	1	1	READ BURST
0	0	1	WRITE BURST

8. MBC/LMI INTERFACE

The MBC/LMI interface consists of the following lines:

MNEMONIC	NAME	DIRECTION
LMB000:310	Local Memory Bus	MBC ↔ LMI
LMRS0	Local Memory Request Service	MBC → LMI
LMDS0	Local Memory Data Strobe	MBC → LMI
LMRDY0	Local Memory Ready	MBC → LMI
LMBSYA0 LMBSYB0 LMBSYC0	Local Memory Busy A, B, and C	LMI ↔ LMI

Data Bus LMB000:310

These bidirectional lines are time-multiplexed to transfer address and control data to the LMI boards at the start of a memory cycle and later to transfer data between the MBC and LMI boards, as shown in Figure 6. Halfword data is always transferred on LMB160:310.

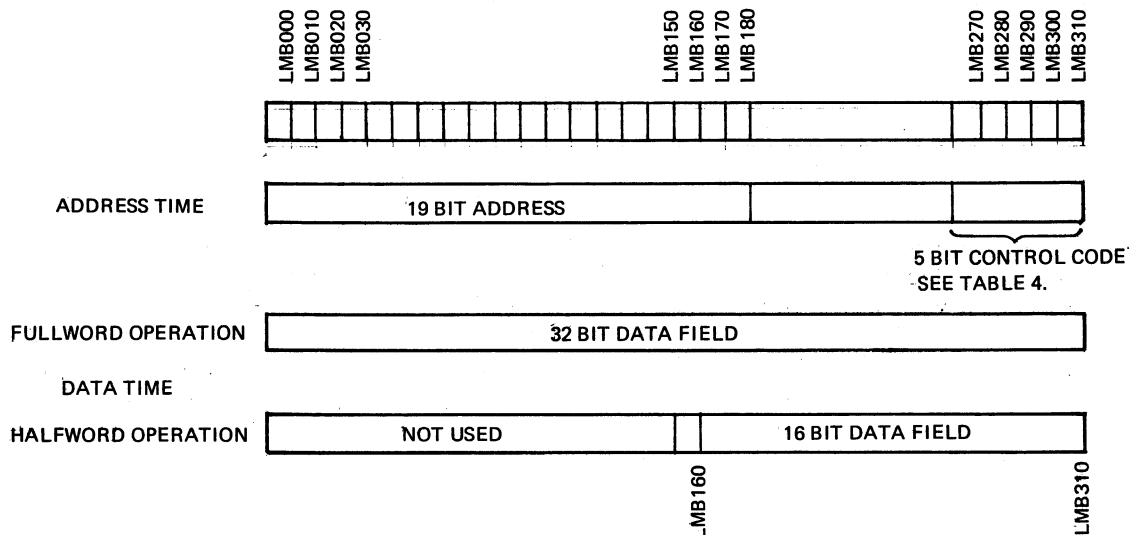


Figure 6. Local Memory Bus Formats

Control Lines

LMRSO	Local Memory Request is made low-active when the MBC is requesting a memory cycle.
LMDSO	The leading edge of Local Memory Data Strobe clocks write data into the LMI Memory Data Register during Write operations.
LMRDY0	The Local Memory Ready pulse is maintained low active when an LMI begins the requested memory cycle and when data for a Read operation is valid on the LMB.
LMBSYB0 } LMBSYC0 }	Local Memory Busy A, B, and C communicate between LMI boards to synchronize their access cycles for an instruction Read operation.

8.1 Types of Memory Operations

Five types of data transfer operations can occur over the Local Memory Bus. Table 4 lists the operations and their identifying control codes. The control code is transmitted over LMB27:31 during address time (see Figure 6) with high-active polarity.

TABLE 4. LOCAL MEMORY CONTROL CODE

LMB27	LMB28	LMB29	LMB30	LMB31	FUNCTION
C001	C011	C021	EX1	SX1	
0	X	1	X	0	WRITE FULLWORD
1	0	1	X	0	READ FULLWORD (FROM CPA or DMA)
1	1	1	1	0	READ FULLWORD (FROM DMA in BURST)
0	1	0	X	0	WRITE HALFWORD
1	X	0	X	0	READ HALFWORD
1	1	1	0	0	INSTRUCTION READ (FROM CPA)
X	X	X	X	1	INSTRUCTION READ (FROM LOOK-AHEAD CACHE)

Write and Read fullword are 32-bit data transfer operations. Note that two codes exist for read fullword—the LMI responds identically to either code. Write and Read halfword are 16-bit data transfers over LMB16:31. An Instruction Read request causes both LMIs to respond with a fullword Read operation. This results in a double fullword Read operation, aligned on double fullword boundaries. The first 32-bits transmitted to the MBC contain the actual instruction halfword requested. 125 nanoseconds later, the second 32-bits of the aligned double fullword is transmitted by the other LMI. Both fullwords are stored in the look-ahead cache.

Note that there are two Instruction Read codes, depending on whether the CPU or the look-ahead cache initiated the request. The LMI boards respond identically to either code.

9. LMI/LMM INTERFACE

The LMI interface with the Local Memory Modules (LMMs) is defined by the following lines:

<u>MNEMONIC</u>	<u>NAME</u>	<u>DIRECTION</u>
MS000:160	Memory Sense Bus	LMI ↔ LMM
MD000:160	Memory Data Bus	LMI ↔ LMM
MAX060, MAX070, MA00:14(0)	Memory Address	LMI → LMM
ERO	Early Read	LMI → LMM
INH0	Inhibit	LMI → LMM
EWRT0	Even Halfword Write	LMI → LMM
OWRT0	Odd Halfword Write	LMI → LMM

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

Data Lines MS000:160 and MD000:16*

These two bidirectional data busses carry 16 bits + parity of read data and write data between an LMI and its two separate banks of memory. The MS Bus and associated memory bank handle the most significant (even) halfword in an aligned fullword, while the MD Bus and memory bank handle the least significant (odd) halfword (see Section 3). The two busses are then linked together within the LMI for a fullword of data (on fullword operations).

Address Lines MAX060, MAX070, MA00:14, (0)

These lines carry the 17-bit address from the LMI to both banks of memory. Contrary to their mnemonic designation, the LMI outputs high-active address information.

Control Lines

ERO	Early Read is a Memory Module control signal which initiates the readout phase of a core memory cycle.
INH0	Inhibit is a Memory Module control signal which initiates the restore/write phase of a core memory cycle.
EWRT0	Even Write is a Memory Module control signal which is maintained low-active when it is desired to write into the even-halfword bank of memory.
OWRT0	Odd Write is a Memory Module control signal which is maintained low active when it is desired to write into the odd-halfword bank of memory.

The Memory Module cycle is basically the same for any request type initiated by the MBC, the only difference is that one or both of EWRT0/OWRT0 are maintained low active if a Write operation is performed. It is important to remember that an LMI cycles two Memory Modules (even and odd halfword) for every access cycle, whether the request is for a halfword or fullword operation.

10. MBC BLOCK DIAGRAM DESCRIPTION (See Figure 7.)

1. The 19 CAXX0 lines from the CPA are the memory address lines. They are stored in 19 tracking latches, the CA ADDR Register.
2. The STK ADDR Register contains the 17-bit address of the four halfwords in the stack with the lowest double-fullword address. The STK A valid flip-flop indicates when this data is valid.
3. STK B Σ is a 16-bit adder that effectively adds 1 to the stack address register to provide the address of the four halfwords of data in the stack with a higher double-fullword address. The STK B valid flip-flop indicates when this data is valid.
4. The comparators, CEQL and CEQU compare the 17 most significant bits of the CA ADDR Register to the STK A and STK B data to determine if the address requested is contained in the stack.
5. The Control Memory Control (CMC) analysis block decodes the Instruction Read code, data read and write codes, and the null state code of the CMC bits.
6. The Memory Contention (MC) circuit resolves contention for the memory between the Processor, the EDMA Bus, and the look-ahead stack. In the case of more than one request to the memory, the Memory Contention (MC) circuit also sets priority. The EDMA Bus has highest priority, the Processor second, and the look-ahead stack lowest. This circuit enables the Local Memory Request Service signal (LMRS), holds the Local Memory Bus Busy state, and is reset by the Cycle Complete (CYCOM) signal.
7. The enabled LMRS logic generates LMRS with the appropriate delays and conditions.
8. The Cycle Complete (CYCOM) logic generation indicates that the present access to memory using the Local Memory Bus is completed.
9. Counter F (CTR F) keeps track of the number of Local Memory Readys (LMRDY) required from the LMI to steer the data and input to the CYCOM logic.
10. The Local Memory Bus (LMB) is a 32-bit bidirectional bus that sends and receives the LMI data to and from the LMIs.

* These bidirectional data busses should not be confused with the unidirectional MS and MD lines on the Memory Modules. The MS and MD lines of the Memory Module are wired together and the combination is then connected to either the LMI MS Bus or LMI MD Bus, as appropriate.

11. When the LMRS signal is sent to the LMIs, the LMB is used for address and control information. The LMB is driven by tri-state multiplexors at each end and received by STTL gates. The address information is low active on the bus while the control bits are high active. The first Local Memory Ready signal from the LMIs indicate that the address has been accepted and the memory cycle has started. The LMB is then used for the transfer of data. The data is high active on the LMB.

The LMB tri-state multiplexor puts the address on the LMB either from the Processor (CAF), the EDMA (DAD), or from the stack request (STB). For Write operations to memory this multiplexor is also enabled and places either the Processor data (CD), or the EDMA data (DMF) on the LMB.
12. The Instruction Stack contains eight halfwords. It has separate Read and Write select lines. It is loaded four halfwords at a time, each time the Processor makes an Instruction Read memory access or each time the stack makes a look-ahead access. The four halfwords come in two 32 bit passes on the LMB. The Write select logic determines the stack address to be loaded and the CTR F logic creates the Load strobes. The Read select lines come directly from the three least significant bits of the address from the Processor, CAF. A tri-state multiplexor, CD B, places the data from the stack onto the CD lines to the Processor during an Instruction Read from an address in local memory.
13. Tri-state multiplexor CD A places data onto the CD Bus to the Processor during a data read from Local Memory or during a read from memory on the EDMA Bus.
14. The Address analysis logic looks at the address requested by the Processor on the CA lines and compares it to four sets of straps to determine if the address is contained in Local Memory, or in one of the three memories that can be placed on the EDMA Bus, or it is beyond the range of memory fitted in the particular machine. The outputs M0, M1, M2, M3, and GTUU (Greater Than Unused) indicate in what region the requested address lies. There are four straps in each set so the one megabyte of memory can be divided up with a resolution of 64 K bytes.
15. Tri-state buffer A places the CAF information on the DMT lines (to DMA transmitters) for Processor requests to memory on the EDMA Bus. Tri-state multiplexor, DMT A, places the data onto the DMT lines when the request is a write to external memory.
16. The DMA transceivers translate the TTL DMT levels into the EDMA Bus levels.
17. The LM data register stores the LM data for answers to EDMA requests from Local Memory.
18. The tri-state multiplexor, DMT B, places the correct half of the LM register onto the DMT lines for these reads.
19. The DXR and DMR lines are signals from the EDMA Receivers.
20. The DMA address register counter stores (and increments when necessary), the 19-bits of address information from the EDMA Bus during accesses to Local Memory.
21. The DMA CTRL register stores the three control bits.
22. The DMA data register is a 32-bit register that stores the data from the DMA Bus that is sent to Local Memory on a write, or stores the read data sent back to the Processor from external memory.
23. The DCOMP A and the DCOMP B circuits are comparators that signal when the EDMA Bus is Writing into memory over an address that is valid in the instruction look-ahead stack.
24. The CPA SEL logic requests the EDMA Bus for transfers to external memory and gets the CPA selected as the transmitting device on the EDMA.
25. CTR B generates the Load and End of Transmit (EOT) signals for the EDMA when CPA is communicating with external memory. It also helps create the signals for data steering and enabling.
26. CTR D receives Load signals from the EDMA Bus and creates the load register signals. It counts the loads and initiates action when required.
27. The Bus Control logic with signals from CTR A, handles the requests for the EDMA Bus and generates the Queue (QUE), Transmit Priority Chain (TPC), and Start of Transmit (SOT) for the bus.
28. The CRDY logic generates CRDY back to CPA at the end of the CPA access.

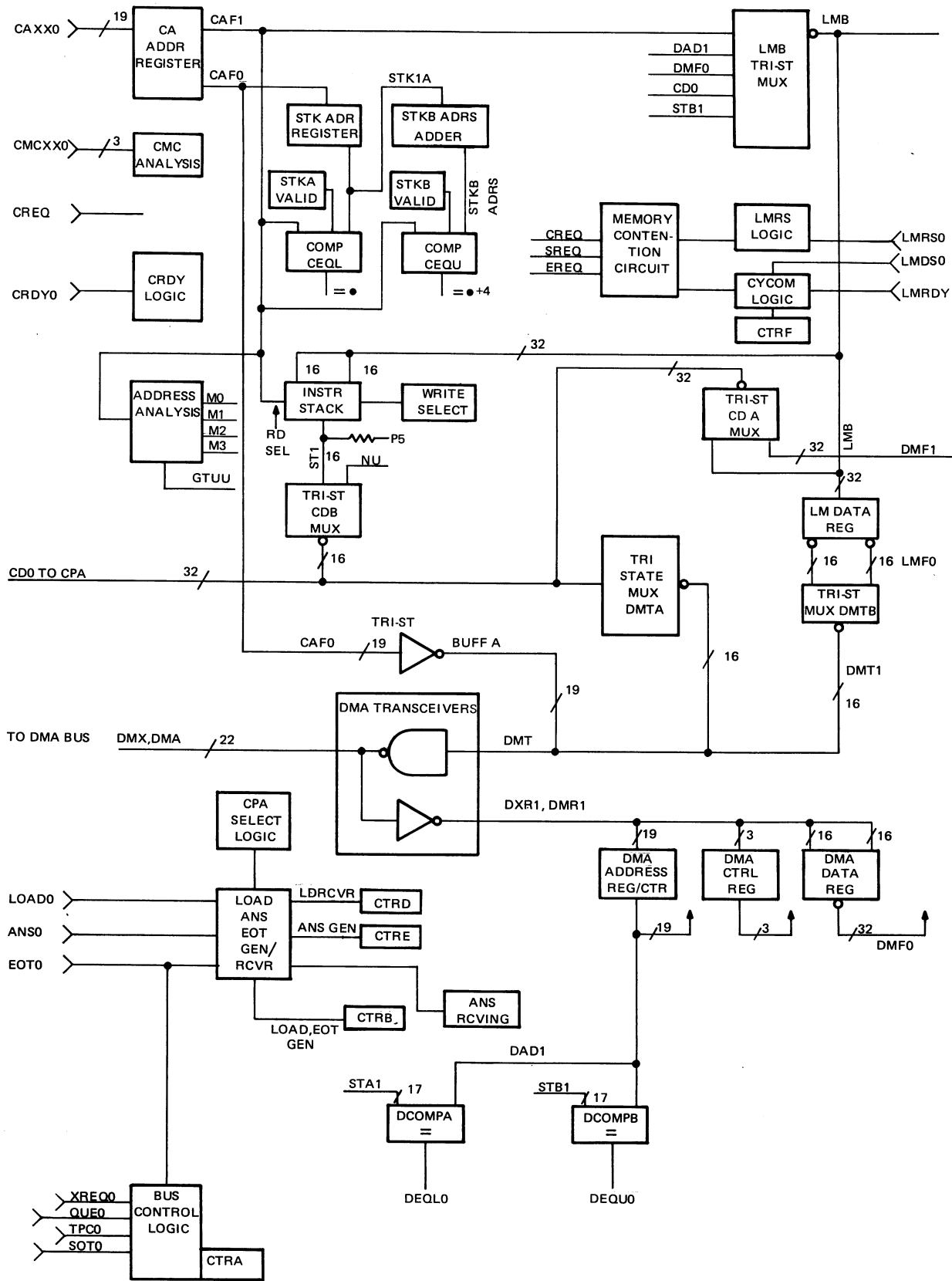


Figure 7. MBC Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

11. MEMORY BUS CONTROLLER (MBC) FUNCTIONAL DESCRIPTION

11.1 CPA to Local Memory Write Halfword of Fullword

11.1.1 Address Transfer Cycle. Refer to timing diagram Figure 8. The address bits from CPA come through tracking latches (Sheet 6) which are tracking because the CREQF (6F6) is still reset. The address is analyzed by the comparators (Sheet 9) and found to lie in the range of the Local Memory (Memory 0). The CMC bits (11H6) are also checked and it is found that this access is not an Instruction Read (IR).

The CREQ pulse is received from the CPA and stored in the CREQF (6F6). A delay line (14A7) is also started to give delayed signals equivalent to CREQF. The setting of CREQF freezes the address in the tracking latch and signals the start of the MBC response.

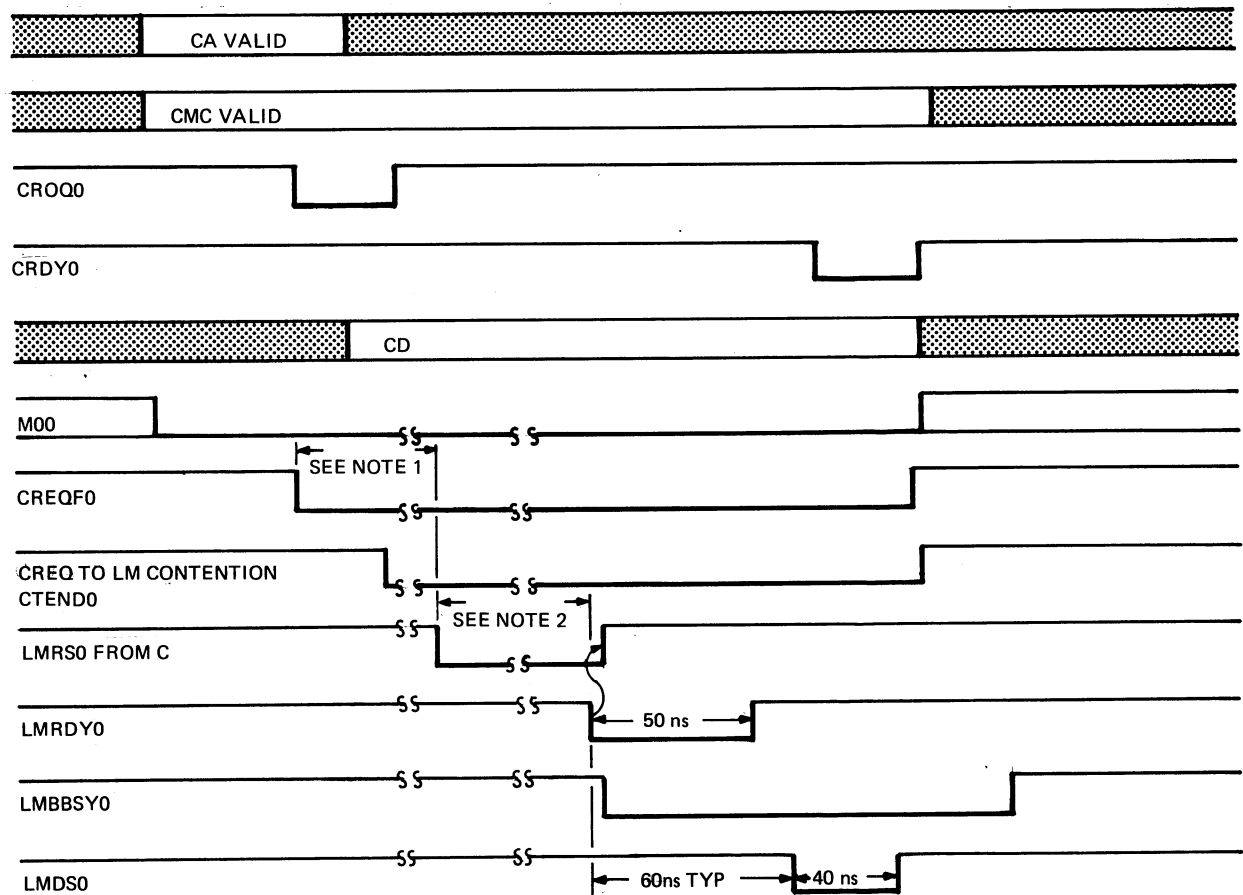
CTEND0 (13E8) is the input to the Memory Contention circuit and it is created from M01, IR0, CRQD001 (CRQD001 is CREQ delayed zero time), and no resetting signal. The CTEND0 signal is immediately sent when CREQ is received, the access is not an Instruction Read (IR0), and the address is in the range of Local Memory (M01).

The LMB multiplexors (Sheet 3) enable and select the CAFxx1 and CAFBxx1 signals with a low active signal on DATTM1 (13M9) and a high active signal of EFBO. DATTM1 is low active during address time except when the look-ahead stack is requesting to be filled from memory (SMZO being low).

The address time ends when the Local Memory Bus Busy flip-flop is set. This occurs when the first Local Memory Ready is received from the LMIs. The command code is also sent on the LMB at the same time as the address. Bit 27 on the LMB (LMB270) is low at this time for write commands.

The AND-OR-Invert gate (13N7) creates Local Memory Request Service (LMRS). The CF1 and EF0 inputs indicate that the Processor (C) and not the EDMA (E) is in control of the memory. LMRS is turned off by LMBBY being set. The D35 is a delayed enable signal from the contention delay line and is discussed in the Memory Contention circuit.

Note in the timing diagram that the gap between CREQ to LM and the LMRS0 from C could be caused by another cycle presently in progress or a request from the EDMA Bus which has a higher priority. The gap between LMRS0 and LMRDY0 is caused by the LMI being addressed while still being busy from a previous access.



NOTE 1: This Delay depends on whether or not the Memory is busy with an EDMA or Stack Request.
 NOTE 2: This Delay depends on whether or not the requested LMI is busy.

Figure 8. CPA → LM Write Timing Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

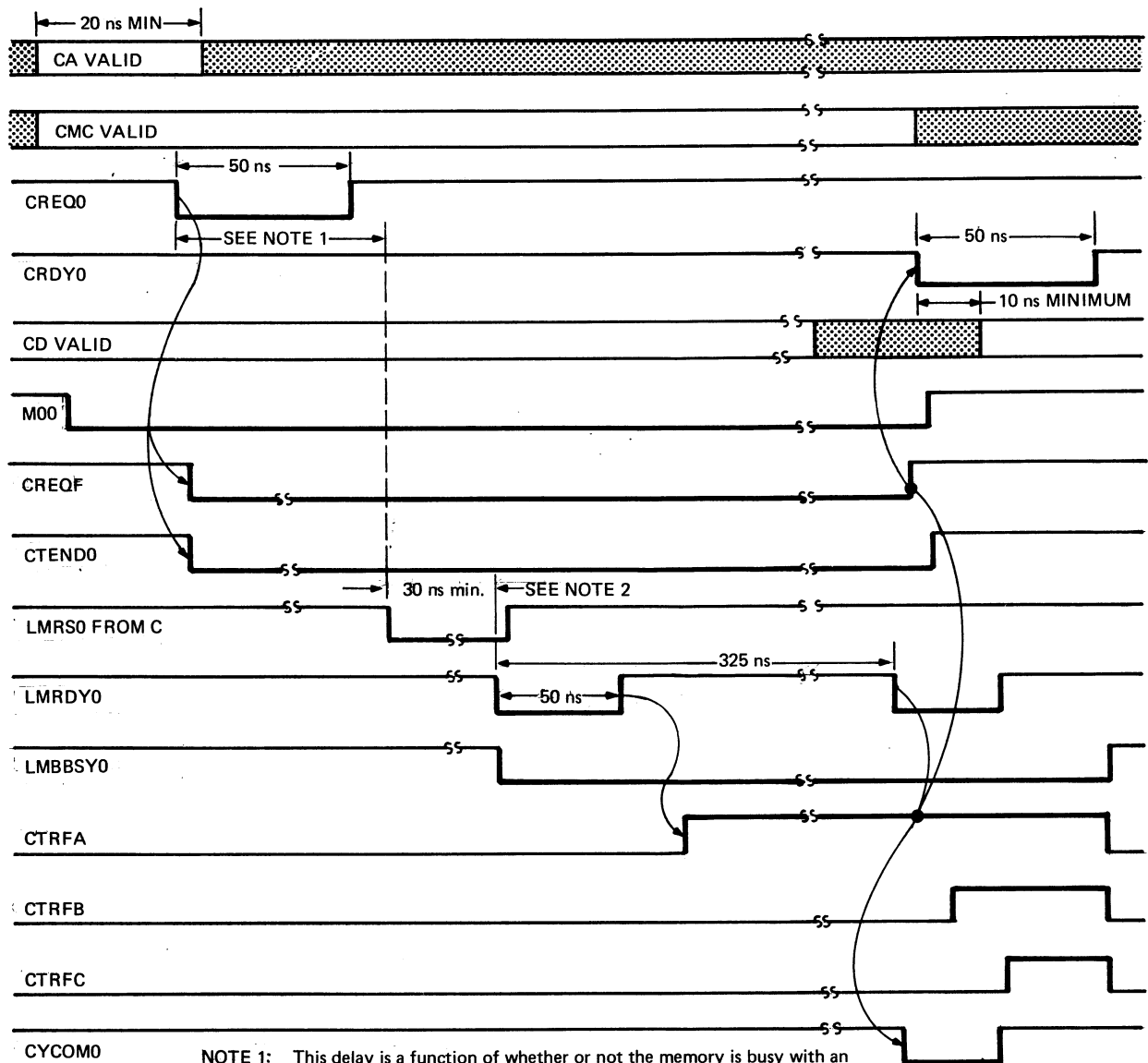
11.1.2 Data Transfer Cycle. During the LMRS signal, a delay line (11L9) is being loaded with a low active signal to generate the data strobes on the LMB and the Local Memory Data Strobe (LMDS).

The conditions necessary to load this delay line with a 0 are: 1. Not a stack request SMX0, 2. Local Memory Request Service (LMRS1), and 3. Data write command code as indicated by a 1 on LMB271. When LMRS is disabled (caused by the LMRDY signal from the LMI which sets the LMBBY flip-flop) the delay line (11L9) starts to reload with a 1 level. Meanwhile, DATTM1 (13L9) has gone high active because of LMBBY, causing WDTM0 (Write Data Time) to go low (11N7). Local Memory Data Strobe (LMDS) is also generated (11N5) by this delay line and occurs during WDTM.

The LMB multiplexors (Sheet 3) are switched at data time because the address enabling signal, DATTM1, goes high and the Write Data Time (WDTM0) signal goes low, enabling the CD lines onto the LMB and disabling the CAF lines. LMDS indicates the end of the cycle and Cycle Complete (CYCOM). CYCOM is generated in parallel on the AND-OR-Invert gate (11D8) by DA701 and DA400. Cycle Complete (CYCOM) resets the Memory Contention circuit. The ready signal to the Processor (CRDY) is generated by the AND-OR-Invert gate (14N8). LMDS causes CRDY through the gate (14K8) which is the OR of LMDS or Instruction Read ready from Local Memory (IRWRT - Instruction Read or Write). The CREQ flip-flop is reset in parallel by the same signal into the RCREQF circuit (14R5). There is no distinction made in the MBC between halfword and fullword accesses to Local Memory.

11.2 CPA to Local Memory Read Halfword or Fullword

11.2.1 Address Transfer Cycle. Refer to timing diagram Figure 9. This cycle is the same as for the data writes to Local Memory except that Bit 27 on the LMB (LMB271) is high at this time indicating a read command.



NOTE 1: This delay is a function of whether or not the memory is busy with an EDMA or Stack request.

NOTE 2: This delay is a function of whether or not Local Memory is busy.

Figure 9. CPA → LM Read Timing Diagram

11.2.2 Data Transfer Cycle. The edge of the first LMRDY signal from the LMI indicates that the address has been accepted and the data cycle follows. At this time the LMB multiplexors in the MBC are turned off waiting for the LMI to send back the data. This is accomplished by DATTM1 (13M9) going to the 1 state when LMBBY (13G9) is set.

Counter F, Stage A (CTF A) (11E4) is set on the trailing edge of LMRDY. The LMI sends another LMRDY when it has the data settled on the lines so Stage A of Counter F is used to differentiate between the first and the second LMRDY.

When the LMI sends data, it is placed on the CD lines through tri-state multiplexors (Sheet 4), CDA. At this time the Read Data (RDDAT0) signal is low (11H5). This is generated by the AND of CMC001 (11F5) and not an Instruction Read from Local Memory (STCD0) (11C5). (STCD0 – Stack to CD enable signal). RDDAT0 and STCD0 enables the multiplexors, and M00 (Memory Zero or Local Memory) or MB00 (Buffered Memory Zero) select the LMB lines.

Cycle Complete (CYCOM0) is generated by the second LMRDY when it is ANDed with CTF A and not an Instruction Read and not a Stack Access (SMX0) to memory (AND-OR-Invert gate) (11D8). CRDY is generated in parallel by CTFA1 (14N8), LMRDY (delayed by 5 nanoseconds extra), and the output of a flip-flop which indicates that the access was not an Instruction Read. The reset of the CREQF flip-flop is generated by CRDY (14R6).

11.3 CPA Instruction Read from Address in Local Memory

11.3.1 Instruction Valid in the Stack. The Control Address flip-flops (CAF) tracking latches are constantly being compared against the STA and STB signals. STA is the output of the register that stores the present base address of the data in the stack (Sheet 7). STB is the output of the adder (Sheet 7) that adds 1 to STA, giving the address of the data in the upper half of the stack. Note that there is no STB28 because this is the complement of STA28. This is the reason there are 17-bits of STA but only 16-bits of the adder are needed to add 1 to it.

The comparisons are done on Sheet 8. Bit 17 is handled in both cases by Exclusive OR gates (8H9). Even if the address compares in one of these two comparator circuits, other signals are needed to obtain an equal output. These signals are STKA, which indicates that the lower half of the stack is valid, STKB, which indicates that the upper half of the stack is valid; and SD200, which is a signal that permits the data in the stack to be loaded and settled from a stack access.

If either of these comparisons show up as equal by the time the CREQ signal reaches the 20 nanosecond tap on the CREQ delay line (14A7), then it is remembered in a cross coupled EQ1 flip-flop (14D7). EQ1, M01, IR1, and C0 (Instruction Read, and Processor not accessing memory) combine in an AND gate to become EQIRC0 (14G6). This combines with CRQD30 (CREQ delayed 30 nanoseconds) to form the CRDY back to the Processor (14M8). The equivalent of the Stack Ready (STKRDY) is generated two other places, once as the true signal (STKRDY1) (14G8) and once as the not signal at (STKRDY0) (14G7). These are used to load the STA register with the present base address of data in the stack (14J7) and to reset STKB (14L6) (indicator of the validity of the upper half of the stack) if the access is to the upper half of the stack, and generate a request to memory at the same time to refill the stack with a Set Stack Request (SSREQA) (14M6).

The data is delivered to the CD lines during this process from the stack (Sheet 5) through CD Multiplexor B, (Sheet 4). These multiplexors are enabled by a STCD0 signal (11C5). STCD0 is low whenever the Processor is doing an Instruction Read (IR) from Local Memory (M01).

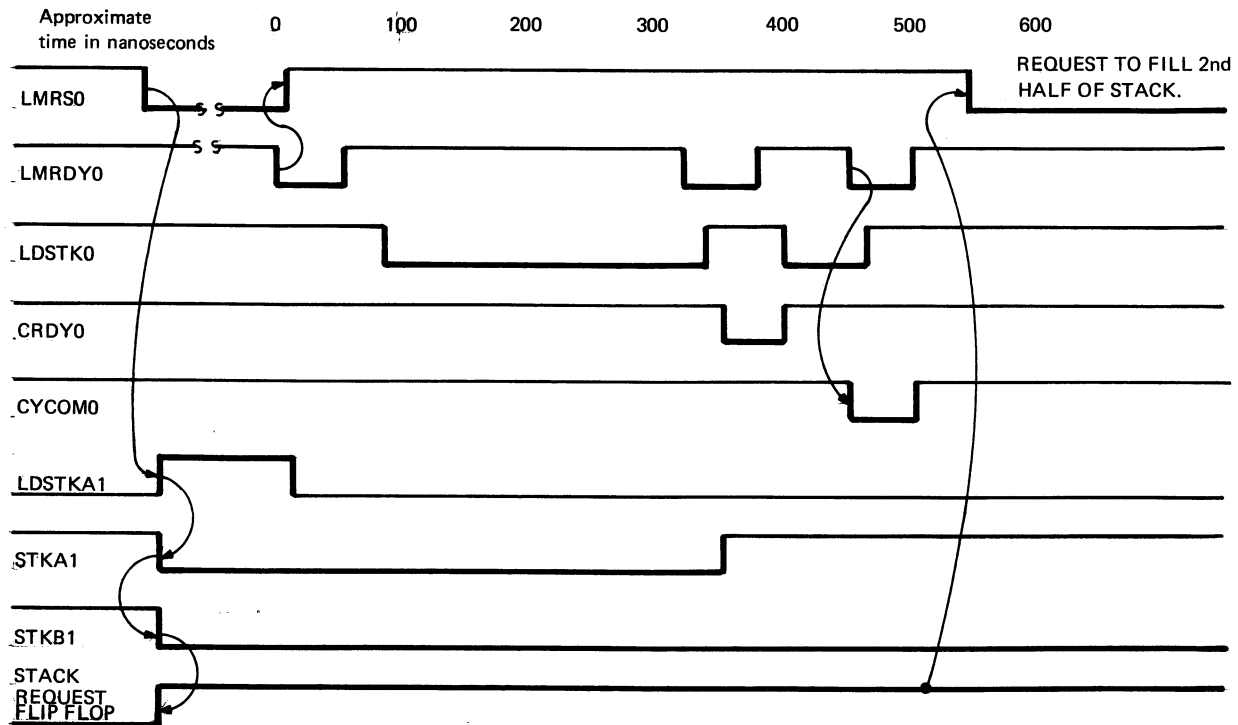
The correct word in the stack is addressed by the three least significant bits of the address.

11.3.2 Address Requested not in Stack or Stack not Valid. Refer to timing diagram Figure 10. If the comparators do not indicate an equal in either half of the stack and the address is in Local Memory (M0) then a request to Local Memory is made. This is accomplished with CTEND0 (13D8) created from not CEQL, not CEQU, M0, and not any other ready or complete presently being generated and the stack not being filled (SD500—Stack Delayed 50 nanoseconds).

The request and address transfer cycle to Local Memory is the same as for data reads or writes. The data cycle requires transferring two 32-bit words as follows. The second LMRDY from the LMIs indicates that the data fullword containing the requested halfword is on the LMB. This is enabled into the stack with the signal LDSTK0 (11C1). The delay line (11B2) creates the proper timing of this signal from the second LMRDY. CRDY (14R8) is generated at this time by the AND of CTFB1 and CTFC0. Counter F, Stage B is set during the second LMRDY and Counter F, Stage C is set at the trailing edge of the second LMRDY. Reset CREQF is generated in parallel by IRWRT0 (Instruction Read or Data Write) (14R6). Cycle Complete (CYCOM) is generated by the third LMRDY from the LMIs by the AND of Counter F Stage C (CTFC) and LMRDY at the AND-OR-Invert gate (11C8).

When the Processor has to go to Local Memory for an instruction, the STA register (Sheet 7) is loaded with the new base Address (LDSTA1) (14J7). The two flip-flops that indicate the valid state of the stack, STKA (14J4) and STKB (14L6), are both reset by C1LMRS0 (14 G5). There are three other signals that can reset these flip-flops through the clear input. These signals are System Clear (SCLR), writing into an address that is valid in the stack with a Write command from the EDMA Bus (DEQ), and writing into an address that is valid in the stack from the Processor (from CMC000, and C1 AND LMRS).

The STKA flip-flop is set again with the IRWRT signal (14H5) when ANDed with C1 (the Processor in charge of the Local Memory) and IR1 (a decode of the CMC bits that indicate an Instruction Read) (11H6). C1LMRS0 also sets the Stack Request flip-flop (13B5) to Local Memory to fill the upper half of the stack with another memory access.



NOTE 1: If Stack was valid but requested instruction is not in Stack, C1-LMRS0 will invalidate stack.

Figure 10. Instruction Read when Stack is Invalid or the Requested Instruction is not Currently in Stack

The data is written through the stack on an Instruction Read by controlling the Write select lines to the stack. These signals are STWB1 (14C1) and STWA1 (14E2). The S1 and SMX0 signals to this logic are from the Memory Contention circuit indicating when the stack is accessing memory. At this time however, since the Processor (C for CPA) is accessing memory, the stack is not. This leaves S1 low and SMX0 high. This gates the state of STA280 through to STWB1 and the state of CAF290 clocked into the flip-flop on the first LMRS.

Stage B of Counter F first enables the true side of the flip flop and then the complement side through to STWA1. Meanwhile STWB1 is directly the inverse of STA280 (14B2) and is actually the equivalent of Bit-28 of the control address from CPA. Recall that STA is the register that contains the base address of the data in the stack and it was just loaded with the present address being requested by CPA (LDSTA1) (14J7) so it reflects the present state of CAF28 for this Instruction Read.

For example, if CAF290 is high indicating a request to memory with an even fullword address, then the first data back from the memory is from the even fullword LMI. Stage B of Counter F (CTF B) is still low when this data is indicated valid by the second LMRDY, and STWA1 is low for the first data fullword and high for the second data fullword from memory.

11.4 Stack Control

- Stack request to memory
- Stack filled from memory
- Stack invalid-valid states

There are two ways that the stack may start a memory access. Since an Instruction Read from CPA fills the bottom half of an empty or invalid stack, a stack request is required to fill the upper half. The C1LMRS0 signal (14G5) is caused by a CPA request. It is generated when the Processor is making an Instruction Read request to Local Memory. The other signal that sets the Stack Request flip-flop (13C5) is SSREQA0 (14N6). This occurs whenever the CPA initiates an Instruction Read to an address in the upper half of the stack. STKB flip-flop is also reset at this time to indicate that this half of the stack is temporarily not valid. The leading edge of Stack Ready (STKRDY1) resets the STKB flip-flop if the address comparison was in the upper half of the stack as indicated by C address Equal to Upper half of stack (CEQU). Note that the lower and upper halves of the stack are defined by the address stored in STA register and not by the physical address in the stack. That is, at one instant of time the bottom half of the stack may be Addresses 0:3 and at another time the bottom half of the stack may be Addresses 4:7. (The data is not moved from one location in the stack to another. Only the STA Address register is reloaded.) The STKA flip-flop refers to validity of the data in the stack that is from the memory location indicated by the STA register, and the STKB flip-flop refers to validity of the data in the stack that is from a memory location indicated by the STB address.

The address sent to the memory during the request from the stack is from the STB addresses. This is put on the LMB multiplexors (Sheet 3) by a Stack Busy signal (SBSY0) 14C1). Note that LMB310 is high indicating a Stack request to memory. SBSY0 is low when the stack is accessing memory (S1) until the Local Memory Bus Busy flip-flop is set (LMBBY0) (indicating that the memory has accepted the address and the data transfer cycle is about to begin.

During the data cycle the LMIs pass back two 32-bit words of data the same as the Instruction Read to memory from the CPA. The Load Stack signal (LDSTK0) (11C1) is generated the same as for the Instruction Reads from CPA since the SXIR signal is the OR of Instruction Read from the CPA and Stack in control of the memory (11D6). Cycle Complete (CYCOM) (11F8) is from the third Local Memory Ready (LMRDY) as indicated by the state of Counter F Stage C (CTF C) the same as for Instruction Reads from CPA to memory.

The Write select lines to the stack to select the addresses that the data goes to are STWB and STWA (14A1) (the same as described in CPA Instruction Reads to memory). The fact that this memory access is from the stack looking ahead to fill the upper half, conditions these select lines. STWB1 is the complement of STA281 since the stack fills the upper half from the opposite double fullword address as the lower half. STWA1 is always low for the first 32-bit data word in a stack access because the even address fullword is always sent back first in a Stack request (recall in a CPA Instruction Read, the first fullword sent back is the one that contains the halfword that was requested and could be the even or the odd fullword). This is done by SMX0 (the signal that enables the Stack address onto the LMB through the LMB multiplexors) presetting the flip-flop (14B2).

The signal SD500 from the delay line (13K6) is used to permit the data to settle in the stack before permitting CPA to use it.

11.5 EDMA Bus Control Circuit (See Figure 11.)

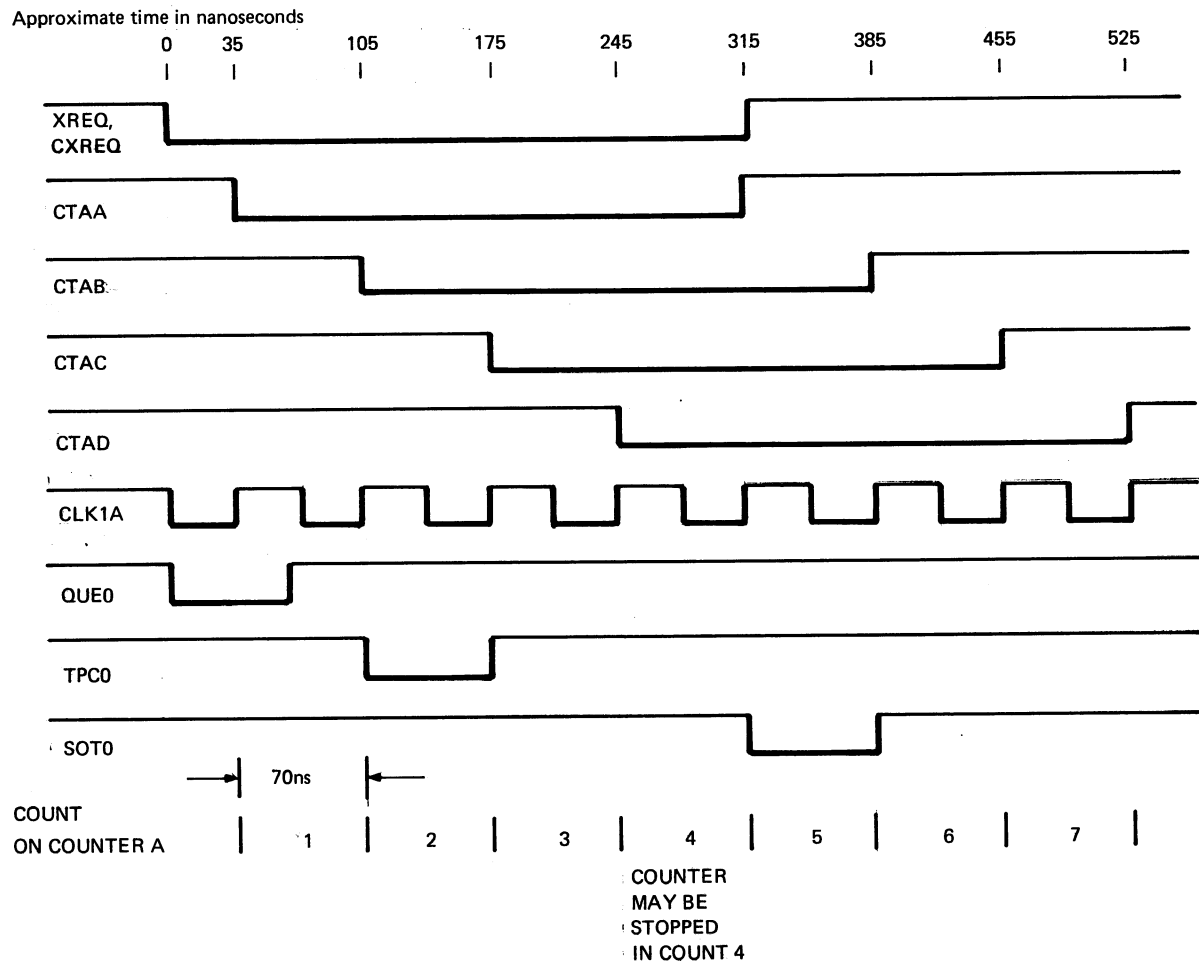


Figure 11. EDMA Bus Control

The EDMA Bus control circuit generates the QUE, Transmit Priority Chain (TPC), and Start of Transmit (SOT) signals to the bus. These signals are created from a 4-bit Johnson Counter which is labeled CTA (10H1). The counter can be stopped at either of two times by signals at the clock oscillator (10C1). At Count zero, Stage A and Stage D of the counter (CTAA and CTAD) are both high. At this time the counter is stopped unless there is a request from the EDMA Bus (XREQ) or a request for the bus from the Processor (CXREQ). This condition is the output of the four input NAND gate (10B1). When one of these requests arrives, the counter clock starts and immediately sends out a QUE0 (10N3). At Count two, the Transmit Priority Chain (TPC) (which is actually the beginning of the priority chain) is sent out (10R7). At Count 4, the counter can again be stopped (CTAA1 and CTAD1 both high) by the STHOLD1 signal (10K3). STHOLD1 is generated by the ORing of five signals. These signals are: 1. DMAACT0, which indicates that the DMA is presently transmitting data and another Start of Transmit (SOT) should not be sent at this time, 2. The Local Memory Request Received signal (LMRQR1) (which indicates that the device queued up is requesting Local Memory) is ANDed with EF0, which is the EDMA selected state of the Memory Contention circuit. This means that if the queued device is requesting Local Memory, Start of Transmit (SOT) is not sent until the EDMA has control of Local Memory in the Memory Contention circuit, 3. BH0 which is the Bus Hold signal indicating that a memory is still using the bus and will momentarily send back answers, 4. A delayed Bus Hold signal to inhibit false indications of Bus Hold being removed, or 5. EDMA Request and Burst Read (ERQBRD0) which indicates that when a Burst Read operation is occurring, another SOT should not be transmitted until the Burst Read access is completed.

Count 5 (CTA Stage B high and CTA Stage A low) (10L4) generates Start of Transmit (SOT0). SOT0 sets the DMAACT flip-flop indicating that the DMA is actively transmitting data at this time. End of Transmit or System Clear resets this flip-flop as the END0 signal (10M9).

11.6 CPA Select on EDMA Bus (See Figure 12.)

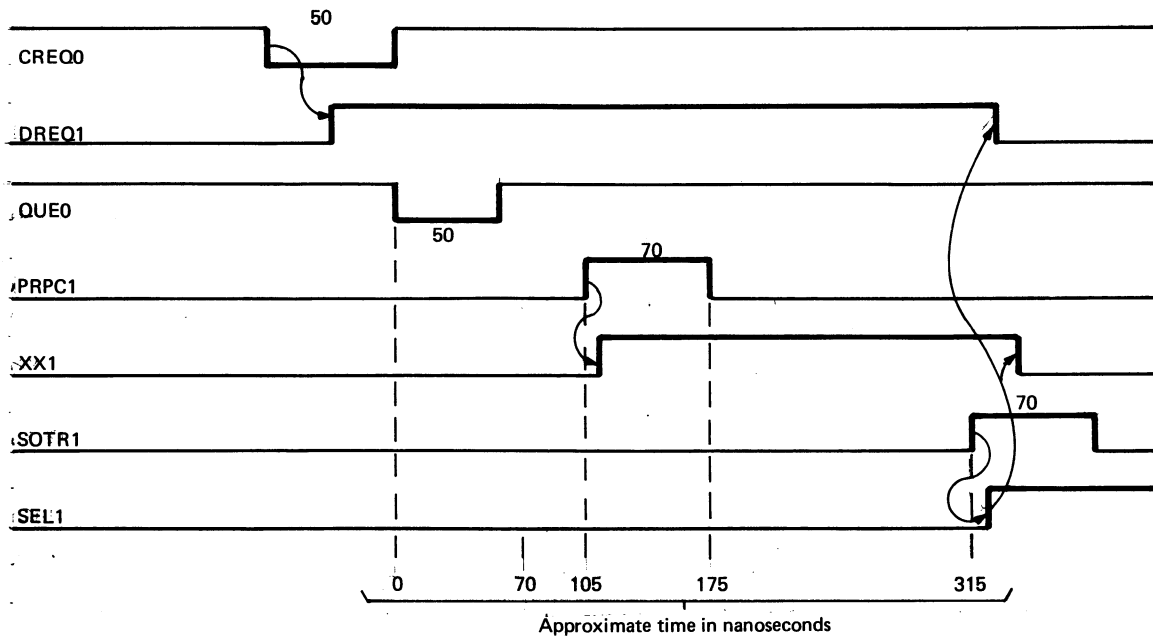


Figure 12. CPA Request to EDMA Bus

When CPA makes a request to the MBC for an address that is not in Local Memory, the address analysis logic indicates if the address is in Memory 1, 2, 3, or not in the system. This is done by the comparators (Sheet 9). The strapping on the comparators draws the division lines through the memory so that when an address (four most significant bits) is greater than one set of straps, but less or equal to the next set of straps, then the address is in that block of memory. When the address is found to be within the range of Memory 1, 2, or 3, the MBC goes out to the EDMA Bus for the access. The DREQ flip-flop (9L3) is set by a pulse when the address is not in the Local Memory addresses (M00) and less than the unequipped memory addresses (LTUU0). When the memory requested is no longer busy as determined by the AND-OR-Invert gate (9J5) then the CXREQ signal is sent to request the EDMA Bus and the SQUEF1 signal (9M5) sent to the select logic (Sheet 10).

The first Select Logic flip flop (10H6) is set on the leading edge of QUE0 when the SQUEF1 signal is present. This information is transmitted to the second flip flop on the trailing edge of QUE0. The leading edge of RPCR1 clocks the XX1 flip-flop (10K6) and the leading edge of Start of Transmit clocks this into the second Select Logic flip-flop (10N6) (SEL1).

When the SEL flip-flop (10N6) is set, the XX1 flip-flop is reset, and the DREQ flip-flop (9L3) is also reset. This ends the select sequence.

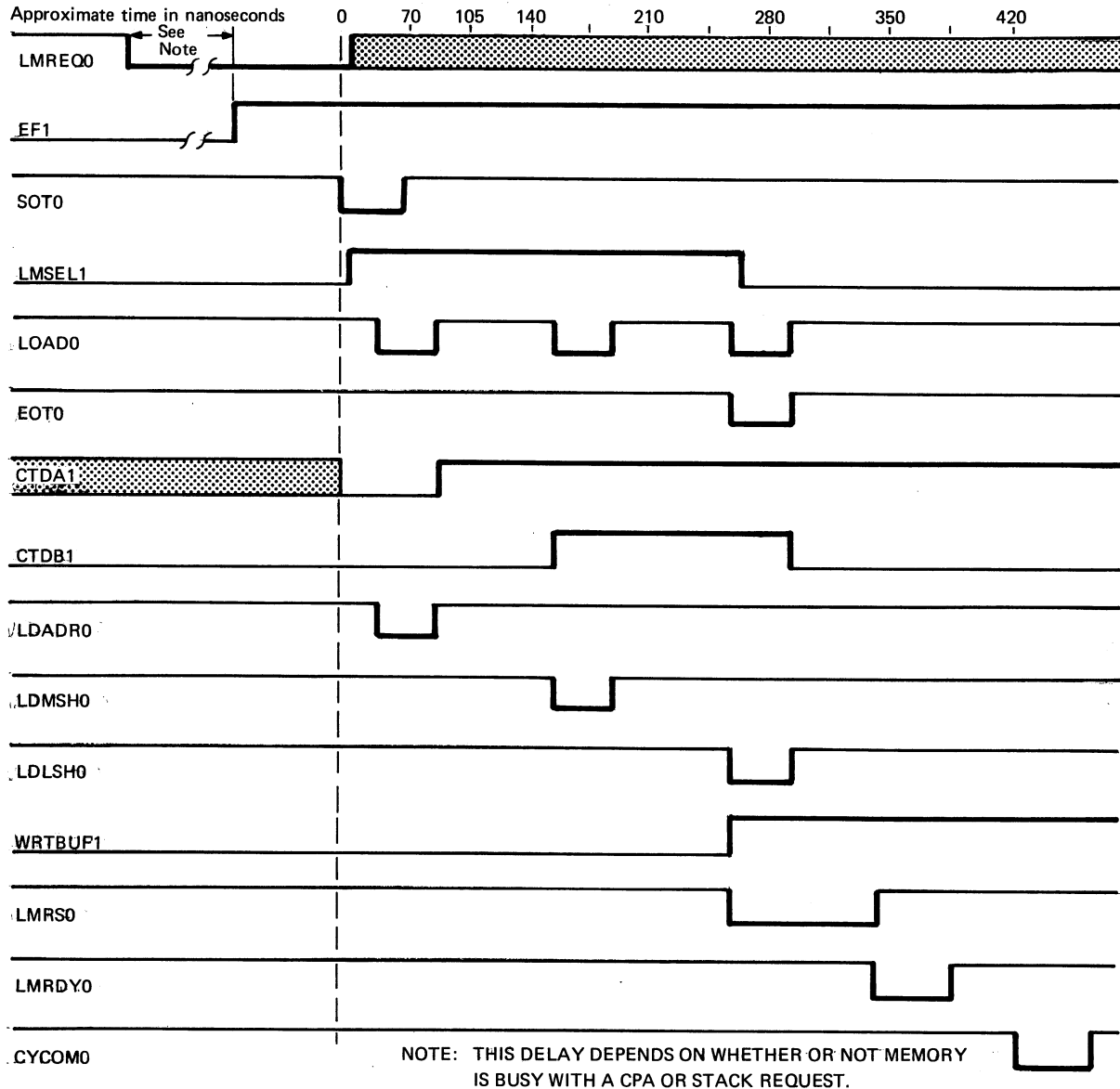


Figure 13. EDMA Bus Write to Local Memory

When LMRQR1 is high with Start of Transmission, the Local Memory Selected flip-flop is set (LMSEL) (12D6). The Start of Transmission (SOT0) initializes Counter D Stage A (CTD A), resets the Write Buffer Full flip-flop, and other flip-flops used.

Counter D keeps track of the loads being received so that the first load from the EDMA Bus loads the DMA Address Counter Register (Sheet 1). This register is loaded with a low on Address Finished (ADRFN1) (12H8) and a clock (CDMAD1) (11G9). The low on ADRFN1 is from Stage A of Counter D (CTD A) not being set yet. The clock signal is from the Load Address input (LDADRO) (11G9). This is generated (12C7) by the first load from the EDMA Bus after SOT0. At the trailing edge of this first load, CTD A is set and the Address cycle is finished. On the second load, a Load Most Significant Half (LDMSH) of the DMF Data Register (Sheet 1) is generated. This is the LDMSH0 signal (12H8) that is ORed (11M4) to actually load the register. Note that there is a load at the same time as the address load but it is of no significance because it is written over by the second one. The second load also toggles the Stage B flip-flop of Counter D (CTD B) (12F8). This enables the third load to set the Write Buffer Full flip-flop (WRTBUF) (12K8) and load the Least Significant Half (LDLSH) of the DMF Data Register with the LDLSH0 signal.

EHOLD0 (11M6) goes high now and the memory access is started. The EDMA Address register (DADxx) is put on the LMB by enabling the multiplexors (Sheet 3) with the DATTM1 signal as described in the CPA write to Local Memory, and selecting the DADxx inputs with the EFB0 signal from the Memory Contention circuit indicating that the EDMA is in control of the Local Memory Bus.

The remainder of the operation is the same as for CPA writes to Local Memory except that the data multiplexors to the LMB are selecting the DMF data inputs with the EFB0 signal. The Local Memory Data Strobe is generated in the same way.

The Local Memory Select flip-flop is reset with the End of Transmit signal but the operation is extended to the end of the cycle on the Local Memory Bus by the cross coupled flip-flop (13B2). This flip flop is set by LMSEL0, enabled to request memory with the DMA control bit that indicated a write, DMMC000, and reset with Cycle Complete (CYCOM). This flip-flop is another condition that delays Start of Transmit by holding the Bus Control counter with ERQWRT0 (10H3).

For halfword writes to Local Memory, Stage B of Counter D is held high by the DMMC021 signal being low. This is the control bit from the EDMA Bus, after it is stored in the register (1K8), that indicates a halfword operation when it is low. This control bit register was loaded at the same time as the address with the leading edge of LDDMC signal (12D7).

11.10 EDMA Bus Read from Local Memory (See Figure 14.)

When the LMRQ signal is sent on the EDMA Bus, the bus control circuit does not send SOT0 until the EDMA has control of the Local Memory Bus in the same manner as for an EDMA write to Local Memory.

SOT0 and LMRQ set the Local Memory Select flip-flop (12D5). The Address transfer cycle is the same as for the Write operation. The SOT0 signal initializes several circuits for the Read operation. These are the Read Buffer Full (RDBUFL) flip-flop (12C2) and the Counter E Stages A and B (12H3 and 12K4).

The gate (11K5) removes EHOLD0 when the ADRFN1 signal goes high. This occurs after the address is transferred to the DAD Registers. At this time the Read Buffer is also empty, reset by SOT0, and not in a Burst Read operation, and DMMC001 is high because it is a Read operation.

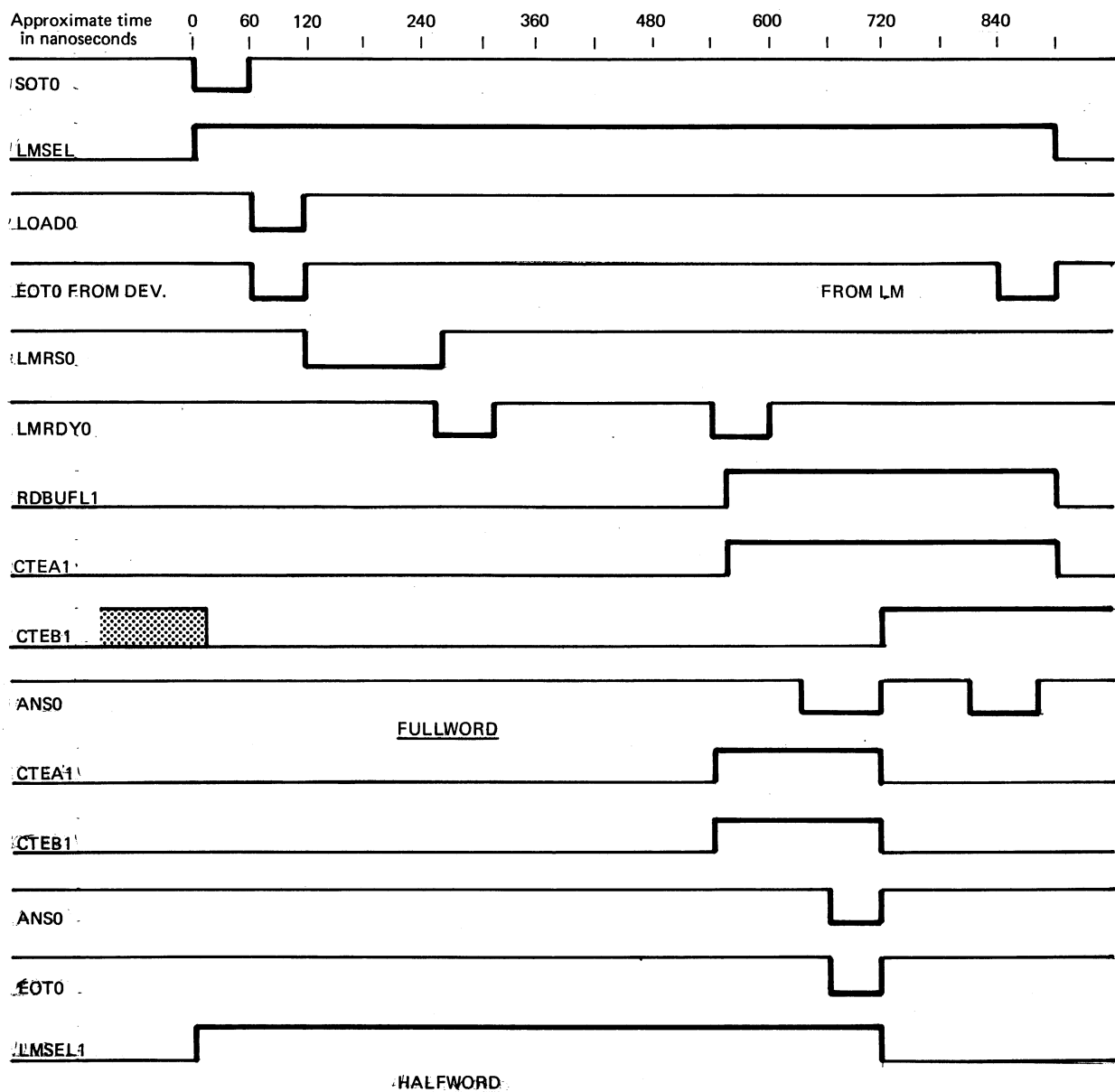


Figure 14. EDMA Bus Read from Load Memory

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The memory operation of Local Memory Request Service (LMRS) and two Local Memory Readys (LMRDYs) continue as usual. The Cycle Complete (CYCOM) signal is generated as with any data read command. CYCOM0 (12B2) is ANDed with LMSEL to set the Read Buffer Full (RDBUFL). The other two inputs to the gate (12D2) are DMMC001 and RDYDAT1. These are the command bits, indicating a Read operation, and a Ready for Data signal respectively, that can be low only during Burst Read operations. This gate therefore starts the delay line controlled clock for Counter E (12H2). This circuit generates the Answers (ANSWT1) (12J2) and enables and steers the LMB register (LMF) (Sheet 2) onto the EDMA Bus. Note that Counter E Stage A enables the multiplexors (Sheet 2) and Counter E Stage B (CTE B) selects the most significant half of the LMB register first and the least significant half second. Meanwhile the EDMA transmitters (Sheet 2) are enabled with the DMA Enable signal (DMENB0). DMENB0 is generated (10E9) by the OR of Counter E Stage A (which is the controlling signal for reads at this time) or Counter B not being at Count 0 (see Sections 11.7 and 11.8).

This description has implied fullword transfers. For halfword operations, DMMC021 is low and the gate (12J3) causes Stage B of Counter E to be set on the first clock. This transfers just the least significant half of the data word.

The gates (12D4 and 12E4) generate End of Transmit (EOT0) along with the second ANS0 except when it is a Burst Read operation. This EOT0 resets LMSEL flip-flop (12D5) ending the cycle.

11.11 EDMA Bus Burst Write to Local Memory (See Figure 15.)

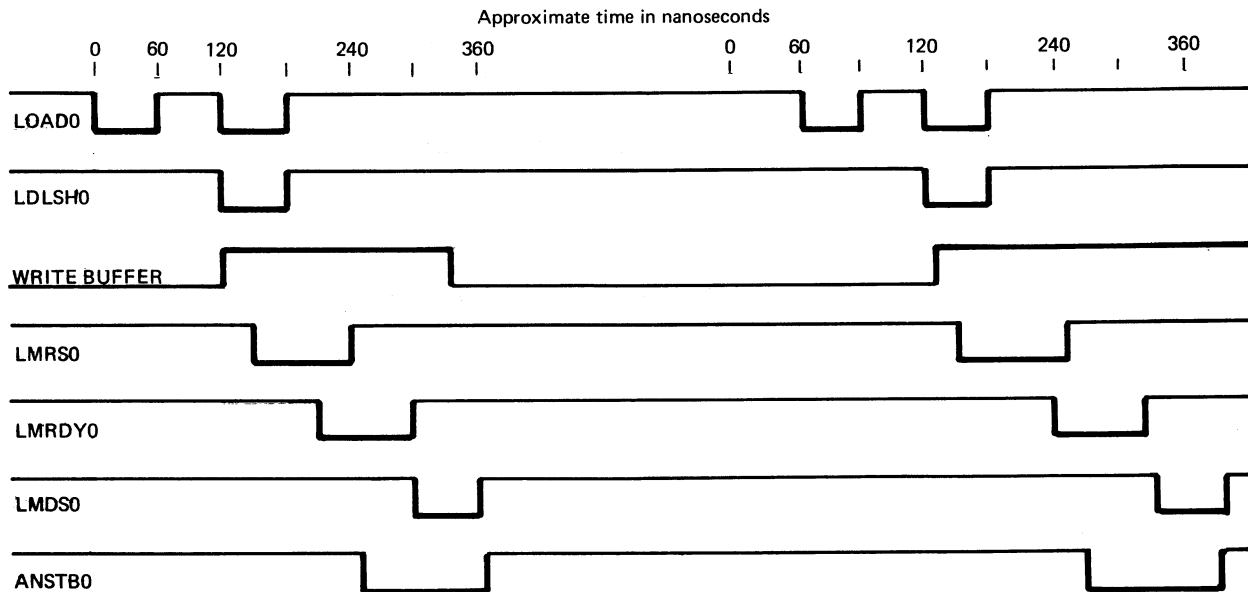


Figure 15. EDMA Burst Write to Local Memory

The Burst Write mode starts the same as the half or fullword write but continues on after the data is sent to memory.

The Burst Write mode to Local Memory operates by referring to the Write Buffer Full flip-flop (WRTBUF) (12K8). As the data is written out to Local Memory, ANSTB0 is generated (11M6) to signal the device that it may send more data. CYCOM0 resets WRTBUF when the Local Memory Data Strobe is sent out. WRTBUF is set when the LDLSH0 signal (12L8) loads the least significant half of the EDMA Data Register (DMF) (Sheet 1).

This operation continues with each CYCOM0 incrementing the EDMA address register counter (Sheet 1) with the CDMAD1 signal (11G9) until the device sends an EOT0.

11.12 EDMA Bus Burst Read from Local Memory

The Burst Read mode starts the same way as the half or fullword read. It does not send out an EOT0 with the ANS0 because of the gate at 12D5 that inhibits EOT0 when Burst Read (BRSTRD1) is decoded from the DMMC bits (12B1). The next memory access is made as soon as the Read Buffer is empty, by raising EHOLD0 (11N6) with the RDBUFL1 signal going low (11N5). The Read Buffer is emptied at the end of the data transfer to the bus.

The other controlling signal is RDYDAT1 (12D1). The flip-flop that enables this signal (12B3) is set by the LOAD0 as received from the EDMA Bus and reset as soon as CTE A is set. CTE A is Stage A of Counter E that generates the ANS0s to the EDMA Bus (12H3).

The gate (12D1) starts the clock for Counter E when the buffer is full, when the EDMA is ready for data, and when it is a Read operation (DMM001 being high). EHOLD0 is removed and the memory access started when the Read Buffer (12C2) is emptied. The operation continues until the device sends an EOT0. The MBC initiates an extra Read to Memory at this time, which is not used.

11.13 Memory Access Contention Control

Sheet 13 contains the Memory Access Contention Control logic. It handles requests from either of three inputs. These are EREQE1 (13C1), CTEND0 (13C3 and 13E8), and the Stack Request flip-flop (1305). EREQE1 is the request from the EDMA Bus and CTEND0 is the request from CPA.

These request signals are low active at the inputs to the first rank of flip-flops (Sheet 13). They also OR into the delay line (13F7). The 5 nanosecond tap on this delay line clocks the requests into the first rank of flip-flops and the 25 nanosecond tap clocks the requests into the second rank of flip-flops. It is at this second rank of flip-flops that the contention is resolved. If the EDMA or EF1 flip-flop is set it has control of the memory. The CF1 flip-flop indicates that CPA has control of the memory unless EF1 is also set. The bottom flip-flop is the Stack request flip-flop. It is the lowest priority and is reset if either the EDMA or CPA flip-flop is set.

The Stack Request flip-flop is set when CPA requests an instruction from the upper half of the stack (SSREQA0) or CPA has gone to memory for an instruction because it wasn't in the stack (CILMRS0).

CTEND0 is low (requesting the memory) when from top to bottom of the AND-OR-Invert gate (13D7) 1. ANDing M01, SD500, (CEQL0 or CEQU0), and CRQD201 – 50 nanoseconds after request to go to memory for Instruction Reads. 2. ANDing CRQD301, GTUU1, and CRQD001 – make a dummy access to memory that is not in this system configuration, to prevent Processor hang-up. 3. CI-feedback latch to store CTEND0 until CYCOM0, DMARDY0, or STKRDY0. 4. ANDing M01, IRO, and CRQD001 – Access to memory quickly if the request is to Local Memory and is not an Instruction Read.

EREQE1 is the OR of three EDMA Bus signals or states. From top to bottom (Sheet 12) they are: 1. LMREQR1 – Local Memory is Requested by the Queued device. 2. LMSEL and Burst Read code until the end of the next Cycle Complete (CYCOM). 3. LMSEL stored until next Cycle Complete and the DMMC bit that indicates a write command.

These requests are reset or at least removed during the Cycle Complete signal.

Local Memory Request Service (LMRS0) is generated at 13M7. The inputs to the AND-OR-invert gate are from top to bottom: 1. SS1 – Stack access after a delay to let the multiplexor settle. 2. Feedback to act as a flip-flop and store the request for timing elsewhere. 3. EF1 AND EHOLD0 – EDMA request and EDMA ready to access memory. 4. EF0 AND CF1 – EDMA not requesting, and CPA is requesting. The D35 signal is from the delay line at 13F7 to cover overlap of flip-flops, etc. The LMBBY0 (13F8) signal indicates that the first Ready signal has been received from the Local Memory and the address cycle is over.

12. LOCAL MEMORY INTERFACE (LMI) BLOCK DIAGRAM ANALYSIS

The LMI boards (two per system) control access to the Local Memory Modules (LMM) from the Memory Bank Controller (MBC) via the Local Memory Bus (LMB). Whenever a request is made by the MBC, the appropriate LMI sends address data and control signals to its Local Memory Modules (LMMs). Halfword and fullword data is steered to or from the LMMs according to the type of memory access. The LMI also contains the parity generation and check circuits for its LMMs.

12.1 LMI Block Diagram (See Figure 16.)

The LMI boards are idle until the Local Memory Request Signal (LMRS0) is generated by the MBC. The LMIs respond by performing a memory cycle. The memory cycle is the same for each type of memory access. They differ in the data steering required. The following steps outline a memory cycle.

1. Examine the address data on the LMB to determine which LMI should respond (refer to Section 3 on interleaved address organization).
2. Latch address and control data into registers of the correct LMI and indicate the start of the cycle by sending LMRDY0 to the MBC.
3. The LMI sends ERO to its LMMs to begin the cycle.
4. If the memory access is a Write operation, load data from the MBC into the write data register on the LMI, steer write data and WRT0 to the LMMs.

If the memory access is a Read operation, accept data from LMMs and store in Restore Cycle Data Registers. Relay this data and LMRDY0 to the MBC.

5. Generate or check parity as required for the type of memory cycle. Indicate parity errors to Processor via DLMP0 and IRLMP0.
6. Send INH0 to the LMMs to begin write/restore phase of cycle.

Note that if the memory access is an Instruction Read, both LMIs respond to the request (LMRS0) and each performs a memory cycle for its block of memory. This is explained in detail in Section 13.

The following paragraphs describe each functional block of Figure 16. Each block includes the pertaining schematic sheet number.

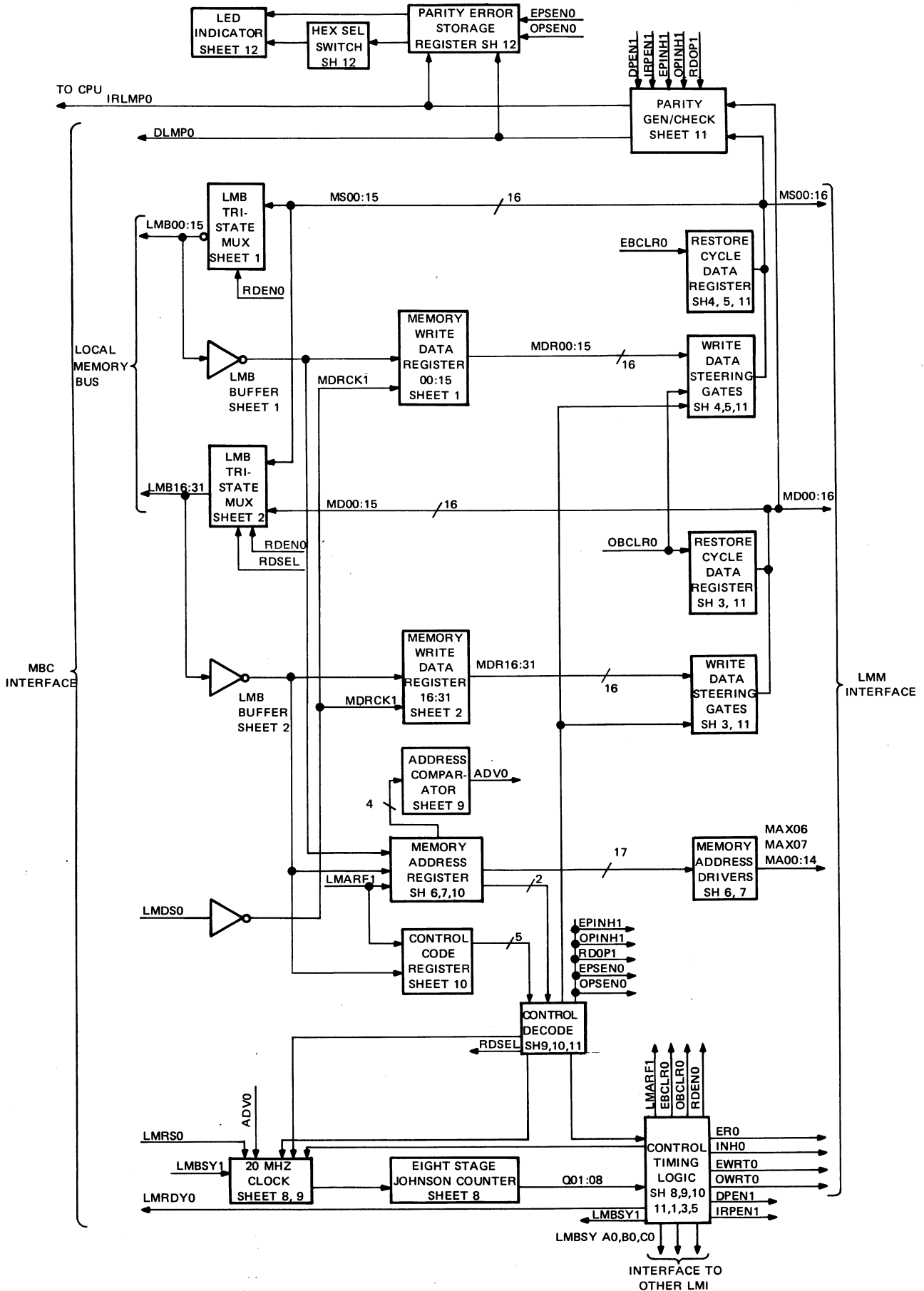


Figure 16. Local Memory Interface Functional Block Diagram

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

12.2 Local Memory Bus Tri-State Multiplexors

The LMI uses the LMB to transmit data from a memory Read operation to the MBC. These multiplexors serve as the LMB drivers for this function. The drivers are normally in a high-impedance state and are enabled for approximately 100 nanoseconds during a memory Read operation by RDEN0.

Note that this function is represented on the diagram as two blocks; one block drives LMB00:15, the other block drives LMB16:31. This shows the data steering function for halfword Read operations. LMB16:31 is used for transmitting all halfword format data between the MBC and LMI. These lines have two possible sources of halfword data (e.g., MS00:15 and MD00:15) corresponding to the most significant (even) halfword and the least significant (odd) halfword, respectively, within any given fullword. The correct selection of the source is made by RDSEL.

12.3 LMB Buffers

These serve to reduce the normalized load on each LMB line to one.

12.4 Memory Write Data Register

The Memory Write Data Register (MDR) receives data from the LMB for use in memory Write operations only. Data is strobed into the register by MDRCK1, which is the buffered and control gated back panel signal LMDS0. The MDR is shown divided into two blocks, and all data for halfword Write operations is stored in MDR16:31.

12.5 Restore Cycle Data Register

The Restore Cycle Data Register is used to store the memory module output data during a Read operation so that it can be written back into memory during the restore phase of a core memory cycle. This is required under two conditions; that is, during any Read operation, when the registers function as described previously, and during a halfword Write operation, when one LMM is being written into while the second is not. Recalling that two Local Memory Modules (LMMs) are cycled in parallel whenever the LMI is active, a halfword Write operation requires that the module which is not being written into be cycled through a Read operation, leaving its data unchanged.

The Restore Cycle Data Registers are normally in a tracking mode and follow the data lines. They are latched by the EBCLR0 and OBCLR0 control signals when the Local Memory Module (LMM) output data becomes valid.

12.6 Write Data Steering Gates

This block steers data to the correct LMM during memory Write operations. Note that the steering gates for the even halfword (MS00:15) have two possible sources of data. MDR00:15 is selected for a fullword Write operation. MDR16:31 is selected if a halfword Write operation to the even halfword is required, since all halfword write data from the MBC is stored in MDR16:31. Write data steering is controlled by four signals called WTE00:03. All Write Data Steering Gates are disabled for memory Read operations.

12.7 Memory Address Register

The Memory Address Register (MAR) stores the 19-bit memory address provided by the MBC. The devices used in the MAR are the tracking latch type, permitting the address to be quickly relayed to the Local Memory Modules (LMMs). It should be noted that the MAR does not latch up an address unless that particular LMI is required to respond to it.

12.8 Memory Address Drivers

The Memory Address Drivers are high current devices used in conjunction with terminated memory address lines to provide good transmission characteristics on these lines. This becomes important in systems that have additional memory located in a separate chassis.

12.9 Control Code Register

The Control Code Register (CCR) works in the same way as the MAR, and is used to store the 5-bit control code that identifies the type of memory access to be made (see Section 8).

12.10 Address Comparator

The Address Comparator takes the four most significant bits of the address from the MAR and compares them against an upper and lower address limit for that LMI which is selected by a strapping feature. The address must be within the allowed limits for the comparator to generate ADV0, which allows the control clock to start a memory cycle.

12.11 Control Decode Logic

The Control Decode Logic accepts the 5-bit control code stored in the CCR, the two least significant bits of the address stored in the MAR, and generates all the combinatorial logic outputs required for LMI control. The more important signals are shown on the block diagram.

12.12 Clock

The Clock is a gated 20 MHz square wave oscillator that provides the time base for operation of the LMI and control of the Local Memory Modules (LMMs). Operation of the clock is conditional on LMRS1, ADV1, STCLKA1, and STCLKB1 all being valid.

12.13 Counter

The Counter is an eight stage Johnson Counter driven by the 20 MHz clock described previously. The outputs of its eight stages are decoded to provide directly the signal timing needed within the LMI and LMMs.

Note that a ninth stage is shown in the schematics. This stage is only used when the system is configured with 64 KB Local Memory Modules (LMMs) and is described in Section 13.7.

12.14 Control Timing Logic

The Control Timing Logic accepts the output from the Counter and Control Decode blocks as inputs and develops all time based control signals required by the LMI and the Local Memory Modules (LMMs). In addition, the Control Timing logic block controls the synchronization of two LMIs for the instruction read cycle. The important signals are shown on the block diagram.

12.15 Parity Generation and Check

The parity logic generates the parity bits during Write operations and checks for correct parity during Read operations. One parity bit is provided for each halfword of memory.

If a parity error is detected during a Read operation, the parity logic signals the Processor via IRLMPO or DLMP0, which indicates parity error in an Instruction Read, or parity error in a data read, respectively.

12.16 Parity Error Storage Register

During a parity fail indication (IRLMPO or DLMP0), the Parity Error Storage Register determines which of the Local Memory Modules (LMMs) being accessed is the source of the parity error and sets a flag to identify that module. The parity error storage register can monitor up to 16 Local Memory Modules (LMMs), which is the maximum number that one LMI can accommodate.

12.17 Hexadecimal Selector Switch and LED Readout

These two blocks are used to read out the contents of the Parity Error Storage Register and determine which of the Local Memory Modules (LMMs) is the source of the parity errors received.

13. LMI DETAILED FUNCTIONAL AND TIMING DESCRIPTION

This section describes in detail the timing of all memory operations referencing Functional Schematics 35-534D08. The reader should be familiar with Sections 1 through 9 before reading this section. The timing given here is for a system using 750 nanoseconds cycle time with 32 KB memories, and assumes that the memory being accessed is not busy when the request is initiated. Timing differences required by other memory modules is explained in Section 13.5.

13.1 Fullword Write (See Figure 17.)

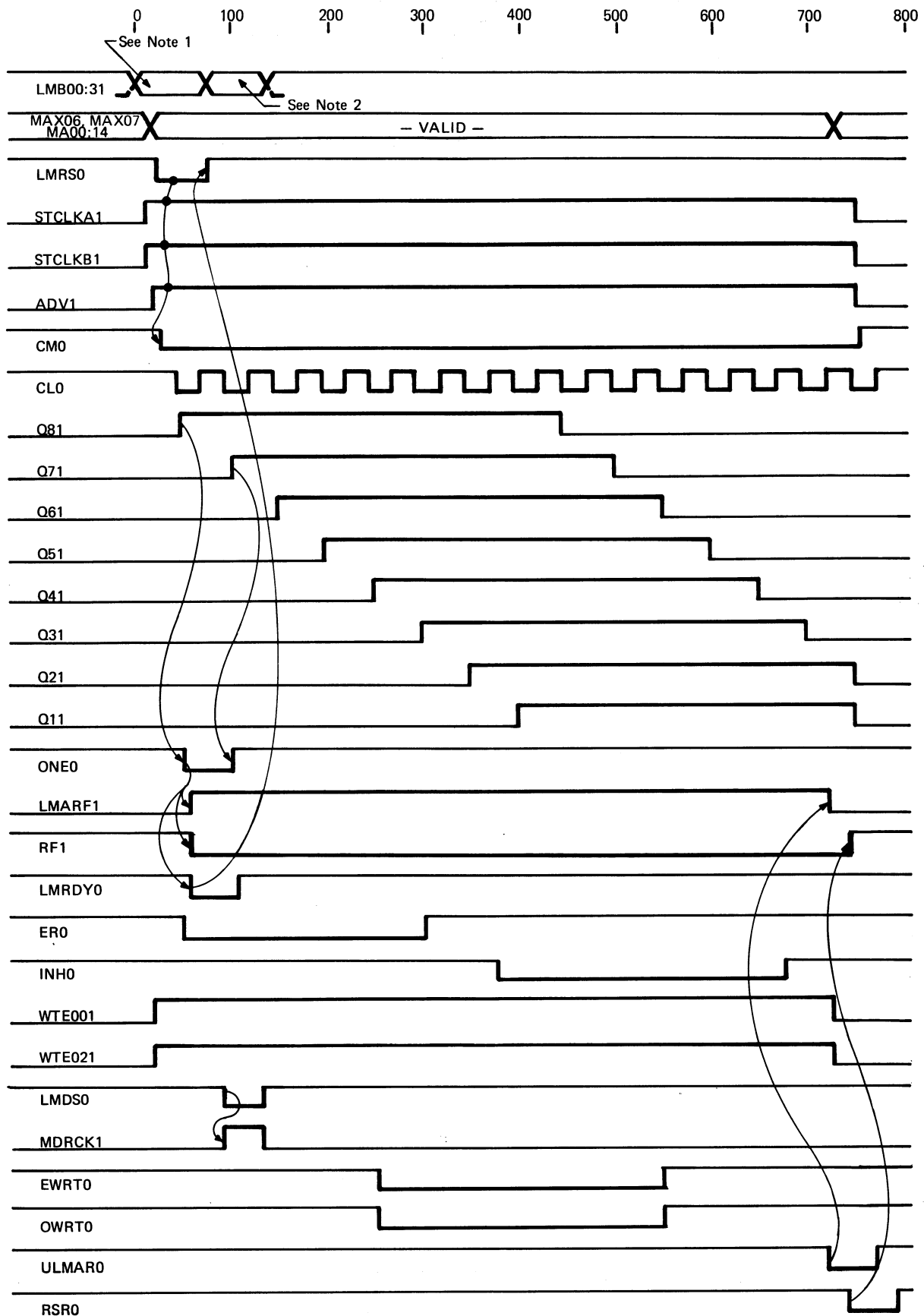
A fullword Write operation begins with address and control data being placed on the LMB by the MBC. The memory address, in low-active polarity, is placed on LMB Lines 00:18; while the 5-bit control code is placed on LMB27:31 in high-active polarity. LMB polarity is thus mixed at this time. Approximately 20 nanoseconds after the LMB data is valid, the MBC generates LMRS0 and holds it low waiting for the LMI to respond.

The 17 most significant bits of the memory address pass through the address tracking latches (Sheets 6 and 7) and are transmitted via the MA Bus to the Local Memory Modules (LMMs). The four most significant bits are also routed to the address comparator (9J3) where they are compared against the prewired upper and lower address limits for the LMI. If the ADRS is within limits, ADV0 (9R3) goes low, forcing ADV1 (8L3) high.

If Bit 18 of the 19-bit address is in the correct state, as determined by a strap at 10F8, FWD0 (10J8) goes low, forcing STCLKA1 (8K2) high. Bit 18 determines which of the two LMIs controls the fullword location desired, since the LMIs are interleaved on fullwords. The Control Decode Logic decodes INSRD1 (9C2) as low, since this is not an Instruction Read operation. This forces STCLKB1 (8K3) high.

Since the MBC is holding LMRS0 low, LMRS1 (8F1) is high, and LMRS1, STCLKA1, STCLKB1, and ADV1 sets CM0 (8N2) low. CM0 starts the clock running (8B5) and the buffered clock output CL0 (8C5) drives the eight stage Johnson Counter (Sheet 8), which is triggered by the clock's negative going edge. The ninth stage, QA1 (8L8), does not enter into the count sequence (see Section 13.5).

Two features of the counter are: a self-start feedback decode (SSFB0) (8R5) which is used in conjunction with a complete decode of one counter state (9B2) to assure that the counter does not lock into an undesired sequence of states, and its modified interconnection of stages Q21 (8M6) and Q11 (8N6) which causes the counter to have only 15 states, instead of the normal 16 (see Timing Diagram, Figure 17).



NOTE 1: During this time, LMB00:18 contains the low-active address, and LMB27:31 contains the high-active control code.

NOTE 2: During this time, LMB00:31 contains 32 bits of high-active data.

Approximate time in nanoseconds

Figure 17. Fullword Write Timing

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

The first clock edge causes ONE0 (9F3) to be generated, which in turn sets LMARF1 (10C1) high and RF1 (8A2) low. LMARF1 latches the address and control data into the Memory Address Register (MAR) and Control Code Register (CCR) respectively. RF1 keeps STCLKA1, STCLKB1, and ADV1 high, while its complement, RF0 (8C2), latches the Local Memory Request flip-flop (8D2) in the high state. These insure that CM0 remains low-active for the duration of the memory cycle. In addition, ONE0 causes LMRDY0 (9K2) to be sent back to the MBC. Upon receipt of LMRDY0, the MBC releases LMRS0. The first clock edge also causes ERO (8F9) to be sent to the Local Memory Modules (LMMs), thus beginning the memory access cycle.

The MBC places 32-bits of high active data on the LMB to be written into memory. It sends LMDS0 to the LMs which becomes MDRCK1 (1R6) and strobes the data into the Memory Data Register (MDR00:31) (Sheets 2 and 3). In the interim, WTE001 (10L2) and WTE021 (10L3) go high, steering the data to the MS and MD Busses (Sheets 3, 4 and 5) and into the memory modules. Approximately 200 nanoseconds after the leading edge of ERO, EWRT0 (9G6), and OWRT0 (9G5) are sent to the Local Memory Modules (LMMs), inhibiting the data currently stored in the accessed address from being placed on the MS and MD Busses. Approximately 325 nanoseconds after the leading edge of ERO, INH0 (8H2) is generated and sent to the Local Memory Modules (LMMs). This begins the write phase of the memory cycle.

After the write phase is completed, ULMAR0 (9R8) unlatches the Memory Address Register (MAR) and the Control Code Register (CCR), allowing address and control data for a new memory cycle to be accepted. Approximately 20 nanoseconds later, RSR0 (9G1) resets RF1 high, which unlatches the Local Memory Request flip-flop and frees STCLKA1, STCLKB1 and ADV1 for a new cycle. The timing of RSR0 is such that CM0 goes high and disables the clock when the counter is in the clear (all low) state, thus leaving the control logic ready to begin another cycle.

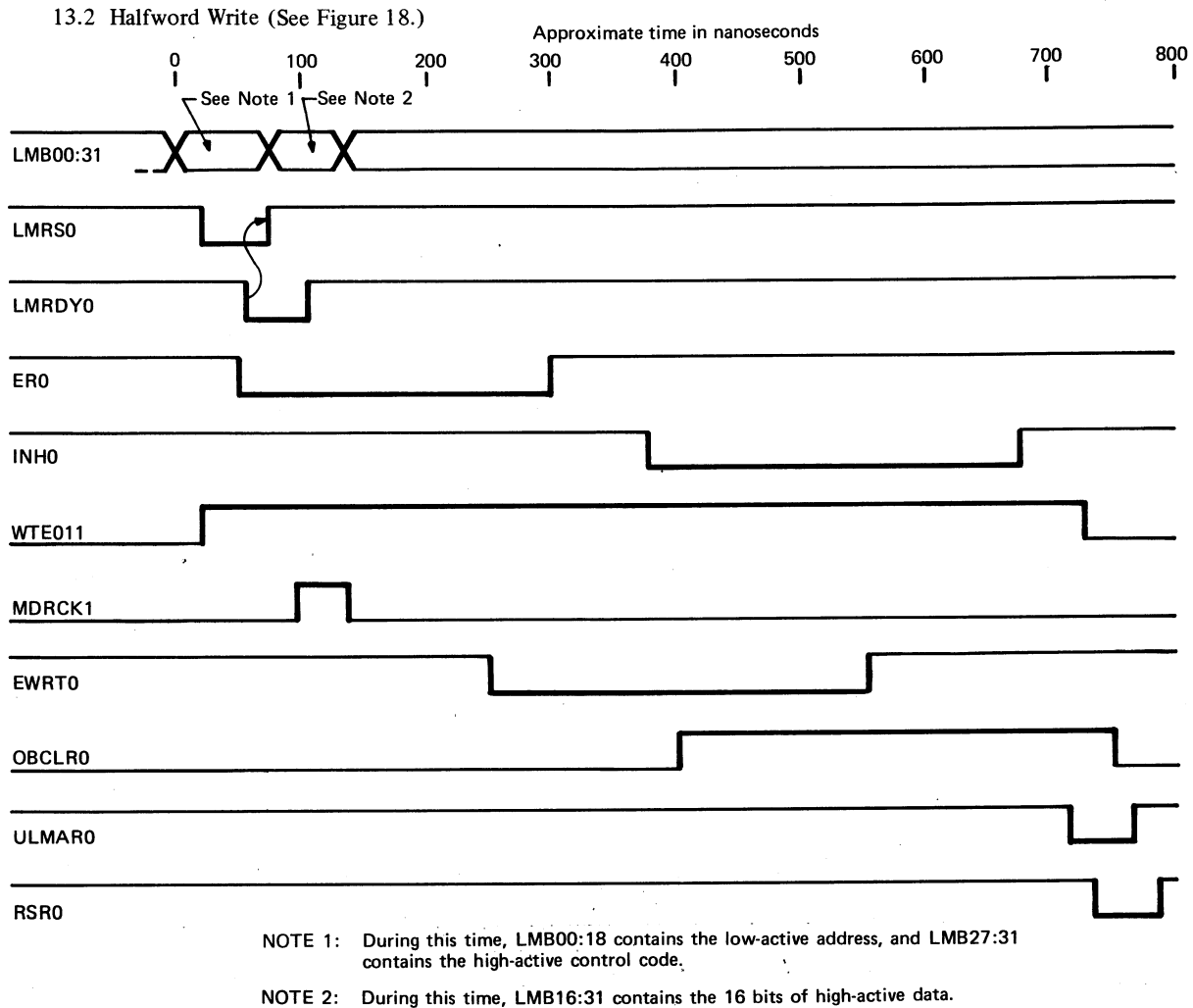


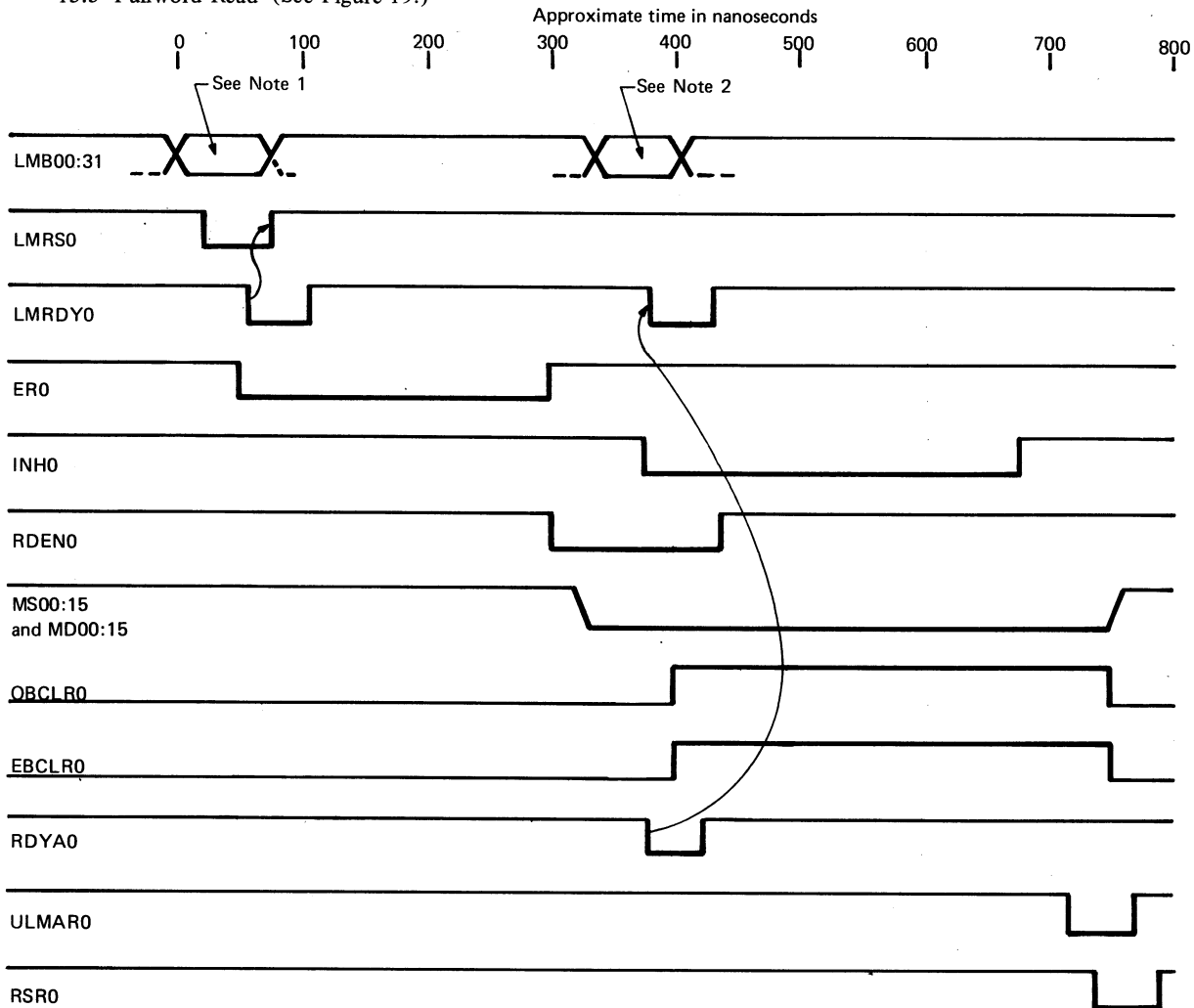
Figure 18. Halfword Write Timing (Most Significant Halfword)

The halfword Write operation is very similar to a fullword write; the difference is essentially in steering the halfword of data from the LMB to the appropriate Memory Module Bus (MS or MD) and restoring the data in the other unmodified halfword of the fullword actually being accessed. (A LMI board always accesses two Local Memory Modules in parallel. When a halfword operation is specified, the other, unused, halfword of the accessed location must remain unchanged.) This section describes a halfword write into the most significant (even) halfword. A least significant (odd) halfword write may be extrapolated directly from the description.

The halfword write cycle begins the same as a fullword write. The description in Section 13.1 up to and including the generation of ERO applies here. After the receipt of LMRDY0, the MBC places the 16-bits of high-active data on LMB16:31 and sends LMRDY0 which becomes MDRCK1(1R6) and strobes the data into MDR16:31 (Sheets 1 and 3). (All halfword data transmitted over the LMB is placed on LMB16:31.) WTE011 goes high and steers the data in MDR16:31 out to MS00:15 (Sheets 4 and 5). WTE001 and WTE021 remain low. Approximately 200 nanoseconds after the leading edge of ERO, EWRT0 (9G6) is generated, inhibiting the data currently stored in the most significant halfword from also being placed on MS00:15. Approximately 350 nanoseconds after the leading edge of ERO, OBCLR0 (3L8) goes high, allowing the data read out of the least significant halfword to be stored in the Restore Cycle Data Register flip flops of MD00:15 (Sheet 3). Approximately 325 nanoseconds from ERO, INH0 (8H2) is generated and sent to the Local Memory Modules (LMMs). During the write phase initiated by INH0, new data is written into the most significant halfword on MS00:15, while the old data read from the least significant halfword on MD00:15 and stored in the Restore Cycle Data Register is rewritten without change.

After the write phase is completed, ULMAR0 (9R8) and RSR0 (9G1) reset the LMI for the start of another cycle as described in Section 13.1.

13.3 Fullword Read (See Figure 19.)



NOTE 1: During this time, LMB00:18 contains the low active address, and LMB27:31 contains the high-active control code.

NOTE 2: During this time, LMB00:31 contains 32 bits of high-active data.

Figure 19. Fullword Read Timing

The Fullword Read cycle begins like a fullword write. The description in Section 13.1, up to and including the generation of ERO, is applicable here. After the receipt of LMRDY0, the MBC releases the LMB and waits for the LMI to respond with data. Approximately 250 nanoseconds from the leading edge of ERO, RDEN0 (IA8) goes low active, enabling the 2:1 multiplexors which serve as LMB drivers (Sheets 1 and 2). RDSEL (10K5) is low, which steers MS00:15 onto LMB00:15 and MD00:15 onto LMB16:31 (Sheets 1 and 2). The output data from the Local Memory Modules (LMMs) appears on MS00:15 and MD00:15 no later than 275 nanoseconds from the edge of ERO, and is steered to the LMB as described previously. At 325 nanoseconds from ERO another LMRDY0 pulse (9K2) is generated by RDYAO (9G3) which indicates to the MBC that data is valid on the LMB. Approximately 25 nanoseconds later, EBCLR0 (5H8) and OBCLR0 (3L8) go high, allowing the data on MS00:15 and MD00:15 to be stored in the Restore Cycle Data Register flip-flops. INH0 (8H2) is generated 325 nanoseconds from ERO, initiating the write phase of the Local Memory Module cycle. This causes the data stored in the Restore Cycle Data Registers to be written back into memory.

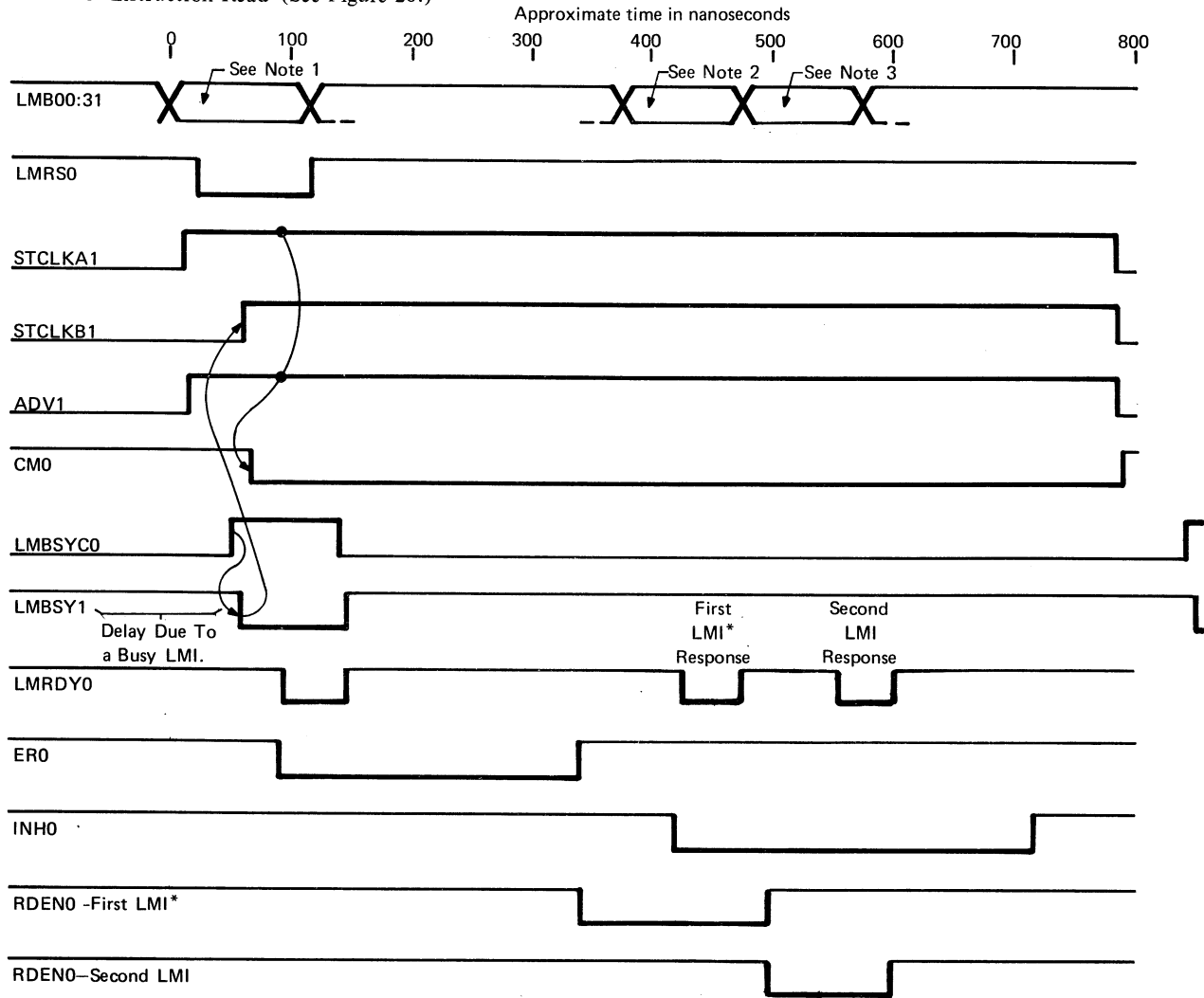
After the write phase is completed, ULMAR0 (9R8) and RSR0 (9G1) reset the LMI for the start of another cycle as described in Section 13.1.

13.4 Halfword Read

In all halfword Read operations, the requested halfword is placed on LMB16:31 (Sheet 2). A halfword read cycle to access the least significant (odd) halfword in a fullword is identical to the fullword read cycle described in Section 13.3. The MBC is this case only uses the data on LMB16:31.

In a halfword read cycle to access the most significant (even) halfword, RDSEL (10K5) is high. All other details of the cycle are identical to a fullword read (Section 13.3). When RDSEL is high, the data on MS00:15 is steered to LMB16:31 (Sheet 2), from which it is accepted by the MBC.

13.5 Instruction Read (See Figure 20.)



* The first LMI controls the address containing the instruction halfword requested by CPA.

Note 1: During this time, LMB00:18 contains the low active address, and LMB27:31 contains the high active control code

Note 2: During this time, LMB00:31 contains a fullword of data from the first LMI.*

Note 3: During this time, LMB00:31 contains a fullword of data from the second LMI.

Figure 20. Instruction Read Timing

The Instruction Read operation is a double fullword (64-bits) read, in which both LMI boards cycle in parallel and each sends a fullword over the LMB in sequence. Two important features of LMI operation during this type of request are: 1. Both LMI boards must be quiescent (i.e., not cycling) before the operation can begin, and 2. The first of the two fullwords of data sent to the MBC contains the instruction halfword requested by CPA.

This cycle begins with address and control data placed on LMB00:18 and LMB27:31 respectively, followed by LMRS0 from the MBC (see Section 13.1 for details). INSRD0 (9E2) goes low, forcing STCLKA1 (8K2) high. If the address is within range, ADV1 (8L3) goes high. Since INSRDI (9C2) is high, STCLKB1 (8L3) does not go high unless LMBSY1 (9G8) is low. As shown in Figure 21, LMBSY1 does not go low unless LMBSYC0 (9H8) from both LMIs is high, indicating that both LMI boards are quiescent. This ensures that the LMI boards will cycle in parallel. When LMBSY1 goes low, STCLKB1 goes high. CM0 (8B5) is forced low and starts the clock, and each LMI begins a fullword Read operation.

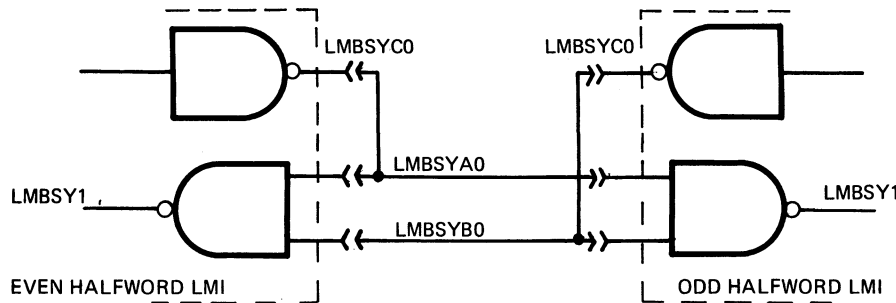


Figure 21. LMSY1 Logic

The LMI that controls the address containing the instruction halfword requested by CPA completes a normal fullword read cycle as described in Section 13.3. Thus, RDEN0 (1A8) goes low 250 nanoseconds from the leading edge of ER0 (8F9). This is followed by a fullword of data not later than 275 nanoseconds from the leading edge of ER0, and LMRDY0 (9K2) generated by RDYA0 (9G3) logic 325 nanoseconds from the leading edge of ER0.

The other LMI also completes a fullword read cycle, but delays placing the data on the LMB as follows: data is received on MS00:15 and MD00:15 not later than 275 nanoseconds from the leading edge of ER0 as before, but RDEN0 does not go low until 400 nanoseconds from ER0, at which time the fullword of data is gated to the LMB (note that RDEN0 in the first LMI has gone high, disabling the outputs). This is followed by a LMRDY0 pulse at 450 nanoseconds from ER0, generated by RDYB0 logic (9E4). Data from each of the two LMI boards is received, a fullword at time, spaced by 125 nanoseconds.

13.6 Parity Generation and Checking

Parity bits MS160 and MD160, correspond to the most significant (even) halfword and least significant (odd) halfwords respectively. The parity bit is generated whenever its corresponding halfword is written. When a Write operation is initiated, the parity generator/checker integrated circuits (11C7) used for the halfwords being written into are enabled via EPINH1 (11D1) and/or OPINH1 (11C7) (e.g., on an even halfword write, only EPINH1 goes low and enable the I.Cs. The data on MS00:15 or MD00:15 generates EPGEN1 or OPGEN1 respectively, which is gated onto MS160 or MD160 by WTE031 (11H1) and written into memory. If the operation is only a halfword write, the parity bit of the other halfword that is accessed is read out of memory, latched into the open-collector flip-flop of that line (11J3 for MS160, 11K6 for MD160) and rewritten into memory. These flip-flops function identically to those of the Restore Cycle Data Registers (see Section 13.3.)

In a Read operation, the parity bit for each halfword is read out of memory, latched into the open-collector flip-flops of MS160 and MD160 and rewritten into memory. As with parity generation, only the parity generator/checker integrated circuits used for the halfwords being requested are enabled with EPINH1 and/or OPINH1. RDOP8 (10H2) goes high, allowing the parity bit as well as the data bits to be checked. A parity error in one of the halfwords is indicated by EPERR0 (11H4) or OPERR0 (11G6). These are ORed to become PERR1 (11J4), which is then gated by either IRPEN1 (11J4) or DPEN1 (11J3) to become IRLMP0 (Instruction Read Parity Error) or DLMP0 (Data Read Parity Error) respectively. Both IRLMP0 and DLMP0 are 50 nanosecond pulses that occur approximately 400 nanoseconds from the leading edge of ER0.

13.7 Local Memory Module Capacity and Timing (32KB 750 nanoseconds, and 32 and 64KB 1 microsecond.)

In addition to the 32KB 750 nanosecond Local Memory Module (LMM) which the previous description is based on, the LMI can also accept 1 microsecond 32KB and 1 microsecond 64KB Local Memory Modules (LMMs). Use of these modules requires several changes in the previous description. They are explained in detail in this section.

Use of the 1 microsecond 32KB Local Memory Module (LMM) requires a different strapping arrangement on the LMI (shown on Assembly Drawing 35-534 E03) as well as timing pulses MCR0A1 (11R2) and MCR0B0 (9M8), which should nominally be 500 and 400 nanoseconds wide respectively. These pulses are adjusted with potentiometers located at 14R and 00R respectively on the LMI. The Difference in the previous descriptions are: 1. INH0 does not occur until 420 nanoseconds from the leading edge of ER0, and remains 300 nanoseconds wide. 2. LMARF1 stays high for an additional 125 nanoseconds, thus the MAR and control code register are valid for that additional time. This implies that all combinatorial outputs based on the control and address bits also remains valid. 3. LMBSYC0 remains low-active for 950 nanoseconds. 4. STCLKC1 (8L4) inhibits restarting of the clock until 1 microsecond after the beginning of the previous memory cycle.

The 1 microsecond 64KB Local Memory Module also requires a different strapping configuration and timing pulses MCR0A1 and MCR0B0. Pulse widths are reduced to 450 and 350 nanoseconds respectively. Omission of a Pulldown strap (8H4) adds stage QA1 (8K8) to the counter, modifying the counter sequence by adding an extra state. This is to accommodate the 64KB module's slower access time of 325 nanoseconds. The effects of these changes on the previous descriptions are: 1. The following signals are delayed 50 nanoseconds with no change in pulse widths – INH0 (8HZ), OBCLR0 (3L8), EBCLR0 (5H8), LMRDY0 – except the first pulse (9K2), RDEN0 (1A8), ULMAR0 (9F1), and RSR0 (9G1). 2. LMARF1 stays high an additional 125 nanoseconds, maintaining address, control code, and combinatorial outputs based on them. 3. LMBSYC0 remains low active for 950 nanoseconds. 4. STCLKC1 inhibits restarting of the clock until 1 microsecond after the beginning of the previous memory cycle.

14. MEMORY SYSTEM TEST FEATURES

14.1 Parity Error Source Indicator (LMI)

The Parity Error Source Indicator (Sheet 12) detects the occurrence of a parity error (DLMP0 or RLMP0) and sets a flag to indicate which of the 16 Memory Modules controlled by the LMI is the source of the data causing the parity failure. The occurrence of DLMP0 or IRLMP0 causes the flip-flop (12N1) to be set, lighting the LED (12R2). The control logic (Sheet 12) resolves the source of the parity error to the particular halfword (hence the particular Memory Module) from which the erroneous data was received. This causes EPSENO (12J6) or OPSENO (12J8) to go low, depending upon whether the error source was an even or odd halfword respectively. MPI (12J6) goes high if either DLMP0 or IRLMP0 are activated. When all three enable inputs on a 3 to 8 decoder (12K7) are activated, it decodes the three Most Significant Address Bits; ML001, MLX061, and MLX071 (12J6) and the corresponding PDXX0 (12L7) output goes low. This sets one of the 16 flag flip-flops (Sheet 12).

To find the source of an error, a Hexadecimal switch (12K4) is rotated so that its outputs select one input at a time to the two 8 to 1 multiplexors (12E3), which turns ON an LED (12J2) whenever the selected flag flip-flop is set. The Hexadecimal switch positions correspond to Memory Module locations as shown in Figure 22. The module that is the source of a parity error turns ON the LED (12J2) when the hexadecimal switch is at that module's position.

In a system using 64KB Memory Modules, a maximum of eight modules are connected to each LMI and the Hexadecimal switch positions correspond to Memory Modules as shown in Figure 23.

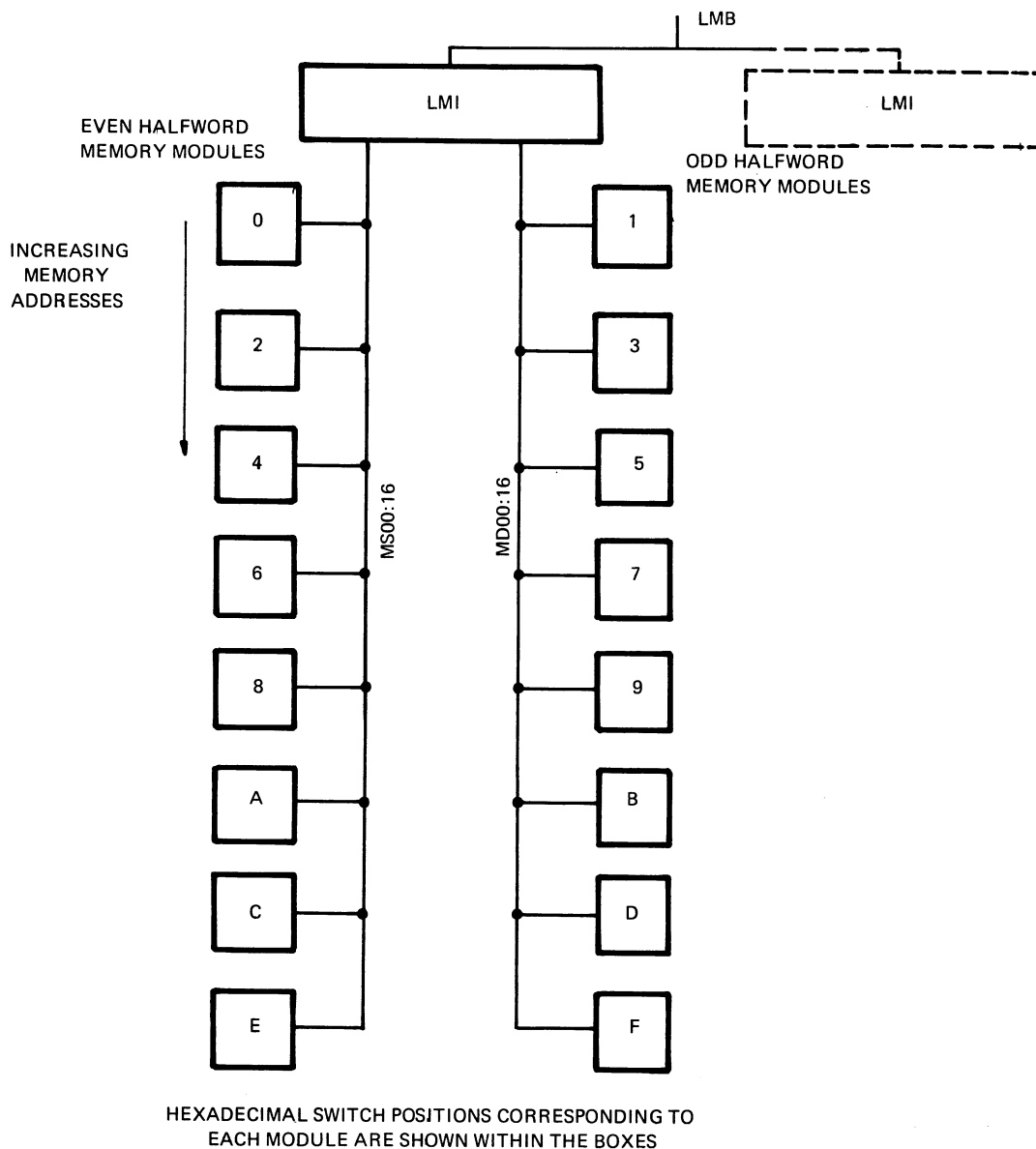
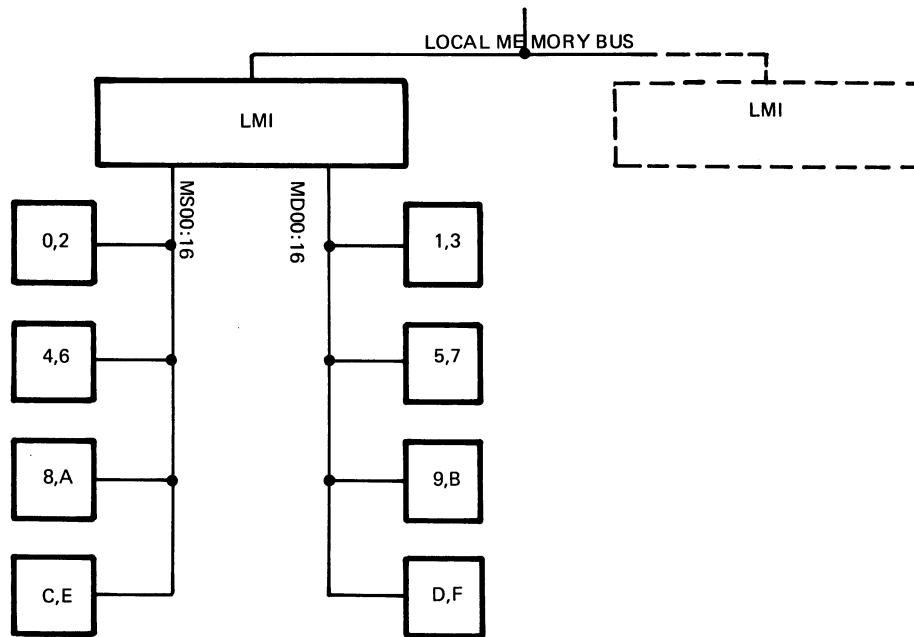


Figure 22. Parity Error Source Indicator – Hexadecimal Switch Positions for 32KB Modules



HEXADECIMAL SWITCH POSITIONS CORRESPONDING TO EACH MODULE ARE SHOWN WITHIN THE BOXES.

Figure 23. Parity Error Source Indicator – Hexadecimal Switch Positions for 64KB Memory Modules

14.2 Force Parity Error Strap (LMI)

A strap (11F5) is provided on the LMI which, when installed, causes the LMI to generate a parity error on each memory read cycle. The type of access (instruction or data read) determines whether IRLMPO or DLMPO is activated.

MEMORY SYSTEM TEST PROGRAMS

The following test programs are recommended to check the Memory System operation:

- | | | |
|-----------|---------------------------------------|--|
| 1. 06-156 | Memory Test Program Parts 1, 2, and 3 | This program checks all available memory in various ways, using only the CPU port of the MBC. |
| 2. 06-159 | System Exerciser | This program checks the EDMA port of the MBC and operates it in contention with the CPU port. This program requires an ESELCH. |

16. TROUBLESHOOTING

16.1 Extender Board Operation

Use of the 28-015 Extender board allows convenient troubleshooting of the Model 8/32 Main Memory. It allows extended operation of the MBC or either LMI. To operate any board on the extender, the extender board must be plugged into the card file flot corresponding to that board. Only one board should be extended from the chassis at a time.

NOTE

In order to operate the MBC on an extender a 35-597 Extender Board Terminator must be plugged into the spare slot of the extender board.

16.2 Troubleshooting Checks

If trouble develops and the memory system is suspect, the following checks are recommended:

1. Check for proper seating of all boards and cable connectors.
2. With the Power switch ON, measure P5, P16, and N16 on all memory chassis.
3. If parity errors are the source of the problem, check the LMI Parity Error Source Indicators to determine if the problem is isolated within a particular Memory Module, i.e., possibly a defective module.
4. Check for correct data and control (handshaking) between the MBC and the LMI boards. This usually isolates the problem to either the MBC or the LMI boards and Memory Modules.
5. If the problem only occurs for one particular state of the fullword address bit, it is associated only with the corresponding LMI and its memory modules. If, in addition, the problem is also restricted to one state of the halfword bit, then it is associated with only one LMI and one bank of memory modules (see Section 3).

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

16.3 Timing Adjustment

A timing adjustment is performed on all LMI boards at the factory to set the speed of the control logic clock. If the clock driver IC or the delay line in the clock circuit is replaced on an LMI board, the clock should be readjusted via the strap at board location 14D as follows (see schematic location 8C3):

1. Place a strap between Pins 14D05 and 14D15.
2. Place Processor in any memory access loop that continually cycles that LMI.
3. Measure clock period signal CL0 on Sheet 8D5 at the second clock pulse or later (not the first pulse) and adjust strap until the period is 49.5 to 51.5 nanoseconds.

17. INSTALLATION

Mechanical installation of Model 8/32 memories is covered by the Installation Specification, 01-078A20. After the equipment is mounted and connected, the following checks should be performed:

1. Unplug all Main Memory System boards (MBC, LMIs, and LMMs)
2. Adjust the power supply voltages as follows:
 P5 to +5.0 Volts
 P15 to + 16.5 Volts
 N15 to - 16.5 Volts
3. Power down and plug in all boards
4. Power up again and check voltages; readjust if necessary
5. Run Memory Test 06-156.

18. MAIN MEMORY SYSTEM MNEMONICS

The following lists provide a brief description of each mnemonic found in the memory system. The source of each signal on functional schematic 35-535D08 (MBC) and 35-534D08 (LMI), is also provided.

18.1 8/32 Memory Bus Controller (MBC) Functional Schematic 35-535D08.

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
ADRFN1	Address cycle of EDMA transfer finished	12J9
ANS0	EDMA Bus Answer	5J2
ANSR1	Answer Received	5G1
ANSTB0	Answer signal to transmitters—B input	11N6
ANSWT1	Answer to EDMA bus transmitter	12J2
BRSTWRT1	Burst mode write decode of EDMA	11M6
BURSTRD1	Burst read decode of DMA control bits	12C1
C28EB0	CPA address Bit 28 equal to upper address in stack	8J8
CA120:300	Control Address lines from CPA	Sheet 6
CAF120:300	Control Address flip-flops address register zero output	Sheet 6
CAF121:151	True side of Control Address flip-flops	Sheet 6
CCYCOM0	C Selected and Cycle Complete	13C6
CD000:310	Control Data lines to/from CPA	Sheet 4
CDMAD1	Clock DMA address register	11G9
CEQL0	CPA address equal to lower address in stack	8K9

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
CEQU0	CPA address equal to upper address in stack	8R9
CF1/0	CPA select flip-flop in memory contention circuit	13F3
C1/0	CPA selected in memory contention circuit buffered	13K3
CILMRS0	CPA access to memory with instruction read and local memory request service	14G5
CLK1A	Clock to EDMA Bus control counter	10F1
CLK1B	Clock to counter B	10E6
CMC000:020	Control Memory control code from CPA	Sheet 11
CRDY0	C Ready signal to CPA	14S8
CREQ0	Control Request from CPA	6G6
CREQF 1/0	Control Request flip-flop	6A1
CRQD 001	CPA Request Delayed 00, 10, 20, etc. nanoseconds	Sheet 14
101		
201		
301		
501		
601		
CTAA1	Counter A Stage A	10J2
CTAB1	Counter A Stage B	10N2
CTAC1	Counter A Stage C	10L2
CTBA1/0	Counter B Stage A	10E6
CTB000	Counter B not at count time zero	10C8
CTBB1/0	Counter B Stage B	10F6
CTDA1/0	Counter D Stage A	12G6
CTAD1/0	Counter A Stage D	10J1
CTDA1-LMSEL	Counter D Stage A and local memory selected	12H6
CTEA1/0	Counter E Stage A	12J3
CTEB1/0	Counter E Stage B	12L4
CTFA1	Counter F Stage A	11E4
CTFB1/0	Counter F Stage B	11G4
CTFC1/0	Counter F Stage C	11B4
CTFCLK	Counter F Clock	11C1
CTMOUT0	CPA Request Time out	14N2
CXREQ0	CPA Request to EDMA	9N5
CYCOMB0	Memory Cycle complete buffered	11G9
C28EA0	CPA address Bit 28 equal to lower address in stack Bit 28	8J9
D28EA0	DMA address Bit 28 equal to lower address in stack Bit 28	8E8
D28EB0	DMA address Bit 28 equal to upper address in stack Bit 28	8E9
D35	Memory contention request delayed 35 nanoseconds	13G6
DA000	Data strobe delay line output 00 nanoseconds	11K8
DA300	Data strobe delay line output 30 nanoseconds	11K8

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
DA700	Data strobe delay line output 70 nanoseconds	11L8
DA400	Data strobe delay line output 40 nanoseconds	11L8
DAD121:301	DMA Address register counter	Sheet 1
DATTM1	Data Time on LMB	13M9
DB700	Delay line 70 nanosecond tap	12J1
DEQ0	DMA address equal to address in stack and DMA writing to memory	14H1
DEQL0	DMA address equal to lower address in stack	8C9
DEQU0	DMA address equal to upper address in stack	8G9
DLMPF0	Data Local Memory Parity Fail	14A3
DMAACT1/0	DMA Active	10J4
DMAFW0	Fullword access from CPA to EDMA	2M9
DMA160:170	EDMA Bus	1M9
DMARDY A/B	DMA Ready—creates CRDY	11M5
DMARY0	DMA RDY to CPA	14K8
DMCA0	DMA control address time	10C7
DMENB0	Enable DMA transmitters	10E9
DMF001:311	DMA Data Register flip-flops	Sheet 1
DMMC000:020	DMA Memory Control code	1K9
DMPF0	Data Memory Parity Fail	14C4
DMT001:171	Signals to DMA Transmitter	Sheet 2
DMX120:150	EDMA Bus lines	5L2
DPF1	Data Parity Fail	14C3
DREQ1/0	CPA request to EDMA Bus	9L4
DXR121:151	DMX signal out of receiver	Sheet 5
EFB0	EDMA flip-flop in memory contention circuit-buffered	13K2
EF1/0	EDMA selected flip-flop in memory contention circuit	13G1
EHOLD0	EDMA Hold memory access	11N6
ENB0	Enable memory busy on EDMA Bus	10N8
END0	End selected condition	10N9
ENEOT1	Enable End of Transmit	10F8
EOT0	End Of Transmit	12F4
EOT0	End Of Transmit	10G9
EQ1	CPA address equal to an address in stack	14E6
EQIRC0	Equal and instruction read and CPA not still accessing memory	14F6
EREQE1	EDMA request to memory contention circuit	13E1
ERQWRT0	EDMA request and write command	13C1
GTUU1	Greater Than Unused memory lower limit	9E2
GXX1	Combination of ready signals	14K2
IR 1/0	Instruction read decode of CMC bits	11H7
IRWRT0	Instruction read or data write—ready	14K8

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
LDADRO	Load from EDMA Bus receiver	12E7
LDDMC1	Load DMA control bit register	12E7
LDLMF1	Load Local Memory flip flops—Data Register	11G9
LDLSH0	Load Least Significant Half of DMA Data Register	12K8
LDLSH1	Load Least Significant Half of DMA Data Register	11N3
LDMSH0	Load Most Significant Half of DMA Data Register	12J8
LDMSH1	Load Most Significant Half of DMA Data Register	11N4
LDSTA1	Load Stack address register STA	14K7
LDSTK0	Load Stack	11C1
LMB271	Local Memory Bus Bit-27	11J7
LMB000:316	Local Memory Bus signals to Local Memory Interface	Sheet 3
LMBBY1/0	Local Memory Bus Busy	13G8
LMDS0	Local Memory Data Strobe	11N5
LMF000:310	Local Memory Bus flip-flops Data Register	Sheet 2
LMRDY0	Local Memory Ready	11A7
LMREQ0	Local Memory Requested	10C4
LMRS0	Local Memory Request Service	13N8
LMSEL1/0	Local Memory Selected	12D5
LMTMOT0	Local Memory Time Out	14S4
LOAD0	EDMA Bus Load	5J2
LOADR1	Load Received	5G1
LOADT0	Load signal to EDMA transmitter	10F7
M00	Same as memory zero except local memory zero active	9G8
M01	Memory Zero, Local Memory One active	9G7
M1BZR0	Memory 1 Busy from EDMA Bus receivers	9J4
M2BZR0	Memory 2 Busy from EDMA Bus receivers	9J4
M3BZR0	Memory 3 Busy from EDMA Bus receivers	9J4
MB00	Same as memory zero except buffered zero active	9G7
MSTP0A:3A	Memory Address strapping upper address of each memory most significant bit	Sheet 9
MSTP0B:3B	Same as MSTP0A:3A except next MSB	Sheet 9
MSTP0C:3C	Same as MSTP0B:3B except next MSB	Sheet 9
MSTP0D:3D	Same as MSTP0C:3C except next MSB	Sheet 9
P5RB	P5 through Resistor B	1N7
P5RX	P5 through Resistor X	8H6
P5RY	P5 through Resistor Y	9A1
P5RZ	P5 through Resistor Z	12B3
QUE0	Queue	10N3
RDBUFL1/0	Read Buffer Full	12D2
RDDAT0	Read Data	11H5
RDYDAT	EDMA Ready for Data	12D1

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
RPC1	Receive Priority Chain	10N1
RSREQ0	Reset stack request signal	13B6
S1	Stack selected in memory contention circuit	13M5
SBSY0	Stack Busy on LMB	14C1
SCLRBO	System Clear buffered	10H9
SD1	Stack select delayed	13M5
SEL1/0	Selected (CPA)	10N6
SLMSELO	Set Local Memory Select flip flop	12D5
SOTO	Start Of Transmit	10M4
SPECIR1	Special instruction read stored signal	14K9
SMX0	Stack multiplexor control	13J4
SS1	Stack Selected	13M5
SSREQA0	Set Stack Request	14N6
ST161:311	Stack output data	Sheet 5
STA121:281	Stack Address register bottom of stack	Sheet 7
STA280	Stack Address register LSB	7N2
STB121:271	Stack Address - middle of stack	Sheet 7
STCDO	Stack to CD lines	11C5
STHOLD	Start of Transmit Hold	10K3
STKA1/0	Valid status of half of stack that contains upper address	14M6
STKRDY1	Stack Request Ready to CPA	14J8
STWA1	Stack Write Select A	14E2
STWB1	Stack Write Select B	14C1
SQUEF1	Set Queue flip-flop	9N5
SXIR	Stack multiplexor control or Instruction read	11D6
T00A	Time Zero from delay line	10A3
T80A	Time 80 nanoseconds delayed from delay line input	10B2
WDTM0	Write Data Time	11M7
WRTBUF1/0	Write Buffer Full	12L8
XREQ0	EDMA Bus request	10A1
XX1/0	Ready to be started	10L6

18.2 8/32 Local Memory Interface (LMI) Functional Schematic 35-534D08

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
ADV0	Output of comparators (Sheet 9) indicating that requested address is within block of memory controlled by this LMI.	9N3
C001:021	Buffered CMC bits from CPA which are part of the control code transmitted from the MBC to LMI boards indicating CPU requests to local memory.	Sheet 10
CL0	Clock Output—pulse train drives Johnson Counter on Sheet 11.	8C6
CL1	Complement of CL0, used to time RSR0 pulse.	8D5
CM0	Cycle Memory—enables the clock and is held low throughout the memory cycle.	8N1
DLMP0	This low-active pulse is generated when an LMI detects a parity error on any operation except instruction read.	11N8
EBCLRO	Even Buffer Clear—is normally low and is asserted high during a Read operation to latch data into the Restore Cycle Data Registers for the even (most significant) halfword.	5H8
EPERRO	Asserted low-active when a parity error is detected on a Read operation from the even halfword memory bank of an LMI.	11G4
EPINH1	High active to inhibit parity generator/checking logic for even halfword memory bank of an LMI.	11D1
ER0	Early Read control signal for local Memory Modules begin the memory cycle read phase.	8F9
EWRT0	Even Halfword Write—is asserted low-active when it is desired to write into an even-address halfword.	9G6
EX1/0	Bit 4 of MBC-LMI control code which indicates an EDMA request for local memory.	10C7
FW1/0	MAR output bit which selects the LMI controlling requested address.	10E8
FWD1/0	Strap-selected level function which is low-active when this LMI is selected by FW1/0.	10G8
HEX011:041	The output of a Hexadecimal rotary switch which select a particular PFxx1 line for display on an LED.	Sheet 12
HW1/0	MAR output bit which is the least significant address bit and is used to select one of the two halfword memory banks controlled by an LMI on halfword memory operations.	10H6
ID021	An intermediate combinatorial decode used in various functions.	10H3
ID030	An intermediate combinatorial decode used in various functions.	10L4
ID041	An intermediate logical decode signal used in various functions.	9F2
INH0	Inhibit control signal of Local Memory Modules begins the restore/write phase of a memory cycle.	8H2

This information is proprietary and is supplied by INTERDATA for the sole purpose of using and maintaining INTERDATA supplied equipment and shall not be used for any other purpose unless specifically authorized in writing.

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
INSRD1	Combinatorial decode indicates access is an Instruction Read.	9C2
IRLMP0	This low-active pulse is generated when an LMI detects a parity error during an instruction Read operation.	11M4
LMARF1/0	Level signal which changes mode of memory address register and control code register between tracking and latched.	10C1
LMB000:310	32 Bit wide bidirectional Local Memory Bus used to transmit data and addresses between the MBC and LMI boards.	Sheets 1 and 2
LMBSY A0 B0 C0 }	Three control lines used to synchronize LMI cycles for an Instruction Read (double fullword read).	9G8
LMRDY0	Ready signal from LMI to MBC indicates that LMI has responded to request and is ready to accept write data, or that read data is valid on the LMB.	9K2
LMDS0	Local Memory Data Strobe-loads data transmitted from the MBC into the MDR.	
LMRS0	Request line from MBC—indicates that MBC is requesting the memory to cycle.	8D1
MA000:140	Memory Address bits—concatenated with MAX060 and MAX070 to form the address sent from the LMI to the memory modules.	Sheets 6 and 7
MAX060 and 070	Extended Memory Address bits—see MA000:14.	
MCLRO	Memory Clear—buffered System Clear which is sent to all Local Memory Modules.	8E7
MCROA1	Timing pulse used to control signals only when 1 microsecond cycle time local Memory Modules are used.	11N1
MCROB0	Timing pulse used to control signals only when 1 microsecond cycle time local Memory Modules are used.	9M8
MD000:160	Memory Data—is the bidirectional data bus which transmits data between the LMI and the least significant halfword (odd) bank of Local Memory Modules (LMMs).	Sheet 3
MDR001:311	Output of the Memory Data Register which store data for memory write operations only.	Sheets 3 and 4
ML000:010	Latched Memory Address bits—output of MAR.	6C7
MLX060 and 070	Latched Extended Memory Address bits. Output of the tracking latches used for the MAR.	Sheet 8
MP1	Memory Parity Fail—high active when either DLMPO or IRLMPO are asserted.	12LI
MS000:160	Memory Sense—is the bidirectional data bus which transmits data between the LMI and the most significant halfword (even) bank of memory modules.	Sheets 4 and 5
MSTRP0	Timing pulse used to control signals only when 1 microsecond cycle time Local Memory Modules are used.	9M9
NULMAR0	Used to unload memory address register and control code register at the end of the cycle for 750 nanoseconds cycle time memories.	9F1

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
OBCLR0	Odd Buffer Clear—is normally low and is asserted high during a read operation to latch data into the Restore Cycle Data Registers for the odd (least significant) halfword.	3L8
ONE0	Decode of the first state of the counter (Sheet 8) used to initiate various memory cycle signals.	9F3
OPERR0	Asserted low-active when a parity error is detected on a Read operation from the odd halfword memory bank of an LMI.	11G6
OPINH1	High active to inhibit parity generation/checking logic for odd halfword memory bank of an LMI.	11D5
OWRTO	Odd Halfword Write—is asserted low-active when it is desired to write into an odd-address halfword.	9G5
PD000:150	Each of these decoder output lines correspond to one of the up to 16 local Memory Modules than an LMI can control. A line is asserted low-active when a parity error is detected on a Read operation to its corresponding memory module.	Sheet 12
PF001:151	The 16 outputs of the Parity Error Storage register. A line is set high active when a parity error is received from the Local Memory Module that particular flip flop represents.	Sheet 12
Q010:080	Output of Counter flip-flops stages 1 thru 8 (flip-flop Q outputs).	Sheet 8
Q011:081	Same as above but complemented (flip-flop Q outputs).	Sheet 8
QA1/0	Output of an additional (ninth) stage of the Johnson Counter, used only when memory modules are 64 KB, 1 microsecond cycle time type to accommodate longer access time of this module.	8L8
RDENO	Read Enable—goes low to activate the multiplexors/drivers of the LMB on Read operations.	1A8
RDOP1	This signal is asserted high-active on any type of Read operation.	10H2
RDSEL	Selects the correct data for placement on LMB16:31 at LMB2:1 multiplexors/driver (Sheet 2). RDSEL is high for an even halfword read and low for an odd halfword or fullword read.	10K6
RF0	Complement of RF1, Captures LMRS0 in tracking latch.	8C2
RSR0	Initializes the LMI at the end of a memory cycle.	9G1
SCLR0	System Clear	8A8
SSFBO	Counter Self-Start Feedback—insures counter does not remain in an undesired stable sequence of statues.	8R5
SX1/0	Bit 5 of MBC-LMI control code which indicates an Instruction Read initiated by the look-ahead cache on MBC.	10C5
ULMAR0	Pulse to Unlatch Memory Address register and control code register at the end of memory cycle.	9R8
WTE001	This data steering signal is asserted high-active for a halfword Write operation into the most significant (even) halfword.	10L2

<u>MNEMONIC</u>	<u>MEANING</u>	<u>SCHEMATIC LOCATION</u>
WTE011	This data steering signal is asserted high-active for a fullword Write operation.	10L4
WTE021	This data steering signal is high-active for a halfword write into the least significant (odd) halfword or for a fullword write.	10L3
WTE031	This signal is asserted high-active on all Write operations to enable the appropriate parity bit(s) to be written into memory.	11H1
XRPC	Pullup Resistor C	8N5
XRPD	Pullup Resistor D	12K3
XRPG	Pullup Resistor G	9E4

BACK PANEL MAP

Main wiring map table with columns for TITLE, BD.LOC., TERM. NO., and various component labels (CHB, DCS, CPU, ALU, IOU, I/O) across rows 1-41.

NOTES: 1. BACK PANEL - 35-541 WIRE WRAP RUN

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT...

LEASED FOR PRODUCTION

PRE PRODUCTION APPROVAL, INIT. DATE, DEV. DATE, PROD. DATE

NAME: G. MELTON, G.J. HOMEFIELD, T. FIFE, S. MESSINA; TITLE: DRAFT, ENGR, DIR ENG; DATE: 6-10-74, 10-10-74

TITLE: 8/32 BACKPANEL LOWER; SHEET OF: 1-4; PART NO: 03053; DWG NO: 01-078R03D08

BACK PANEL MAP

Main table with columns for LMM, LMI, and MBC components, organized by row and column. Includes component titles, board locations, and terminal numbers.

NOTES
1. BACKPANEL - 35-540
2. SLOT ASSIGNMENTS SHOWN FOR THE BASIC 128 KB MEMORY SYSTEM.

THE LMI BOARDS ARE RE-ASSIGNED WHEN THE SYSTEM IS EXPANDED WITH CORRESPONDING CHANGES IN MA, MD, & MS STRAPPING OPTIONS.

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

Administrative form with fields for NAME, TITLE, DATE, and SHEET OF. Includes names like G. MELTON and T. EYFE.

BACK PANEL MAP

Table with columns for CON N., TITLE, LMM, and ROW. It contains a grid of component identifiers (e.g., PS, GND, MD150, MS010, LMB300) organized into four main sections (07, 06, 05, 04) and four sub-sections (03, 02, 01, 00). Each section has two rows and two columns of components.

NOTES
1. BACK PANEL - 35-542

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

NAME: G. MELTON, G.J. HOMEFIELD, T. FYFE, S. MESSINA
TITLE: DRAFT, CHK, ENGR, DIR ENR
DATE: 8/32
TITLE: BACK PANEL MEMORY EXP.
NO: 03053
SHEET OF: 3-4



REVISIONS

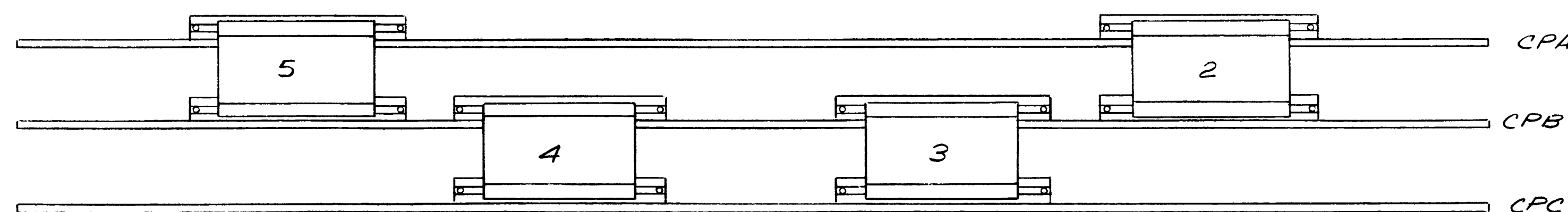
ADDED: "MS160" ON SHT. 2 TO CPA, PIN 240-L.
 3/4 (1) 03088 23 3-24-75 R01
 ADDED: SHT. 1 "PBPCC" MNEMONIC TO PIN 128-0. P228-0 SLOTS 00 THRU 02. SHT. 2 "LMB5Y0" TO PIN 139-0 SLOTS 02 THRU 07. "LMB5Y0" TO PIN 239-0 SLOTS 02 THRU 07.
 CHANGED: PIN 137-0 SLOT 01 WAS "LMB5Y0", PIN 137-0 SLOTS 02 THRU 07 WERE LMB5Y0.
 3/4 (1) 03088 23 3-24-75 R02
 ADDED: SHEET 4.
 3/4 (1) 03088 52 3-24-75 R03

CONN 5	
2	1
16 UIR300	UIR310
15 UIR280	UIR290
14 UIR260	UIR270
13 UIR240	UIR250
12 YD300	YD310
11 YD280	YD290
10 YS300	YS310
09 YS280	YS290
08 SX300	SX310
07 SX280	SX290
06 GND	ININTO
05 GND	MAIO
04 GND	IREQO
03 GND	DEEQO
02 GND	PSW210
01 GND	GND
00 GND	PASSIA

CONN 3	
2	1
24 GND	GND
23 CSAD141	GND
22 CSAD121	CSAD151
21 GND	CSAD131
20 CSAD101	CSAD111
19 CSAD081	GND
18 CSAD061	CSAD091
17 GND	CSAD071
16 CSAD041	CSAD051
15 GND	GND
14 GND	GND
13 GND	SPARE
12 GND	PCLKO
11 PSW251	SECLK1
10 PSW260	PSW270
09 GND	S2B0
08 ASELO31	ASELO41
07 ASELO11	ASELO21
06 GND	ASELO01
05 BSELO31	BSELO41
04 BSELO11	BSELO21
03 GND	BSELO01
02 SSELO31	SSELO41
01 SSELO11	SSELO21
00 GND	SSELO01

CONN 4	
2	1
24 CSD301	CSD311
23 CSD281	CSD291
22 GND	CSD271
21 CSD261	CSD251
20 CSD241	GND
19 CSD221	CSD231
18 GND	CSD211
17 CSD201	CSD191
16 CSD181	GND
15 CSD161	CSD171
14 GND	CSD151
13 CSD141	CSD131
12 CSD121	GND
11 CSD101	CSD111
10 GND	CSD091
09 CSD081	CSD071
08 CSD061	GND
07 CSD041	CSD051
06 GND	CSD031
05 CSD021	CSD011
04 CSD001	GND
03 GND	CSWPO
02 GND	INCLKO
01 GND	CLKO
00 GND	GND

CONN 2	
2	1
16 MCO20	MCO30
15 MCO00	MCO10
14 GND	S2B0
13 GND	PCLKO
12 GND	LUTY1
11 GND	RX3D0
10 GND	SPARE
09 GND	CS000
08 GND	SRT1
07 GND	GND
06 GND	GND
05 BSELO31	BSELO41
04 BSELO11	BSELO21
03 GND	BSELO01
02 SSELO31	SSELO41
01 SSELO11	SSELO21
00 GND	SSELO01

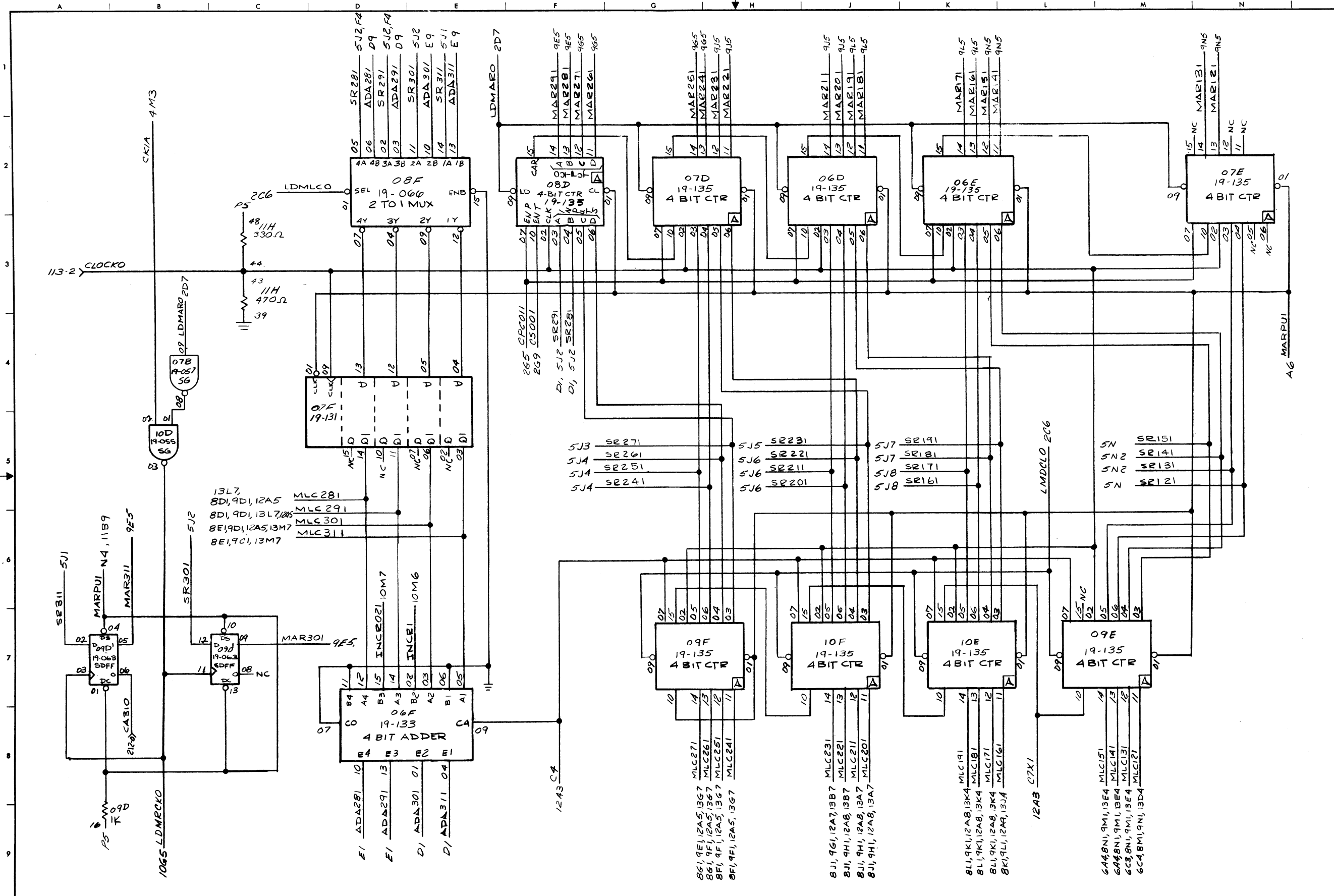


THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

SCALE-	NAME	TITLE	DATE	TITLE
TOLERANCE: XXX ± .005 XX ± .02 X ± .03 ANGLES ± 1° UNLESS OTHERWISE SPECIFIED	G. MELTON G. J. HOMEFIELD H. NAWER R. BARBER S. MESSINA	DRAFT CHK ENGR QA MGR	2-14-75 2-18-75	CPA/CPB, CPB/CPC FRONT END CABLING
				TASK 03088 SHEET OF 4-4

BRUNING 44-231 24538

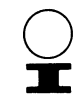
REVISIONS
 DELETED: 19-117 @ 08E, 19-057 @ 07B, 19-055 @ 10D
 ADDED: MNEMONICS 'LDMRCKO' @ B9, 'C5001' @ FA, 'C7K1' @ LB.
 2-3-75 RO1
 AREA B2: DELETED GATE 07B (19-057) AND RELATED CIRCUITRY WHICH SHOWED; 07B 12 TO 07F09 CK0B, 07B13 TO 10D02, MNEMONIC CKIA, MNEMONIC CLOCKS R3300 & R7012 RESISTORS WERE NOT SPEC'D. AREA F8: 09F 10 WAS TO 06F09, MNEMONIC C4 WAS NOT SPEC'D.
 WS 12-16-75 RO2



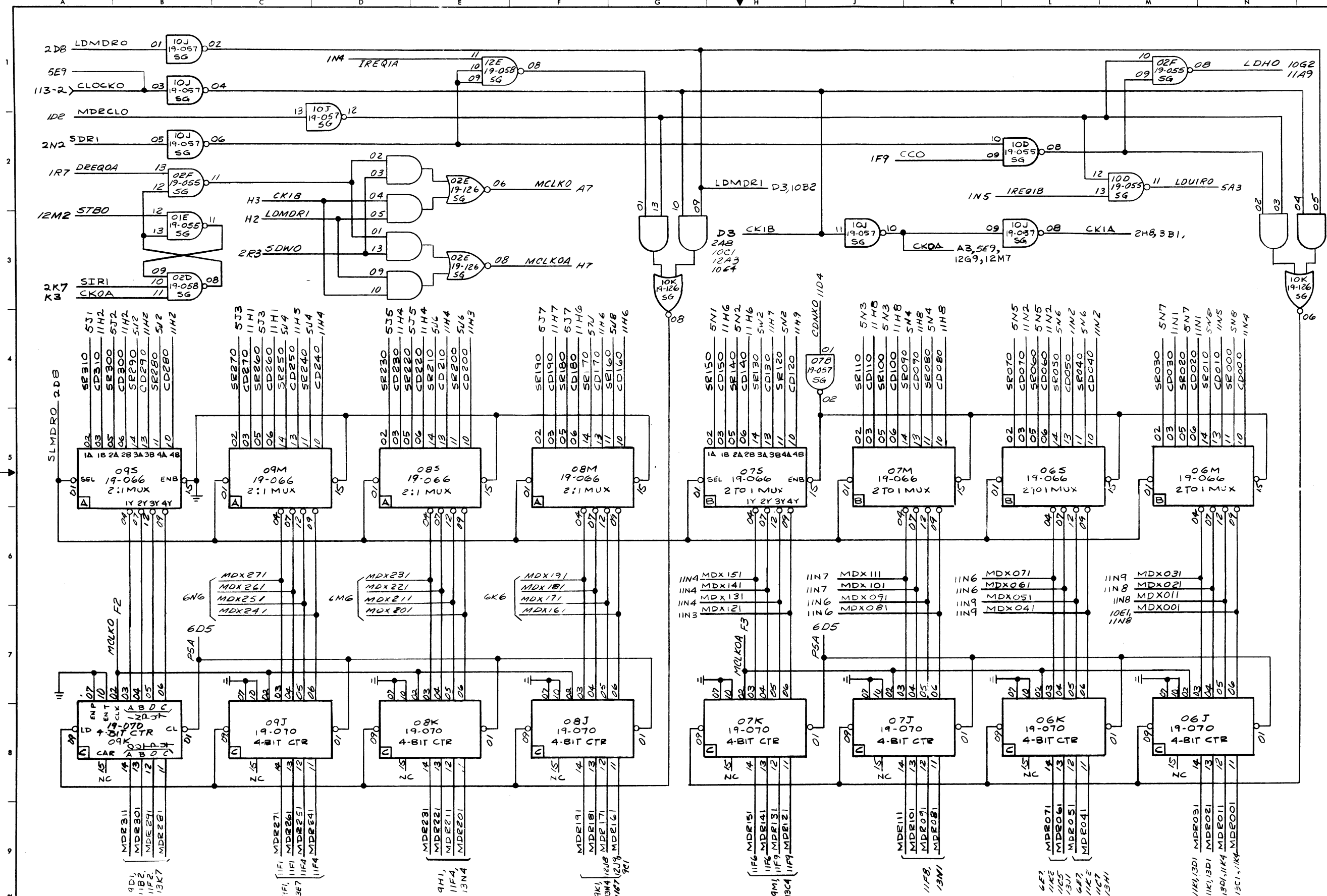
NOTES

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
G. MELTON	DRAFT		8132 CPA
	CHK		
	ENGR		
	DIR ENG		
TASK NO. 03088		SHEET OF 3-13	
DWG NO. 35-536 R02D08			



BRUNING 44-231 10042



REVISIONS	
ADDED: 2-19-55 @ 02F-547 LOC B2 & M1.	
CHANGED: 'MCLKO & MCLKOA WERE CONNECTED.	
9/4/55 13058-31 2-3-75 R01	
ADDED: PSA MNEMONIC AT LOC B7 & M1.	
9/4/55 13088-48 2-3-75 R02	
ADDED 2DB TO LDMRO 2AB TO CKIB	
SLMDRO WAS LDMRO	
9/4/55 13088-55 2-3-75 R03	
10C1E 12A3 WENT TO 10J0B	
9/4/55 13088-72 2-3-75 R04	
ADDED 10E4 TO CKIB	
9/4/55 13088-2521 5-28-75 R05	
AREA M3, CKIA WAS CROSS-REFERENCED TO 3B2	
US 12698 - 12-16-75 R06	

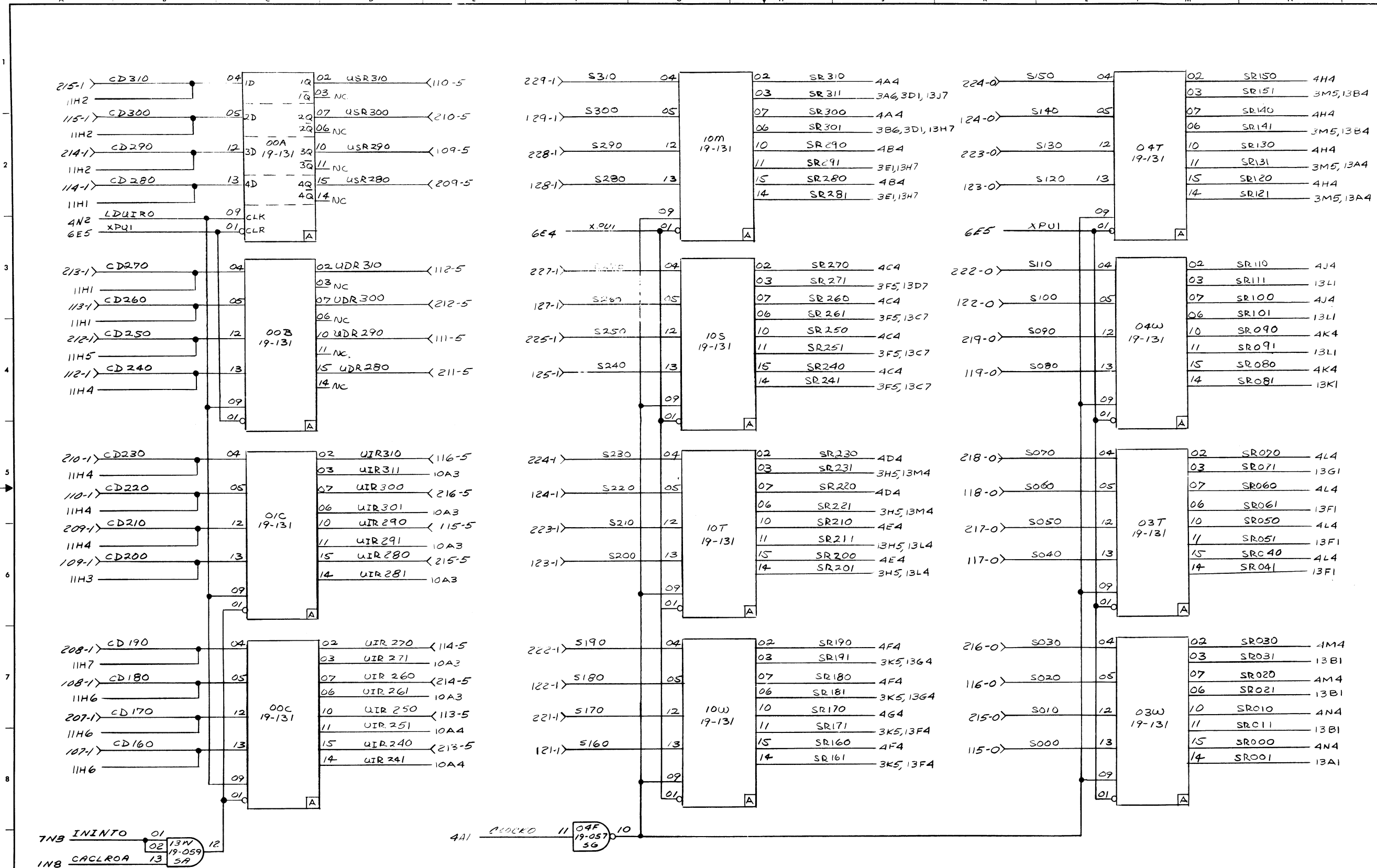
NOTES

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
G. MELTON	DRAFT		8/32 CPA
	CHK		
	ENGR		
	DIR ENG		

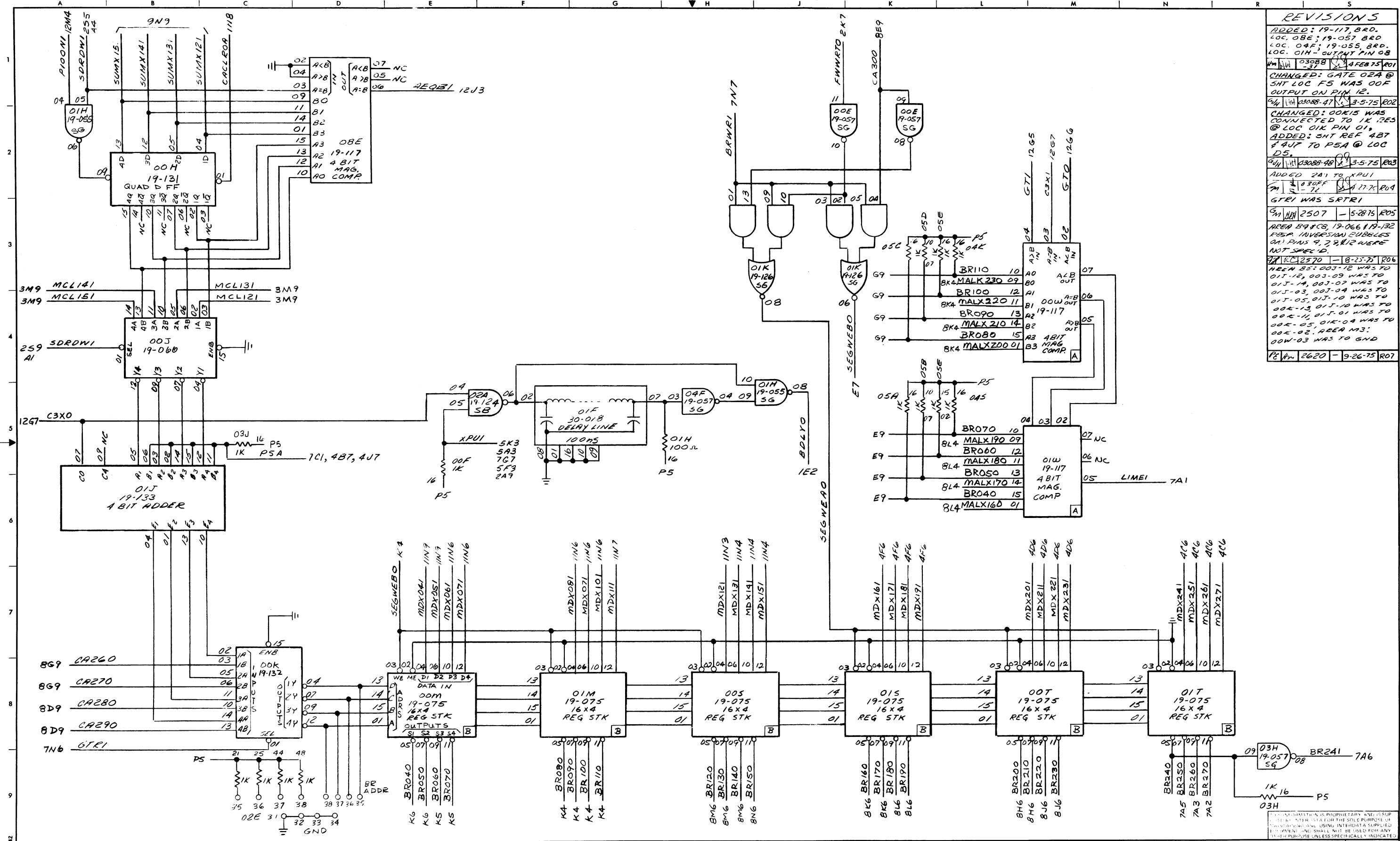
THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

BRUNING 44-231 16042

REVISIONS	
AREA F9: OOD-03 WAS IR271. OOD-02/05 WAS SIR1. OOD-04 WAS IR281. OOD-06 WAS TO 13W-01902.	
PE	03088-57 2-4-75 ED1
CHANGED: 04F11 WAS C'KOA' MNEMONIC.	
REV	03088-57 2-4-75 RO2



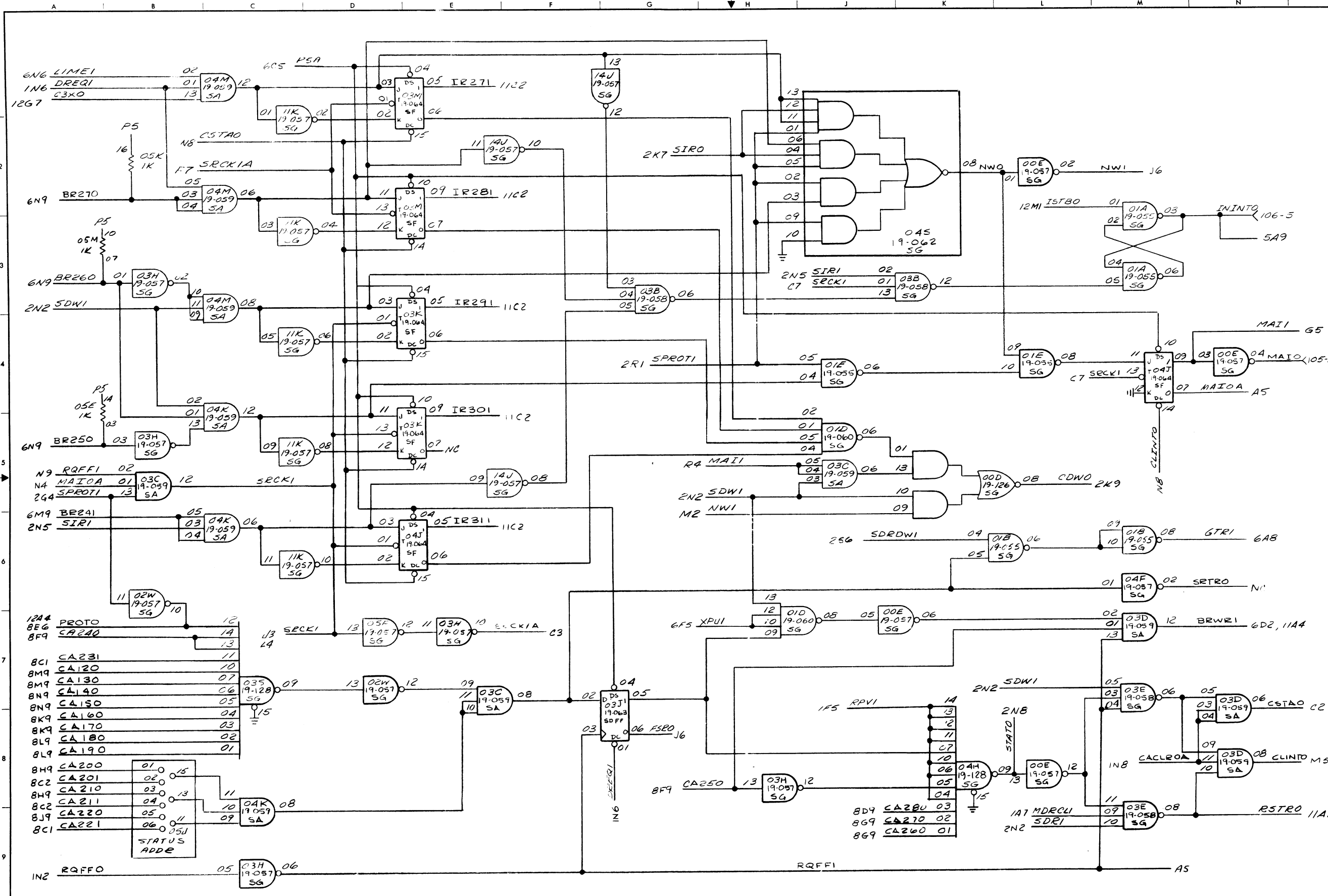
BRUNING 44-231 16042



REVISIONS	
ADDED:	19-117, BR0.
	LOC. 08E; 19-057 BR0
	LOC. 04F; 19-055, BR0.
	LOC. 01H - OUTPUT PIN 08
CHANGED:	GATE 02A @
	SHT LOC F5 WAS 00F
	OUTPUT ON PIN 12.
ADDED:	00E 19-057 SG
CHANGED:	00K15 WAS
	CONNECTED TO 1K.PES
	@ LOC 01K PIN 01.
	ADDED: SHT REF 4B7
	F AUT TO PSA @ LOC
	D5.
ADDED:	2A1 TO XPUI
AREA:	594C8, 19-066, 19-132
	PESA INVERSION BUBBLES
	ON PINS 9, 2, R12 WERE
	NOT SPEC'D.
AREA:	55: 003-12 WAS TO
	01J-18, 003-09 WAS TO
	01J-14, 003-07 WAS TO
	01J-03, 003-04 WAS TO
	01J-05, 01J-10 WAS TO
	00K-13, 01J-10 WAS TO
	00K-14, 01J-01 WAS TO
	00K-05, 01K-04 WAS TO
	00K-02, AREA M3:
	00W-03 WAS TO GND
PIN:	2620 - 9-26-75 R07

NOTES			
NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
	DRAFT		8/32 CPA
	CHK		
	ENGR		
TASK NO.	03088	SHEET OF	6-13
DIR ENG	35-536 R07 008		

BRUNING 44-231 1604Z

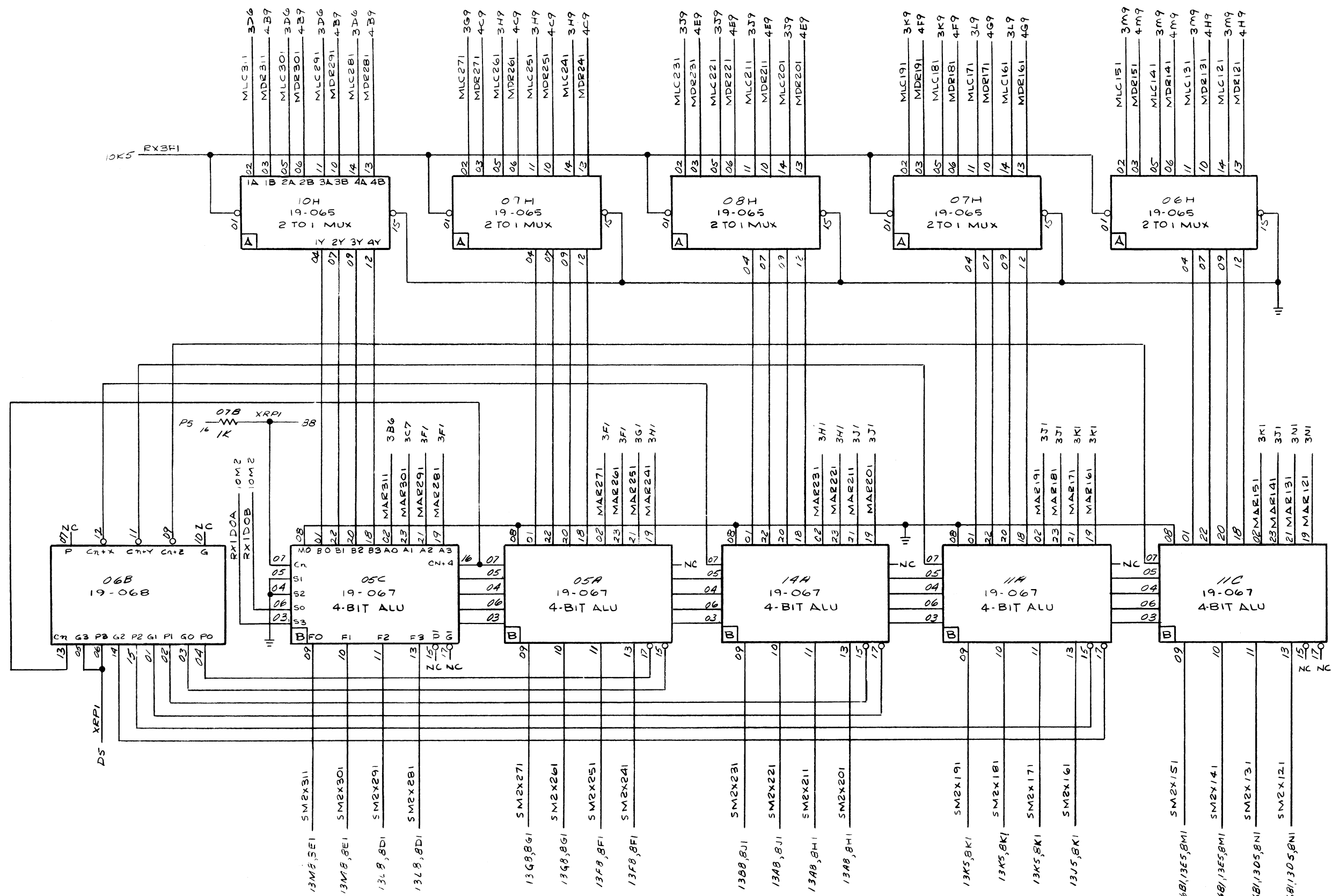


REVISIONS			
CHANGES EXTENSIVE, SEE ECR. (1)			
9/4	03088-31	2.4.75	RO1
ADDED IC'S IN AREA D7			
E67, 03M13 WAS GOING TO			
03K01, 03M01 WENT TO			
SPEAFI.			
9/4	03088-48	3.6.75	RO3
ADDED: 19-057 @ 11K			
SNT LOC CE, THRU C9.			
CHANGED: 09M13 WAS			
COUNT TO 03M06; 09M09			
WAS COUNT TO 03M07;			
09M10 WAS COUNT TO 03M06.			
9/4	03088-55	4.8.75	RO4
ADDED 2GATE'S 01B (L, ME)			
09M11 WAS SUTRI TO 6A6			
9/4	2507	5.28.75	RO5
RELM B1: 09M-13 WAS TO			
PE	0620	9.26.75	RO6

NOTES

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
G. MELTON	DRAFT		B/32 CPH
	CHK		
	ENGR		
	DIR ENG		
TASK NO	03088	SHEET OF	7-13
DWG NO	35-536 R06008		

BRUNING 44-231 16042



BRUNING 44-231 1604Z

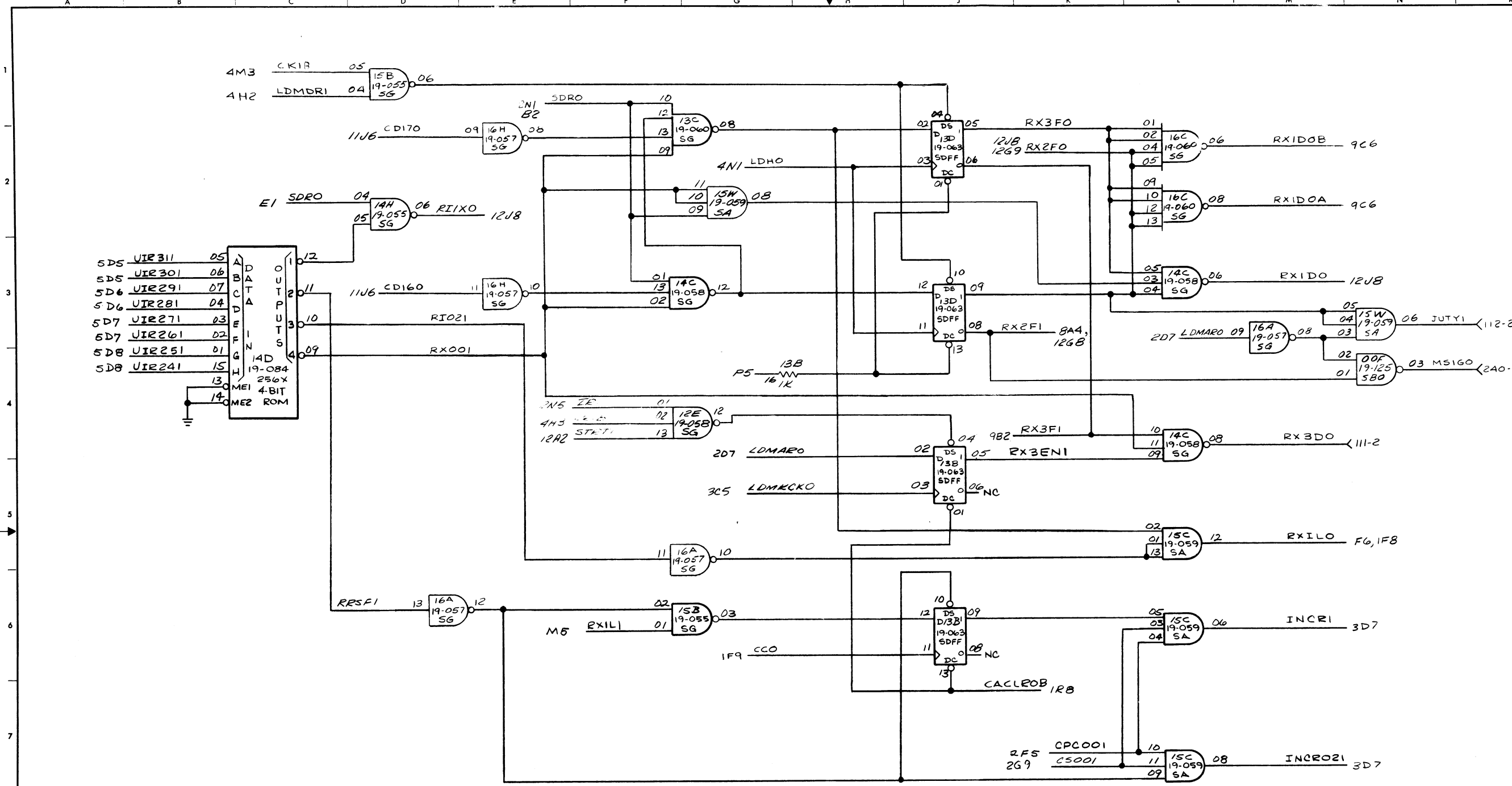
NOTES

NAME	TITLE	DATE	TITLE
G MELTON	DRAFT	7-2-74	FUNCTIONAL SCHEMATIC
CHK	ENGR		8/30 CPA
DIR ENG			
TASK NO. 03088		SHEET OF 9-13	
REV. NO. 35-536 D08			

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA ON THE SOLE PURPOSE OF THE CONTRACT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

REVISIONS

AREA B1: REMOVED
 1REQ A TO 13A-0110.
 CKIA 13 TO 13A-13
 LDMDR1 WAS TO 13A-01
 13D-04 WAS TO 13A-
 08. AREA 15: 15C-10
 14C-01 WERE N.C.
 AREA 11: 13C-13 WAS
 MEXCH. AREA F5:
 14C-13 WAS MDX01.
 AREA C3: 14D-12 WAS
 N.C. AREA H3: 13D-03 &
 11 WAS TO DD1. AREA
 J5: 13B-03 WAS TO
 LDMAR0. 13B-02 WAS
 TO 16A-04. 16A-03 WAS
 TO CKIA. AREA J6:
 13B-10 WAS TO 13B-01,
 13. AREA L7: 15C-11 &
 03 WERE TO INHMLCO
 PIN 107.2 AREA J6:
 13B-11 WAS TO 15B-06.
 15B-04 WAS TO IRI.
 15B-05 WAS TO DREQI.
 AREA L2: 16C-05 & 13 &
 14C-03 WERE RX001.
 16A-09 WAS BMXSLBI
 AREA N4: 15W-06 WAS
 MSIGO PIN 240-1. 00F-
 03 WAS JUTY1 PIN 112-2
 CHANGED: 16A09 WAS
 'BMXNBI' MNEMONIC.
 CHANGED: GATE @ 00F
 SAT LOC N4 WAS 13-05G.
 15B05 WAS CKIA
 16H WAS REPLACED BY
 14H IN LOC D2
 ADDED GATE 10D TO
 LOC 64; 13B04 WAS
 DSTE0 TO 12M2
 AT LOC 65 GATE 12D
 WAS 10D; 12E12 WAS 10D06;
 12E02 WAS 10D05; 10D04
 SIRIA; 12E01 & 12E15
 WAS NOT SPEC.
 AREA G2, 15W-09 & 15W-
 11 WERE REVERSED.



NOTES

NAME	TITLE	DATE	TITLE FUNCT.
G. MELTON	DRAFT	6-28-74	8/32 CPA
	CHK		
	ENGR		
	DIR ENG		

TASK NO. 03088	SHEET OF 10-13
FILE NO. 35-536 E09D08	

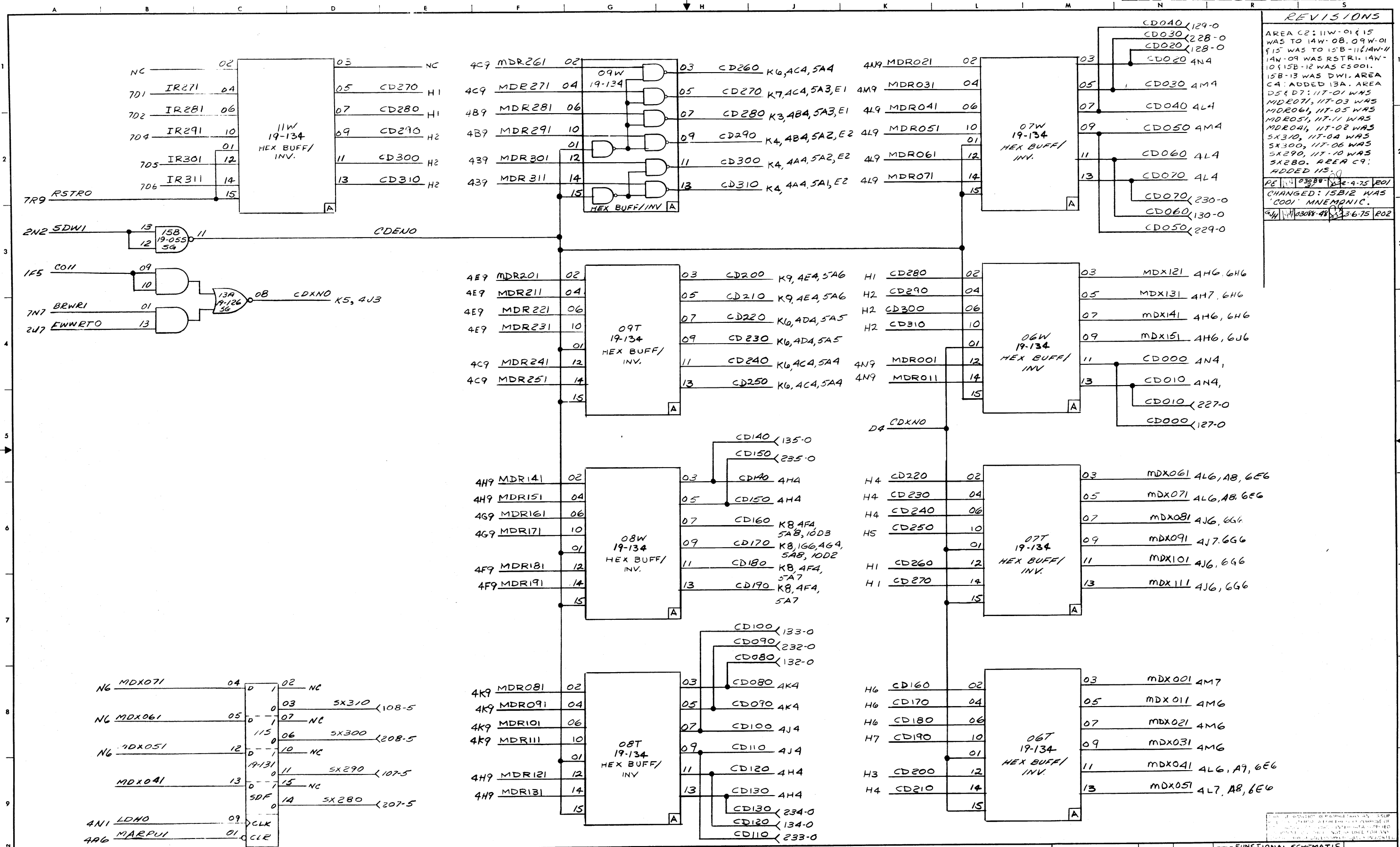
BRUNING 44-231 16042

ALL INFORMATION IS UNCLASSIFIED AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF REPAIR, MAINTENANCE AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

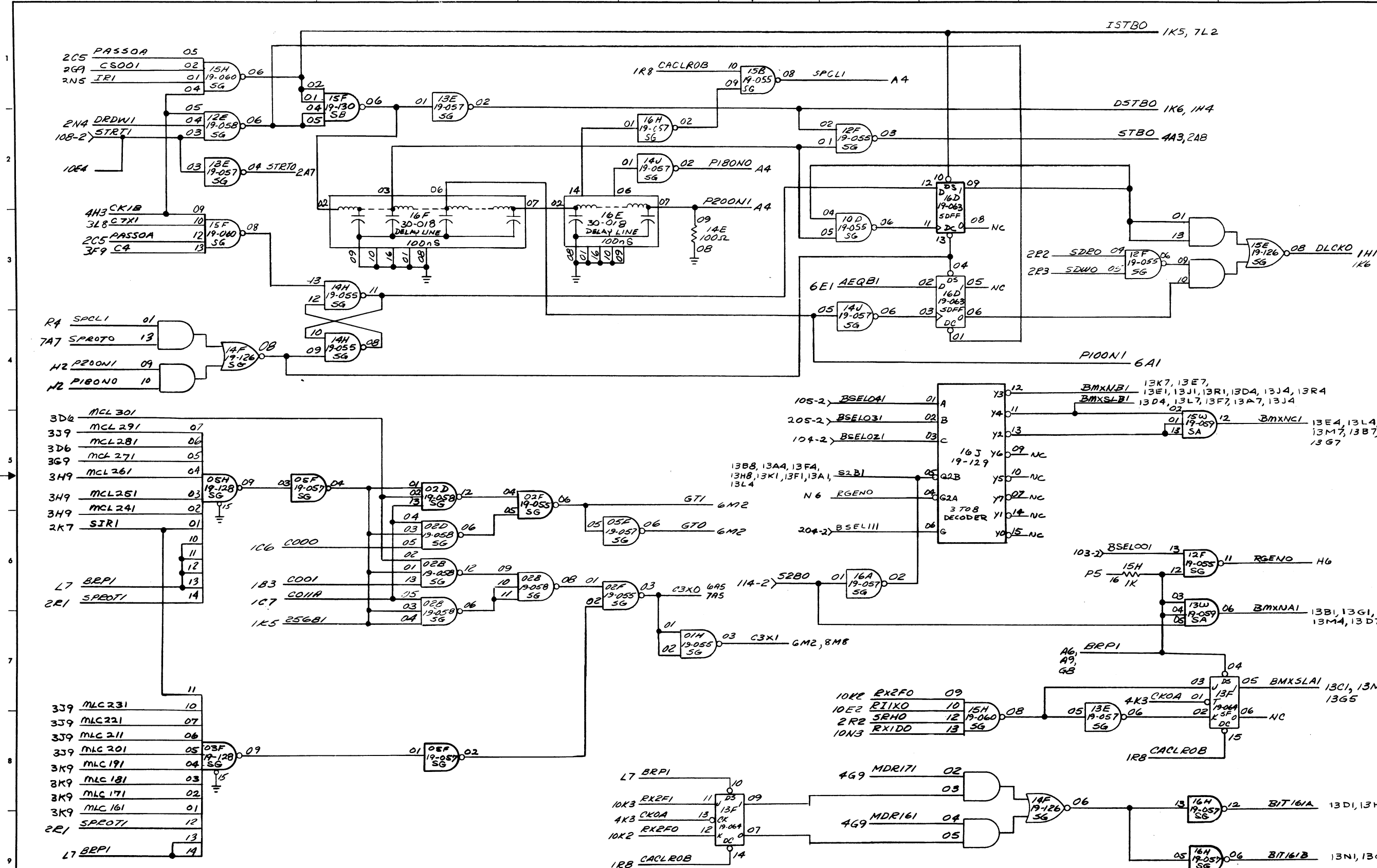
REVISIONS

AREA C2: 11W-01 & 15 WAS TO 14W-08. 09W-01 & 15 WAS TO 15B-11 & 14W-11. 14W-09 WAS RSTR. 14W-10 & 15B-12 WAS CS001. 15B-13 WAS DWI. AREA C4: ADDED 13A. AREA D5 & D7: 11T-01 WAS MDR071, 11T-03 WAS MDR061, 11T-05 WAS MDR051, 11T-11 WAS MDR041, 11T-02 WAS SX310, 11T-04 WAS SX300, 11T-06 WAS SX290, 11T-10 WAS SX280. AREA C9: ADDED 15.

RE 03088 12-9-75 1001
 CHANGED: 15B12 HAS 'COO1' MNEMONIC.
 9/4/75 13088-48 3-6-75 1002

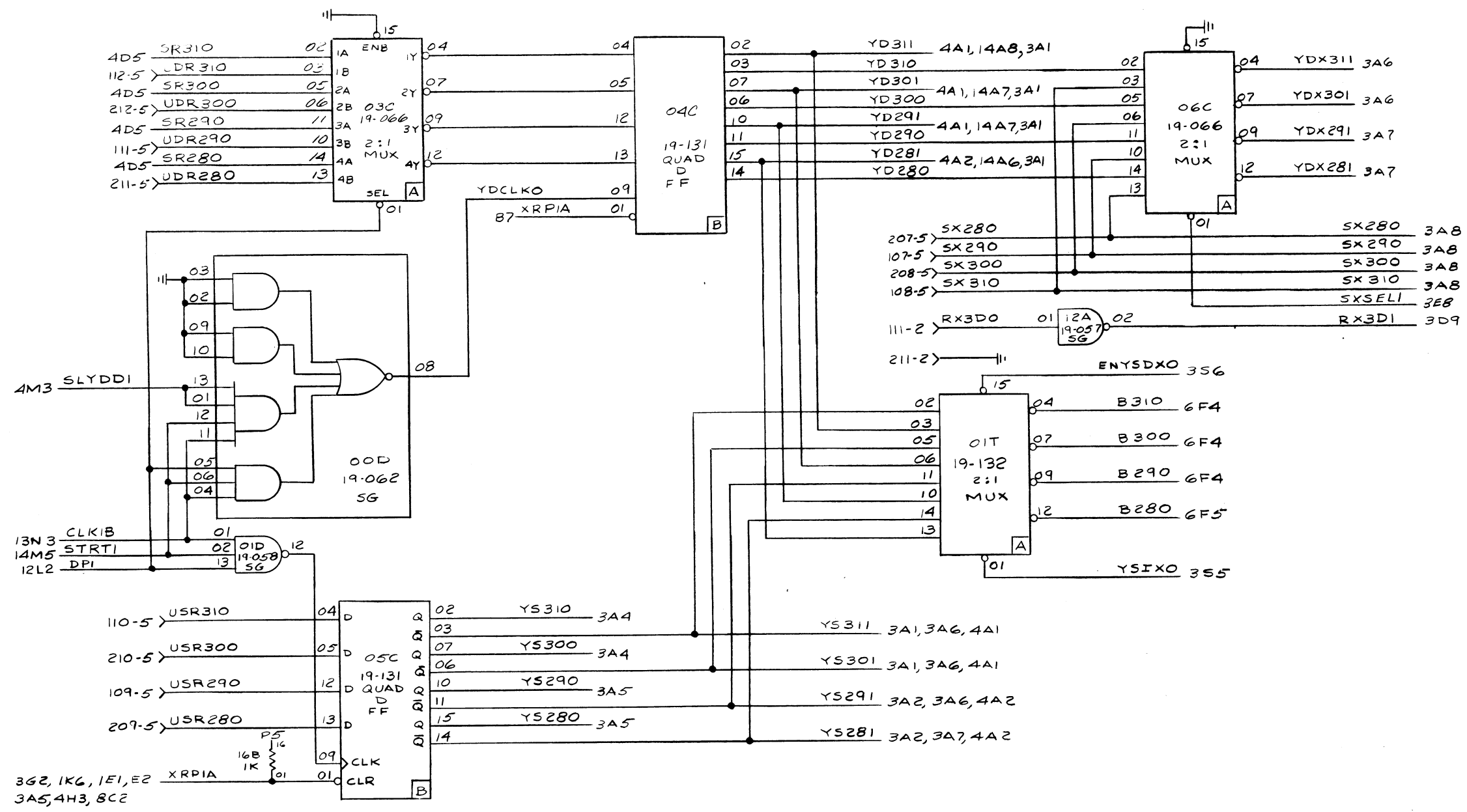


BRUNING 44-231 16042



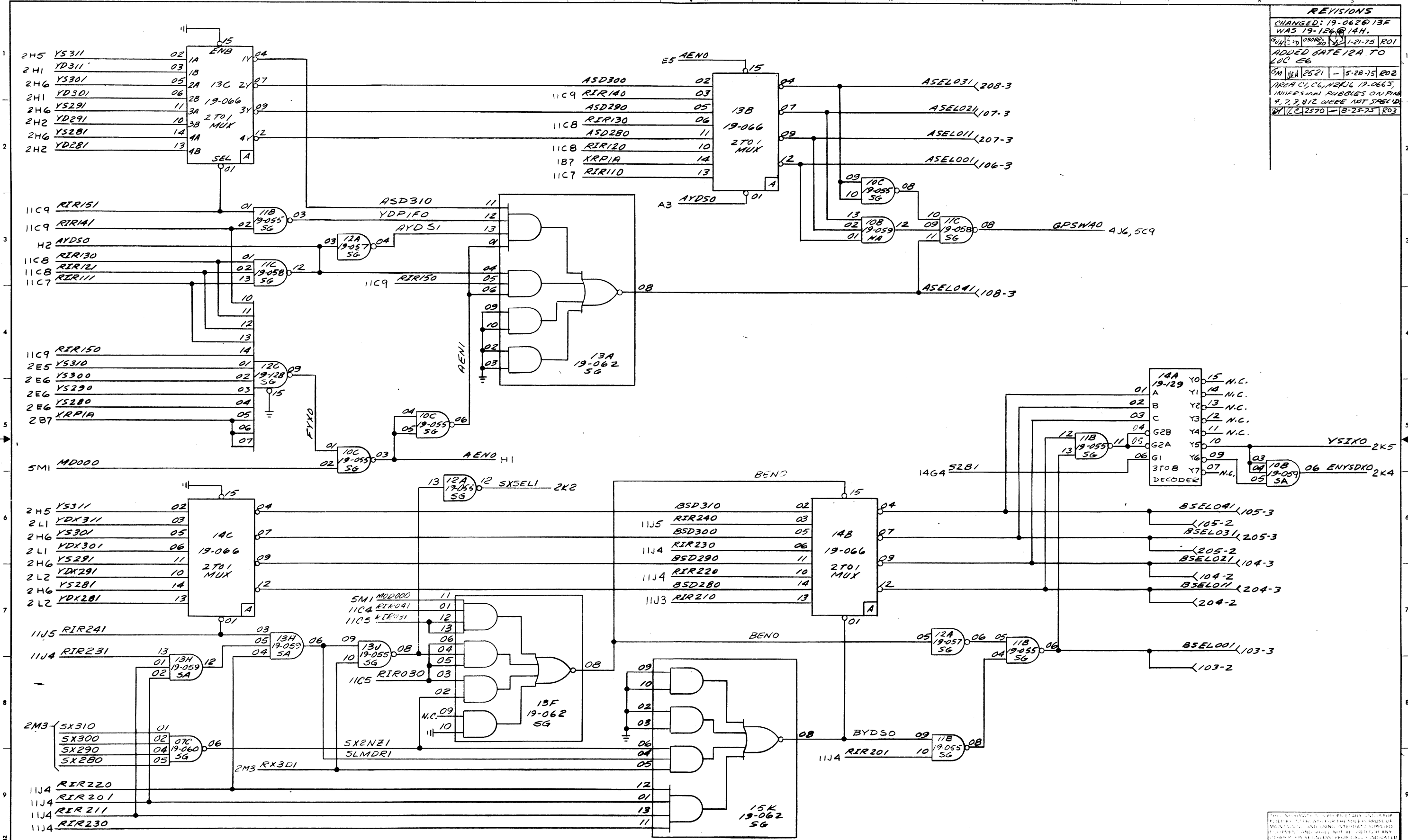
REVISIONS	
CHANGES EXTENSIVE SEE ECR.	
01	12-5-75 R01
02	12-6-75 R02
03	12-7-75 R03
04	12-7-75 R04
05	12-7-75 R05
06	12-7-75 R06
07	12-7-75 R07
08	12-7-75 R08
09	12-7-75 R09
10	12-7-75 R10
11	12-7-75 R11

REVISION	
CHANGED: 'DPI' LEAD @ LOC AS WAS 'DI'.	
03088 01 21-75 R01	
06C01 WENT TO 12A02 'SXSELI' WAS NOT SPEC	
04M WEN 2521 - 5-28-75 R02	
AREA D2, 4A2, 19-066 5 INVERSION BUBBLES ON PINS 4, 7, 9, 12 WERE NOT SPEC'D.	
AREA 14, 19-132 INVERSION BUBBLES ON PINS 4, 7, 9, 12 WERE NOT SPEC'D.	
37 12-25-70 - 8-25-75 R03	

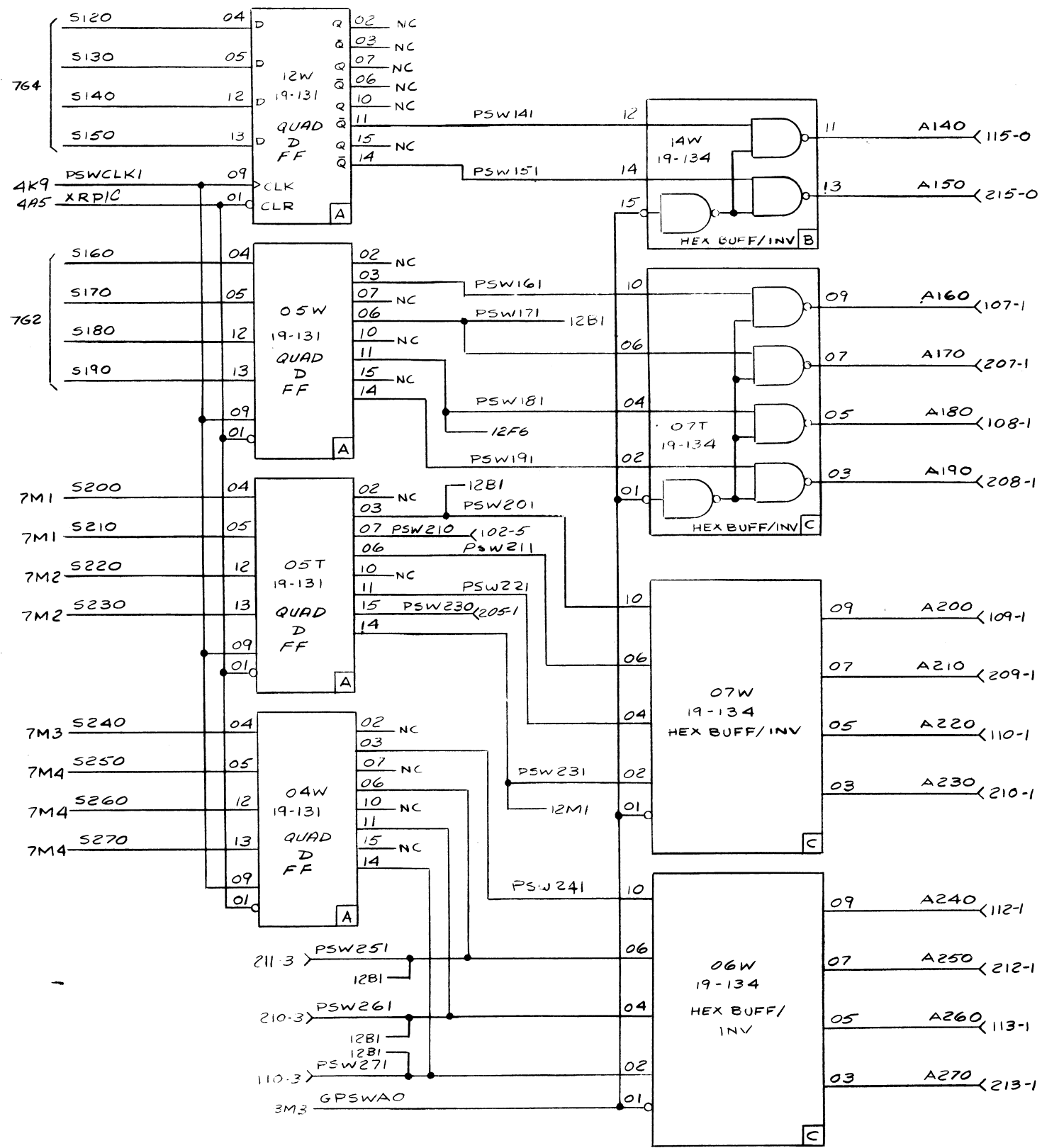


THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED

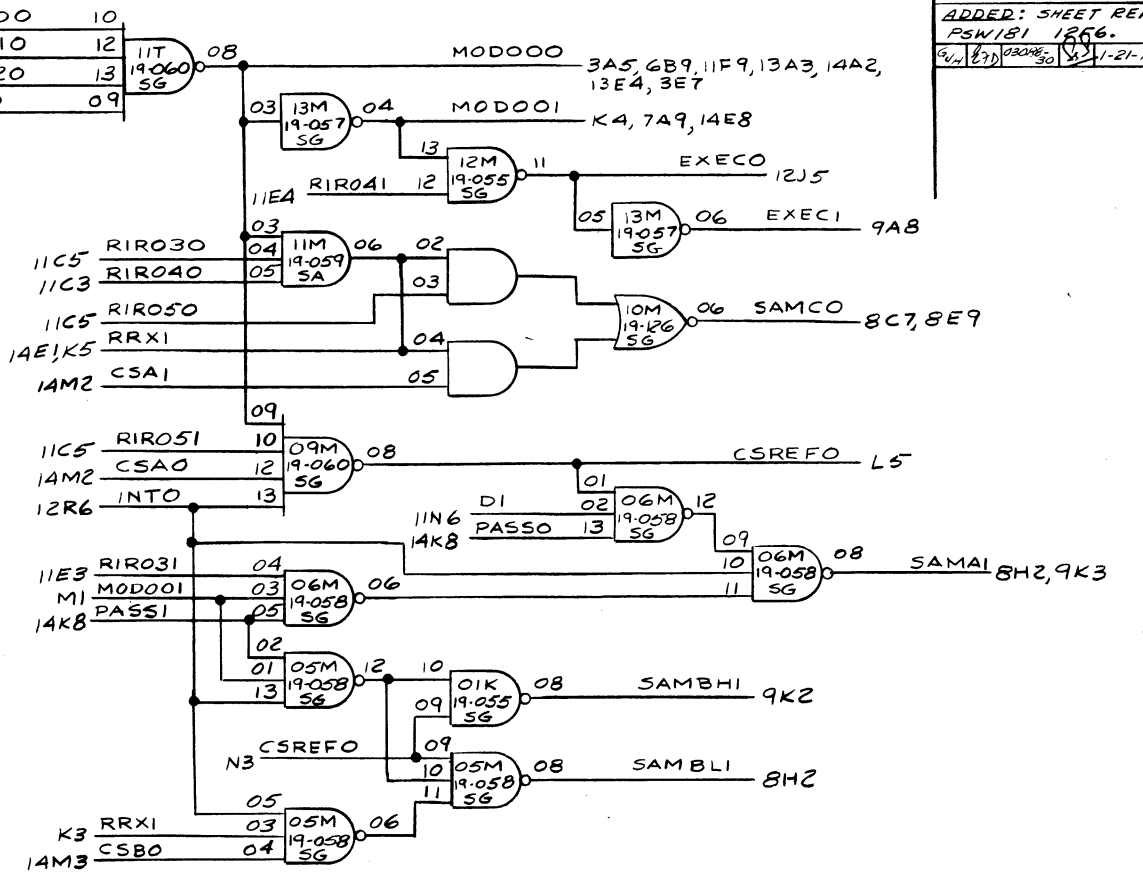
REVISIONS	
CHANGED:	19-0620 13F
WAS:	19-126 14H.
BY:	10800 20
DATE:	7-21-75
BY:	RO1
ADDED:	GATE 12A TO LOC EG
BY:	2521
DATE:	5-28-75
BY:	RO2
AREA:	CI, CL, MRLC 19-0665
MINOR:	MAN RUBBLES ON PINS 4, 7, 8, 12 WERE NOT SPEC'D
BY:	2570
DATE:	8-25-75
BY:	RO3



DRAWING 44-231 16042



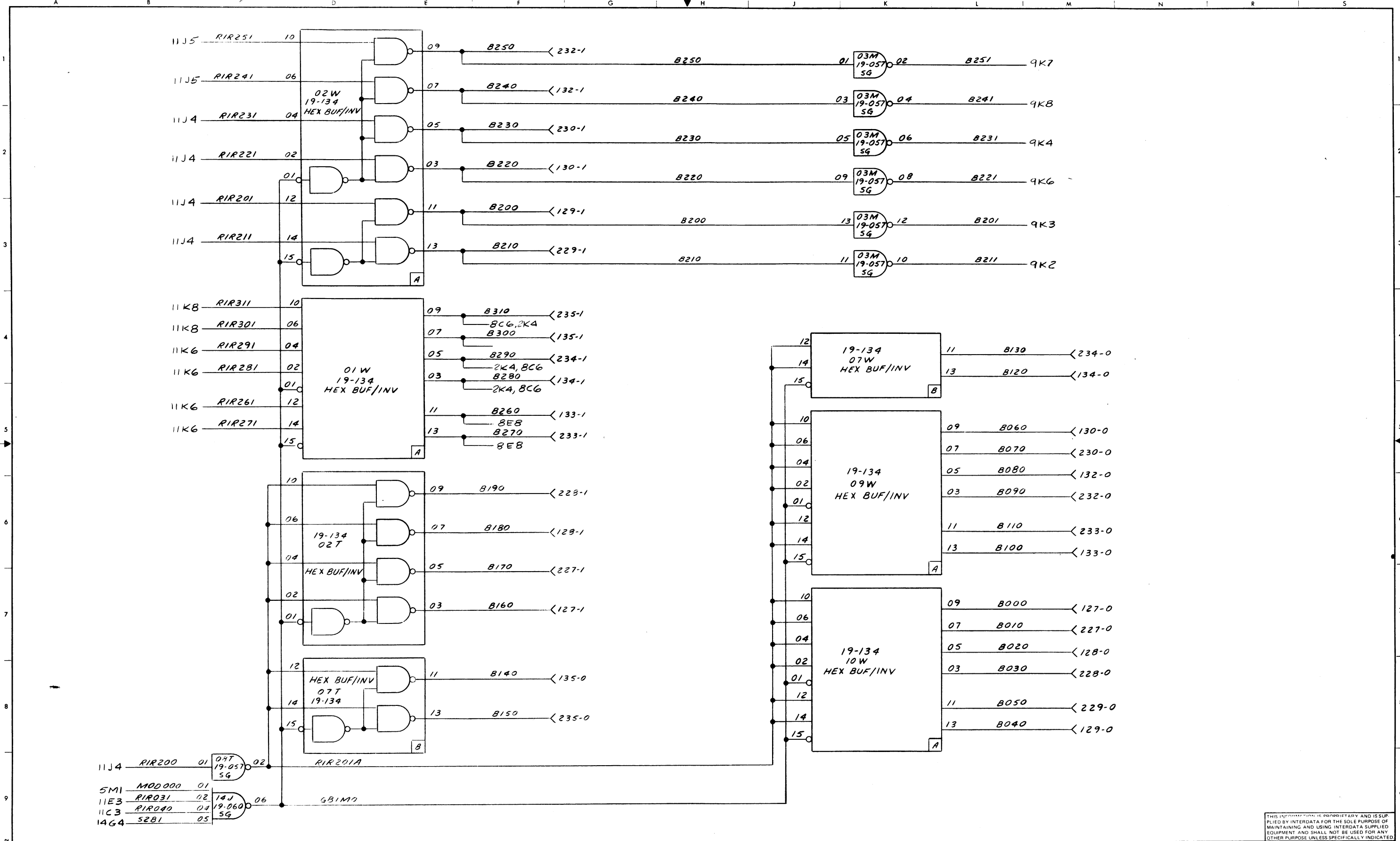
11C1	R1R000	10
11C2	R1R010	12
11C3	R1R020	13
14M3	CSB0	09



REVISIONS	
1	ADDED: SHEET REF TO PSW151, 1256.
2	...
3	...
4	...
5	...
6	...
7	...
8	...
9	...

FUNCTIONAL SCHEMATIC
8/32 CPB
03088
35-537 R01 DOB 5 14



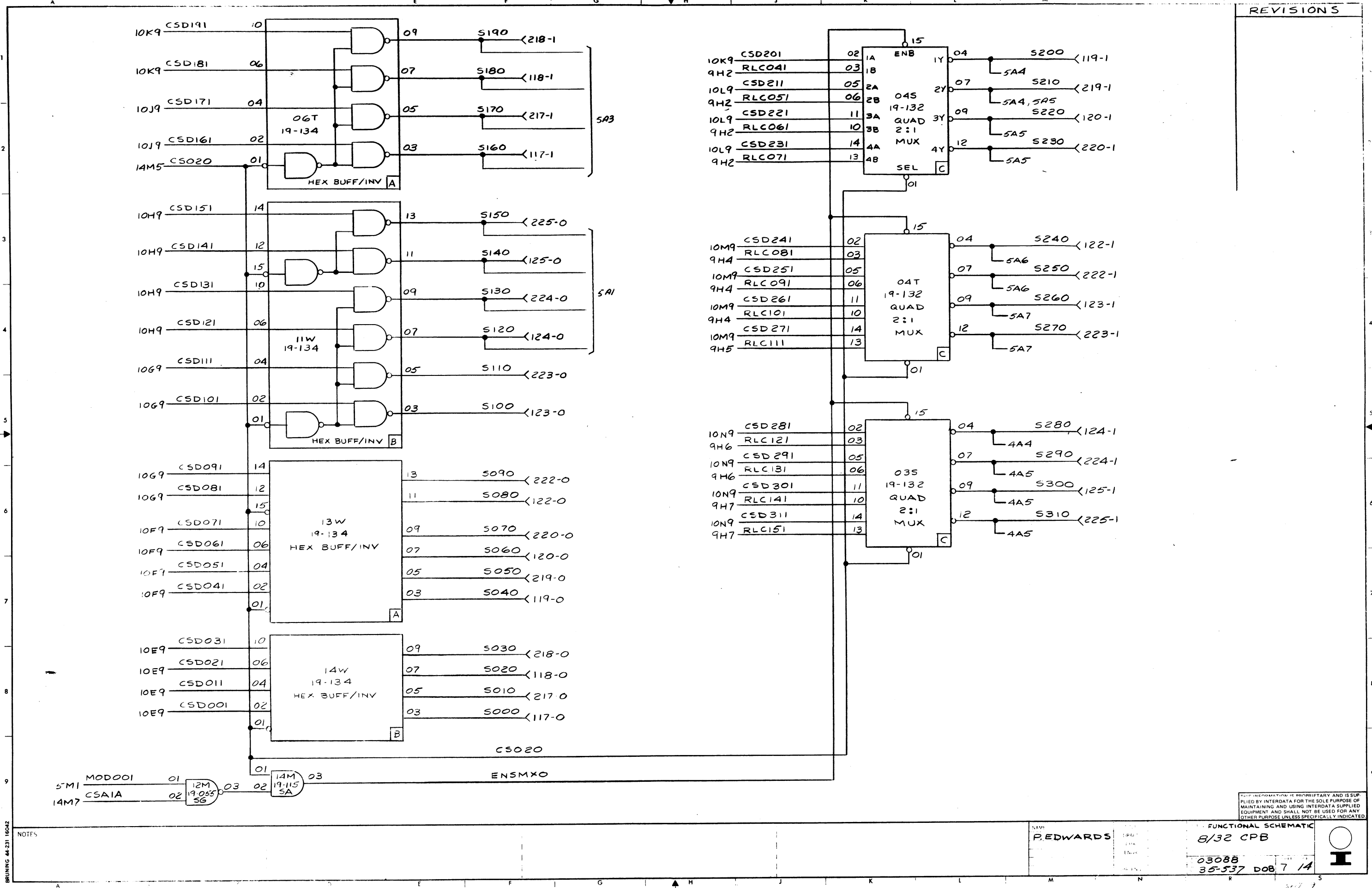


- 11J4 RIR200 01 03T 19-057 SG 02
- 5M1 M00000 01 14J 19-060 SG 06
- 11E3 RIR031 02 14J 19-060 SG 06
- 11C3 RIR040 03 14J 19-060 SG 06
- 14G4 5281 05 14J 19-060 SG 06

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

NOTES		NAME J.R. BIELSKIE	TITLE 10-11-74	TITLE FUNCTIONAL SCHEMATIC 8/32 CPB	SHEET 6 OF 14
		DIR ENG	35-537 D08	6-14	

BRUNING 44-231 1604Z



BRUNING 44231 1604Z

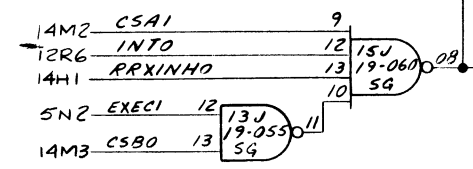
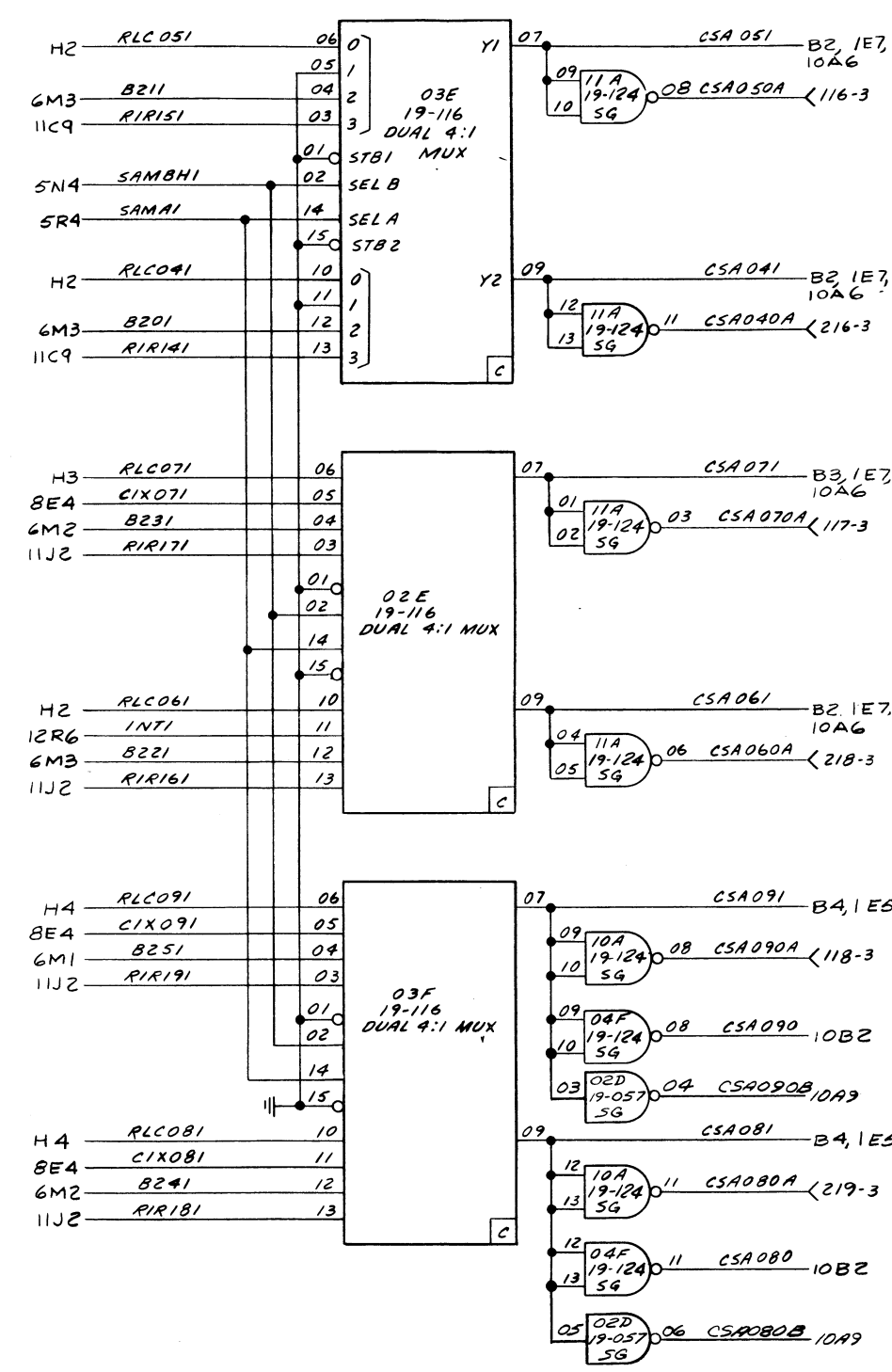
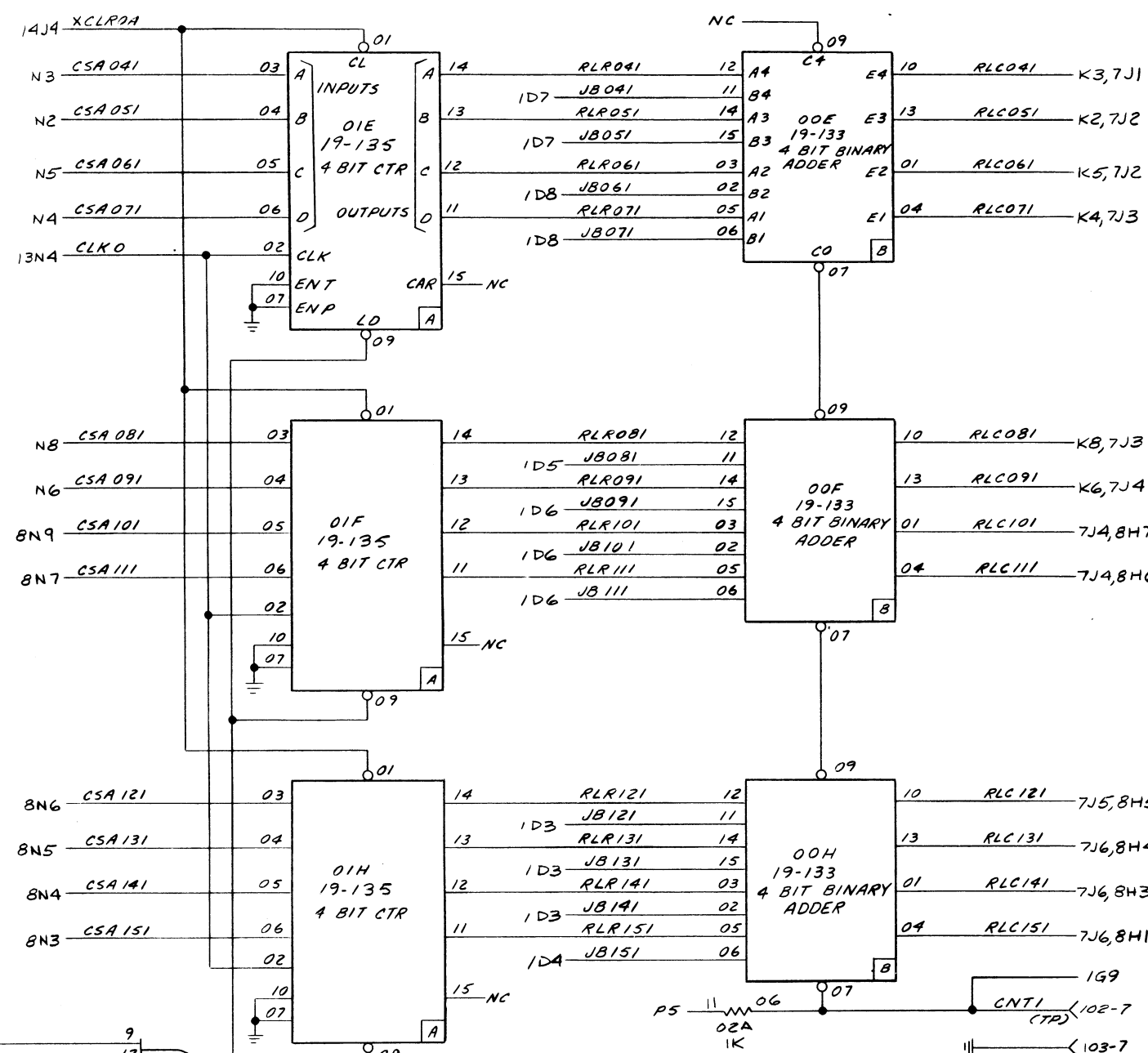
NOTES

NAME P. EDWARDS	DATE 8/32 CPB	FIGURE NO. 03088	REV. 36-537 DOB 7 14
--------------------	------------------	---------------------	-------------------------

FUNCTIONAL SCHEMATIC

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

REVISIONS				
CHANGED: CNT1 (TP) WAS ON PIN 104-7;				
ADDED: SHEET REF TO CNT1 OF 100				
04	3	03088-30	1-21-78	R01
ADDED: GATE @ 02D-OUTPUT 'CSA090B'; GATE @ 02D OUTPUT 'CSA080B'.				
14	10	03088-40	1-23-78	R02

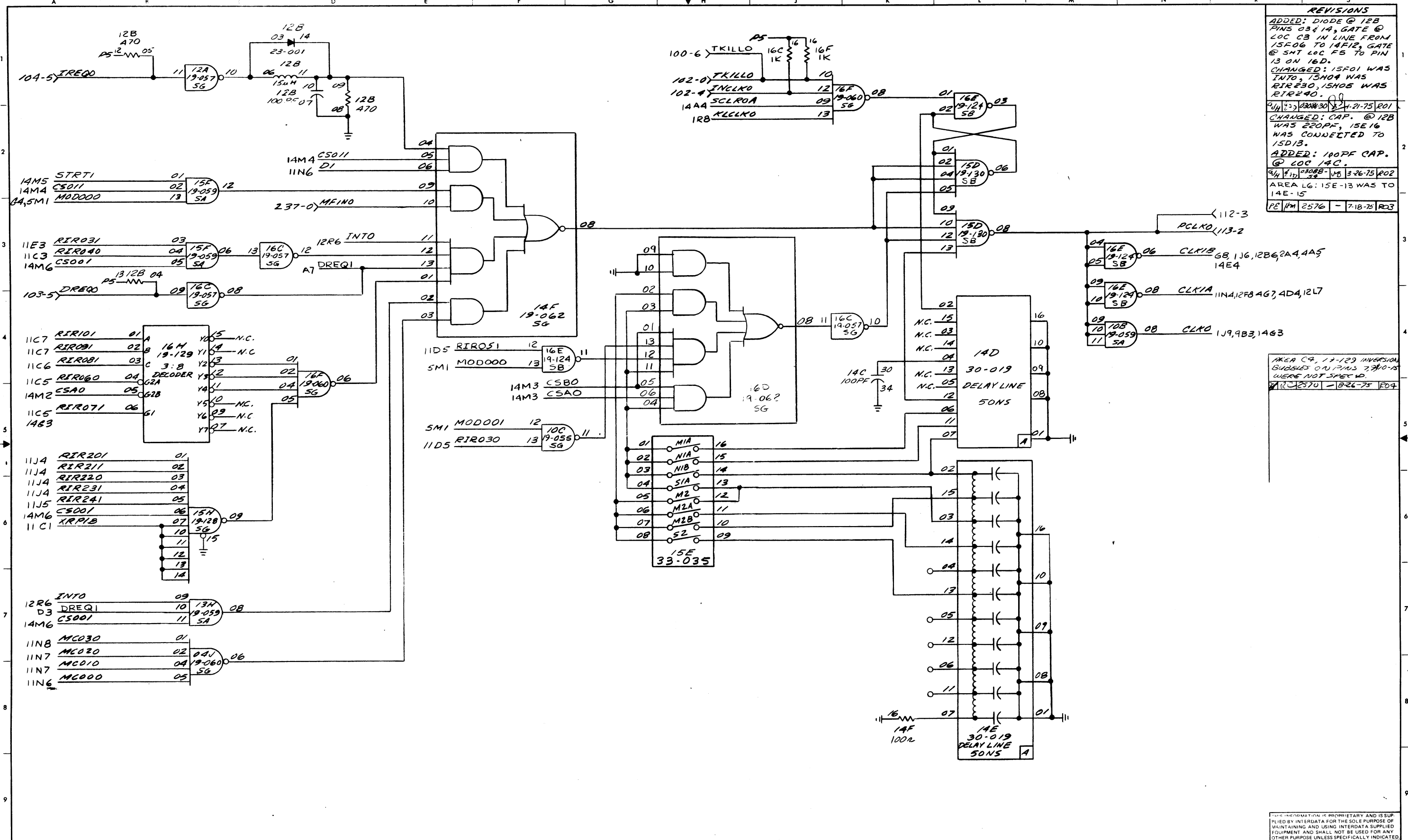


NOTES

NAME	TITLE	DATE	TITLE
J. R. BIELSKIE	DRAFT	10-11-74	FUNCTIONAL SCHEMATIC
	CHK		8/32 CPB
	ENGR		
	DIR ENG		

TASK NO. 03088
SHEET OF 9-14

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED



REVISIONS

ADDED: DIODE @ 12B
PINS 03 & 14, GATE @
LOC C3 IN LINE FROM
15F06 TO 14F12, GATE
& SHT LOC F5 TO PIN
13 ON 16D.

CHANGED: 15F01 WAS
INTO, 15H04 WAS
R1R240.

CHANGED: CAP. @ 12B
WAS 220PF, 15E16
WAS CONNECTED TO
15D13.

ADDED: 100PF CAP.
@ LOC 14C.

AREA LG: 15E-13 WAS TO
14E-15

PE/AM 2576 - 7-18-75 R03

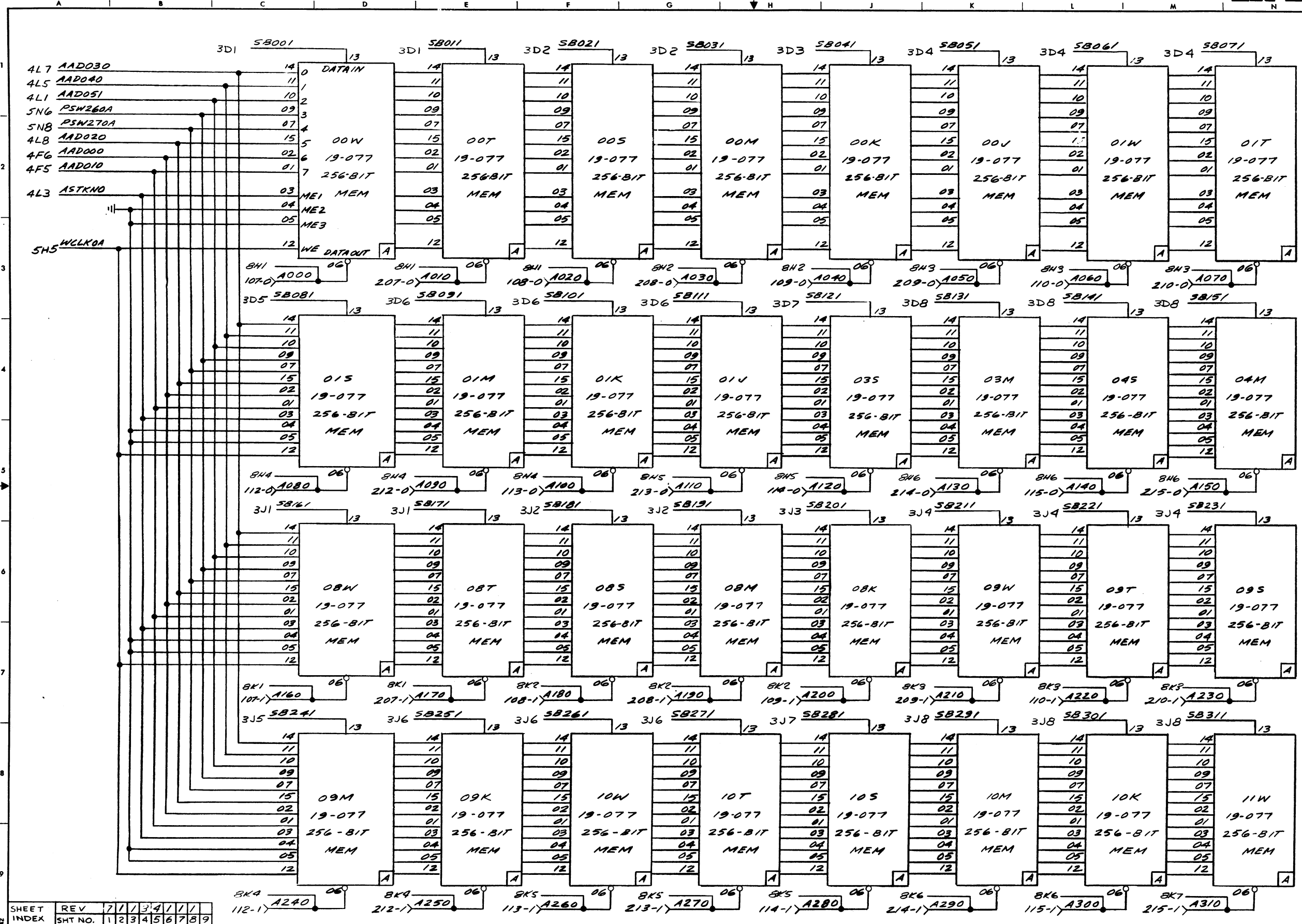
AREA C9, 17-129 INVERSION
BUBBLES ON PINS 7, 8, 10-15
WERE NOT SPEC'D.

PE/AM 2570 - 8-26-75 R04

NOTES

NAME	E. ROE	TITLE	8/32
DATE	10-12-74	CHK	CPB
ENGR		DIR ENGR	
0308B		35-537 R04D0813-14	

BRUNING 44-231 16642



REVISIONS			
PRE PRODUCTION APPROVAL	DEV	DATE	INIT
		11/15/74	11-15-74
CHANGED: SHEETS 3, 4, 5 WERE R00 REV.			
ADDED: SHEETS 6-9; REF TO SHT 8 ON '000' TRPH 'A310' LEADS.			
CHANGED: SHEETS 4 & 5 WERE R01 REV.			
CHANGED: SHEETS 4 & 5 WERE R02 REV.			
LEASER FOR PRODUCTION			
ENG. H. H. HAN			
DATE 11/17/74			
REV LEVEL OF SHT 5 WAS R03			
REV LEVEL OF DCS LOC 59 WAS R00, REV LEVEL OF CPC WAS R06			
ENTIRE SHT 19-077'S INVERTION BUBBLES PIN 6, W/RS A07 SPEC'D. REVISED SHTS 12, 6 & 7.			
REVISED SHT 8			

THE REVISION LEVEL OF THIS SHEET IS CONSIDERED TO BE THE REVISION LEVEL OF THE DOCUMENT

PRINTED CIRCUIT BOARDS AGREEING WITH THIS SCHEMATIC MUST BE AT LEAST THE FOLLOWING REVISION LEVEL

8132 CPC	35-555 R08
8132 CPC/DCS	35-555 F01 R04

SHEET INDEX	REV	7	1	3	4	1	1
	SHT NO.	1	2	3	4	5	6

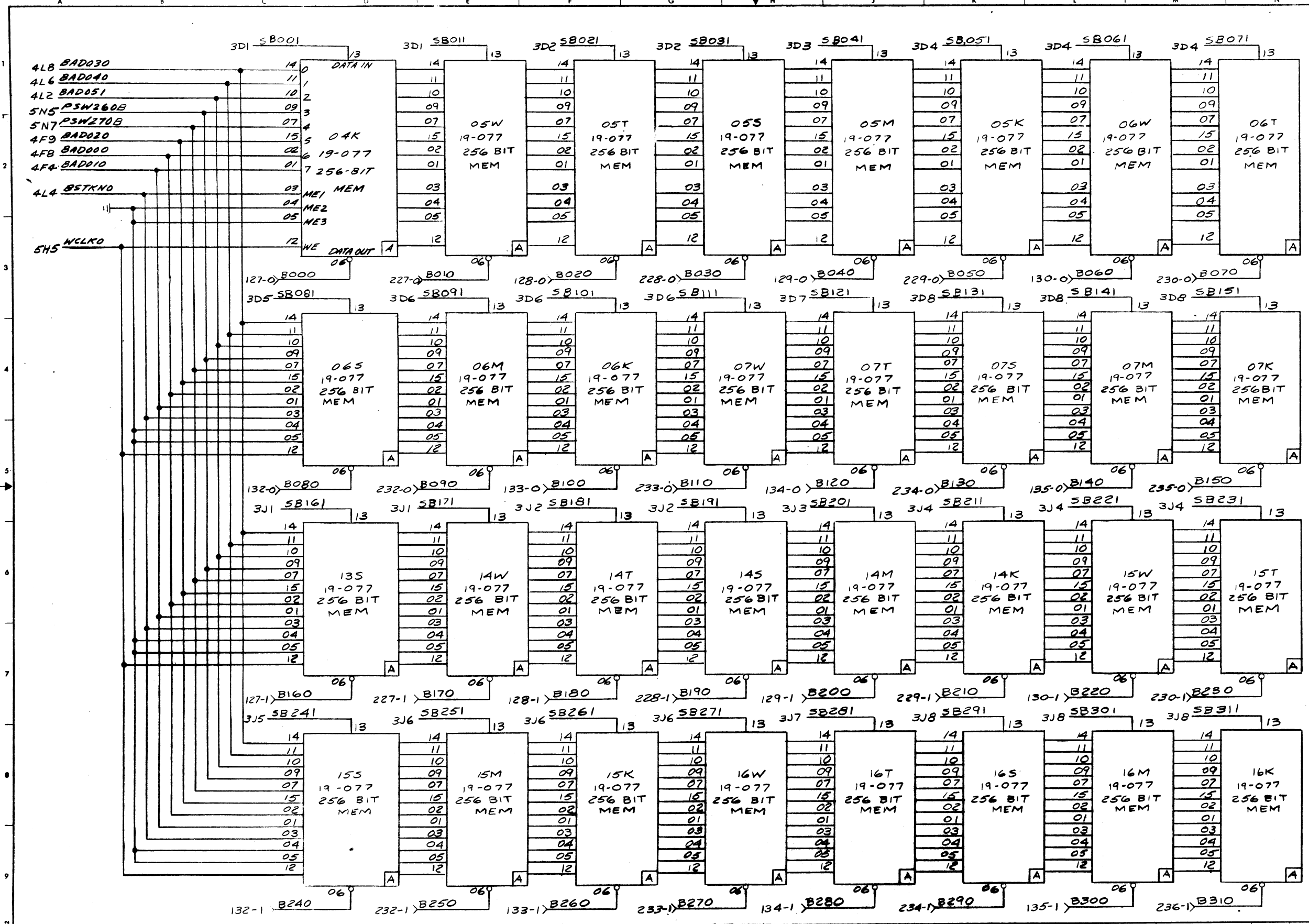
NOTES

NAME	TITLE	DATE	TITLE/FUNCTIONAL SCHEMATIC
G. MELTON	DRAFT	10/15/74	8/32 CPC
G.V. HOMEFIELD	CHK	1-23-75	
E. DEANGELO	ENGR		
R. BARKER	QA		
R.E. JONES	DIR ENGR		

TASK NO. 03088	SHEET OF 1-9
DRW NO. 35-555 R07 D08	

BRUNING 44-231 16042

REVISIONS	
ENTIRE SMT. 19-077'S INVERSION BUBBLE ON PIN 6, WAS NOT SPEC'D.	
1	18-25-70
2	18-25-75
3	18-25-75

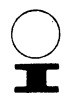


GRUNING 44-231 16042

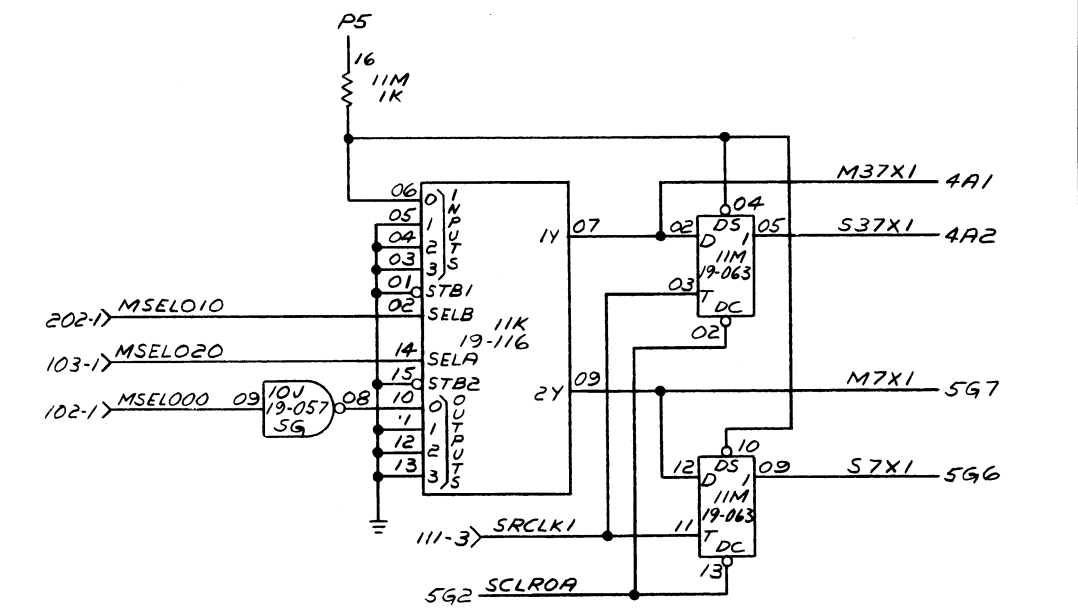
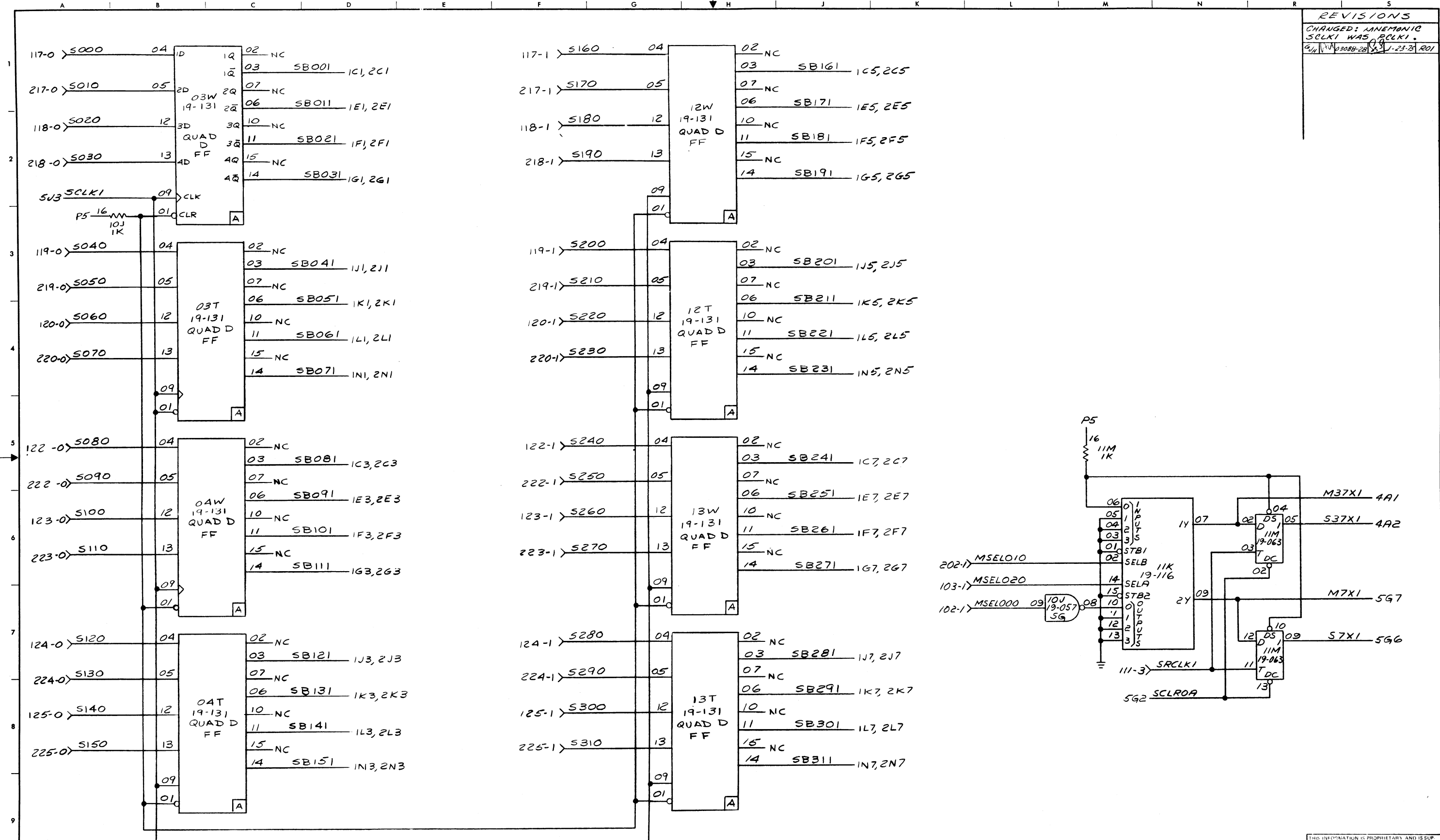
NOTES

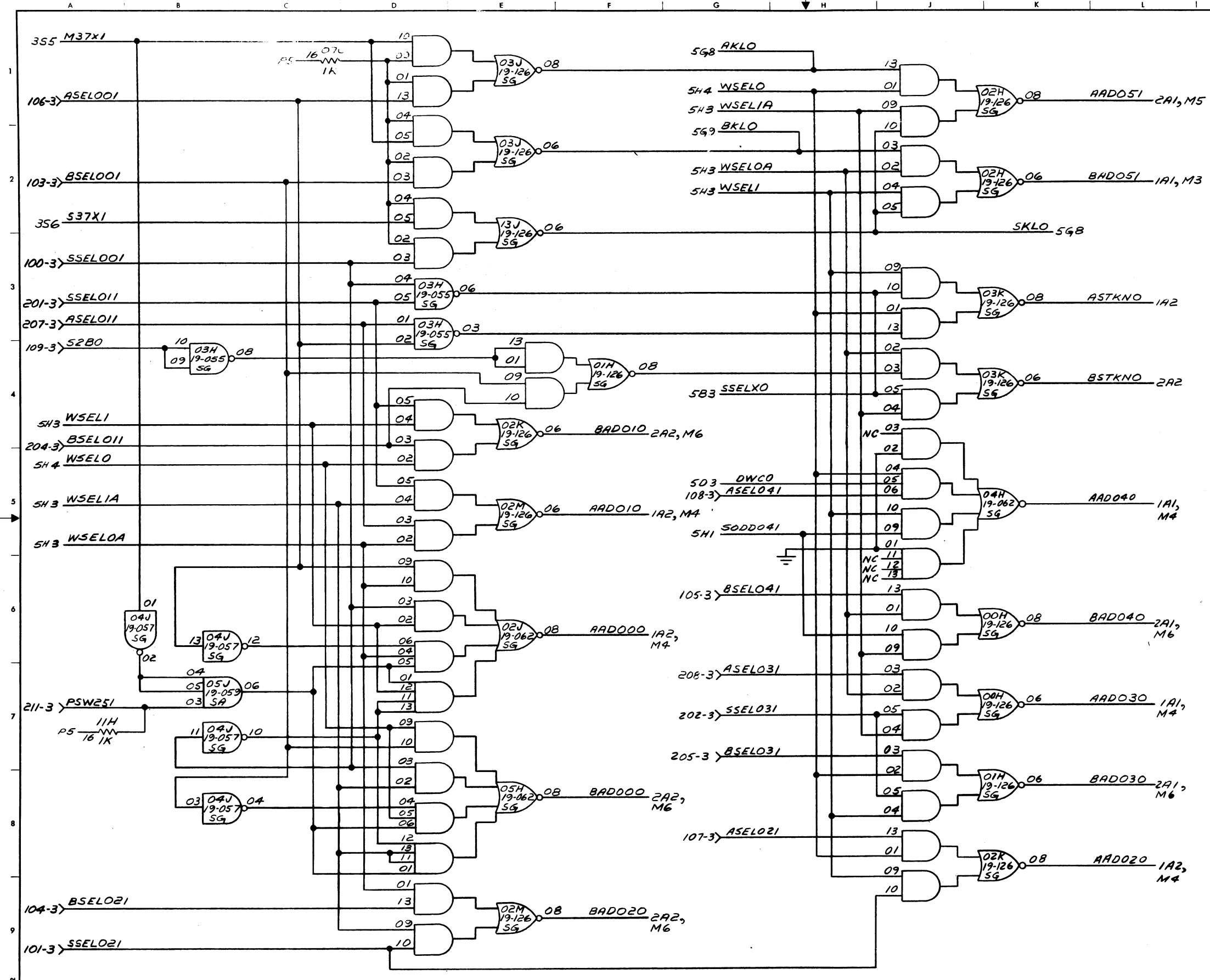
NAME	TITLE	DATE	FUNCTIONAL SCHEMATIC
	8/32 CPC		

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.



REVISIONS
 CHANGED: JANE MANIC
 SCLK1 WAS BCLK1.
 6/11/71 03088-28 1-23-71 R01





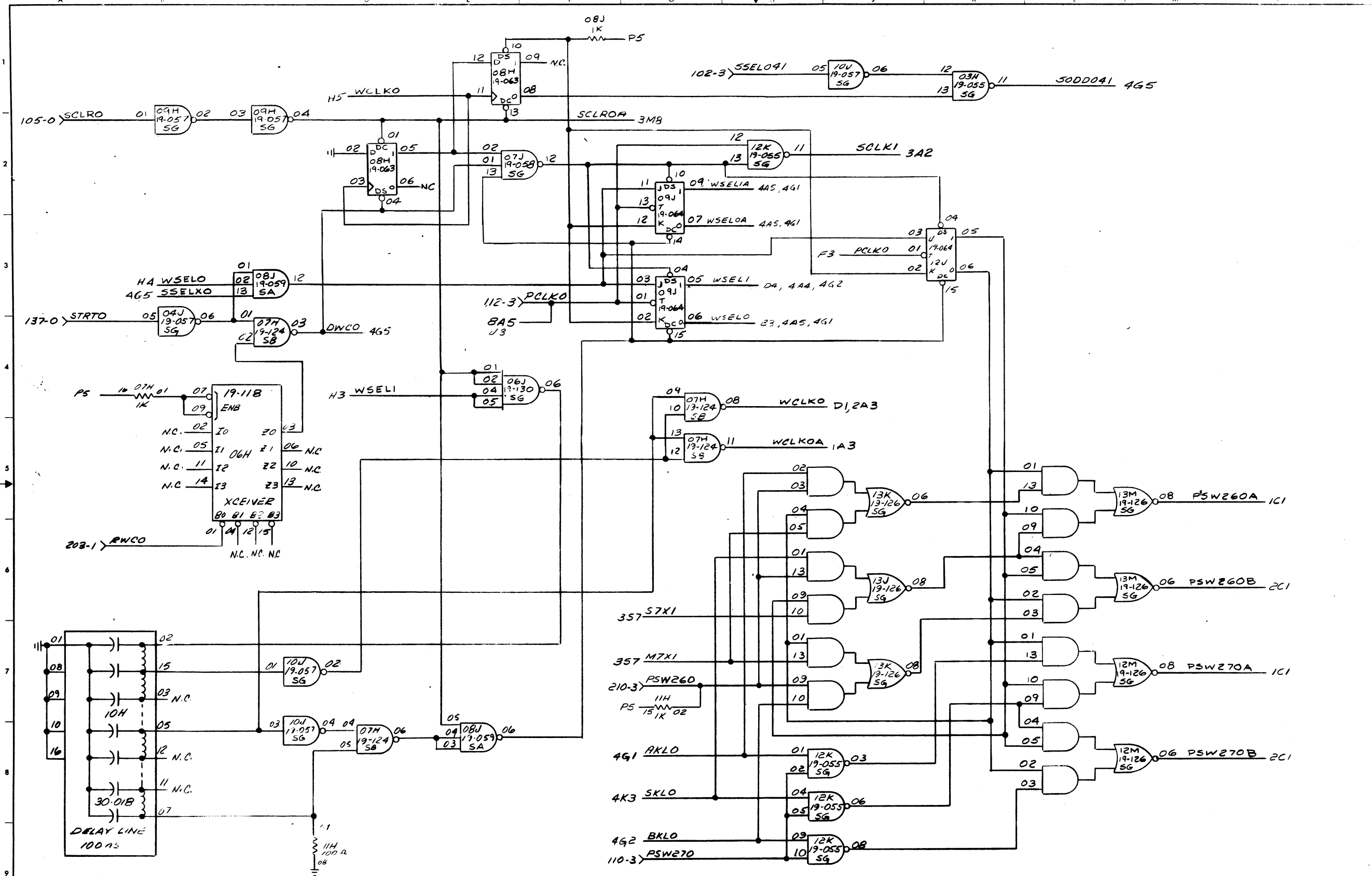
REVISIONS

CHANGED: MNEMONIC 'SODDO41' WAS 'SSELO41';
 ADDED: GATE @ SMT LOC 34-19-055 @ 03H; 19-126 @ 01H SMT LOC F4;
 DELETED: 2-33-035 SWITCHES @ 16A & 16B
 12-23-75 R01
 ADDED: RES ON 'PSW251' LEAD; @ 11H, RWS OF 16.
 1-23-75 R02
 WSELO WENT TO 5L4 WSELIA, WSELOA & WSELI WENT TO 5L3
 4-10-75 R03

NOTES

NAME	TITLE	DATE
	FUNCTIONAL SCHEMATIC	
	8/32 CPC	
	TASK NO. 03084	SHEET OF 4-9
	35-555 R03 DOB	

BRUNING 44-231 16042



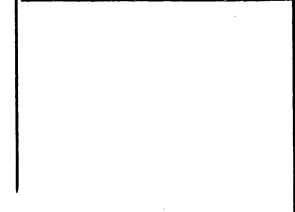
REVISIONS

1. LABEL: 13K10 WAS
 CORR TO M7X1 NET; 13K13
 WAS CORR TO BKLO NET;
 DELETED: 19-057 GATE ON
 'PCLKO' NET;
 ADDED: GATE @ LOC 12K
 19-055; GATE @ LOC 10J
 19-057; GATE @ LOC 03H
 19-056.

2. ADDED: 1K PULL-UP @
 LOC 11H ON PSW260.

3. DELETED IC 09H, 19057
 ADDED: 12J, 13-064

4. ADDED 08J IN LOC
 EB, 07H04WENT TO 10J03
 10J04 WENT TO 09J15, 07H05
 WENT TO 07H09 & 07H13
 07H04 WENT TO SEL0A,
 06H WAS SKINN THIS



NOTES

NAME: P. EDWARDS

FUNCTIONAL SCHEMATIC

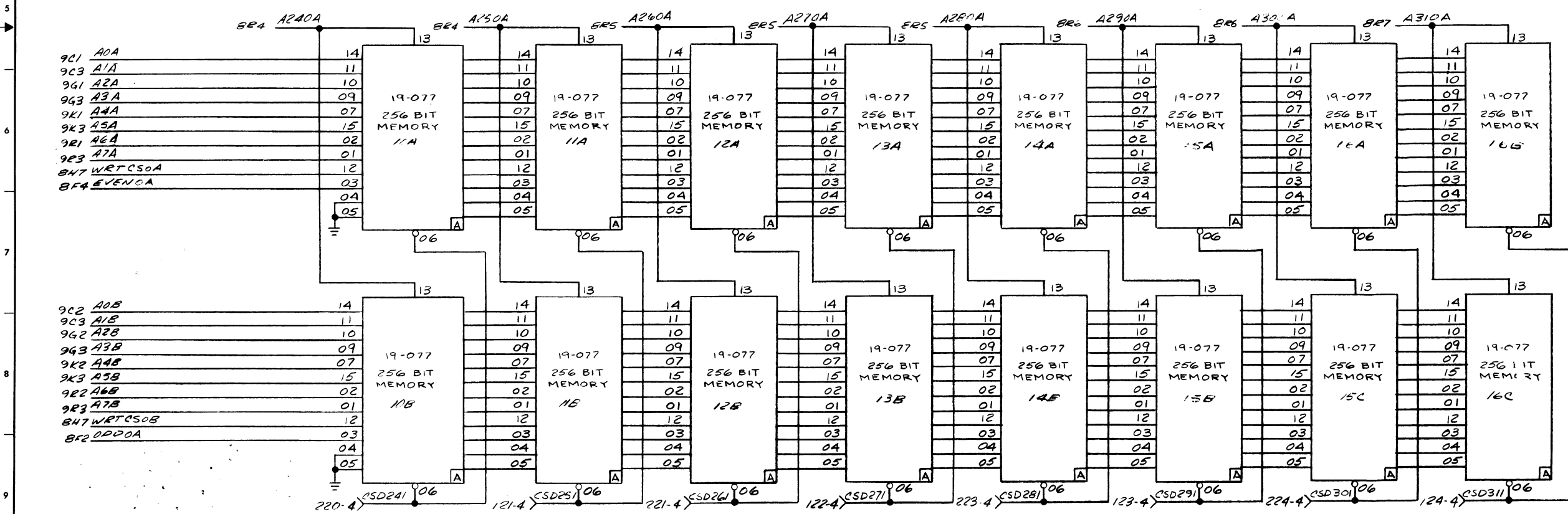
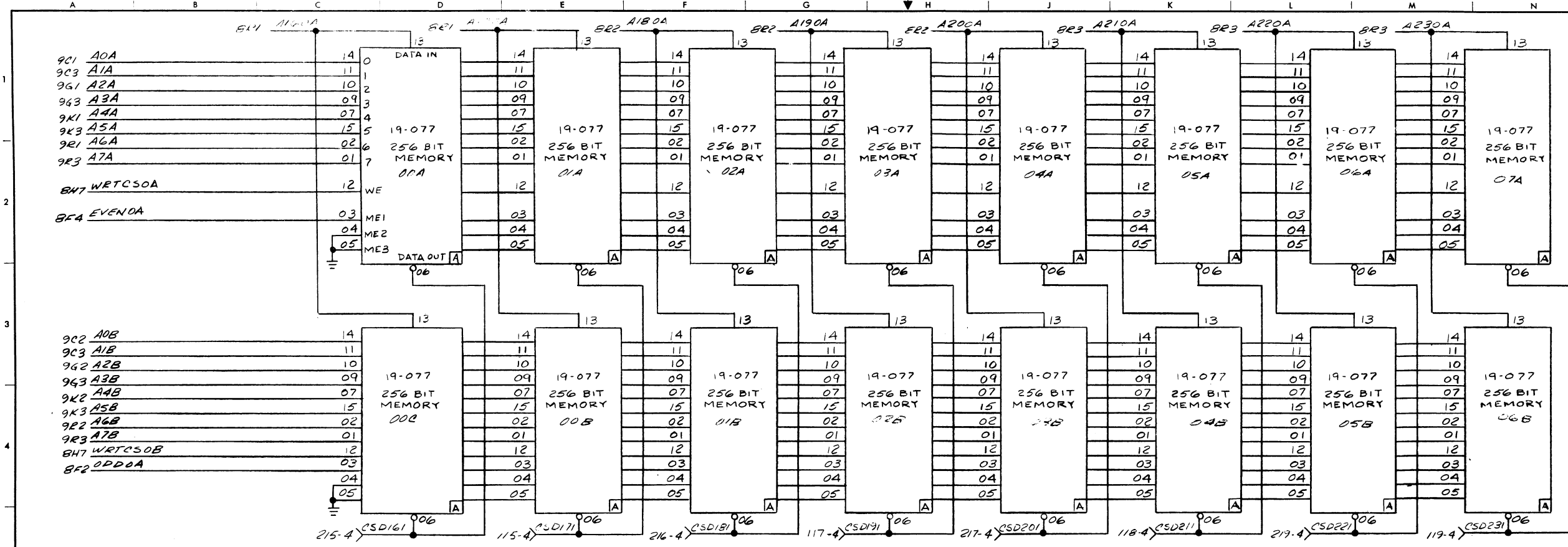
8132 CPC

03088

35-555-RMD08 5 9

THIS INFORMATION IS PROPRIETARY AND IS SUP-
 PLIED BY INTERDATA FOR THE SOLE PURPOSE OF
 MAINTAINING AND USING INTERDATA SUPPLIED
 EQUIPMENT AND SHALL NOT BE USED FOR ANY
 OTHER PURPOSE UNLESS SPECIFICALLY INDICATED

REVISIONS
1. INTAKE SHIT, 19-077'S
INVERSION BUBBLE ON PIN
6 WAS NOT SPEC'ED.
2. 12-2570 - 18-25-73 R01

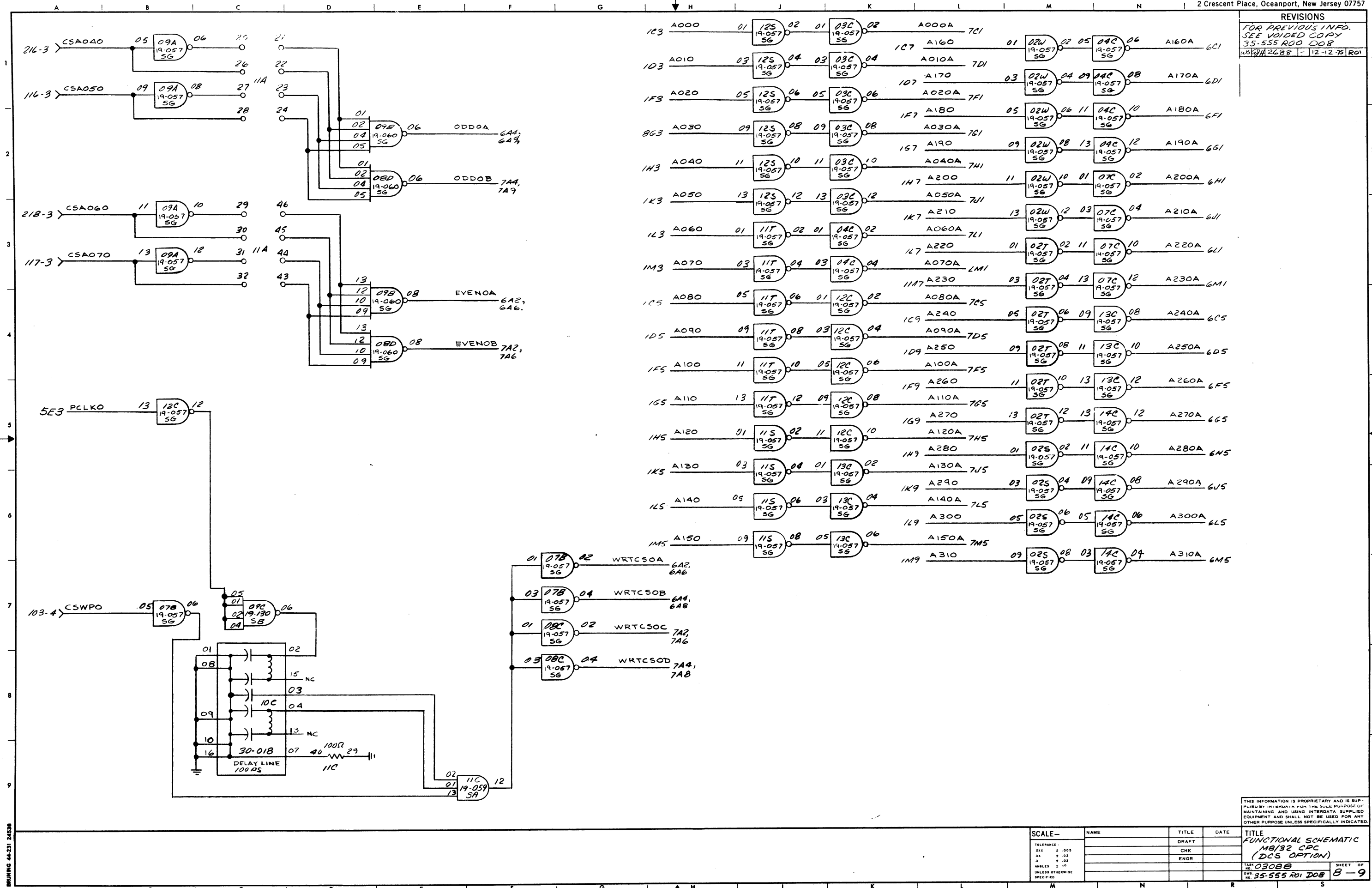


THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

SCALE-	NAME	TITLE	DATE	TITLE
TOLERANCE: XXX ± .005 XX ± .02 X ± .03 ANGLES ± .10 UNLESS OTHERWISE SPECIFIED		DRAFT		FUNCTIONAL SCHEMATIC M8/32 CPC (DCS OPTION)
		CHK		TASK NO. 03088
		ENGR		SHEET OF 6-9
				DWG NO. 35-555 REV D08



TABLE
FUNCTIONAL SCHEMATIC
 MS/32 CPC
 (DCS OPTION)
 03088
 35-555 REV D08 7-9



REVISIONS	
FOR PREVIOUS INFO. SEE VOIDED COPY 35-555 R00 D08	
12-12-78	R01

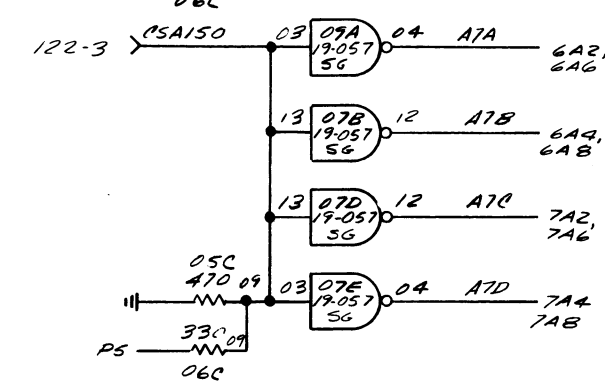
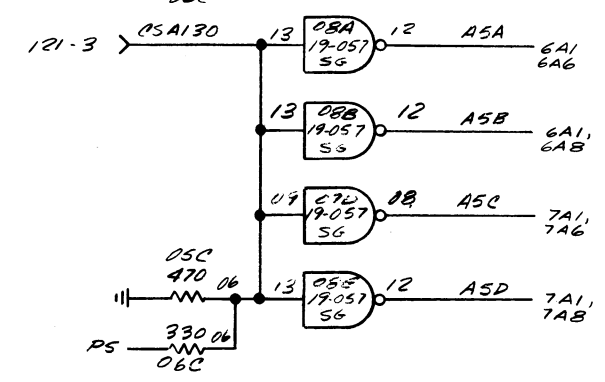
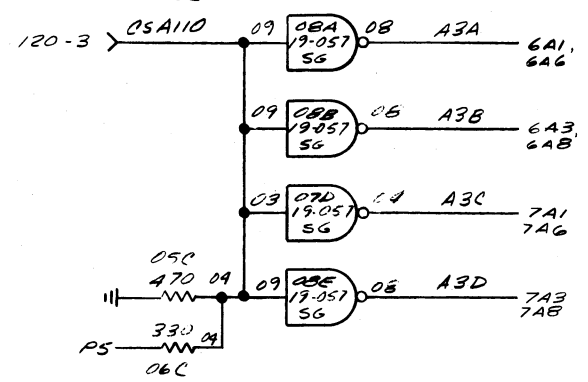
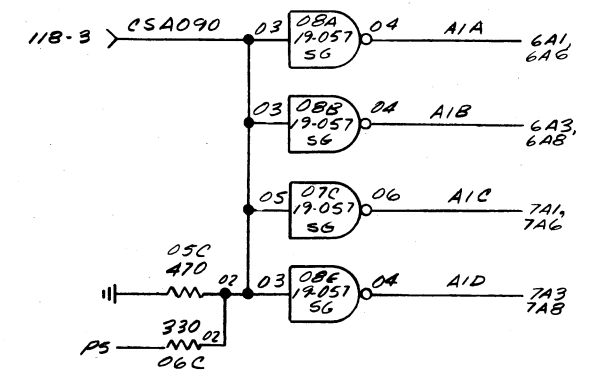
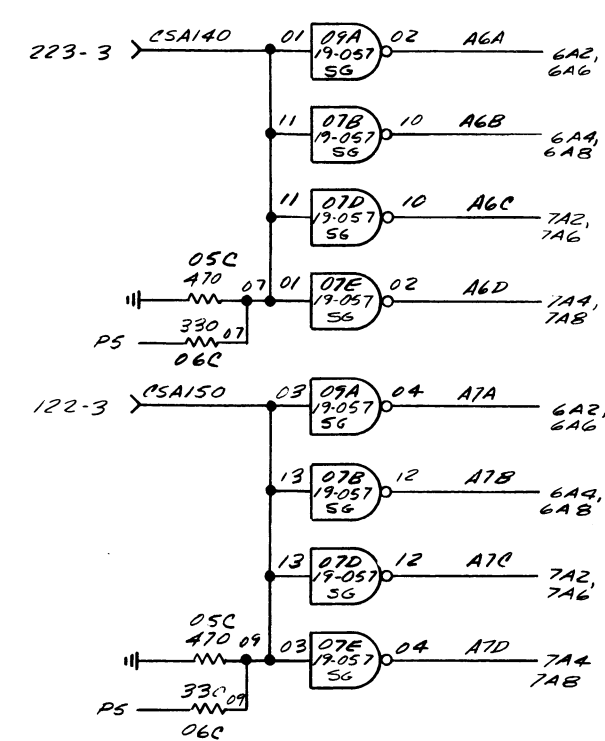
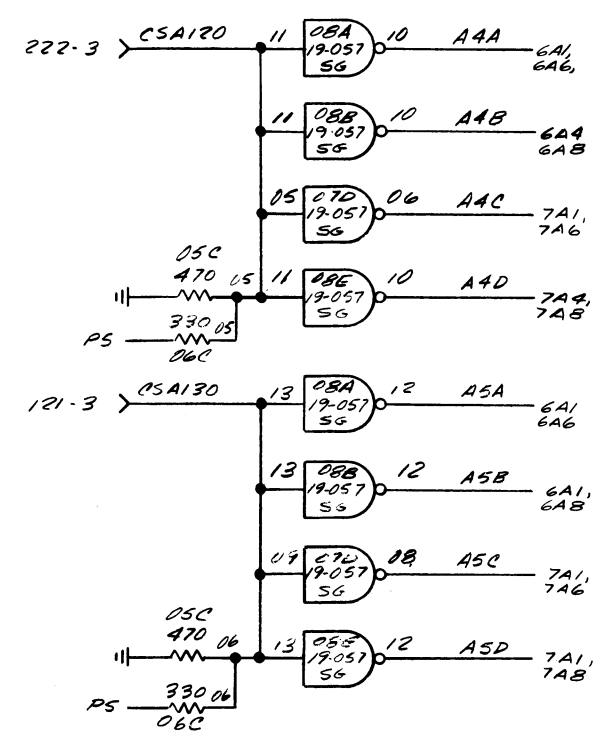
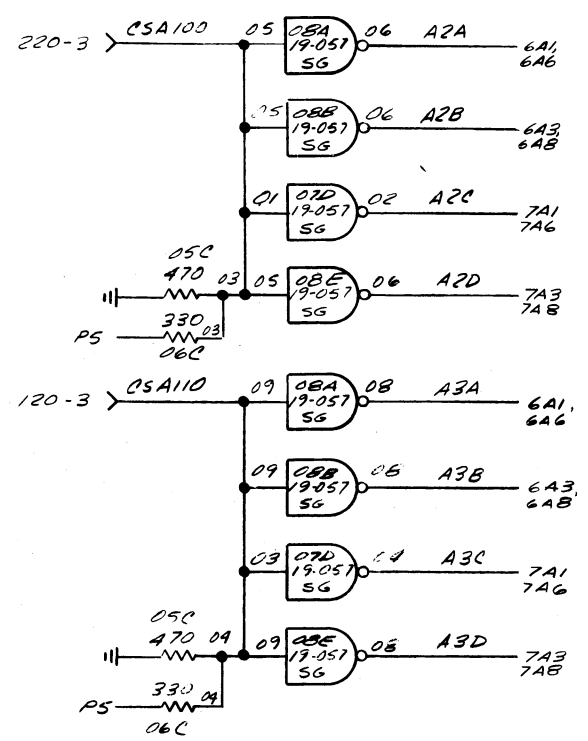
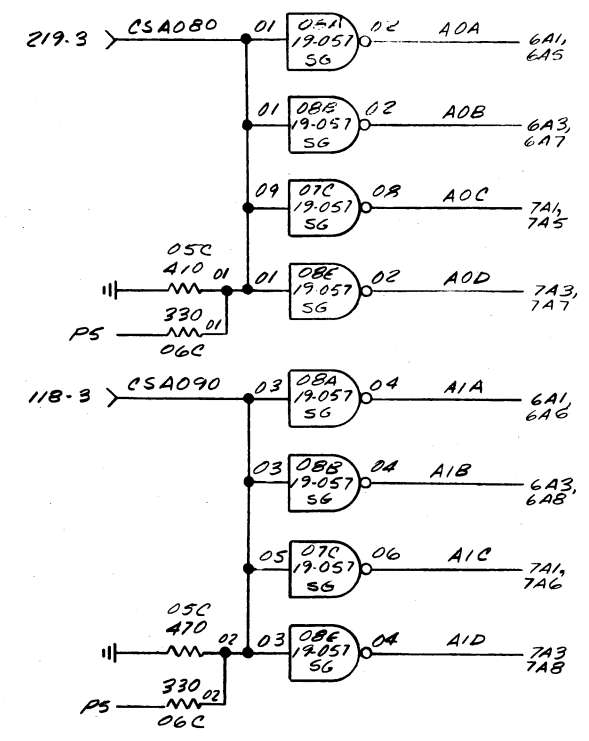
THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

SCALE	NAME	TITLE	DATE
TOLERANCE XXX ± .005 XX ± .02 X ± .03 ANGLES ± 10° UNLESS OTHERWISE SPECIFIED		DRAFT CHK ENGR	

TITLE	TAXI NO.	SHEET OF
FUNCTIONAL SCHEMATIC MB/32 CPC (DCS OPTION)	0308B	8-9
DWG NO. 35-555 R01 D08		

BRUNING 44-231 24538

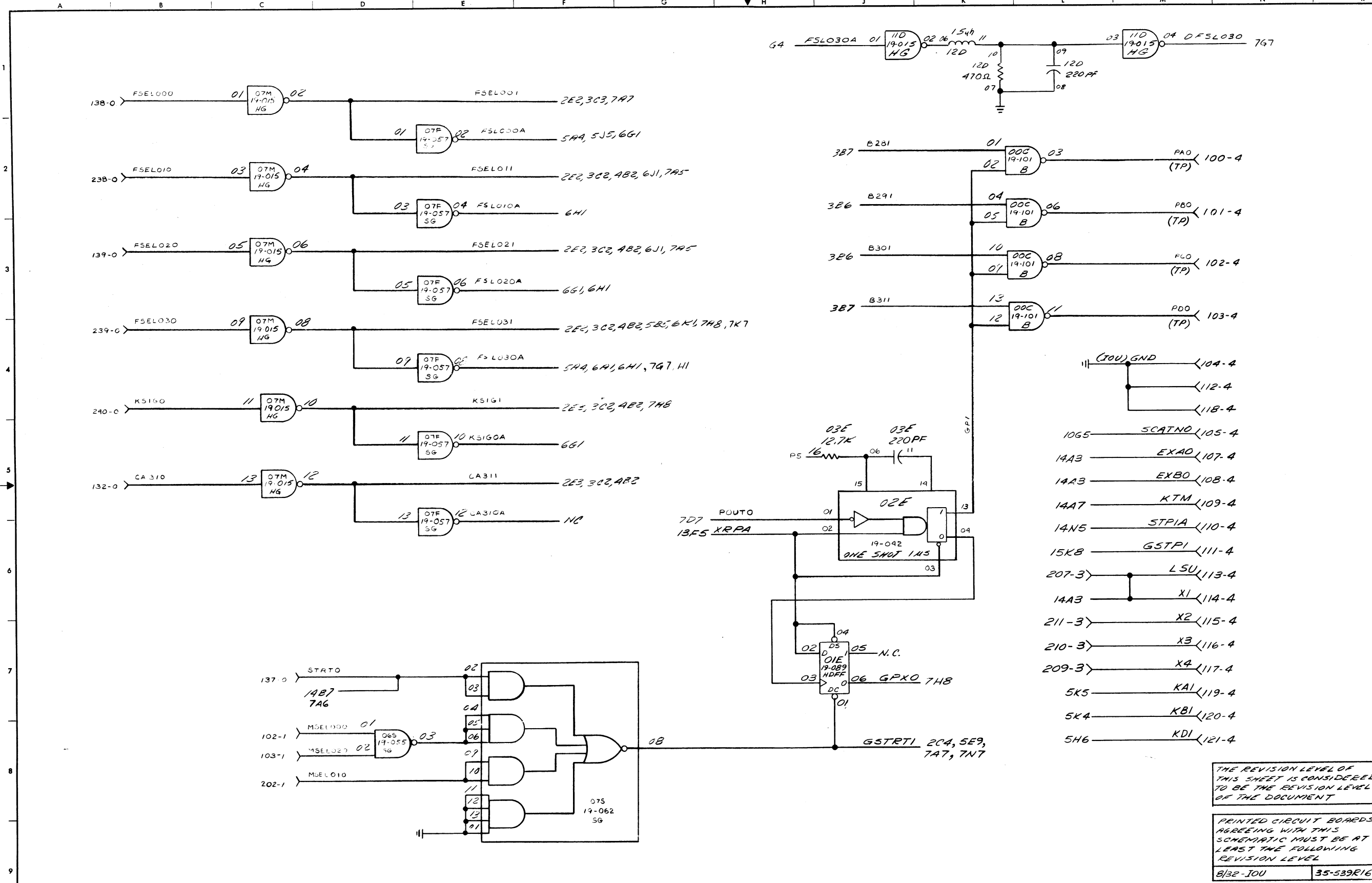
REVISIONS



THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

SCALE-	NAME	TITLE	DATE	TITLE
TOLERANCE:		DRAFT		FUNCTIONAL SCHEMATIC
XXX 1.000		CHK		MB/32 CPC
XX 1.02		ENGR		(DCS OPTION)
X 1.03				TASK NO. 03088
ANGLES 1:10				DWG. NO. 35-555 DOB 9-9
UNLESS OTHERWISE SPECIFIED:				SHEET OF 9-9

BRUNING 44-231 24538



REVISIONS table with columns for PRE PRODUCTION APPROVAL, INT DATE, and various revision notes including 'GATE 05H OUTPUT B, CAP 220PF, RES 470, INDUCTOR MOVED TO SMT 7, DELETED GATE 01K OUTPUT B, GM [S] 03088-18 11/11/74' and 'CHANGED: SMT 16 WAS RO1, SMT 19 WAS RO3, SMT 12 WAS RO1'.

THE REVISION LEVEL OF THIS SHEET IS CONSIDERED TO BE THE REVISION LEVEL OF THE DOCUMENT

PRINTED CIRCUIT BOARDS AGREEING WITH THIS SCHEMATIC MUST BE AT LEAST THE FOLLOWING REVISION LEVEL

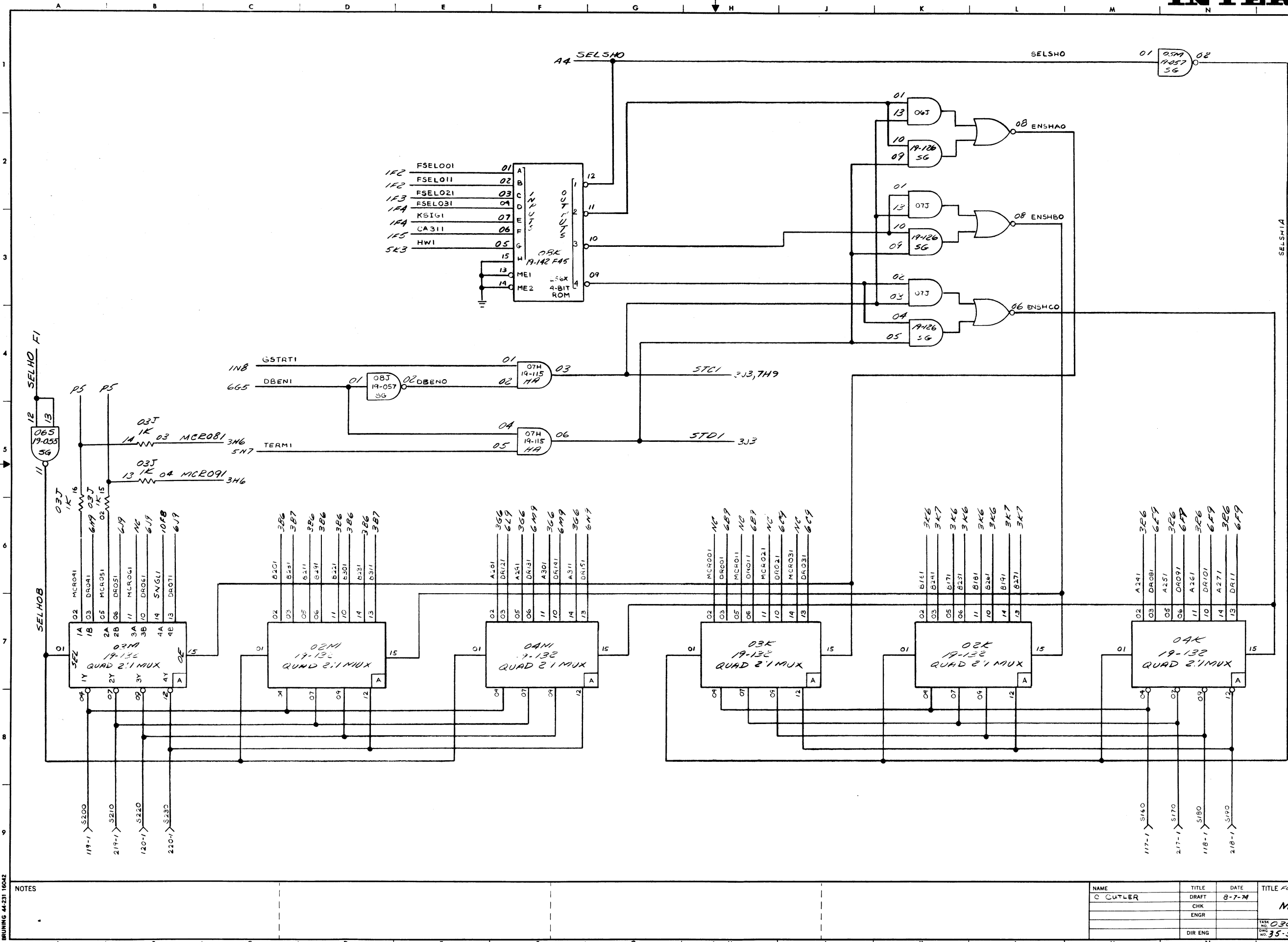
B/32-10U 35-539R16

SHEET INDEX table with columns for REV, SHT. NO, and a grid of numbers 1 through 16.

NOTES section for additional circuit information.

NAME, TITLE, DATE, TITLE FUNCTIONAL SCHEMATIC table with entries for G. J. HOMEFIELD, G. SYROY, S. C. MESSINA, R. A. BARKER, and M8/32-10U.



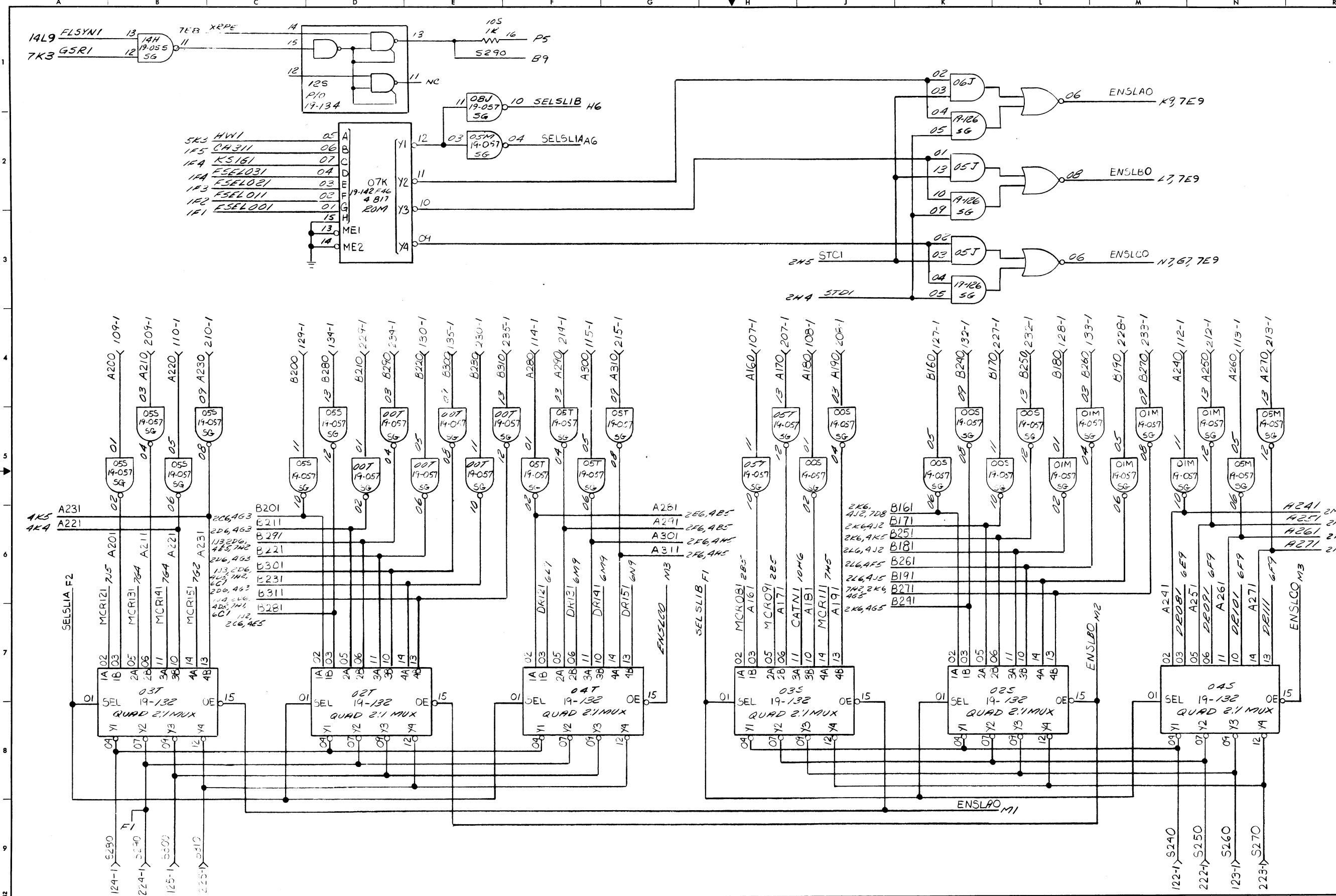


REVISIONS			
ADDED SHEET COORDINATE IN AREA 145			
G	03088	16-12-74	201
M	-18		
ADDED GATE AREA A5 19-055 IN 065.			
M	19-055-32	14-75	202
ROM # WAS 17-084 F78			
G	03088	15-75	203
M	-25		
AREA BA, 8B, 19-132'S INVERSION BUBBLES ON PINS 9, 2, 8, 12 WERE NOT SPEC'D			
M	19-055-32	18-25-75	204

NOTES

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
C. CUTLER	DRAFT	8-7-74	M8/32-100
	CHK		
	ENGR		
	DIR ENG		
PART NO. 03088		SHEET OF 2-16	
REV. NO. 35-539 R04.008			

BRUNING 44-231 1604Z



REVISIONS

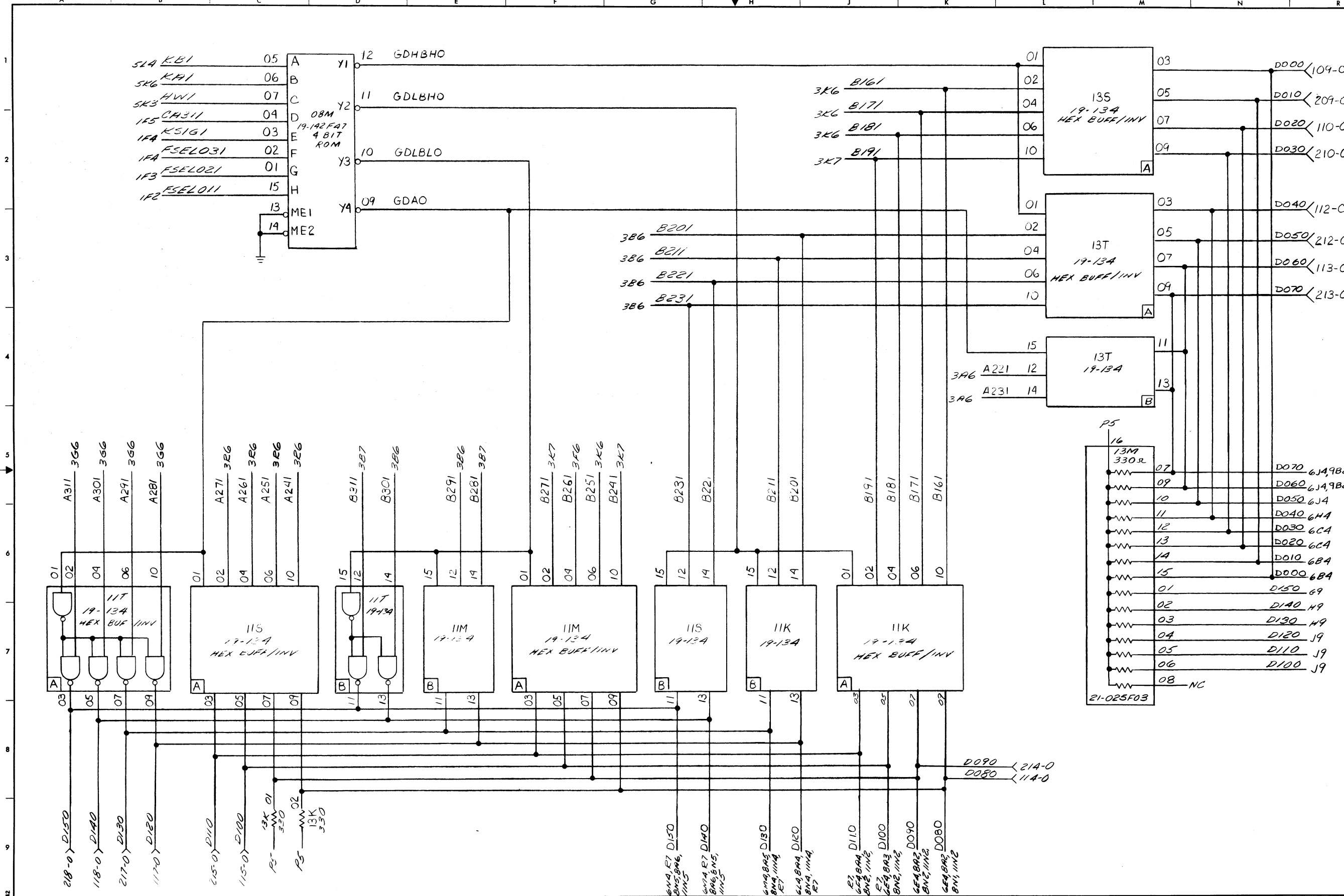
ADDED: NEW SHEET LOCATIONS IN AREA M2	
GM FJS 03088-18	12-14 R01
ADDED: 19-134 TO AREA D1	
GM FJS 03088-29	1-9-74 R02
ADDED: 19-057 GATE OBU, LOC E11	
GM FJS 03088-32	1-14-75 R03
ADDED: GATE 19-055 @ JHT LOC B1, BOARD LOC 14H.	
GM FJS 03088-38	1-27-75 R04
RUM # WAS 19-08FF79	
GM FJS 03088-65	1-15-75 R05

NOTES

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
C. OUTLER	DRAFT	8-12-74	M8/32-IOU
CHK	ENGR		
DIR ENG			
TASK NO. 03088	SHEET OF 3	DWC NO. 35-539 R05 008	

BRUNING 44-231 16042

REVISIONS			
ROM # WAS 17,084 F80			
GM # JG 03088	4/15/75	PO1	

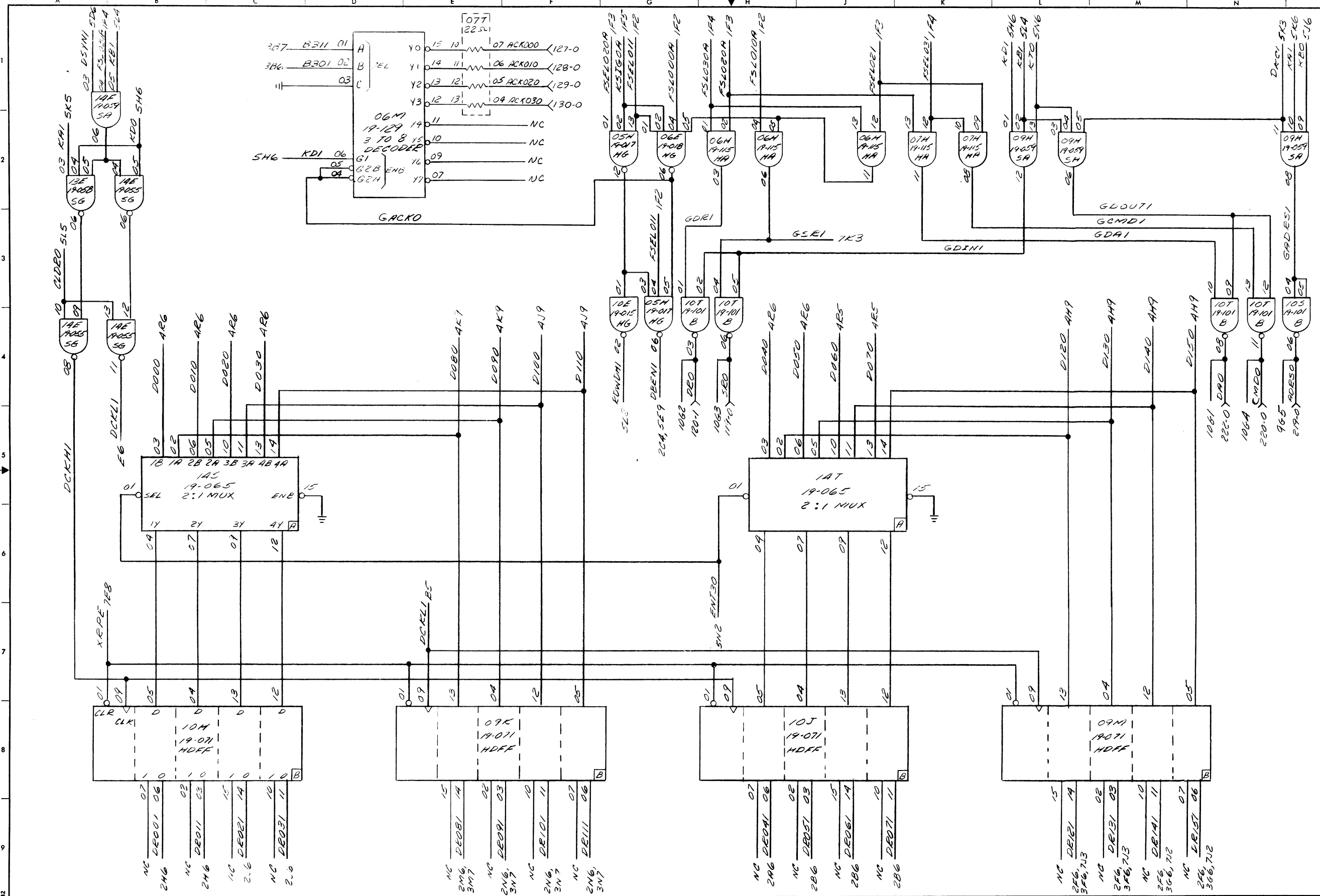


NOTES

NAME	TITLE	DATE	TITLE
C. CUTLER	DRAFT	8-9-74	FUNCTIONAL SCHEMATIC
	CHK		M8/32-10U
	ENGR		
	DIR ENG		

TASK NO. 03088 SHEET OF 4-16
 NO. 35-539 ROM DOB

BRUNING 44-231 16042



REVISIONS

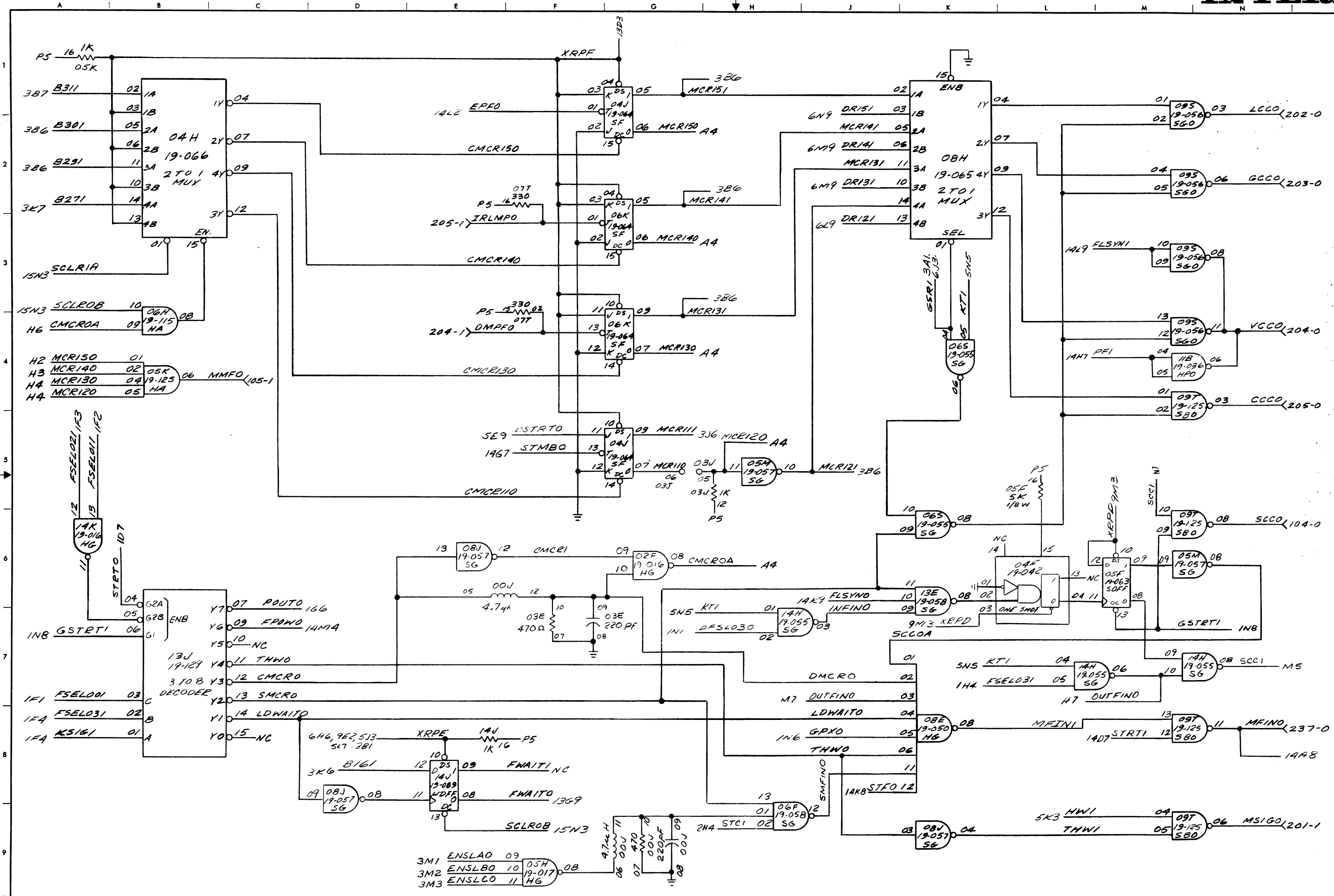
AREA D2, 19-129 WAS SHOWN REVERSED
ALL 25 TO 8-23-75 R01
AREA M1, DAC1 WAS MCI.
CORINSA3 WAS SET.
DATE 2605 9-9-75 R02

BRUNING 44.231 16042

NOTES

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
G. MELTON	DRAFT		M8/32-100
	CHK		
	ENGR		
	DIR ENG		
TASK NO. 03088	SHEET OF 6-16		
DRW NO. 35-539 R02 008			

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTENANCE AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED



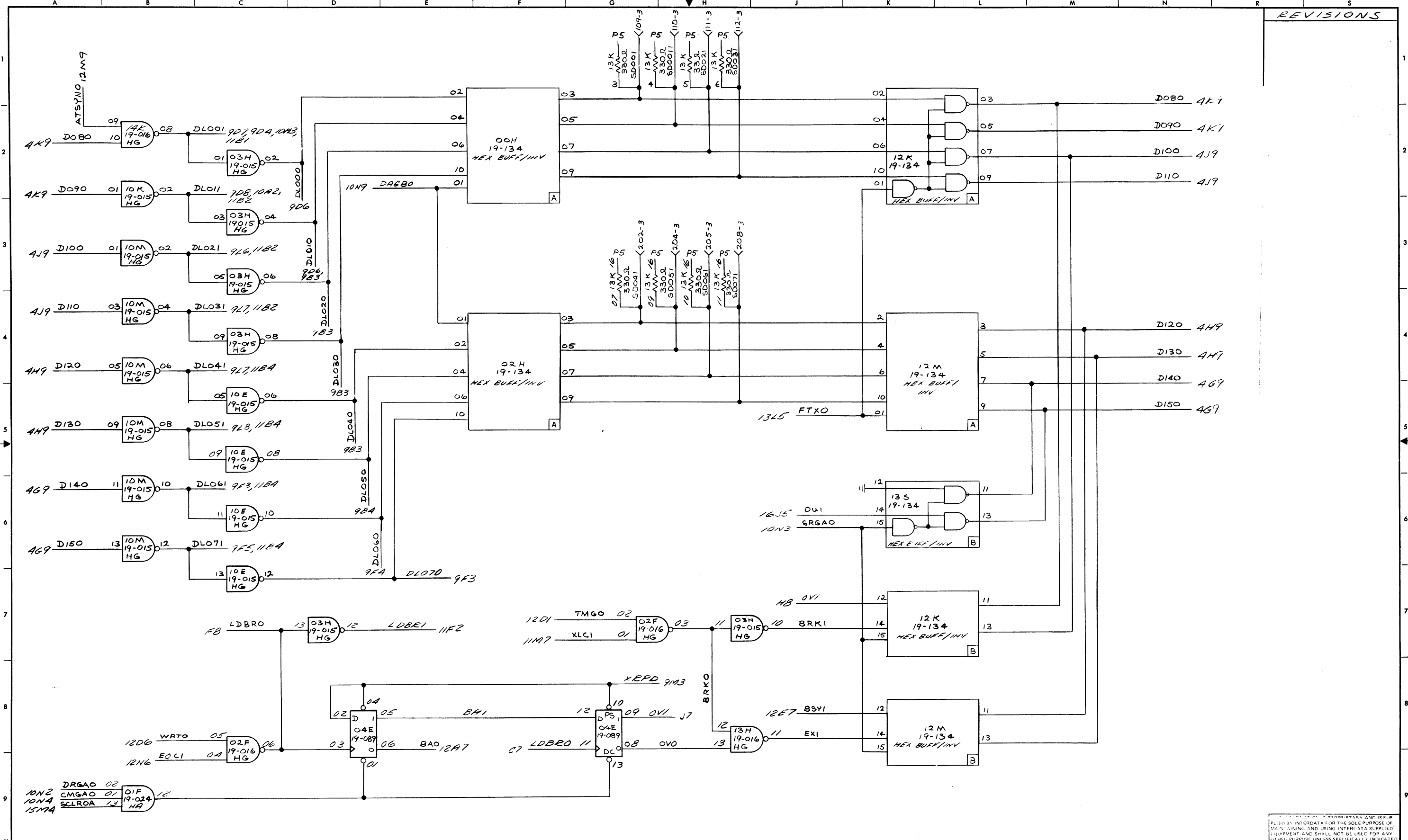
REVISIONS	
ADDED GATE 05H OUTPUT 8, GATE 06F OUTPUT 12, 470 RES., 220PF CAP, 4.7uH INDUCTOR. ADDED MNEMONIC STFO TO J8 AND DSTFO IN PLACE OF XRPF AND STMH0 IN PLACE OF STFO	G 03088 09 M 18-12-74 R01
CHANGED PER MARKED PRINTS	G 03088 09 M 18-12-74 R02
ADDED IC 44H TO SHEET. 09T09 WAS GOING TO 05F09 NOW TO 6STET1 09T10 WAS GOING TO 09T09 NOW TO SCC1. 05F08 NOW GOES TO 14H09, 13E09 IS NOW INFIND WAS KTO. 08E03 WAS GOING TO 08E04 RESISTOR GOING TO 04FIS WAS 10K	G 03088 09 M 18-12-74 R03
09T WAS 19-056	G 03088 09 M 18-12-74 R04
14H02 WENT TO FSELO30A ADDED 19-036 TO LOC. M4	G 03088 09 M 18-12-74 R05
13J04 WENT TO 13J05	G 03088 09 M 18-12-74 R06

NOTES

NAME	TITLE	DATE	TITLE/FUNCTIONAL SCHEMATIC
E. ROE	DRAFT	9-13-74	M8/32-IOU
	CHK		
	ENGR		
	DIR ENG		

TASK NO. 03088 SHEET OF 7-16
Dwg. No. 35-539 R06 008

BRUNING 44-231 1604Z

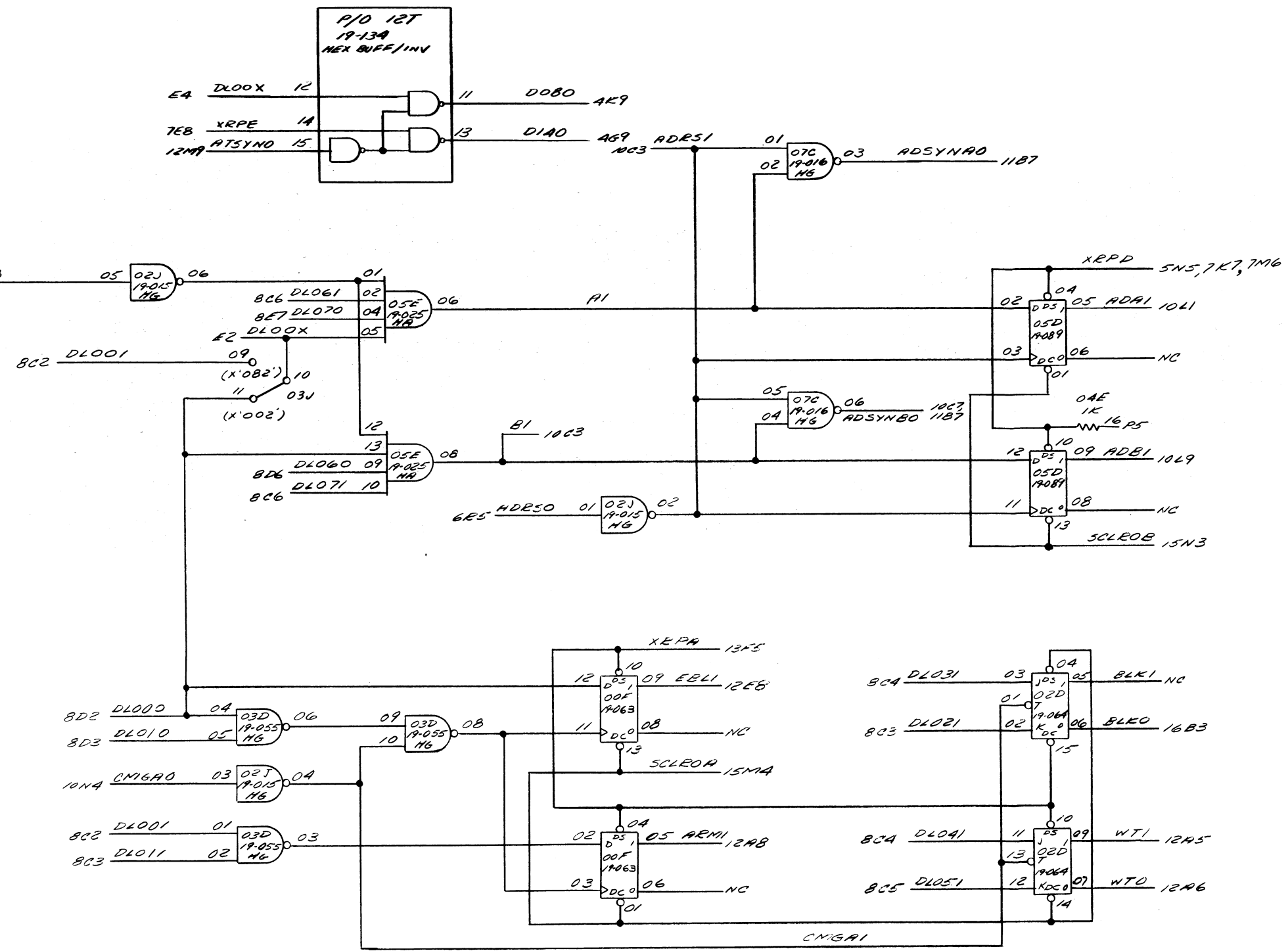


NOTES		NAME	TITLE	DATE	TITLE	FUNCTIONAL SCHEMATIC	
			DRAFT		M8/32-10U		
			CHK		TASK NO. 03088		
			ENGR		DWC NO. 35-539		SHEET OF 8-16
			DIR ENG				

BRUNING 44-231 16042

REVISIONS			
CHANGED PER MARKED PRINTS			
G	03088	18-12-74	R01
M	19		
GATES 03D PIN WAS 19-016 FLIPFLOPS AT 05B#00F PIN WAS 19-089.			
27	0257E	7-10-75	R02

4R6	D060	01
4R6	D070	02
8D3	DLO10	03
8D4	DLO20	04
8D5	DLO30	05
8D5	DLO40	06
		12
8D6	DLO50	11



NOTES

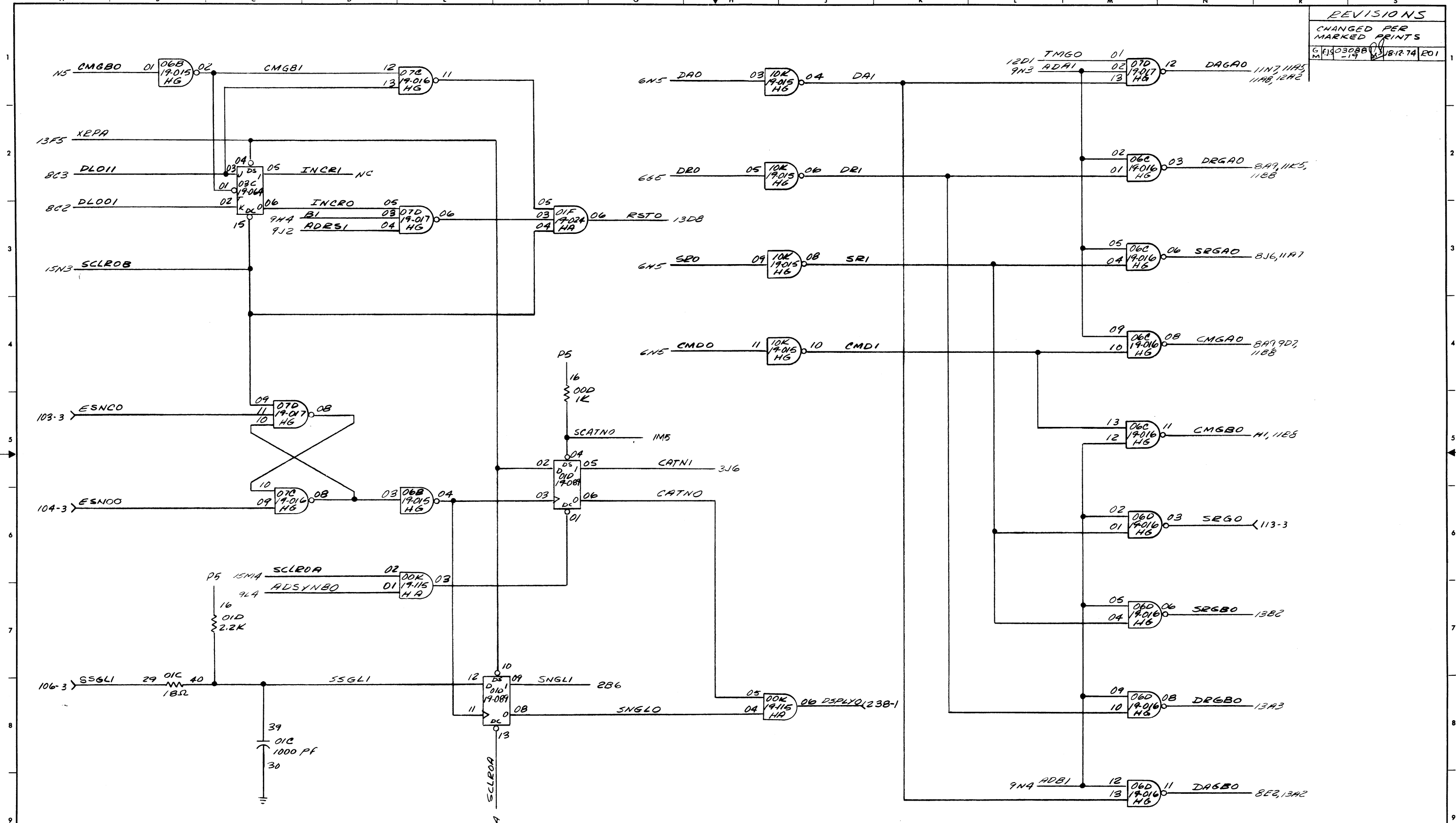
THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF YOUR TRAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

NAME	TITLE	DATE	TITLE FUNCTIONAL SCHEMATIC
	DRAFT		M8/32-100
	CHK		
	ENGR		
	DIR ENG		

TASK NO. 03088	SHEET OF 9-16
DOC NO. 35-539 R02 008	

BRUNING 44-231 1604Z

REVISIONS			
CHANGED PER MARKED PRINTS			
G	16	03088	181274
M		-19	181274
			201

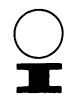


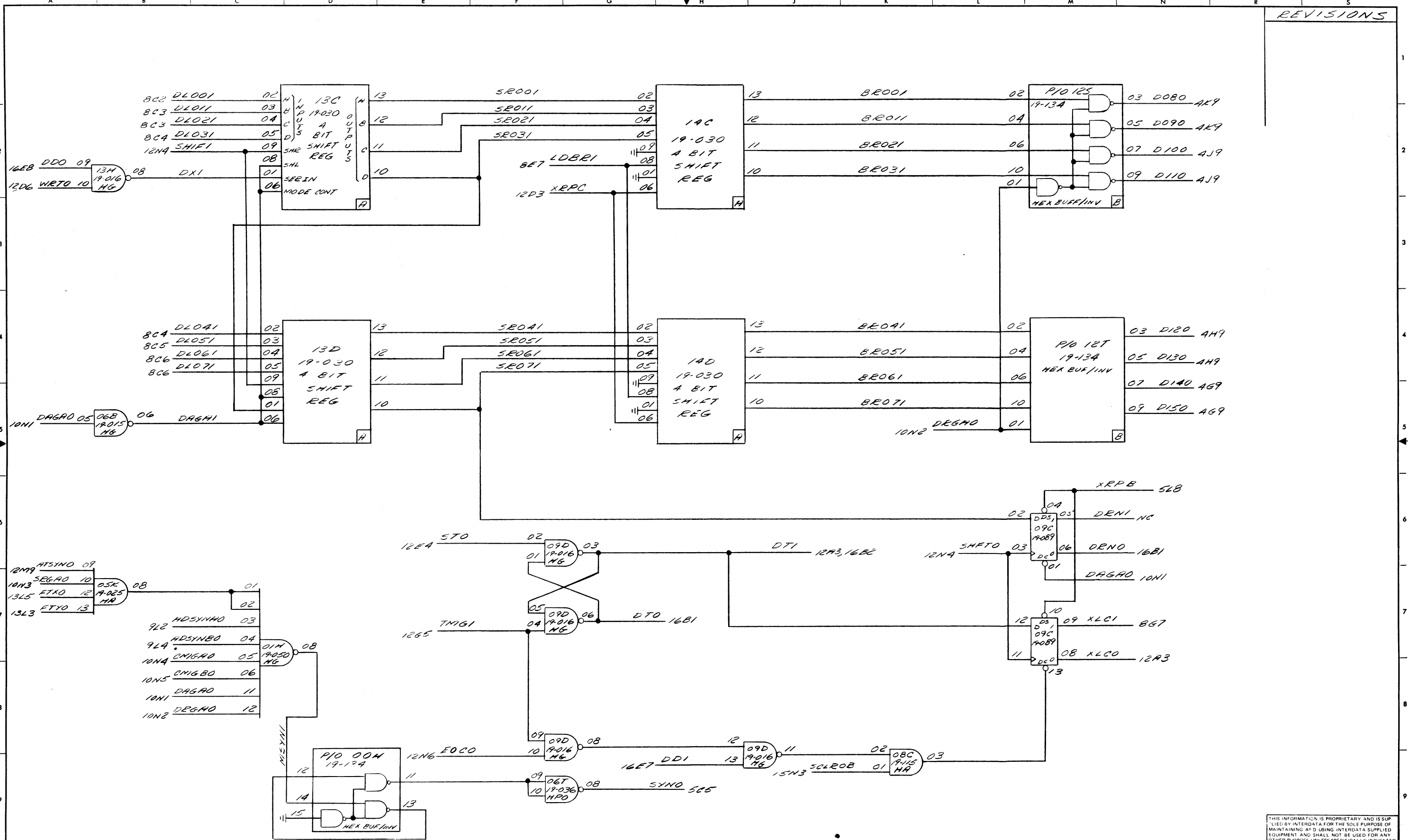
BRUNING 44-231 1604Z

NOTES

NAME	TITLE	DATE	TITLE
	DRAFT		FUNCTIONAL SCHEMATIC
	CHK		M8/32-IOU
	ENGR		
			TASK NO. 03088
			SHEET OF 10-16
	DIR ENG		DWG NO. 35-539 R01 008

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED





NOTES

12M9 A1SYND 09
 10N3 SEGRD 10
 13L5 FTKD 12
 13L3 FTYD 13

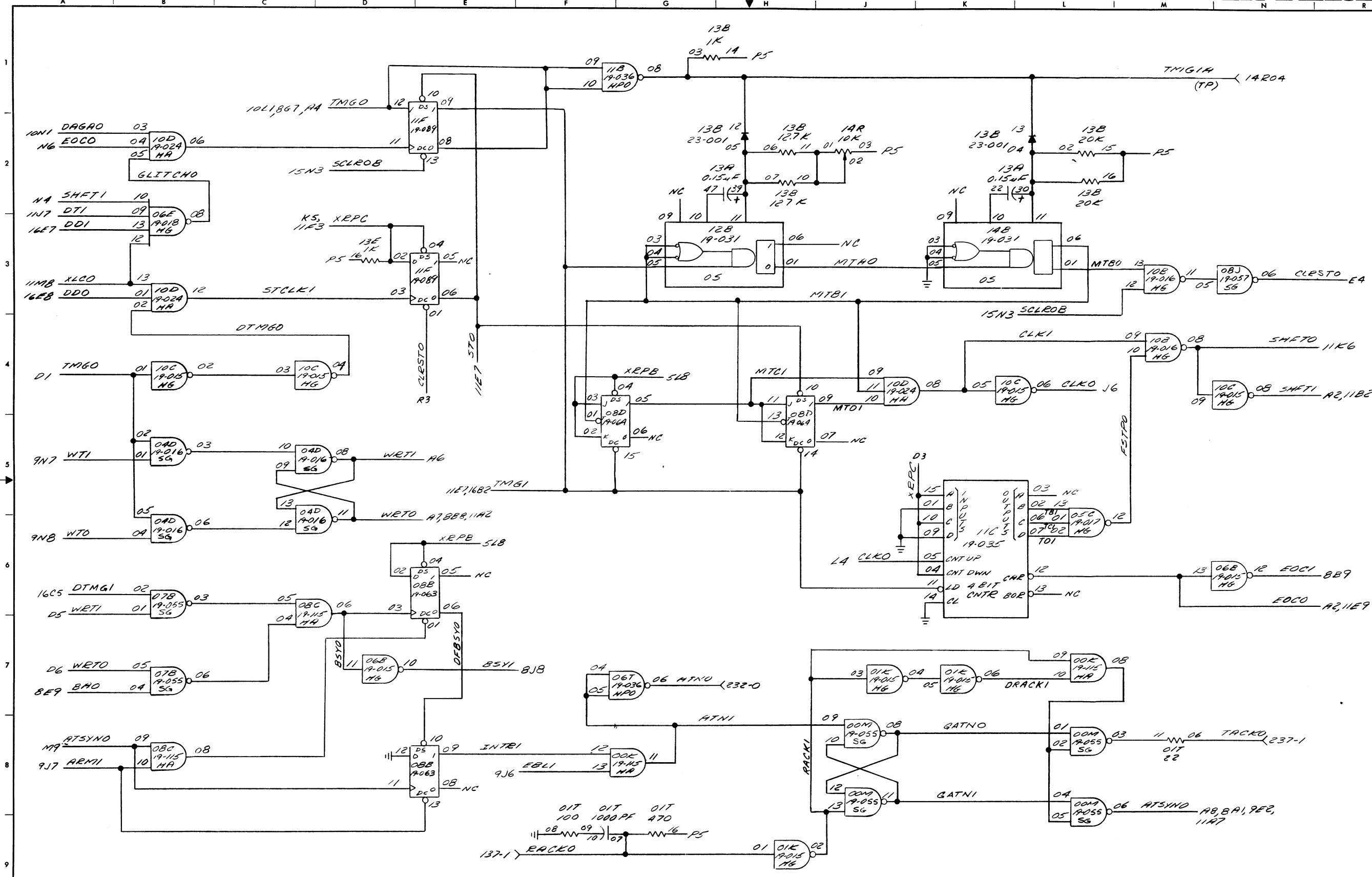
922 ADSYNHD 03
 924 ADSYNBD 04
 10NA CMIGRD 05
 10N5 CMIGBD 06
 10N1 DAGAD 11
 10N2 DEGHD 12

12N6 EOCO
 16E7 DDI
 15N3 SCLROB
 08C 17-115 HA

NAME	TITLE	DATE	TITLE
	DRAFT		FUNCTIONAL SCHEMATIC
	CHK		M8/32-IOU
	ENGR		TASK NO. 03088
	DIR ENG		SHEET OF 11-16
			DWG NO. 35-539 DOB

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED





REVISIONS

CHANGED PER MARKED PRINTS
G.M.J.S. 03088 12-12-74 R01
CHANGED 07802 TO DTMG1, 04D GATES WERE 19-016, 07B GATES WERE 19-016
G.M.J.S. 03088 12-18-75 R02
FLIP-FLOPS AT 08B PIN WAS 19-089 GATES ORD. PIN WAS 19-055
J.A.L.C. 2572 - 7-10-75 R03

NOTES

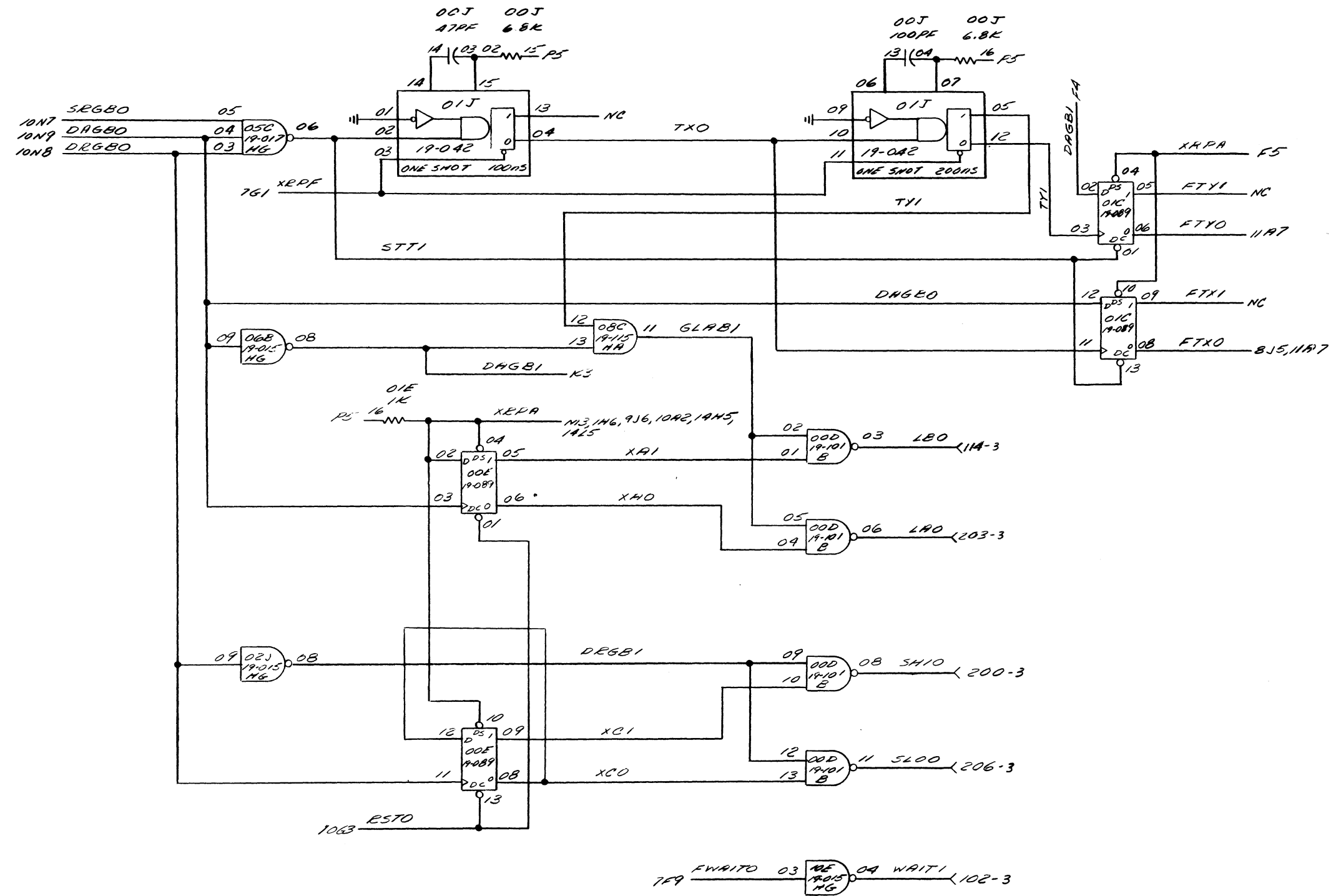
NAME	TITLE	DATE	TITLE
	DRAFT		FUNCTIONAL SCHEMATIC
	CHK		M8/32-IOU
	ENGR		
	DIR ENG		

TASK NO. 03088 SHEET OF 12-16
 DWG NO. 35-539 R03 008

THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

BRUNING 44-231 16042

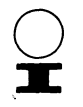
REVISIONS	
00E12	WAS GOING TO 00E10
03088	11/18/75 E02



THIS INFORMATION IS PROPRIETARY AND IS SUPPLIED BY INTERDATA FOR THE SOLE PURPOSE OF MAINTAINING AND USING INTERDATA SUPPLIED EQUIPMENT AND SHALL NOT BE USED FOR ANY OTHER PURPOSE UNLESS SPECIFICALLY INDICATED.

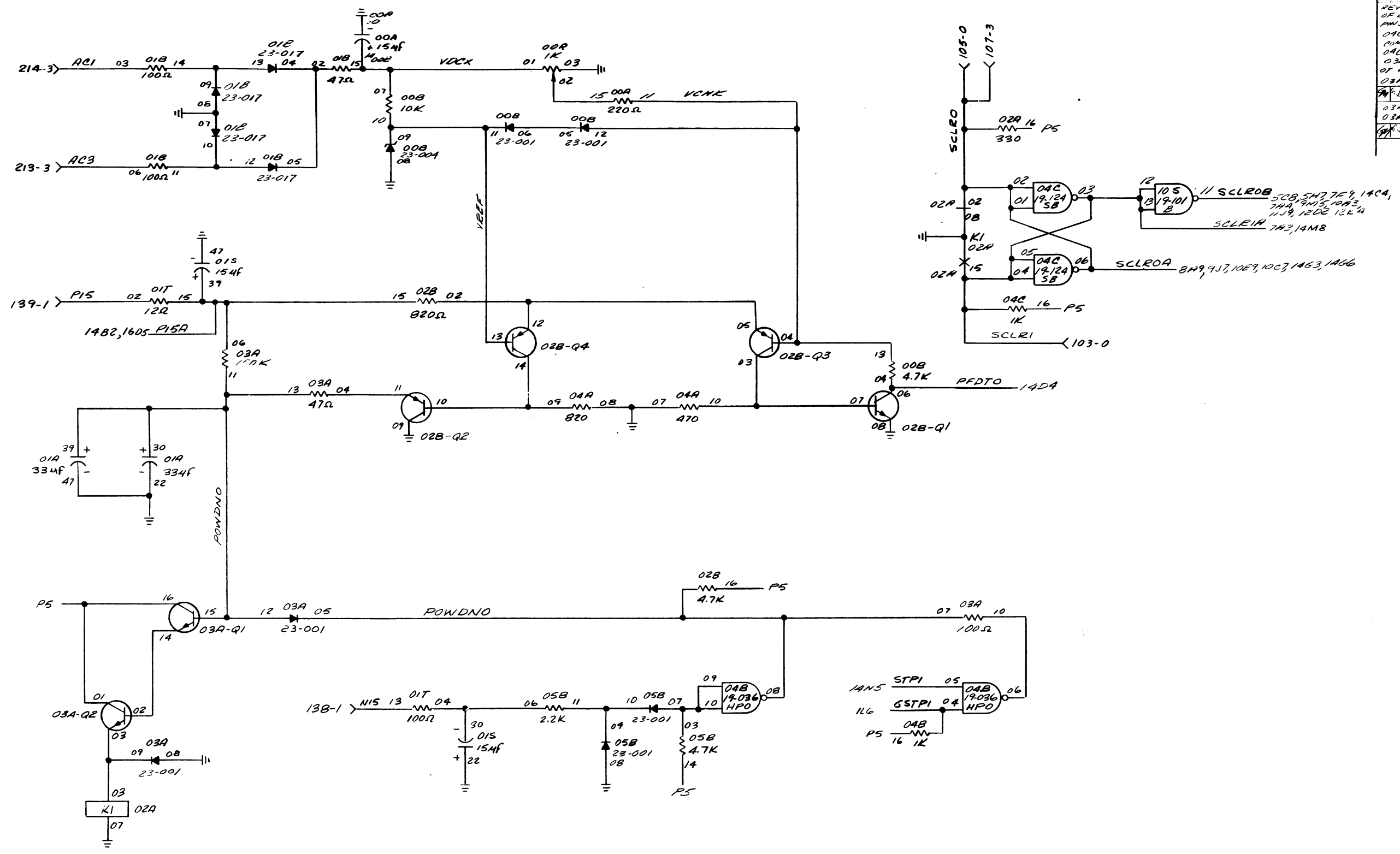
NOTES

NAME	TITLE	DATE	TITLE
			FUNCTIONAL SCHEMATIC
			MB/32-100
			TASK NO. 03088
			SHEET OF 13-16
			DIR ENG



DRAWING 44-231 16042

REVISIONS			
CHANGED PER MARKED PRINTS			
0308B	08	02/17/74	R01
REVERSED THE POLARITY OF DIODES 01B BETWEEN PMS 07, 08, AND 01 & 10			
01C AT LOC 23 WAS 19-016			
PMM, 04C01 & 02, PMM.			
04C05 & 04, RES. AT			
03A07 WAS 470, MADE			
OF 20800 WAS 23-001			
03A06 WAS 470			
0308B	08	04-15-75	R02
03A07 WAS 470 RES			
03A11 WAS 160K RES			
0308B	08	05-27-75	R13



NOTES

NAME	TITLE	DATE	TITLE
	DRAFT		FUNCTIONAL SCHEMATIC
	CHK		MB/32-10U
	ENGR		
	DIR ENG		

TASK NO. 0308B
 SHEET OF 15-16
 Dwg NO. 35-539 RD 3 008

DRAWING 44-231 1604Z

REVISIONS

CHANGED PER MARKED PRINTS			
GM	03088	18-12-74	R01
ADDED ALL OF THE CIRCUITRY COMING OFF OF TMG1			
GM	03088	1-18-75	R02
AREA C3 & B5, HD WAS SHOWN AS 19-057.			
GM	2605	9-2-75	R03

