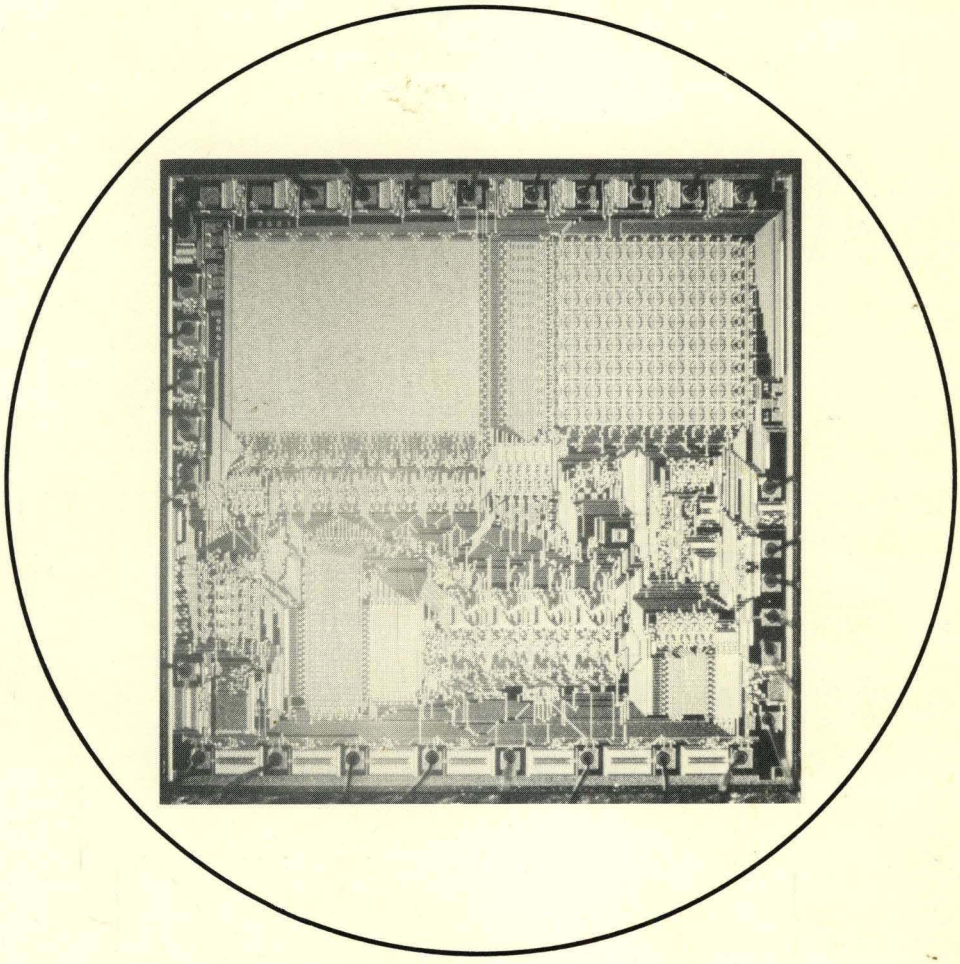
The Engineering Staff of
TEXAS INSTRUMENTS INCORPORATED
Semiconductor Group



TMS 1000 SERIES MOS/LSI ONE-CHIP MICROCOMPUTERS

TEXAS INSTRUMENTS
INCORPORATED

TABLE OF CONTENTS

1. INTRODUCING A ONE-CHIP MICROCOMPUTER
   - 1.1 Description . . . 3
   - 1.2 Applications . . . 4
   - 1.3 Design Support . . . 6

2. TMS1000-SERIES OPERATION
   - 2.1 ROM Operation . . . 7
   - 2.2 RAM Operation . . . 7
   - 2.3 Arithmetic Logic Unit Operation . . . 7
   - 2.4 Input . . . 8
   - 2.5 Output . . . 8
   - 2.6 The Instruction Programmable Logic Array . . . 9
   - 2.7 Timing Relationships . . . 9
   - 2.8 Software Summary . . . 9
   - 2.9 Sample Program . . . 11
   - 2.10 Power-On . . . 12

3. ELECTRICAL AND MECHANICAL SPECIFICATIONS
   - 3.1 Absolute Maximum Ratings . . . 14
   - 3.2 Recommended Operating Conditions . . . 14
   - 3.3 Electrical Characteristics . . . 14
   - 3.4 Schematics of Inputs and Outputs . . . 15
   - 3.5 Internal or External Clock . . . 15
   - 3.6 Terminal Assignments . . . 16
   - 3.7 Mechanical Data . . . 16

LIST OF ILLUSTRATIONS

Figure 1 . . . TMS1000-Series Logic Blocks . . . 3
Figure 2 . . . Block Diagram of Typical Application—Terminal Controller . . . 5
Figure 3 . . . TMS1000-Series Algorithm Development . . . 6
Figure 4 . . . ALU and Associated Data Paths . . . 7
Figure 5 . . . Machine Instruction Flowchart—BCD-Addition Subroutine . . . 13

TMS 1000 NC, TMS 1200 NC MICROCOMPUTERS

1. INTRODUCING A ONE-CHIP MICROCOMPUTER

1.1 DESCRIPTION

The TMS1000 series is a family of P-channel MOS four-bit microcomputers with a ROM, a RAM, and an arithmetic logic unit on a single semiconductor chip. The TMS1000 family is unique in the field of microprocessors because this device is a single-chip binary computer. A customer's specification determines the software that is reproduced during wafer processing by a single-level mask technique that defines a fixed ROM pattern. This versatile one-chip computer is very cost effective and capable of performing a myriad of complex functions.

Key features of the TMS1000 series are:

- 8192-bit Read-Only Memory (ROM) on chip
- 256-bit Random-Access Memory (RAM) on chip
- Arithmetic Logic Unit (ALU) and 2 four-bit working registers on chip
- Conditional branching and subroutines
- Four-bit parallel data input
- 11 latched control/data-strobe outputs in a 28-pin package
- 13 latched control/data-strobe outputs in a 40-pin package
- 8 parallel data outputs and output programmable logic array (PLA)
- Programmable instruction decoder
- On-chip oscillator, or external synchronization if desired
- Single-power-supply operation (15 V)

TMS1000 SERIES	
DEVICE	PACKAGE
TMS1000NC	28-Pin DIP
TMS1200NC	40-Pin DIP

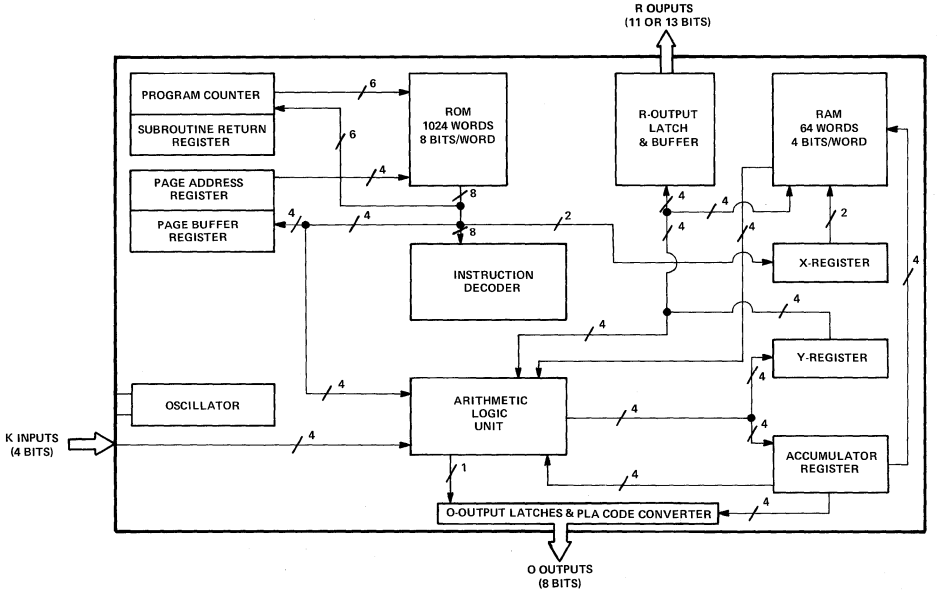


FIGURE 1—TMS1000-SERIES LOGIC BLOCKS

TENTATIVE DATA

This document provides tentative information on a new product. Texas Instruments reserves the right to change specifications for this product in any manner without notice.

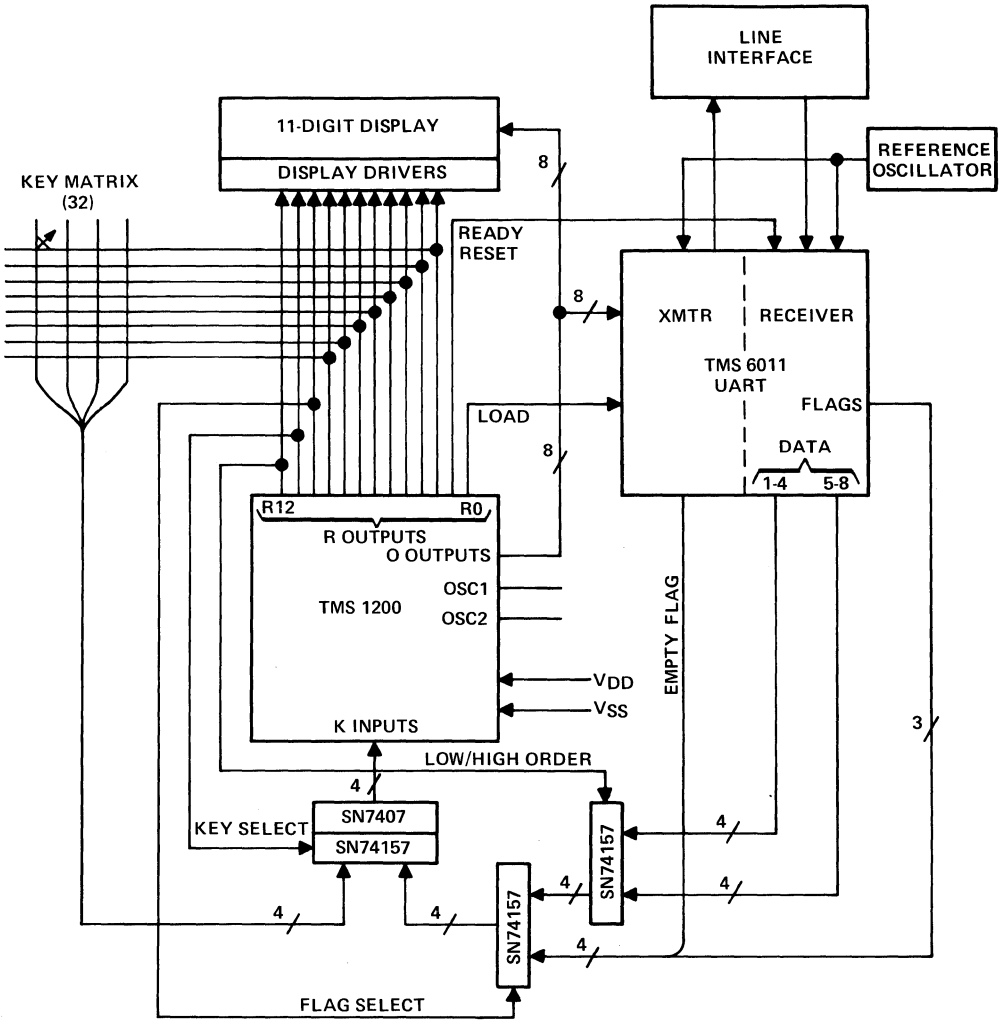
The microcomputer's ROM program controls data input, storage, processing, and output. Data processing takes place in the arithmetic logic unit. K input data goes into the ALU, as shown in Figure 1, and is stored in the four-bit accumulator. The accumulator output accesses the output latches, the RAM storage cells, and the adder input. Data storage in the 256-bit RAM is organized into 64 words, four bits per word. The four-bit words are conveniently grouped into four 16-word files addressed by a two-bit register. A four-bit register addresses one of the 16 words in a file by ROM control.

The O outputs and the R outputs are the output channels. The eight parallel O outputs are decoded from five data latches. The O outputs serve many applications because the decoder is a programmable logic array (PLA) that is modified by changing the gate-level mask tooling. Each of the thirteen R outputs of the TMS1200NC and the eleven R outputs on the TMS1000NC has an individual storage element that can be set or reset by program control. The R outputs send status or enable signals to external devices. The R outputs strobe the O outputs to displays, to other TMS1000 series chips, or to TTL and other interface circuits. The same R outputs multiplex data into the K inputs whenever necessary.

There are 43 basic instructions that handle I/O, constant data from the ROM, bit control, internal data transfer, arithmetic processing, branching, looping, and subroutines. The eight-bit instruction word performs 256 unique operations for maximum efficiency. Section 2.7 defines the standard instruction set, which is optimized for most programs. Microprogramming for special applications is possible, and the operations of the instruction set can be modified by the same mask-tooling step that programs the ROM and the O output PLA.

1.2 APPLICATIONS

One major advantage of the TMS1000 series is flexibility. The TMS1000 series is effective in applications such as printer controllers, data terminals, remote sensing systems, cash registers, appliance controls, and automotive applications. A data terminal is a useful example. In Figure 2, a sample interconnect diagram shows how the R outputs control a universal asynchronous receiver/transmitter (UART), display scan, and keyboard scan. The ROM controls data output to the appropriate display digit or to the transmitter section of the UART. A routine in the ROM program controls selection of incoming data through the K-input ports. Two dedicated R outputs (load and ready reset) control the UART's transmit and receive modes. The remaining R outputs both scan the display and select inputs. The SN74157 TTL devices multiplex eight bits of the incoming data word, four bits of UART status and the four key input lines. Through the TMS1000 series' versatility, a wide range of systems realize reduced costs, fewer parts, and high reliability.



NOTE: Discrete components for level shifting and other functions are not shown

FIGURE 2–BLOCK DIAGRAM OF TYPICAL APPLICATION–TERMINAL CONTROLLER

1.3 DESIGN SUPPORT

Through a staff of experienced application programmers, Texas Instruments will, upon request, assist customers in evaluating applications, in training designers to program the TMS1000 series and in simulating programs. TI will also contract to write programs to customer's specifications.

TI has developed an assembler and simulator for aiding software designs. These programs are available on nationwide time-sharing systems and at TI computer facilities.

A TMS1000 series program (see flowchart, Figure 3) is written in assembly language using standard mnemonics. The assembler converts the source code (assembly language program) into machine code, which is transferred to a software simulation program. Also the assembler produces a machine code object deck. The object deck is used to produce a tape for hardware simulation or a tape for generating prototype tooling.

The TMS1000 series programs are checked by software and hardware simulation. The software simulation offers the advantages of printed outputs for instruction traces or periodic outputs. The hardware simulation offers the designer the advantages of real-time simulation and testing asynchronous inputs. A software user's guide is available.

After the algorithms have been checked and approved by the customer, the final object code and machine option statements are supplied to TI. A gate mask is generated and slices produced. After assembly and testing, the prototypes are shipped to the customer for approval. Upon receiving final approval, the part is released for volume production at the required rate as one unique version of the TMS1000 family.

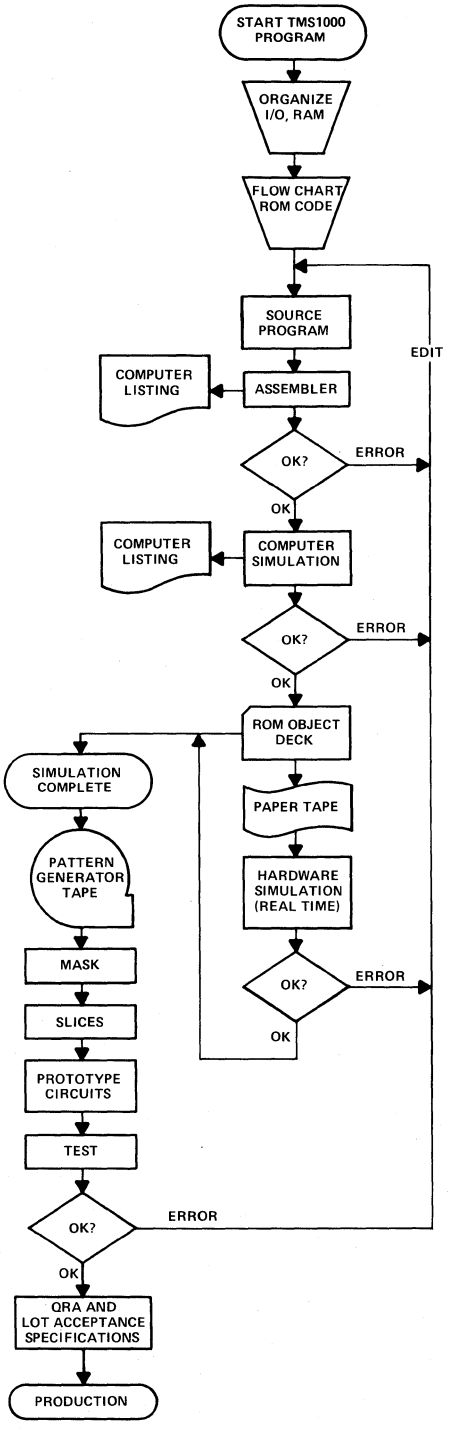


FIGURE 3–TMS1000-SERIES ALGORITHM DEVELOPMENT

2. TMS1000-SERIES OPERATION

2.1 ROM OPERATION

The sequence of the 1024 eight-bit ROM instructions determines the device operation. There are 16 pages of instructions with 64 instructions on each page. After power-up the program execution starts at a fixed instruction address. Then a shift-register program counter sequentially addresses each ROM instruction on a page. A conditional branch or call subroutine instruction may alter the six-bit program-counter address to transfer software control. One level of subroutine return address is stored in the subroutine return register. The page address register (four bits) holds the current address for one of the 16 ROM pages. To change pages, a constant from the ROM loads into the page buffer register (four bits), and upon a successful branch or call, the page buffer loads into the page address register. The page buffer register also holds the return page address in the call subroutine mode.

2.2 RAM OPERATION

There are 256 addressable bits of RAM storage available. The RAM is comprised of four files, each file containing 16 four-bit words. The RAM is addressed by the Y register and the X register. The Y register selects one of the 16 words in a file and is completely controllable by the arithmetic unit. The TMS1000 series has instructions that: Compare Y to a constant, set Y to a constant, increment or decrement Y, and/or perform data transfer to or from Y. Two bits in the X register select one of the four 16-word files. The X register is set to a constant or is complemented. A four-bit data word goes to the RAM location addressed by X and Y from the accumulator or from the constants in the ROM. The RAM output words go to the arithmetic unit and can be operated on and loaded into Y or the accumulator in one instruction interval. Any selected bit in the RAM can be set, reset, or tested.

2.3 ARITHMETIC LOGIC UNIT OPERATION

Arithmetic and logic operations are performed by the four-bit adder and associated logic. The arithmetic unit performs logical comparison, arithmetic comparison, add, and subtract functions. The arithmetic unit and interconnects are shown in Figure 4. The operations are performed on two sets of inputs, P and N. The two four-bit parallel inputs may be added together or logically compared. The accumulator has an inverted output to the N selector for subtraction by two's complement arithmetic. The other N inputs are from the true output of the accumulator, the RAM, constants, and the K inputs. The P inputs come from the Y register, the RAM, the constants, and the K inputs.

Addition and subtraction results are stored in either the Y register or the accumulator. An arithmetic function may cause a carry output to the status logic. Logical comparison may generate an output to status. If the comparison functions are used, only the status bit affects the program control, and neither the Y register's nor the accumulator register's contents are affected. If the status feedback is a logic one, which is the normal state, then the conditional branch or call is executed. If an instruction calls for a carry output to status and the carry does not occur, then status will go to a zero state for one instruction cycle. Likewise, if an instruction calls for the logical-comparison function and the bits compared are all equal, then status will go to a zero state for one instruction cycle. If status is a logic zero, then branches and calls are not performed.

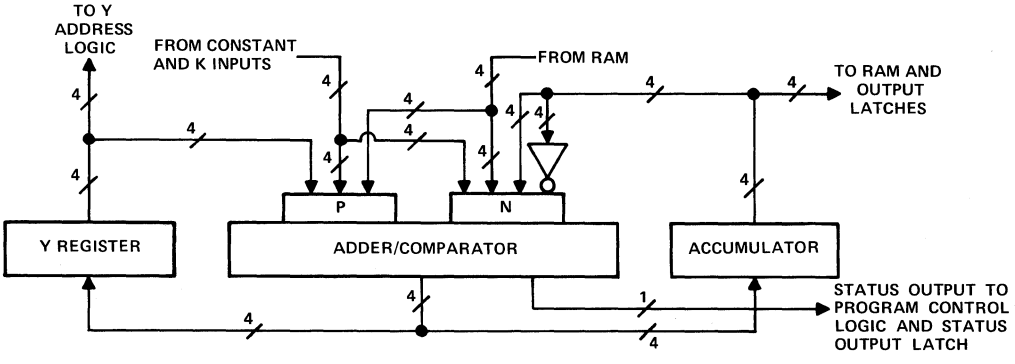


FIGURE 4–ALU AND ASSOCIATED DATA PATHS

2.4 INPUT

There are four data inputs to the TMS1000-series circuit, K1, K2, K4, and K8. Each time an input word is requested, the data path from the K inputs is enabled to the adder. The inputs are either tested for a high level ($\approx V_{SS}$), or the input data are stored in the accumulator for further use. The R outputs usually multiplex inputs such as keys and other data. Other input interfaces are possible. An external device that sends data out to the K-input bus at a fixed rate may be used with the TMS1000 series when an initiating "handshake" signal is given from an R output. Data from the K inputs is stored periodically in synchronization with the predetermined data rate of the external device. Thus, multiple four-bit words can be requested and stored with only one R output supplying the control signal.

2.5 OUTPUT

There are two output channels with multiple purposes, the R outputs and the O outputs. Thirteen latches store the R output data. The eight parallel O outputs come from a five-bit-to-eight-bit code converter, which is the O-output PLA.

The R outputs are individually addressed by the Y register. Each addressed bit can be set or reset. The R outputs are normally used to multiplex inputs and strobe O output data to displays, external memories, and other devices. Also, one R output can strobe other R outputs that represent variable data, because every R output may be set or reset individually. For example, the Y register addresses each latch in turn, the variable data R outputs are set or reset, and finally, the data strobe R latch is set.

The eight O outputs usually send out display or binary data that are encoded from the O output latches. The O latches contain five bits. Four bits load from the accumulator in parallel. The fifth bit comes from the status latch, which is selectively loaded from the adder output (see Figure 4). The load output command sends the status latch and accumulator information into the five output latches. The five bits are available in true or complementary form to 20 programmable-input NAND gates in the O output PLA. Each NAND gate can simultaneously select any combination of O0 thru O7 as an output. The user defines this PLA's decoding to suit an optimum output configuration. As an illustration, the O output PLA can encode any 16 characters of eight-segment display information and additionally can transfer out a four-bit word of binary data.

2.6 THE INSTRUCTION PROGRAMMABLE LOGIC ARRAY

The programmable instruction decode is defined by the instruction PLA. Thirty programmable-input NAND gates decode the eight bits of instruction word. Each NAND gate output selects a combination of 16 microinstructions. The 16 microinstructions control the arithmetic unit, status logic, status latch, and write inputs to the RAM.

As an example, the "add eight to the accumulator, results to accumulator" instruction can be modified to perform a "add eight to the Y register, results to Y" instruction. Modifications that take away an instruction that is not used very often are desirable if the modified instructions save ROM words by increasing the efficiency of the instruction repertoire. A programmer's reference manual is available to explain PLA programming and the TMS1000-series operation in detail.

2.7 TIMING RELATIONSHIPS

Six oscillator pulses constitute one instruction cycle. All instructions are executed in one instruction cycle. The actual machine cycle period is determined by either a fixed external resistor and capacitor connected to the OSC1 and OSC2 pins (refer to Section 3.5), or an external clock input frequency.

2.8 SOFTWARE SUMMARY

The following table defines the TMS1000 series' standard instruction set with a description, mnemonic, and status effect. The mnemonics were defined for easy reference to the functional description. Eighteen mnemonics use an identifier to indicate the condition that satisfies the status requirement for a successful branch or call if the instruction is followed immediately by a branch or call command. "C" means that if the instruction generates a carry (status = one), then a following branch or call is executed. If a branch instruction does not follow or if there is no carry (status = zero), then the program counter proceeds to the next address without changing the normal counting sequence. "N" means that if no borrow (equal to a carry in two's complement aritmetic) is generated, an ensuing branch or call is taken. "Z" indicates that if the two's complement of zero in the accumulator (instruction CPAIZ) is attempted with a branch or call following, then the branch or call is taken. "1", "LE", "NE", and "NEZ" are used to indicate conditions for branch and call for seven test instructions. The test instructions do not modify data at all; tests are used solely in conjunction with subsequent branches or calls.

If an instruction that does not affect status is placed between an instruction that does affect status and a branch or call instruction, then the branch or call is always performed. This is true because status always returns to its normal state (status = one) after one instruction cycle, and branches and calls are taken if status equals one.

TMS1000-SERIES STANDARD INSTRUCTION SET

FUNCTION	MNEMONIC	STATUS EFFECTS		DESCRIPTION
		C	N	
Register to Register	TAY			Transfer accumulator to Y register.
	TYA			Transfer Y register to accumulator.
	CLA			Clear accumulator.
Transfer Register to Memory	TAM			Transfer accumulator to memory.
	TAMIY			Transfer accumulator to memory and increment Y register.
	TAMZA			Transfer accumulator to memory and zero accumulator.
Memory to Register	TMY			Transfer memory to Y register.
	TMA			Transfer memory to accumulator.
	XMA			Exchange memory and accumulator.
Arithmetic	AMAAC	Y		Add memory to accumulator, results to accumulator. If carry, one to status.
	SAMAN	Y		Subtract accumulator from memory, results to accumulator. If no borrow, one to status.
	IMAC	Y		Increment memory and load into accumulator. If carry, one to status.
	DMAN	Y		Decrement memory and load into accumulator. If no borrow, one to status.
	IA			Increment accumulator, no status effect.
	IYC	Y		Increment Y register. If carry, one to status.
	DAN	Y		Decrement accumulator. If no borrow, one to status.
	DYN	Y		Decrement Y register. If no borrow, one to status.
	A8AAC	Y		Add 8 to accumulator, results to accumulator. If carry, one to status.
	A10AAC	Y		Add 10 to accumulator, results to accumulator. If carry, one to status.
	A6AAC	Y		Add 6 to accumulator, results to accumulator. If carry, one to status.
	CPAIZ	Y		Complement accumulator and increment. If then zero, one to status.
Arithmetic Compare	ALEM	Y		If accumulator less than or equal to memory, one to status.
	ALEC	Y		If accumulator less than or equal to a constant, one to status
Logical Compare	MNEZ		Y	If memory not equal to zero, one to status.
	YNEA		Y	If Y register not equal to accumulator, one to status.
	YNEC		Y	If Y register not equal to a constant, one to status
Bits in Memory	SBIT			Set memory bit.
	RBIT			Reset memory bit.
	TBIT1		Y	Test memory bit. If equal to one, one to status.
Constants	TCY			Transfer constant to Y register.
	TCMIY			Transfer constant to memory and increment Y.
Input	KNEZ		Y	If K inputs not equal to zero, one to status.
	TKA			Transfer K inputs to accumulator.
Output	SETR			Set R output addressed by Y.
	RSTR			Reset R output addressed by Y.
	TDO			Transfer data from accumulator and status latch to O outputs.
	CLO			Clear O-output register.
RAM 'X' Addressing	LDX			Load 'X' with a constant.
	COMX			Complement 'X'.
ROM Addressing	BR			Branch on status = one.
	CALL			Call subroutine on status = one.
	RETN			Return from subroutine.
	LDP			Load page buffer with constant.

NOTES: C-Y (Yes) means that if there is a carry out of the MSB, status output goes to the one state. If no carry is generated, status output goes to the zero state.

N-Y (Yes) means that if the bits compared are not equal, status output goes to the one state. If the bits are equal, status output goes to the zero state.

A zero in status remains through the next instruction cycle only. If the next instruction is a branch or call and status is a zero, then the branch or call is not executed.

2.9 SAMPLE PROGRAM

The following example shows register addition of up to fifteen BCD digits. The add subroutine (flow charted in Figure 5) can use the entire RAM, which is divided into two pairs of registers. The definition of registers, for the purpose of illustration, is expanded to include the concept of a variable-length word that is a subset of a 16-digit file. Addition proceeds from the least-significant digit (LSD) to the most-significant digit (MSD), and carry ripples through the accumulator. The decrement-Y instruction is used to index the numbers in a register. The initial Y value sets the address for the LSD's of two numbers to be added. Thus, if Y equals eight at the start, the LSD is defined to be stored in M(X,8), [M(X, Y) ≡ contents of RAM word location X equals 0, 1, 2, or 3, and Y equals 0 to 15]. If Y is eight initially, M(X,7) is the next-most-significant digit.

RAM DATA MAP BEFORE EXECUTING SAMPLE ROUTINE

FILE ADDRESS	REGISTER	Y-REGISTER ADDRESS															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X = 00	D	OV 0	MSD 9	8	7	6	5	4	3	LSD 2							
X = 01	E	OV 0	MSD 1	2	3	4	5	6	7	8	9	0	1	2	3	4	LSD 5
X = 10	F	OV 0	MSD 5	4	3	2	1	0	9	8	7	6	5	4	3	2	LSD 1
X = 11	G	OV 0	MSD 8	7	6	5	4	3	2	LSD 1							

In the preceeding RAM register assignment map, registers D and G are nine digits long, and registers E and F are 16 digits long. The sample routine calls the D plus G → D subroutine and the E plus F → E subroutine. After executing the two subroutines, the RAM contents are the following:

RAM DATA MAP AFTER EXECUTING SAMPLE ROUTINE

FILE ADDRESS	REGISTER	Y-REGISTER ADDRESS															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X = 00	D	OV 1	MSD 8	6	4	1	9	7	5	LSD 3							
X = 01	E	OV 0	MSD 6	6	6	6	6	7	7	7	6	6	6	6	6	6	LSD 6
X = 10	F	OV 0	MSD 5	4	3	2	1	0	9	8	7	6	5	4	3	2	LSD 1
X = 11	G	OV 0	MSD 8	7	6	5	4	3	2	LSD 1							

NOTE: Cross-hatched areas indicate locations in the RAM that are unaffected by executing the example routine.

	LABEL	OPCODE	OPERAND	COMMENT
MAIN PROGRAM PRESETS Y, AND CALL SUBROUTINES		TCY	8	Transfer 8 → Y
		CALL	ADGD	Add: D + G → D
		TCY	15	Transfer 15 → Y
		CALL	AEFE	Add: E + F → E
		.		
		.		
		.		
MULTIPLE ENTRY POINTS FOR SUBROUTINES	ADGG	LDX	3	3 → X; Set up for D + G → G.
		BR	BCDADD	Branch to BCD add.
	AEFF	LDX	2	2 → X; Set up for E + F → F.
		BR	BCDADD	Branch to BCD add.
	AEFE	LDX	1	1 → X; Set up for E + F → E.
		BR	BCDADD	Branch to BCD add.
	ADGD	LDX	0	0 → X; Add D + G → D.
BASE SUBROUTINE CONTAINS LOOPING AND BCD CORRECTION	BCDADD	CLA		Clear accumulator (A).
	LOOP	COMX		$\overline{X} \rightarrow X$.
		AMAAC		M(X,Y) + A → A; A contains possible carry if in loop.
		COMX		$\overline{X} \rightarrow X$.
		AMAAC		Add digits: M(X, Y) + [M($\overline{X}$,Y) + Carry] → A.
		BR	GT9	Branch if sum >15.
		ALEC	9	If A ≤ 9, one to status.
		BR	LT10	Branch if sum < 10.
	GT9	A6AAC		Sum > 9, A + 6 → A; BCD Correction.
		TAMZA		Transfer corrected sum to memory, 0 → A.
		IA		1 → A; to propogate carry
	DECY	DCYN		Y − 1 → Y; index next digit.
		BR	LOOP	If no borrow, continue.
		RETN		If borrow, return to instruction after call.
	LT10	TAMZA		Sum < 9, A → M(X,Y); 0 → A; No carry propogated.
		BR	DECY	

Note that there are four entry points to the base subroutine (ADGG, ADGD, AEFF, AEFE). The main program can call two of the other possible subroutines that store the addition results differently. These subroutines have applications in floating-point arithmetic, multiplication, division, and subtraction routines.

2.10 POWER-ON

The TMS1000 series has a built-in power-on latch, which resets the program counter upon the proper application of power. After power-up the chip resets and begins execution at a fixed ROM address. The system reset depends on the ROM program after the starting address. For power supplies with slow rise times or noisy conditions, an external network connected to the test pin may be necessary.

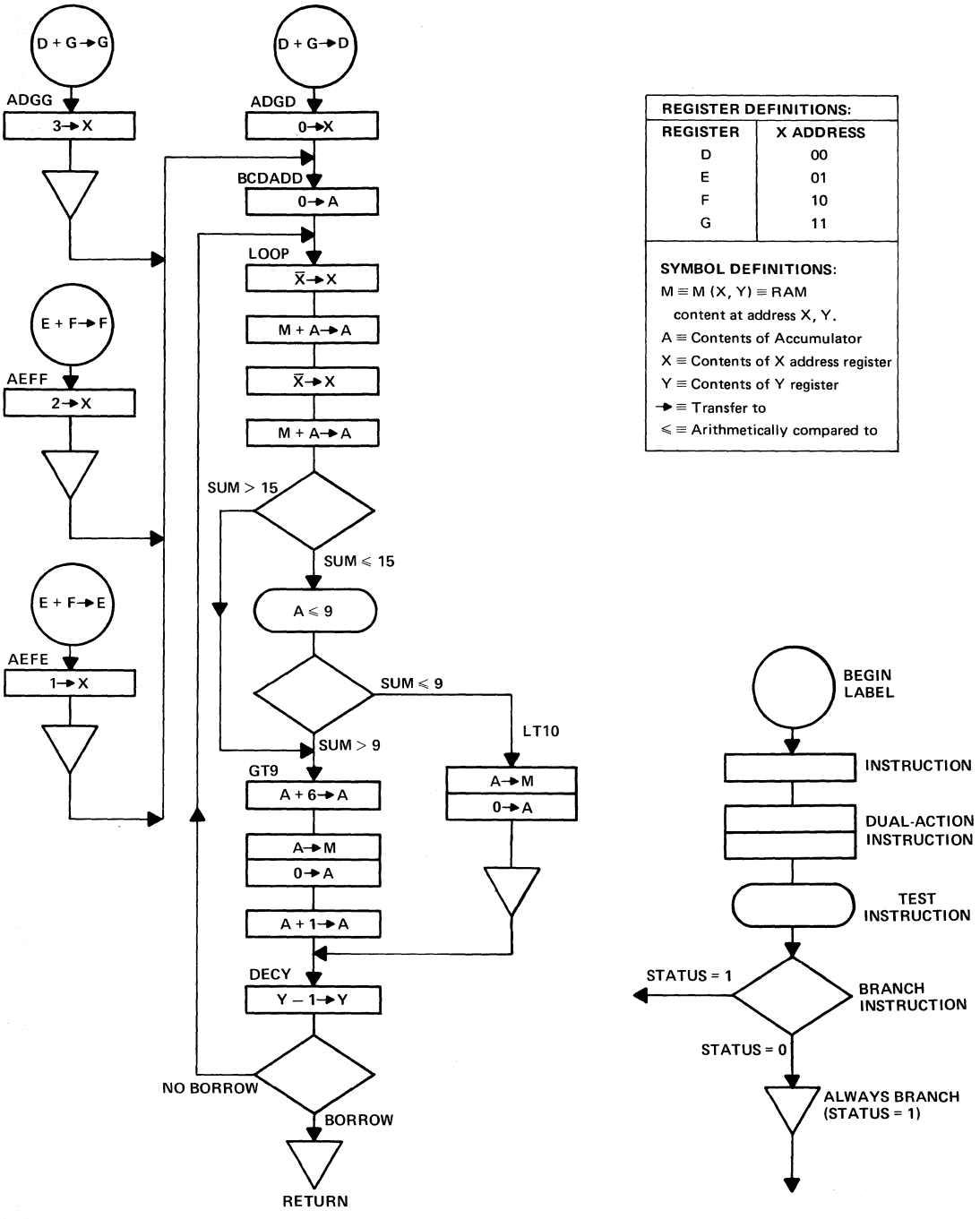


FIGURE 5–MACHINE INSTRUCTION FLOWCHART–BCD-ADDITION SUBROUTINE

3. TMS1000-SERIES ELECTRICAL AND MECHANICAL SPECIFICATIONS

3.1 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)

Voltage applied to any device terminal (see Note 1) . . . . . −20 V
Supply voltage, V_{DD} . . . . . −20 to 0.3 V
Clock input voltage . . . . . −20 to 0.3 V
Data input voltage . . . . . −20 to 0.3 V
Continuous power dissipation: TMS1000NC . . . . . 400 mW
TMS1200NC . . . . . 600 mW
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −55°C to 150°C

3.2 RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{DD} (see Note 2)	−14	−15	−17.5	V
High-level input voltage (see Note 3)	−1.3		0	V
Low-level input voltage (see Note 3)	V_{DD}		−4.6	V
Oscillator frequency	100		400	kHz
Instruction cycle time	15		60	μs
Capacitance for internal oscillator, C_{ext} (see Section 3.5)	10			pF
Operating free-air temperature	0		70	°C

3.3 ELECTRICAL CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (UNLESS OTHERWISE NOTED)

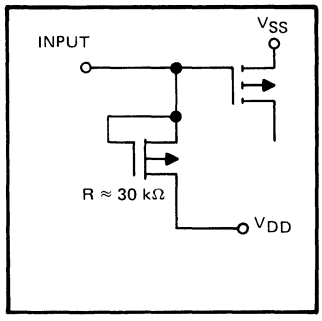
PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_I	Input current, K inputs		V_I = 0 V	50	200	300	μA
V_{OH}	High-level output voltage (see Note 3)	O outputs	I_O = −10 mA	−1.1	−0.6		V
		R outputs	I_O = −2 mA	−0.75	−0.4		
I_{OL}	Low-level output current		$V_{OL} = V_{DD}$			−100	μA
$I_{DD(av)}$	Average supply current from V_{DD} (see Note 4)		All outputs open		−6	−10	mA
$P_{(AV)}$	Average power dissipation (see Note 4)		All outputs open		90	175	mW
f_{osc}	Internal oscillator frequency		R_{ext} = 45 kΩ, C_{ext} = 47 pF	250	300	350	kHz
C_i	Small-signal input capacitance, K inputs		V_I = 0, f = 1 kHz		10		pF

†All typical values are at V_{DD} = −15 V, T_A = 25°C

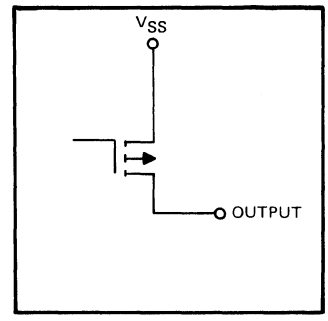
NOTES: 1. Unless otherwise noted, all voltages are with respect to V_{SS}. Exceeding the limits given under absolute maximum ratings may cause permanent damage to the circuit. These are stress limits only, and functional operation of the circuit at these or any other conditions different from those given in the recommended operating conditions of this specification is not implied.
2. Ripple must not exceed 0.2 volts peak-to-peak in the 150-kHz-to-200-kHz range.
3. The algebraic convention where the most-positive (least-negative) limit is designated as maximum is used in this specification for logic voltage levels only.
4. Values are given for the open-drain O and R output configurations. Pull-down resistors are optionally available on all outputs and increase I_{DD} (see Section 3.4).

3.4 SCHEMATICS OF INPUTS AND OUTPUTS

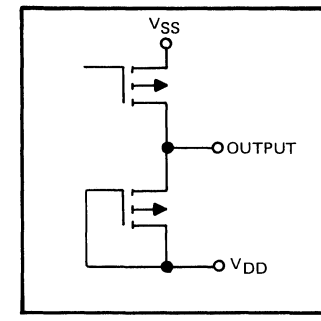
TYPICAL OF ALL K INPUTS



TYPICAL OF ALL O AND R OPEN-DRAIN OUTPUTS



TYPICAL OF ALL O AND R OUTPUTS WITH OPTIONAL PULL-DOWN RESISTORS



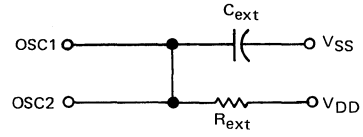
The width of the pull-down resistors is mask alterable and provides the following nominal short-circuit output currents (outputs shorted to V_{SS}):

O outputs: 100, 200, or 300 μA
R outputs: 50, 100, or 150 μA

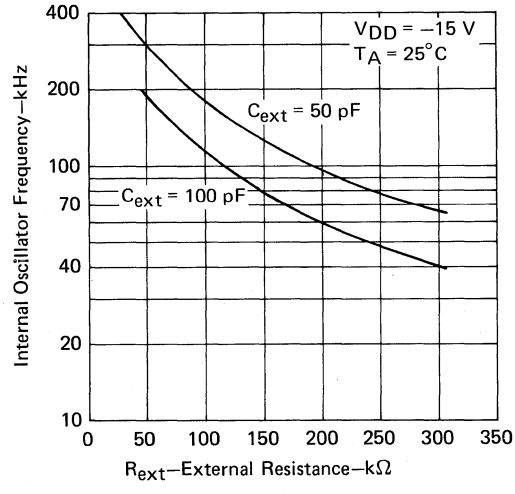
3.5 INTERNAL OR EXTERNAL CLOCK

If the internal oscillator is used, the OSC1 and OSC2 terminals are shorted together and tied to an external resistor to V_{DD} and a capacitor to V_{SS}. If an external clock is desired, the clock source may be connected to OSC1 with OSC2 shorted to V_{SS}.

CONNECTION FOR INTERNAL OSCILLATOR

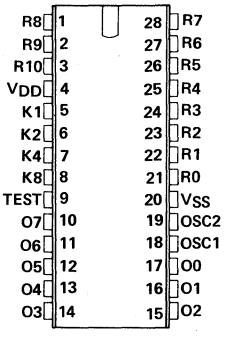


TYPICAL INTERNAL OSCILLATOR FREQUENCY
vs
EXTERNAL RESISTANCE

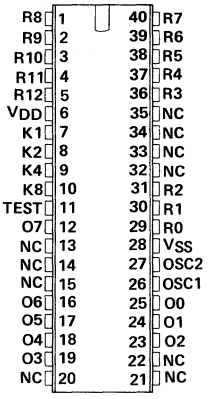


3.6 TERMINAL ASSIGNMENTS

TMS1000NC



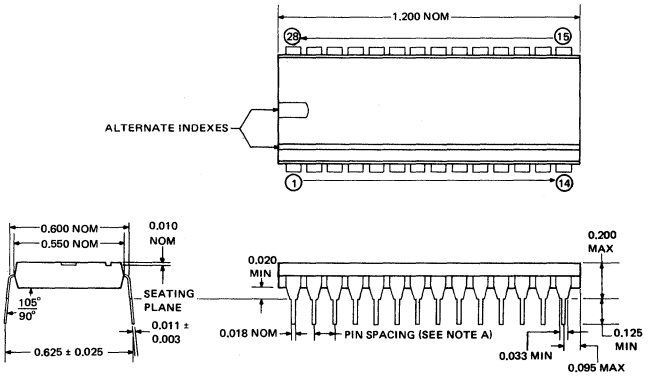
TMS1200NC



NC—No internal connection.

3.7 MECHANICAL DATA

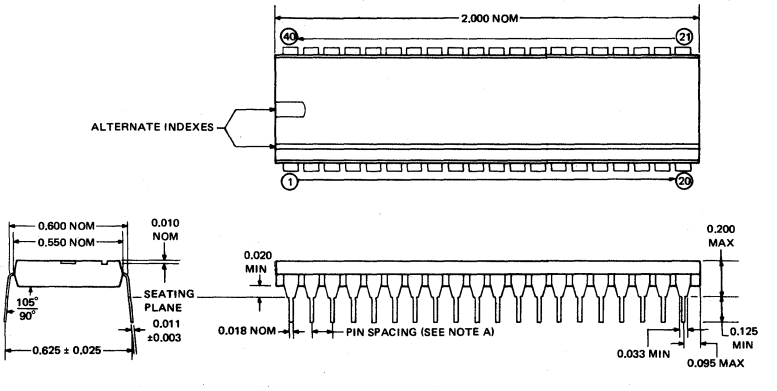
TMS1000NC—28-PIN PLASTIC PACKAGE



NOTES: A. The true position pin spacing is 0.100 between centerlines. Each pin centerline is located within 0.010 of its true longitudinal position relative to pins 1 and 28.

B. All dimensions are in inches unless otherwise noted.

TMS1200NC—40-PIN PLASTIC PACKAGE



NOTES: A. The true-position pin spacing is 0.100 between centerlines. Each pin centerline is located within 0.010 of its true longitudinal position relative to pins 1 and 40.

B. All dimensions are in inches unless othewrise noted.