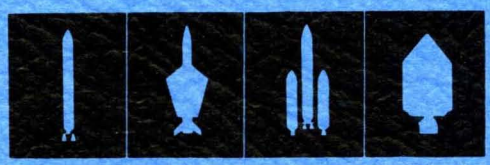
MAGIC III
COMPUTER FAMILY



AC ELECTRONICS DIVISION GENERAL MOTORS CORPORATION MILWAUKEE, WISCONSIN

MASTER NAVIGATORS ... THROUGH TIME AND SPACE

MAGIC III

COMPUTER FAMILY

DATA RIGHTS CLAUSE

The data in this document shall not be disclosed outside Hobbs Associates and the U. S. Government, or be duplicated, used, or disclosed in whole or in part for any purpose other than to evaluate the information, provided that if a contract is awarded to AC Electronics, as a result of or in connection with the submission of such data, Hobbs Associates and the U. S. Government shall have the right to duplicate, use, or disclose this data to the extent provided in the contract. This restriction does not limit the right of Hobbs Associates and the U. S. Government to use information contained in such data if it is obtained from another source.

AC Electronics Division
General Motors Corporation
Milwaukee, Wisconsin 53201

EP-7076 3 February 1967

SECTION I

INTRODUCTION

The Magic III family of computers is a third-generation product of AC Electronics independent research and development program in microelectronic, airborne, general-purpose, digital computers.

- The Magic I computer, developed for ballistic missile guidance and completed in 1962, represented the first computer of its type to utilize micro-integrated circuitry exclusively for all logic functions.
- An advanced version, Magic II, developed for the ASN-47 (LAINS) system, was successfully qualified and extensively flight tested in 1965 and 1966.
- In 1963, AC Electronics recognized a need for a functionally modular family of miniature computers, capable of operating in adverse environments, which would fill the computational requirements of a broad spectrum of applications, including advanced avionics, missiles, and space systems. The Magic III computer family was developed to fill this need. Table 1-1 illustrates the three basic categories of the Magic III family.

The Magic III computer family ranges from a limited-function, simple, serial machine with core memory, capable of performing a typical inertial navigation and display problem, to high-performance parallel machines with fixed memories capable of operating in advanced data processing systems which incorporate optimal Kalman filtering techniques. Computer weights range from a minimum of 5 pounds upwards, depending on available power sources, input/output organization, and storage. The family is compatible in the sense that standardized modules can be interconnected to fabricate a computer having a prespecified memory type and computing capability.

AC Electronics is presently under contract to deliver a large number of computers with 4 different basic designs on 5 different customer-funded programs. Construction of a corresponding family of input/output modules and programming of computer diagnostic self-test programming routines is being pursued concurrently.

Individual computer modules are fabricated using circuit board techniques newly developed by AC Electronics, which eliminate the blind circuit connections inherent in the conventional multilayer circuit board processes. Logic functions are implemented with one custom-designed set of flat-pack micrologic elements. This combination of circuit boards and micrologic results in considerable economy of space as well as high reliability.

These hardware development projects are supported by system and design-aid programming studies. AC Electronics has developed, and is continuing to develop, design aid programs to assist in the design activity. These programs provide an accurate and fast means of accomplishing routine and tedious tasks. Real-time system programs have been developed to handle typical military problems.

PROCESSOR		
SIMPLE (010)	SERIAL (020)	PARALLEL (030)
• Few Instructions • Limited Capability • Speed* 19.5/175.8 • Low Cost	• Many Instructions • Large Capability • Speed* 15.1/121.2 • Moderate Cost	• Many Instructions • Extensive Capability • Speed* 4.5/34.5 • Higher Cost
(311)† • Modular Inertial Navigator • Tactical Missile	(321)† • Ships Navigation • Missile Ground Checkout	(331)† • Bomber Fire Control • Command and Control
MEMORY		
TOROID (001)	WAFFLE IRON (002)	SHMOO (003)
• Fully Writeable • Up to 32,768 Words in 1,024-Word, 2,048-Word, or 4,096-Word Modules	• Permanent • Card Changeable • Up to 32,768 Words in 2,048-Word Modules	• Nondestructive Readout • Electrically Alterable • Up to 32,768 Words in 4,096-Word Modules

* Numbers shown are for add/multiply time in microseconds.
† Toroid memory.

Table 1-1. Magic III Computer Family Characteristics and Applications

SECTION II

COMPUTER PROGRAMS

2.1 DIAGNOSTIC PROGRAMS

Diagnostic software is used to verify proper computer operation. If malfunctions occur, diagnostic type software can often aid in malfunction isolation by providing additional tests for automatic isolation. Consequently, a combination of built-in test circuitry and software testing can yield additional malfunction isolation very effectively.

Programmed diagnostics include a self-test routine which primarily checks the Arithmetic Unit/Instruction Processor Unit, a sum check routine which checks for failures in the program memory, the COP routine which is a summary failure routine, an output reasonableness routine, and special error and recovery routines which are called into operation when errors are sensed. These diagnostic programs are employed with all Magic III family computers.

2.2 ORDER CODE SIMULATOR

The Order Code Simulator (OCS) is an important programming aid that was developed and utilized in the preparation, checkout, and evaluation of programs written for the Magic I, IB, II, and III digital computers. Order code simulation has proven very successful for the coding, checkout, and evaluation of operational programs. The OCS serves as a useful tool because of the relative ease provided for program preparation compared with machine language programming and because of the variety of features that are included for debugging purposes. The OCS has the capability to accept symbolically coded programs as input, translate this input into machine language, and then simulate the response the Magic computer would exhibit upon execution of the program. Figure 2-1 illustrates the various steps in the OCS process.

Many other benefits are realized through the use of an OCS program. These include simplified coding of the operational programs (that is, symbolic coding in contrast to absolute machine language), off-line program checkout and evaluation, and preparation and checkout of programs prior to or in parallel with computer fabrication.

Within the simultating computer (for example, IBM 7090 or 7040), is a block of cells corresponding to the Magic computer memory. The machine language program being

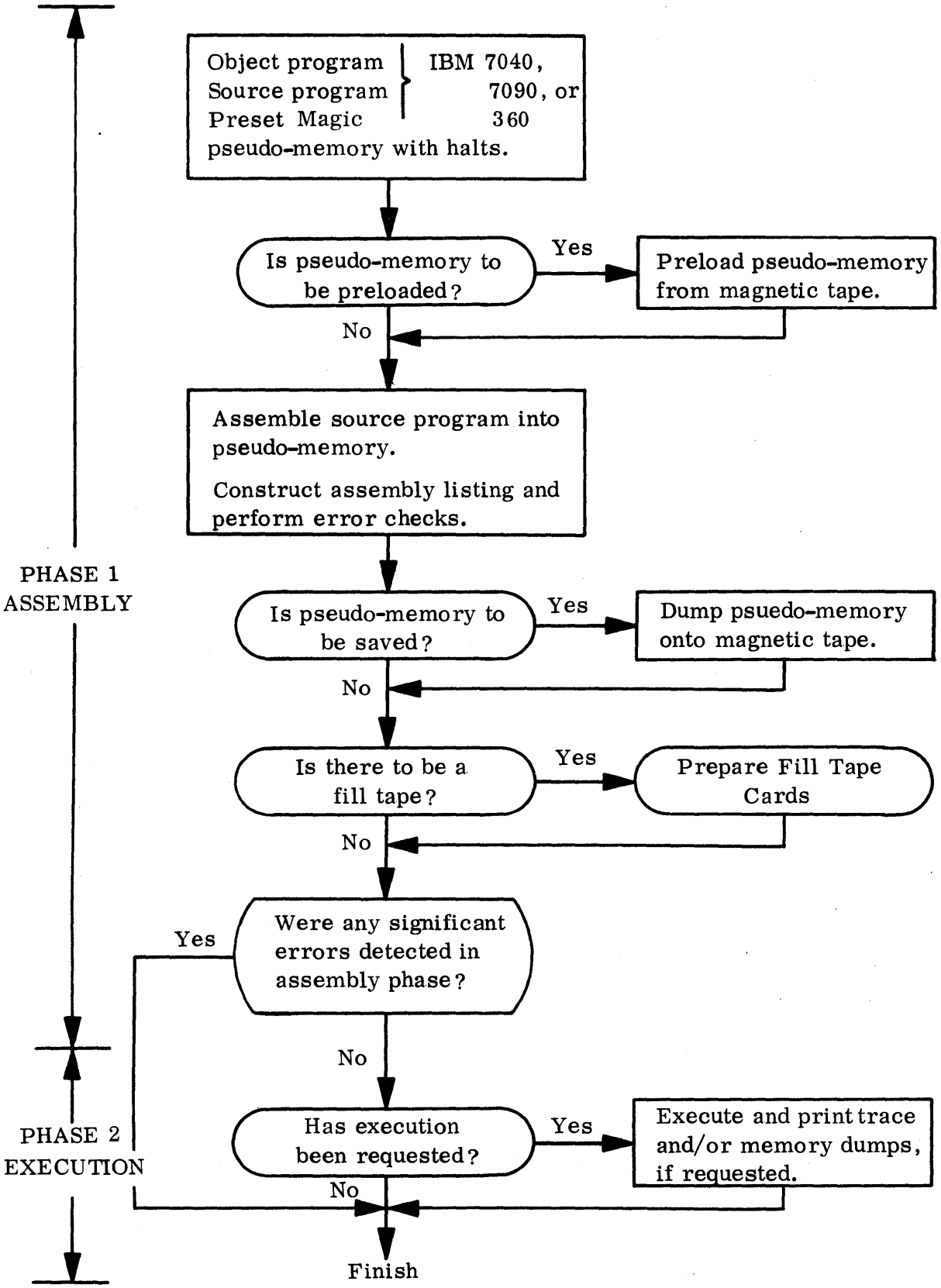


Figure 2-1. Sequence of Steps in the OCS Process

assembled is placed in the simulated memory in the same form (bit configuration) as it will be in the Magic computer memory. The program to be assembled and simulated consists of symbolic representations of instructions and data in combination with pseudo-operations that specify debugging instructions and control functions. These pseudo-operations are nongenerative in nature; that is, they do not transform into equivalent machine language instructions.

Symbolic assembly and simulated program execution are accomplished in two distinct phases of OCS operation as shown in Figure 2-1. The first phase, which provides the assembly process, initially consists of preloading the pseudo-memory with halt instructions; and if the appropriate pseudo-operation is coded, an assembled program previously stored on magnetic tape is written into the simulated memory in Magic machine language. While the program is being assembled, error checks are performed and listings generated including appropriate error messages. These assembly-phase listings include all information contained on the input card deck plus the results of the assembly; that is, the location and instruction in an octal format. Also, a dictionary type equivalence table listing the symbols and their locations is provided.

In the first phase, at the completion of the assembly into machine language, the entire simulated memory can be stored on magnetic tape for future use. If no significant errors are detected in the first phase of OCS operation, the program will proceed to the second phase, which is the simulated program execution.

During the simulated program execution phase of OCS operation, the assembled source program is executed just as it would be done in the Magic computer. In this phase, debugging operations such as program trace or memory dump are executed and associated listings are generated.

SECTION III

MAGIC 311 COMPUTER

3.1 INTRODUCTION

The Magic 311 is a member of the Magic III family of general-purpose digital computers currently being developed at AC Electronics. The Magic 311 computer is a serial, binary, whole-number machine with a toroidal core memory consisting of 6,144 words; each word consists of 12 bits plus 1 parity bit. Data words are 24 bits long, and instruction words are 12 bits long. A comprehensive summary of the computer's characteristics is provided in Table 3-1.

Individual computer modules are fabricated using AC Electronics newly developed circuit board techniques, which eliminate the blind circuit connections inherent in the conventional multilayer circuit board processes. Logic functions have been implemented with readily available flat-pack integrated circuit elements. This combination of circuit boards and flat-packs results in considerable economy of space as well as high reliability.

The choice of memory was based primarily on a desire for low cost and reliable operation over a wide temperature range. The availability of lithium-ferrite cores has permitted operation of the core stacks in the conventional coincident-current mode without taking elaborate precautions to compensate for temperature. Also, the stack design is simplified with a resultant increase in reliability and decrease in cost.

The input/output employs proven solid-state conversion and signal processing techniques. The inputs to these converters are multiplexed electronically under the control of the Central Processing Unit (CPU).

The order code was selected to provide the necessary computational capability along with simple logic design. It was selected on the basis of an instruction utilization analysis of the ASN-47 system program.

3.2 COMPUTER ORGANIZATION

The Magic 311 is conventional in its organization, except that certain hardware economies have been provided at the expense of computer speed. The computer Central Processing Unit consists of the Arithmetic Unit, Instruction Processor Unit, and the Memory Unit.

Computer Type:	Serial, binary, whole number
Data Word:	24 bits (including sign)
Instruction Word:	12 bits
Addressing:	Direct and relative
Memory Type:	Lithium-ferrite toroidal core, coincident-current mode
Memory Size:	6, 144 words (12 bits plus parity)
Speed:	
a. Memory Cycle:	2.6 μs
b. Memory Access:	6.5 μs for 12-bit word
c. Add:	19.5 μs
d. Multiply:	32.5 to 175.8 μs
e. Divide:	332 μs
f. I/O:	19.5 μs
I/O Capability:	
a. Pulse inputs	
b. Digital data inputs and outputs	
c. Discrete signal inputs and outputs	
d. DC inputs and outputs	
e. Synchro inputs and outputs	
f. Self-check inputs	

Table 3-1. Magic 311 Characteristics

3.2.1 CENTRAL PROCESSING UNIT

A block diagram of the Central Processing Unit is shown in Figure 3-1. Instructions of 12 bits are stored in a single memory location, and data words of 24 bits each are located in adjacent memory locations. The memory contains a total of 6,144 words of 12 bits and 1 parity bit.

Memory cycles are initiated by the computer timing counter. A memory cycle is approximately 2.6 microseconds, during which the contents of the memory location specified by the address register is serially shifted 12 clock-times. This procedure is continually repeated. If an instruction does not require an operand memory cycle, the memory cycle will still be executed, but the contents of the readout memory location will not be used.

Computer processing phases are broken into three parts: instruction procure, first half-word processing, and second half-word processing. These three phases are referred to as C0, C1, and C2, respectively. A typical instruction sequence would be to read an instruction from memory into the M register, then shift the M register serially into the operation code register, the address register, and the input/output register. A first half-word memory cycle is executed, and the contents of the memory location are used according to the contents of the operation code register. A second half-word memory cycle completes the processing of the full word. For the long instructions, multiply and divide, phases C1 and C2 are alternately repeated until the instruction is complete.

Two 24-bit registers are used to retain the results of arithmetic operations. The A register has inputs from the arithmetic adder, the B register, the instruction counter, the bias register, and the interface unit. The A register output is sent to the adder, the instruction counter, the M register, the B register, the bias register, and the input/output unit. The B register communicates only with the A register. The bias register is used to alter the operand address so that operands may be stored in any word in memory. The input/output register retains the address portion of the instruction for input/output functions and is used as a counter to time the length of shift, multiply, and divide instructions. It also serially shifts out the address to the instruction counter for instruction modification.

An interrupt feature is incorporated in the computer to force the program to enter a subroutine that performs computations that must be done at a specified rate. Provision has been made for the computer to recognize a single interrupt signal. Upon completion of the present instruction, a special mode will ensue in which the contents of the instruction counter are stored in the least significant half of operand address ONE, and the contents of the bias register are stored in the most significant half of operand address ONE. The program is forced to take the next instruction from

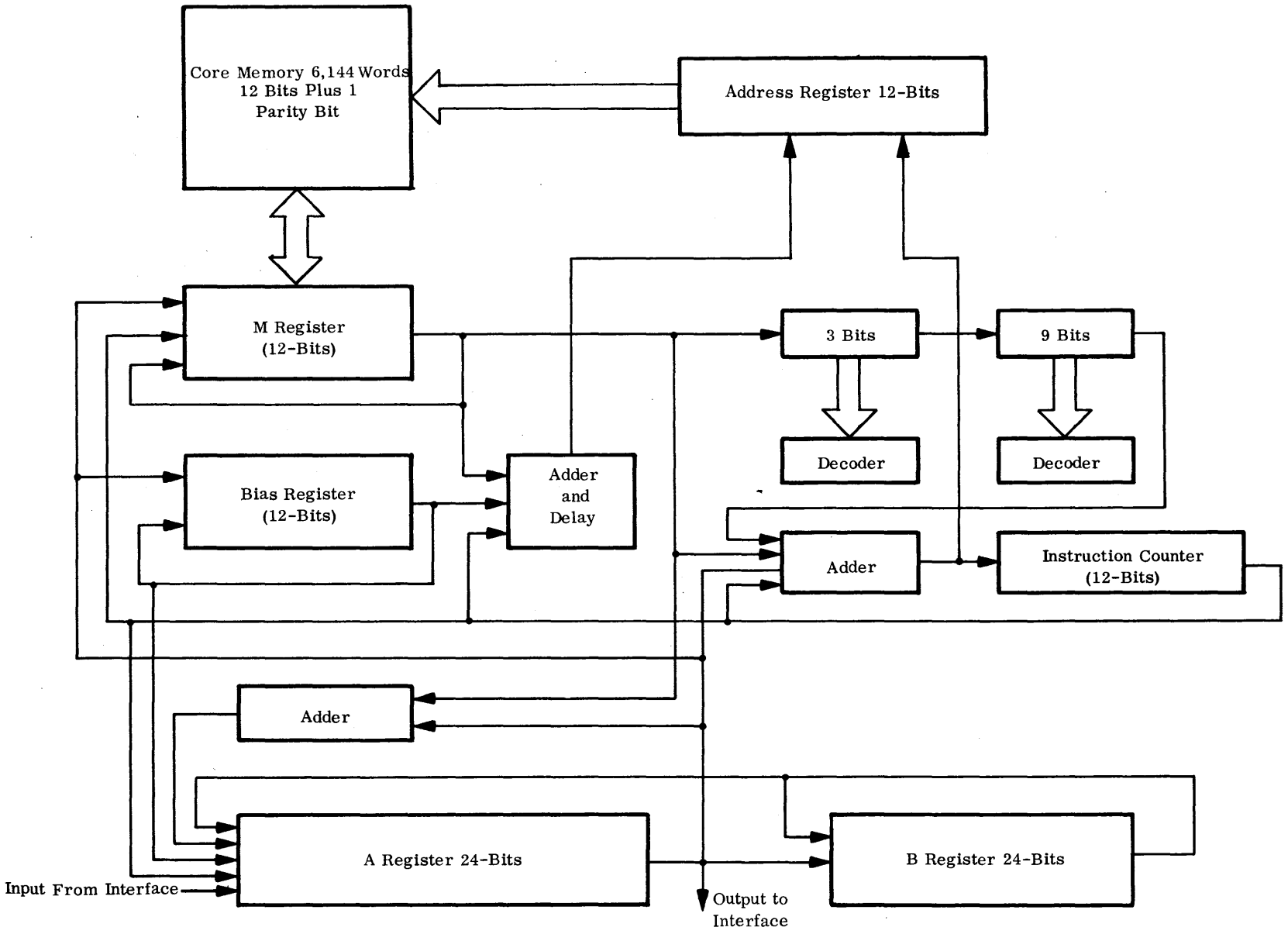


Figure 3-1. Central Processing Unit Block Diagram

memory location 512. It is up to the interrupt subroutine to save the contents of the A and B register and to return them to their prior state before exiting the subroutine. Return to the main program is accomplished by programming an LDA with direct address of ONE. This loads the instruction counter with the contents of the least significant half of operand address ONE and the bias register with the contents of the most significant half of address ONE.

3.2.2 ORDER CODE DESCRIPTION

A summary of the Magic 311 order code is shown in Table 3-2. Mnemonic codes, execution times, and descriptions of the operations are given below. The execution times include access time for one operand and the instruction.

The term "effective address," as used in the following description, designates the actual memory location that is accessed. The address is determined in one of three ways (two for MSK), depending upon the portion of the instruction code called the "basis" field. When bit 9 is a ZERO, the addressing is direct for 256 words in memory; that is, the effective address is that specified by the address field. When bit 9 is a ONE and bit 8 is a ZERO, the instruction address field is treated as a 7-bit two's complement number and added to the contents of the instruction counter to form the effective address. Similarly, if bit 9 is a ONE and bit 8 is a ONE, the address field is added to the contents of the bias register to form the effective address.

The effective address is delayed 1 bit-time as it enters the memory address register; thus, the low-order bit in the memory address register becomes ZERO. To obtain the second half-word of the operand, the low-order bit of the memory address register is complemented to form a ONE.

3.2.3 MEMORY ORGANIZATION

The Magic 311 memory is a conventional coincident-current toroidal core design. The organization of this memory subsystem is depicted in Figure 3-2.

A permanent (wired-in) program is optionally available for Magic 311. The method selected provides this feature by placing cores only in the positions where ONE's are to be stored. No inhibit winding is required in the program portion of the memory. A conventional inhibit winding is provided in the portion of each mat that is used for the scratch pad memory.

MNEMONIC CODE	INSTRUCTION FORMAT													NOTES
	Operation Code Field					Address Field								
	12	11	10	9	8	7	6	5	4	3	2	1		
RTE	0	0	0	0	0	0	Shift Number (47 maximum)							[1] Address 0 — A and B are interchanged. 1 — I C to least significant half of A, most significant half of A is ZEROed 2 — B R to most significant half of A, least significant half of A is ZEROed 3 — Input test word [2] Address 0 — B is ZEROed 1 — Least significant half of A to I C 2 — Most significant half of A to B R 3 — Output test word and test for 1010... [3] Address 0 — Sets memory address bit 13 1 — Round A 2 — Resets memory address bit 13 [4] Address 0 — Least significant half of I C 1 — Most significant half of B R
LRS	0	0	0	0	0	1								
IOC	0	0	0	0	1	0	0	32 Input Words [1]						
						0	1	32 Output Words [2]						
						1	0	32 Discrete Inputs						
						1	1	32 Discrete Outputs [3]						
MSK	0	0	0	1	0	Relative to I C +63								
					1	Relative to B R -64								
TRS	0	0	1	0	0	128-Word Transfer Table								
JOA				0	1	+63, -64								
JOM				1	0									
JOZ				1	1									
STO	0	1	0	Basis Field		0	0	Direct Address 256 Words						
LDA [4]	0	1	1			0	1							
ADD	1	0	0			1	0	Relative to I C + 63, -64						
SUB	1	0	1											
MPY	1	1	0			1	1	Relative to B R + 63, -64						
DIV	1	1	1											

Table 3-2. Order Code Summary

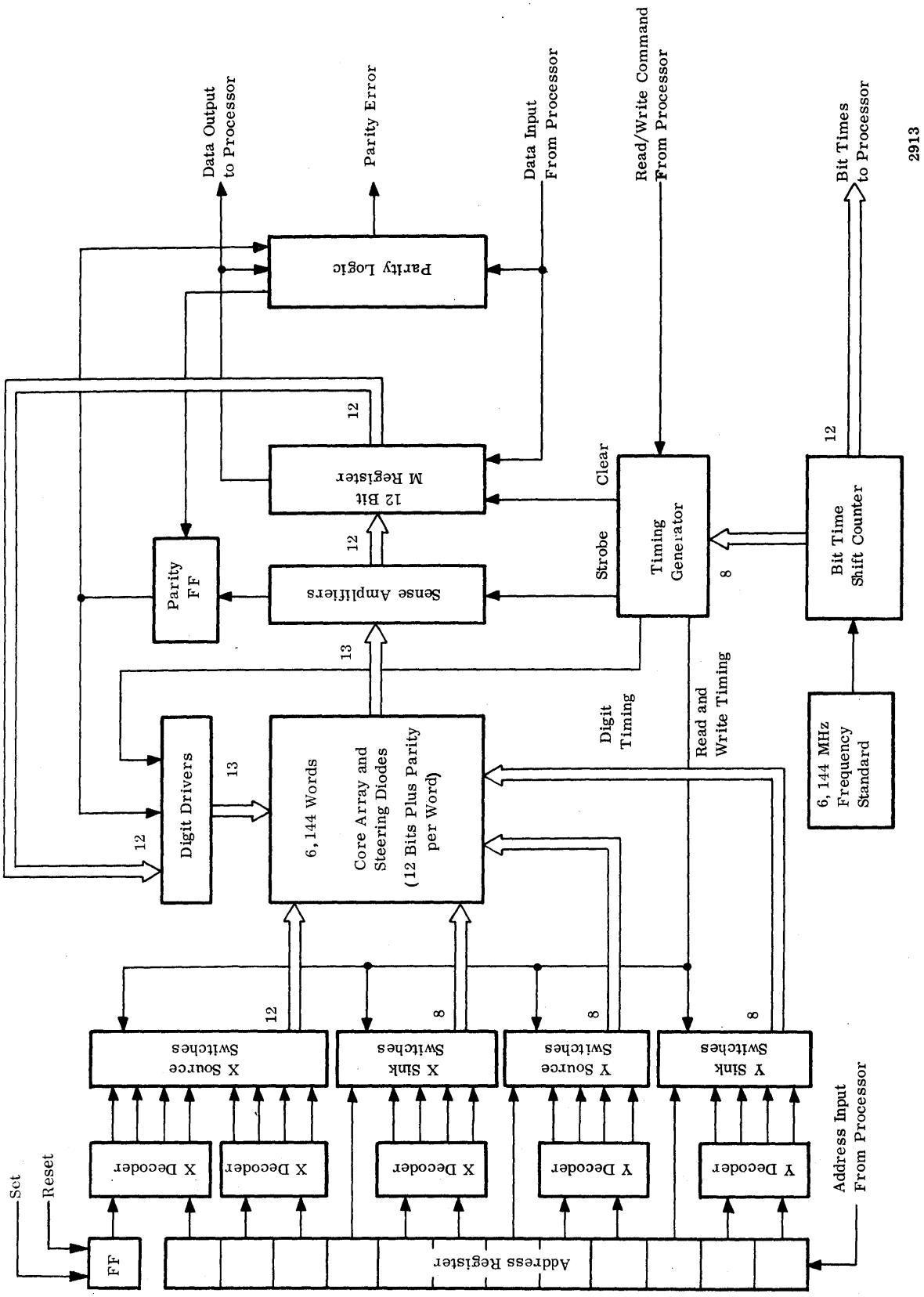


Figure 3-2. Memory Organization for 6,144-Word Memory

The Magic 311 employs a 12-bit word, which represents either one instruction or one-half data word. An additional bit has been added and is used as a parity for every word stored in memory. The logic required to generate parity during a write cycle, and to check parity for a read cycle, is contained within the memory subsystem.

The cycle time of the memory is approximately 2.6 microseconds. The serial transfer rate is 3.072 megahertz, requiring approximately 3.9 microseconds to serially shift a 12-bit word in or out of the M register and a new address into the memory address register. Thus, the complete cycle time is approximately 6.5 microseconds. The basic timing of the memory is derived from a 6,144-megahertz frequency standard. The cycle is broken up into intervals of approximately 0.16 microseconds.

SECTION IV

MAGIC 321 COMPUTER

4.1 INTRODUCTION

The Magic 321 general-purpose machine is a serial, whole-number, programmable computer with a toroidal core memory which is expandable to 32, 768 words. Each data word is 31 bits plus 1 parity bit; there are two 15-bit instructions per data word. A high degree of maintainability has been achieved through the combination of the special mechanical design, the built-in test equipment, and the automatic fault isolation programs. A comprehensive summary of the computer's characteristics is provided in Table 4-1.

4.2 COMPUTER ORGANIZATION

The Magic 321 is organized in four sections: the Instruction Processor Unit, the Arithmetic Unit, the Input/Output Unit, and the Memory Unit. A simplified block diagram illustrating the organization is shown in Figure 4-1. The primary registers and communication paths are shown. For simplicity, the registers and control logic used to assure the proper timing of the computer operations have been omitted. The following is a brief description of the operation. Words are read from the memory and directed to the instruction buffer register in the Instruction Processor Unit for execution. The two instructions per word are executed in sequence. For those instructions requiring operands from memory, the address portion of the instruction is routed through the Instruction Processor Unit adder where it may be modified by one of the three index registers. The modified address, called the effective address, is then sent to the memory address register.

In the case of input/output instructions and shift instructions, the address portion of the instruction is also directed to the control counter input/output address register. Program transfers are executed in one of two ways. The first method uses indirect addressing, whereby the address of the transfer instruction specifies a location in memory, which, in turn, dictates the location from which the next instruction pair will be fetched. A block of locations in memory called the transfer table is reserved for this purpose. The second way in which program transfers can be effected is by modifying the instruction counter with the address field of an instruction. Similarly, the contents of any index register may be modified by the address field of an instruction.

Numbers involved in arithmetic computations are read in parallel from memory and are transferred serially from the M register to the Arithmetic Unit. The results of arithmetic operations can be transferred in parallel from the M register to a hold register in memory, and then placed in memory. For most arithmetic operations, the

Computer Type:	Serial, binary, whole number, programmable
Data Word:	31 bits plus 1 parity bit
Instruction Word:	15 bits; two per data word
Addressing:	Direct, indirect, relative
Memory Type:	Lithium-ferrite, toroidal core; coincident current mode
Memory Size:	Expandable in blocks of 4,096 words up to a total of 32,768 words

Speed:

a.	Memory Cycle:	3 μs
b.	Add:	15.1 μs
c.	Multiply	121.2 μs (average)
d.	Divide:	323.2 μs
e.	I/O:	15.1 μs

I/O Capability:

a. Digital data inputs and outputs
b. Discrete signal inputs and outputs
c. Analog to digital conversion
d. Digital to analog conversion

Cooling: Forced air, conduction, and radiation

Physical*:

a. Weight: 23 pounds
b. Size: 7.5 × 7.6 × 13.5 inches
c. Power: 120 watts

*Physical parameters are for a typical 8,192-word memory.

Table 4-1. Magic 321 Computer Characteristics

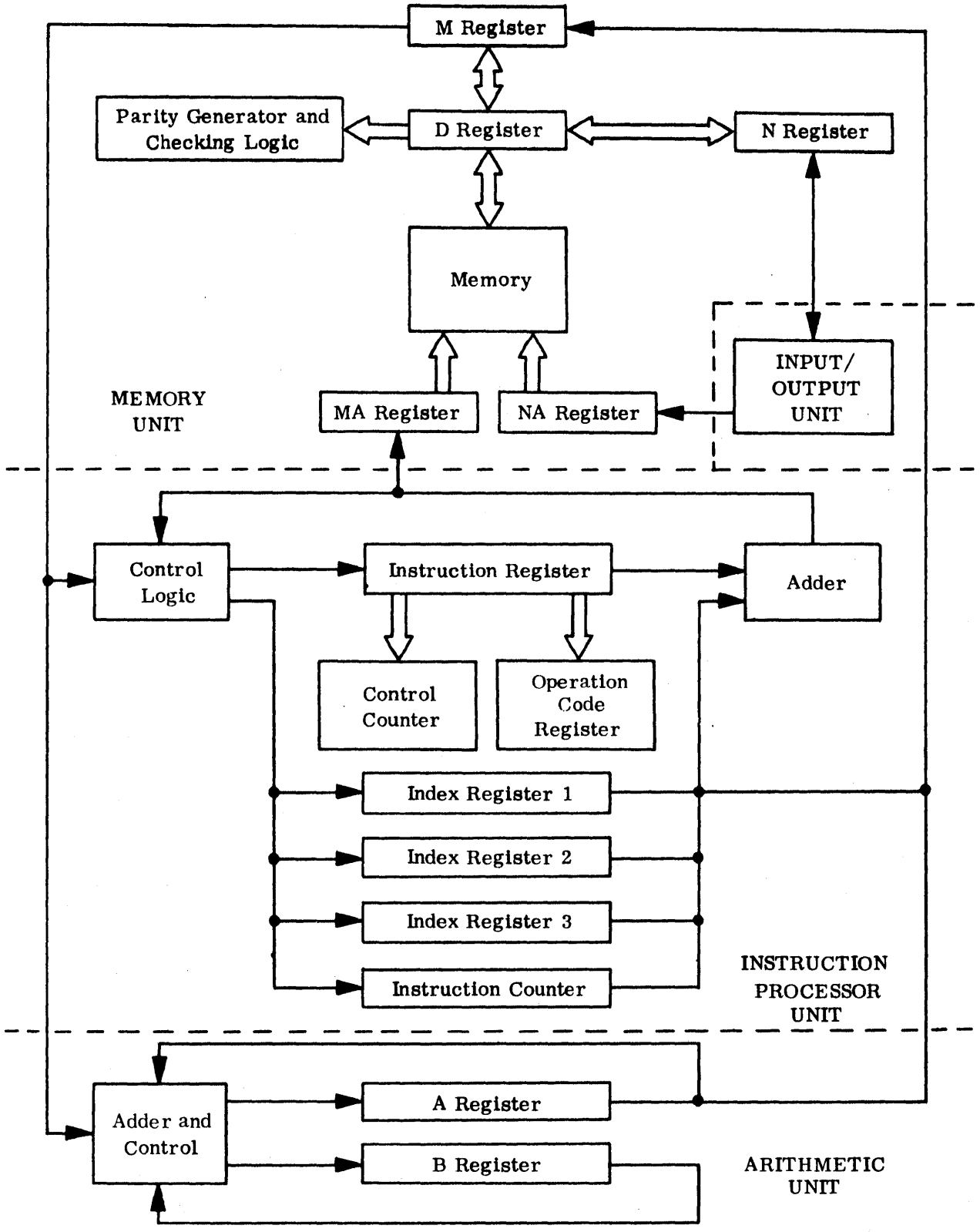


Figure 4-1. Magic 321 Computer Organization

B register is considered to be an extension of the A register. All communication with the other components of the computer and the B register is done via the A register. Normal binary arithmetic is performed with negative numbers being represented in two's complement form.

Input/output operations can be performed under control of the main program, or independent of the main program through the automatic input/output facility. Main-program-controlled input/output operations can be divided into two categories: data communication and discrete communication. For all main-program-controlled communication, the address part of the instruction dictates the source or destination of the data to be transmitted or received. Selection and gating of the appropriate input or output is performed in the Input/Output Unit. The address part of each data input/output command provides addressing capability for 256 quantities. The address part of each discrete input/output command provides addressing capability for 512 quantities. The automatic input/output accesses the memory during the extra memory cycle, which is available between consecutive main-program memory cycles. In this way, the normal program execution can be freed from time dependence on external equipment.

Information rates within the computer are generally at a serial clock rate of 3.072 megahertz. The memory is accessed using a parallel readout technique with a 3-microsecond cycle time. Words are serialized and transmitted to the other parts of the computer at a 3.072-megahertz rate.

The organization of individual sections of the computer is discussed in greater detail in the following paragraphs.

4.2.1 INSTRUCTION PROCESSOR UNIT

The function of the Instruction Processor Unit is to accept instruction and address information from the memory and to process this information to provide command and timing signals for controlling the flow of information within the rest of the computer. The information processed is of two types: instruction and address. Two instructions are contained in each instruction word, which is normally obtained from the memory every third memory cycle. Addresses are obtained from the memory during execution of indirect transfers and index register operations.

In normal operation (no program transfers or interrupts), an instruction pair is obtained from memory, both instructions are executed, and the next instruction pair is obtained from the next sequential memory location. When a transfer-type instruction is executed, the contents of the instruction counter and memory address register are replaced either by the output of the operand address adder, or by the contents of a transfer table word from memory, thereby causing a transfer of control to a new sequence of instructions. When a transfer is made to a subroutine, the program coding must provide for saving the instruction counter contents in memory for a return transfer at the end of the subroutine. Index registers required by the subroutine may also be stored and loaded under program control.

Any instruction may be coded in either the first or second segment of an instruction word, with the exception of INP, which must be first-coded. However, if a first-coded instruction is an unconditional transfer or jump instruction, or a satisfied conditional transfer or jump, the second-coded instruction following the transfer or jump will not be executed.

4.2.2 ARITHMETIC UNIT

The Arithmetic Unit provides the gating, control, and temporary storage required to perform the arithmetic instructions included in the order code. Three 37-bit registers are used. They are designated A, B, and M. A serial adder is the basic arithmetic element.

The A register is the principal arithmetic register and holds the complete results, or the most significant half of the result of all arithmetic operations. The B register holds the least significant half of the result of double precision operations. The M register is used in the multiply and divide operations to recirculate the multiplicand or divisor.

The multiply operation is a variation of the method described by Booth and Booth[1] and assumes operands in two's complement form.

During the first word time of execution, the multiplier is serially transferred from the A register to the B register and the first partial product is entered in the A register. The multiplier is subsequently examined bit-by-bit, least significant bit first, and one of the following operations is performed for each multiplier bit.

1. The multiplicand (in the M register) is added to the partial product (in the A register) and the right-shifted sum replaces the previous partial product. The least significant bit of the sum is placed in the B register, replacing a used multiplier bit.
2. The multiplicand is subtracted from the partial product and the right-shifted difference replaces the previous partial product. The least significant bit of the difference replaces a used multiplier bit.
3. The combined A and B registers are shifted one position to the right.

---

1 A. D. Booth and K. H. V. Booth. Automatic Digital Computers. London: Butterworth Scientific Publications, Ltd., 1956. New York: Academic Press, Inc., Publishers.

The determination of which operation is to be performed is controlled by two multiply control flip-flops that detect the multiplier bits and control the operation of the adder. The add and shift operation is performed when a ONE bit is followed by a ZERO bit if the previous operation was a subtract, and also when an isolated ONE bit (both preceded and followed by a ZERO bit) occurs, if the previous operation was an add. The subtract and shift operation is performed when a ZERO bit is followed by two or more ONE bits if the previous operation was an add, and also when an isolated ZERO bit occurs if the previous operation was a subtract. The shift operation is performed otherwise.

Since an add or a subtract operation requires one word time, while a ONE bit right shift requires only one bit time, the execution time of the multiply command is a function of the configuration of the multiplier. In the worst case (alternating ONE's and ZERO's), 17 word times are required, 16 for additions and 1 for shifts. In the best case (all ZERO's), 1 word time is required for shifting. An average execution time of 12 word times is assumed, including a half word time for instruction access.

In the division process, the quotient is formed at the rate of one bit per word time, starting with the sign bit. An addition or a subtraction of the divisor (in the M register) to or from the partial remainder (in the A register) is performed during each word time, and the left-shifted sum or difference replaces the previous contents of the A register. A comparison of the sign bits of the partial remainder and the divisor at the end of each word time determines both the quotient bit and the operation to be performed during the following word time. The quotient bits are placed in the least significant end of the B register as they are generated, and precessed to the left. The least significant half divisor bits, which spill from the most significant end of the B register, enter as the least significant bit of the A register at the beginning of each word time. At the end of 31 word times, the complete quotient is in the B register and the residue is in the A register. During the 32nd word time, the A and B registers are interchanged so that the quotient will be available to be used as an operand for the next operation.

In shifting operations, the contents of the address part of the instruction are transferred to the control counter in the Instruction Processor Unit immediately prior to execution and subsequently are counted down at a 3.072-megahertz per second rate. During the time that the counter is nonzero, the contents of the combined A and B registers are shifted to the right. After the counter reaches zero, the registers remain static until the following word time. Therefore, the number of bit positions shifted will correspond to the initial contents of the control counter. Left shifts are accomplished by rotation of the combined A and B registers.

To provide error isolation in the Arithmetic Unit, parity checking is done on all shift registers and the adder. For the M register, the parity is checked in parallel. For both the A Register and the B Register, parity is generated at the input to the register (between the input logic and the high-order flip-flop) and held in a flip-flop. When the data is shifted out, a parity detector flip-flop on the output of the register is compared with the previously generated parity bit. For the adder, a parity bit generated from the adder output is compared with the modulo 2 sum of the parity bits of the M register and the A register, and a parity bit generated from the output of the carry flip-flop.

4.2.3 MEMORY ORGANIZATION

The memory selected for the Magic 321 computer is a conventional, coincident-current, toroidal core design. It is designed to be expandable in blocks of 4, 096 words to a total capacity of 32, 768 words. Each word stored in the memory consists of 31 data bits; a 32nd bit is used as parity. The organization of the memory is depicted in Figure 4-2.

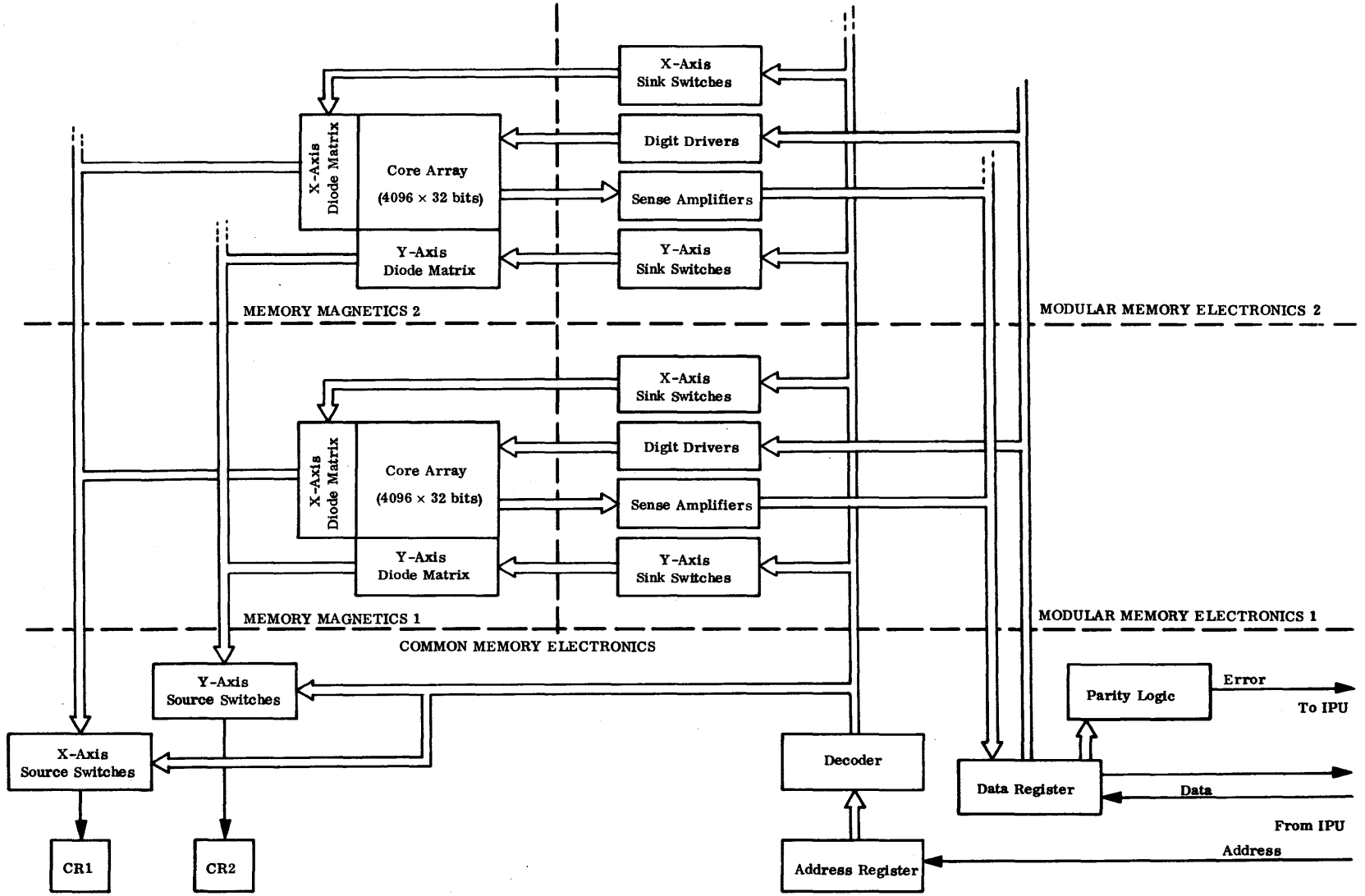
Three basic memory types of assemblies are required. They are the memory magnetics (MM), modular memory electronics (MME), and the common memory electronics (CME). One modular memory electronics assembly and one memory magnetics assembly are required for each 4, 096 words of memory employed, while two common memory electronics assemblies are required per computer.

The large expansion capability of the memory is achieved with a minimum penalty in hardware by incorporating all of the electronics that can be shared in the common memory electronics assemblies. The common memory electronics contains the memory timing, the address register and its decoding logic, the data register, the parity generation and verification logic, and the drive current generators together with the source selection current switches.

Each memory magnetics assembly contains a core memory stack and the drive line selection diodes. The stack contains 32 mats (one for each bit) of 4, 096 cores each. All of the electronics (exclusive of the selection diodes) which are unique to a specific 4, 096-word memory block are incorporated in a modular memory electronics assembly. Each added increment of memory capacity thereby creates automatically a proportionate increase in the nonsharable memory electronics. Included in the modular memory electronic assembly are drive line current sink switches, and the sense electronics and digit driver current switches required for each bit position.

The degree of sharing that is possible is in part determined by the physical size of the modules and in part by the memory device characteristics. The packing density achievable today with miniature memory cores and integrated electronic circuits minimizes problems due to the inductance and shunt capacity of interconnecting leads. The very low capacity of the epitaxial diodes used for steering purposes in the memory array permits many more drive lines to be coupled to a source switch than has heretofore been possible. The recent development of wide-temperature-range, lithium-ferrite memory cores minimizes the effects of temperature gradients, thus relaxing the requirements for precise temperature compensation and permitting the use of common current regulators and/or supply voltages which are compensated on the basis of the mean temperature of the memory arrays.

The computer power supply is specifically designed to protect the contents of the computer memory. In the event of the failure of the power supply, a signal is sent to the memory to disable the current regulators. With the drive currents turned off, it is impossible for the memory contents to be altered.



3201

Figure 4-2. Magic 321 Memory Organization

SECTION V

MAGIC 331 COMPUTER

5.1 INTRODUCTION

The Magic 331 general-purpose machine is a parallel, binary, whole number computer with a toroidal core memory consisting of up to 32,768 words; each word is 31 bits plus 1 parity bit. Data words are 30 bits long plus 1 sign bit. Two 15-bit instructions are stored in each program word. A comprehensive summary of the computer's characteristics is provided in Table 5-1.

5.2 COMPUTER ORGANIZATION

The computer consists of four sections: the Instruction Processor Unit, the Arithmetic Unit, the Input/Output Unit, and the Memory Unit. A simplified block diagram illustrating the organization is shown in Figure 5-1. The primary registers and communication paths are shown. Communication among the various units of the computer is done in a parallel format, 31 bits at a time. For simplicity, several registers and control counters used to assure the proper timing of the computer operations have been omitted. The following is a brief description of the operation.

There are two memory data registers, M and N, that communicate with memory data register, D. The M register provides for the parallel transfer of instructions to the Instruction Processor Unit, and operands to the Arithmetic Unit. The N register provides the communication link between the Input/Output Unit and the Memory Unit and is also used for computer intercommunication.

Each memory data register has a memory address register associated with it. The address register for Arithmetic Unit and Instruction Processor Unit communication provides addressing capability for access to the entire memory. The address register for Input/Output Unit communication provides for accessing only the variable portion of memory.

For Arithmetic Unit and Instruction Processor Unit operation, a basic three-memory-cycle operation is used. An instruction word containing two instructions is read from memory, placed in the memory data register, transferred to the M register, and then transferred to the instruction register in the Instruction Processor Unit during the first memory cycle. The two instructions are executed in sequence during the next two memory cycles. For those instructions requiring operands from memory, the address portion of the instruction is routed to the Instruction Processor Unit adder where it may be modified by one of the three index registers. The modified address, called the "effective" address, is then sent to the memory address register.

Computer Type:	Program controlled, binary, whole number, parallel
Data Word:	30 bits plus 1 parity and 1 sign bit
Instruction Word:	Two 15-bit instructions per data word
Addressing:	Direct, indirect, relative; 3 index registers
Memory Type:	Lithium-ferrite, four-wire, coincident-current, toroidal core
Memory Size:	Expandable in blocks of 4,096 words to a total capacity of 32,768 words

Speed:

a.	Clock Rate:	1 MHz
b.	Memory Cycle:	3 μs
c.	Add:	4.5 μs
d.	Multiply:	34.5 μs
e.	Divide:	94.5 μs
f.	I/O:	4.5 μs

I/O Capability:

The I/O is automatic and separate from the Arithmetic Unit. It provides for contact-closure discretes and serial or parallel digital data transfer. A full complement of analog to digital and digital to analog and ac/dc synchro and resolver inputs and output are optionally available.

Cooling: Forced air and conduction

Physical*:

a.	Weight:	23 pounds
b.	Size:	7.5 × 7.63 × 10 inches (0.35 ft^3)
c.	Power:	115 watts

*Physical parameters are for a typical 8,192 word memory and digital I/O packaged in a 3/4 ATR case measuring 7.5 × 7.63 × 10 inches.

Table 5-1. Magic 331 Computer Characteristics

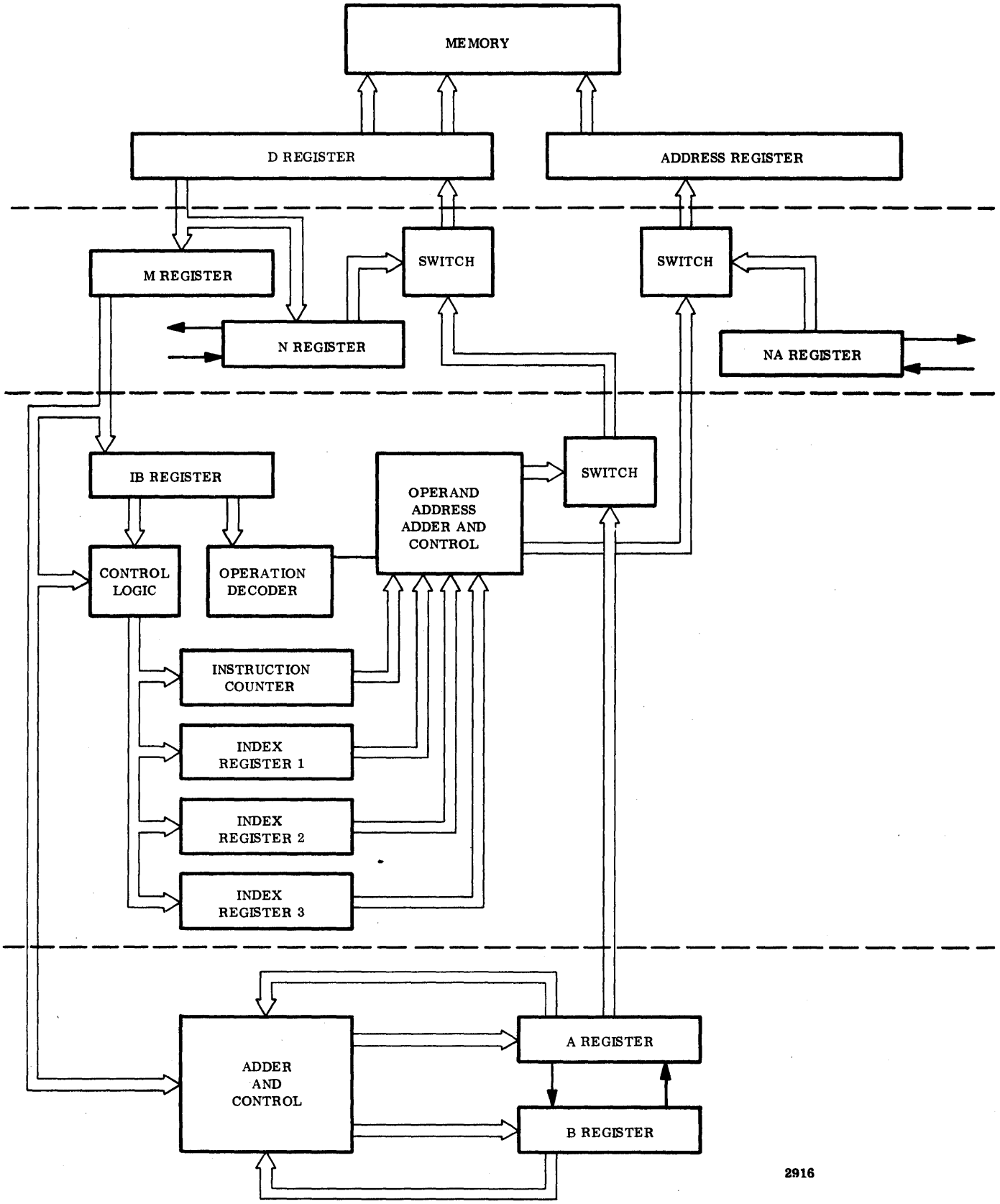


Figure 5-1. Computer Organization

In the case of some input/output instructions and the shift instructions, the address portion of the instruction is also directed to the control counter input/output address register. Program transfers are executed in one of two ways. The first method uses indirect addressing, whereby the address of the transfer instruction specifies a location in memory, which, in turn, dictates the location from which the next instruction pair will be fetched. A block of locations in memory, called the transfer table, is reserved for this purpose. The second way in which program transfers can be effected is by modifying the instruction counter directly with the instruction address field. Similarly, the contents of any index register may be modified by the address of field of certain instructions.

Memory data required for arithmetic computations is transferred in parallel from the Memory Unit via the M register to the A register in the Arithmetic Unit. When the results of arithmetic computations are to be stored in memory, the data is transferred in parallel from the A registers to the Memory Unit. Normal binary arithmetic is performed with negative numbers represented in two's complement form.

Information is generally transferred within the Instruction Processor Unit, Arithmetic Unit, and Input/Output Unit in parallel at a clock rate of 1 megahertz. The memory is accessed using a parallel readout technique with a 3-microsecond cycle time. Information transfers to and from other subsystems are accomplished serially at a 384 or 768 kilohertz rate.

5.2.1 INSTRUCTION PROCESSOR UNIT

The function of the Instruction Processor Unit is to accept instruction and address information from the memory and to process this information to provide command and timing signals for controlling the flow of information within the rest of the computer.

The information processed is of two types: instruction and address. Two instructions are contained in each instruction word, which is normally obtained from the memory every third memory cycle. Addresses are obtained from the memory during execution of indirect transfers and certain index register operations.

Normal operation (no program transfers, interrupts, or automatic input/output) of the Instruction Processor Unit is governed by the operation cycle counter which has three states, C0, C1, C2. During C0, an instruction pair is obtained from memory, during C1, the first instruction is executed, and during C2, the second instruction is executed. During the following C0, the next instruction pair is obtained from the next sequential memory location. When a transfer-type instruction is executed, the contents of the instruction counter and memory address register are replaced either by the output of the operand address adder, or by the contents of a transfer table word from memory,

thereby causing a transfer of control to a new sequence of instructions. When a transfer is made to a subroutine, the program coding must provide for saving the instruction counter contents in memory for a return transfer at the end of the subroutine. Index registers required by the subroutine must also be stored and loaded under program control.

Any instruction may be coded in either the first or second segment of an instruction word, with the exception of INP, which must be first coded. However, if a first-coded instruction is an unconditional transfer or jump instruction, or a satisfied conditional transfer or jump, the second-coded instruction will not be executed.

A fourth state, C3, of the operation cycle counter occurs upon request of the Input/Output Unit. During C3, a word is either read from memory or written into memory. Information is transferred in parallel, via the memory data register, either to or from the N register.

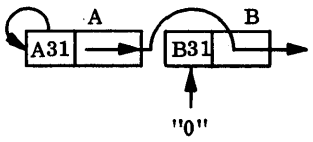
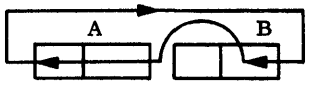
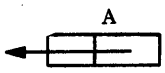
C3 occurs after C2 and is followed by C0. The duration of C3 is one memory cycle time or 3 microseconds. The automatic input/output produces one of two Instruction Processor Unit signals. The first signal requests a memory access during C1 or C2, the second requests a C3. If an excessive amount of time elapses with no available C1 or C2 memory cycle, the Input/Output Unit requests a C3. Since the computer typically spends half the time performing multiplications, and since the input/output memory cycle need not occur at precisely timed intervals, the use of C3 will be rare.

Table 5-2 provides a convenient summary of the Magic 331 order codes, the execution times which are required, and the operations which are performed. Table 5-3 provides a summary of the detailed instruction format including the binary order codes and the comments regarding the use of the address fields. Table 5-4 lists the address allocations for the basic memory module.

As shown in Table 5-4, certain memory locations are reserved for interrupt control. The interrupt subroutines start in locations 5,000 to 5,004. Information necessary for the operation of the recovery subroutines is stored in the fault register. This register is available to the program through the INP command. The flip-flops contained in this register are delineated in Table 5-5.

5.2.2 ARITHMETIC UNIT

The Arithmetic Unit provides the capability to perform additions, subtractions, multiplications, division, left shifts, and right shifts, and also provides communication with the Input/Output Unit. Three 31-bit registers, including the M register shown in the memory section, are required. The information exchange among these registers and the Input/Output Unit is done in parallel. During the multiply operation, the M register holds the multiplicand, and during division it holds the divisor. During addition and subtraction it holds the augend and the subtrahend.

CATEGORY	OPERATION	TIME (μs)	CODE	DESCRIPTION
Arithmetic	Add	4.5	ADD	(A) + (Z) → (A)
	Round	4.5	RND	(A) + (B30) → (A)
	Subtract	4.5	SUB	(A) - (Z) → (A)
	Multiply	34.5	MPY	(A) × (Z) → (AB)
	Divide	94.5	DIV	(AB) ÷ (Z) → (A)
Data Transfer, Logical, and Shifting	Load A from Memory	4.5	LDA	(Z) → (A)
	Store A into Memory	4.5	STO	(A) → (Z)
	Exchange A and B	4.5	XAB	(A) ⟷ (B)
	Mask A from Memory	4.5	MSK	(A) ⊗ (Z) → (A)
	Long Right Shift	4.5 to 64.5	LRS	
	Long Left Shift	4.5 to 64.5	LLS	
	Shift Left A	4.5 to 64.5	SLA	
Transfer and Index	Unconditional Transfer	4.5	TRA	(Z) → (IC)
	Transfer on Minus A	4.5	TRM	(Z) → (IC) if A31 = 1
	Jump on I less than Memory	4.5	JOI	IA + (IC) → (IC) if (Z) > (I)
	Modify I from Address	4.5	MIA	IA + (I) → (I)
	Modify I on Minus A	4.5	MIM	IA + (I) → (I) if A31 = 1
	Load I from Memory	4.5	LDI	(Z) → (I)
	Store I into Memory	4.5	STI	(I) → (Z) or 1 + (IC) → (Z)
Input/Output	Input	15.0	INP	Addressed Input → (A)
	Output	4.5	OUT	(A) → Addressed Output
	Discrete Input	4.5	DSI	Signal → (A31)
	Discrete Output	4.5	DSO	Signal → Interface

A = A Register
A31 = High order position of A Register (Sign)
B = B Register
I = Instruction Counter or Index Register
IA = Instruction Address Field
IC = Instruction Counter
(X) = Contents of X
Z = Effective Address
⊗ = Logical Product
→ = Replaces

Table 5-2. Magic 331 Order Code Summary 2923

MNEMONIC CODE	INSTRUCTION FORMAT																REMARKS
	OPERATION CODE FIELD							ADDRESS FIELD									
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
INP OUT	0 0	0 0	0 0	0 0	0 0	0 0	0 1	.	.	.	.	.	.	.	.		Address used for input/output address.
SLA	0	0	0	0	0	1	0	.	.	.	.	.	.	.	.		Address used for shift control.
RND XAB	0 0	0 0	0 0	0 0	0 0	1 1	1 1	0 1	.	.	.	.	.	.	.		Address not used.
LRS LLS	0 0	0 0	0 0	0 0	1 1	0 1		.	.	.	.	.	.	.	.		Address used for shift control.
DSI DSO	0 0	0 0	0 0	1 1	0 1			.	.	.	.	.	.	.	.		Address used for input/output address.
TRM TRA	0 0	0 0	1 1	0 1				.	.	.	.	.	.	.	.		Address determines one of 2,048 words in the transfer table of the memory module from which the instruction was fetched.
LDI STI	0 0	1 1	0 0	0 1	Basis*			.	.	.	.	.	.	.	.		Direct address, basis used as part of address. Basis 00 → instruction counter.
MIM MIA	0 0	1 1	1 1	0 1	Basis			.	.	.	.	.	.	.	.		Address used as modifier for specified register. Address interpreted as TWO's complement number.
JOI	1	0	0	0	Basis*			.	.	.	.	.	.	.	.		Address interpreted as TWO's complement number.
MSK	1	0	0	1	Basis			.	.	.	.	.	.	.	.		These addresses are relative to the contents of the register selected by the basis field. The address is interpreted as a TWO's complement number. Basis 00 → 0400 octal Basis 01 → index register 1 Basis 10 → index register 2 Basis 11 → index register 3
ADD	1	0	1	0	Basis			.	.	.	.	.	.	.	.		
SUB	1	0	1	1	Basis			.	.	.	.	.	.	.	.		
SUB	1	0	1	1	Basis			.	.	.	.	.	.	.	.		
LDA	1	1	0	0	Basis			.	.	.	.	.	.	.	.		
STO	1	1	0	1	Basis			.	.	.	.	.	.	.	.		
MPY	1	1	1	0	Basis			.	.	.	.	.	.	.	.		
DIV	1	1	1	1	Basis			.	.	.	.	.	.	.	.		

*Basis used as part of address as well as specifying instruction register or index register.

2922

Table 5-3. Order Codes and Address Fields

OCTAL ADDRESS FROM	TO	ALLOCATION
0	3777	Transfer table
0	777	Instruction counter storage
0		Reset interface interrupt
1		Reset power failure interrupt
2		Reset parity interrupt
3		Reset illegal address interrupt
1000	1777	Index register No. 1 storage
2000	2777	Index register No. 2 storage
3000	3777	Index register No. 3 storage
4000		Comparison word for I. C. JOI
4001		Comparison word for IR1 JOI
4002		Comparison word for IR2 JOI
4003		Comparison word for IR3 JOI
5000		I/O interrupt location
5001		Power failure interrupt location
5002		Parity interrupt location
5003		Illegal address interrupt location
5004		Automatic start interrupt location
4000	4777	Directly addressed storage
5000	7777	Relatively addressed storage
14000		Preset location for IR1
10000		Preset location for IR2
30000		Preset location for IR3

Table 5-4. Memory Allocations

FLIP-FLOP	REMARKS
CPI 1	This flip-flop is set when a computer memory parity error occurs and the memory is being read under the control of the computer processor. This flip-flop is reset with a DSO command.
CPI 2	This flip-flop is set under the same conditions as CPI 1, but only if CPI 1 is already set. This flip-flop is reset with a DSO.
API	This flip-flop is set when a computer memory parity error occurs and the memory is being read under the control of the automatic input/output. This flip-flop is reset with a DSO.
CIA 1	This flip-flop is set when an illegal address interrupt occurs and the memory is being accessed by the computer processor. This flip-flop is reset with a DSO.
CIA 2	This flip-flop is set under the same conditions as CIA 1, but only if CIA 1 is already set. It is reset with a DSO.
AIA	This flip-flop is set when an illegal address interrupt occurs and the memory is being accessed by the automatic input/output. It is reset with a DSO.
PFI	This flip-flop is set when a power failure interrupt occurs and is reset only during start-up.
IOI 1	This flip-flop is set when a timed I/O interrupt occurs. It is reset with a DSO.
IOI 2	This flip-flop is set when a timed I/O interrupt occurs and IOI 1 is set. It is reset with a DSO.
OCP 1 OCP 2	Flip-flops OCC 1 and OCC 2 of the operation cycle counter are copies into OCP 1 and OCP 2, respectively, at the occurrence of the last parity error interrupt. There is no reset.
OCA 1 OCA 2	Flip-flops OCC 1 and OCC 2 of the operation cycle counter are copied into OCA 1 and OCA 2, respectively, at the occurrence of the last illegal address interrupt. There is no reset.
FWS	This flip-flop is set if a parity or illegal address error occurs during the first word time of any interrupt subroutine. It is reset with a DSO.

Table 5-5. Fault Register Flip-Flops

At the beginning of all arithmetic operations, it is assumed that one of the operands is in the A register. The A register also holds the computed results of all the single precision operations; for multiplication it holds the most significant half of the product, and for division it holds the quotient. For addition and subtraction it holds the sum or difference. It also transmits and receives information to and from the Input/Output Unit.

The B register holds the multiplier during the multiplication process; at the end of the process it holds the least significant half of the product. At the beginning of division, the B register holds the least significant half of the dividend, and at the end of division it holds the residue. During the long left and long right shift commands, the A and B registers shift together as a double-length register.

Arithmetic overflow (either positive or negative) is detected for addition and subtraction. If overflow occurs, a flip-flop in the fault register is set which may be interrogated by an INP command. The flip-flop is reset by a DSO command.

The speed at which the parallel computer operates is determined principally by the time required for the carry to propagate through all the stages of the adder. In a conventional adder, the carry must pass through the delay of two gates per bit, or a total of 62 gates for a 31-bit adder. Approximately 850 nanoseconds are available in the Magic 331 computer to gate information into the adder, perform addition, and gate the sum into the A register. Therefore, it is necessary to mechanize the adder so that the carry signal passes through fewer gates than conventional adder mechanizations, even though custom-designed integrated logic elements are used instead of conventional micrologic circuits. The Magic 331 adder has a maximum delay of 18 gates. Using 25 nanoseconds per gate as a reasonable value for integrated logic over the entire temperature range, the maximum delay through the adder is 450 nanoseconds. Therefore, 400 nanoseconds are available to gate information into and out of the adder.

Various parallel gating paths have been established in the Arithmetic Unit to provide an efficient flow of data during the execution of arithmetic commands. The main criterion for establishing a gating path was to produce the shortest possible multiplication time. The parallel paths provided are:

- Contents of the A register to the adder,
- Contents of the M register to the adder,
- Complement of the contents of the M register to the adder,
- Sum from the adder to the A register shifted one position to the right (this requires an additional flip-flop on the low-order end of the A register),
- Logical AND of the two adder inputs to the A register,
- B register to A register,
- A register to B register.

Serial connections are provided to shift left the A and B registers, to shift right the combined A and B registers, shift data into the A register from the input/output register, and shift data out of the A register into the input/output register.

Execution of the arithmetic instructions is controlled by the operation cycle counter. For short instructions, such as add, phases C1 or C2 are three clocks or one memory cycle long. The first clock procures the operand from memory when necessary. The second clock is the first execution step of the instruction. The third clock, if needed, is the second execution step of the instruction. For long instructions, the execution starts the same, but the three-clock cycle is repeated by inhibiting C1 or C2 from changing until the instruction is completed.

5.2.3 MEMORY ORGANIZATION

The memory of the Magic 331 computer is of conventional four-wire, coincident-current, toroidal core organization. It is designed to be expandable in blocks of 4,096 words to a total capacity of 32,768 words. Each word stored in the memory consists of 31-data bits; a 32nd bit is used as parity. The organization of the memory is depicted in Figure 5-2.

The large expansion capability of the memory is achieved with a minimum penalty in hardware by incorporating all the electronics that can be shared in the common memory electronics assembly. The common memory electronics contains the memory timing, the address register and decoding logic, the data register, the parity generation and verification logic, and the drive current regulators together with the source selection current switches.

Each memory magnetics assembly contains a $4,096 \times 32$-bit core memory stack, the drive line selection diodes, and all the electronics which are unique to that specific 4,096-word memory block. Each added increment of memory capacity thereby creates automatically a proportionate increase in the nonsharable memory electronics. Included in the memory module are drive-line-current sink switches, the sense electronics and digit driver current switches required for each bit position, and a strobe generator.

The degree of sharing that is possible is in part determined by the physical size of the modules and in part by the memory device characteristics. The packing density achievable today with miniature memory cores and integrated electronic circuits minimizes problems due to the inductance and shunt capacity of interconnecting leads. The very low capacity of the epitaxial diodes used for steering purposes in the memory array permits many more drive lines to be coupled to a source switch than has heretofore been possible. The recent development of wide-temperature-range, lithium-ferrite memory cores minimizes the effects of temperature gradients, thus relaxing the requirements for precise temperature compensation and permitting the use of common current regulators and/or supply voltages which are compensated on the basis of the mean temperature of the memory arrays.

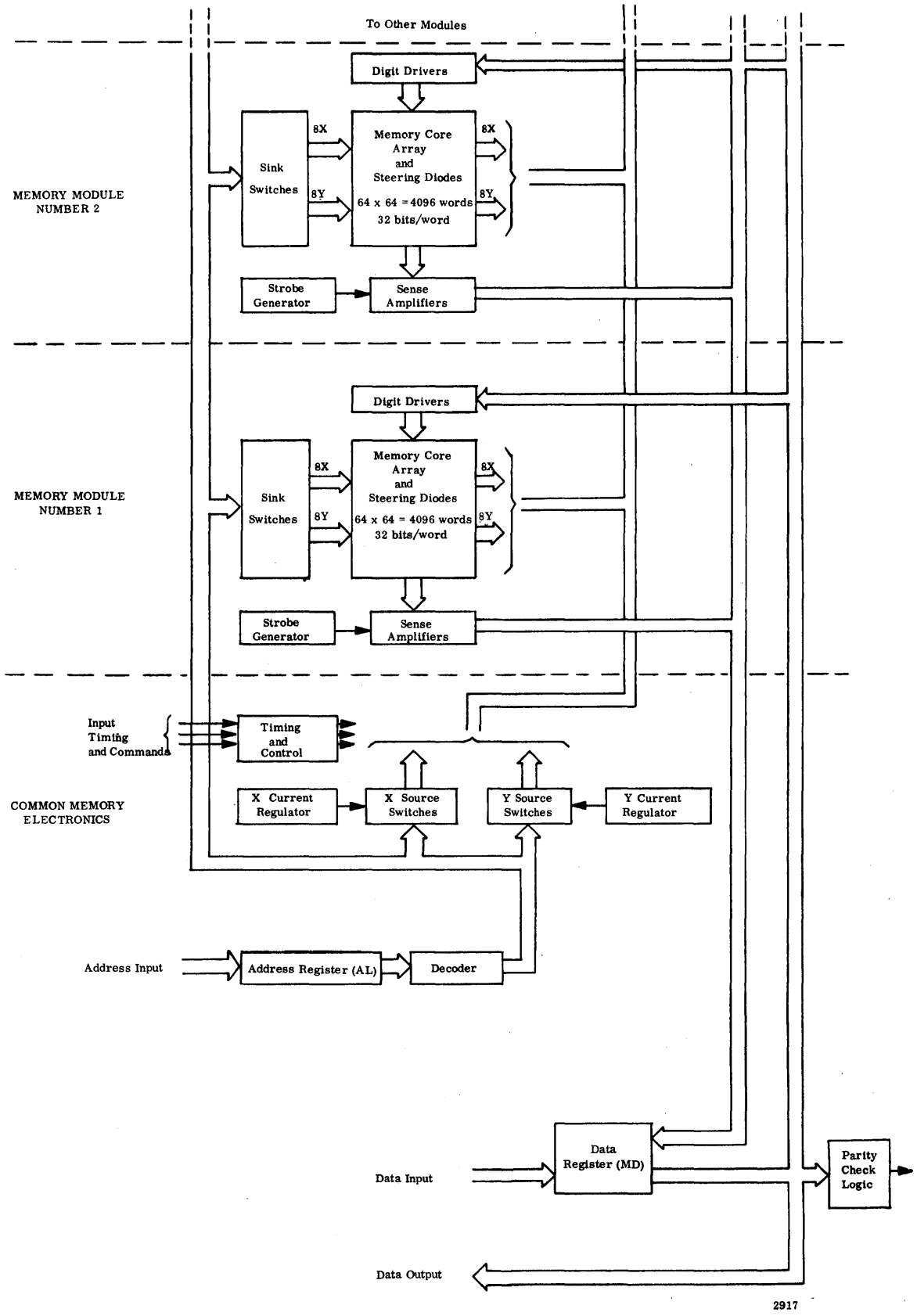


Figure 5-2. Magic 331 Memory Organization