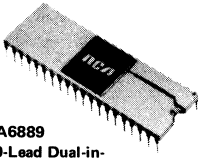
RCA
Solid State Division

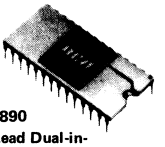
Digital Integrated Circuits

Monolithic Silicon

Developmental Types TA6889, TA6890



TA6889
40-Lead Dual-in-Line Ceramic H-1847



TA6890
28-Lead Dual-in-Line Ceramic H-1848

RCA COS/MOS Microprocessor (COSMAC)

TA6889 – Microprocessor Control IC
TA6890 – Microprocessor Register IC

Features:

- Static COS/MOS circuitry, no minimum clock frequency
- Full military temperature range
- High noise immunity, wide operating voltage range
- TTL compatibility
- 8-bit parallel organization with bidirectional data bus
- Built-in program-load facility
- Any combination of standard RAM/ROM via common interface
- Direct memory addressing up to 65,536 bytes
- Flexible programmed I/O mode
- Program interrupt mode
- On-chip DMA facility
- Four I/O flag inputs directly testable by Branch instructions
- One-byte instruction format with two machine cycles for each instruction
- 59 easy-to-use instructions
- 16 x 16 matrix of registers for use as multiple program counters, data pointers, or data registers

The RCA Dev. Types TA6889 and TA6890, as an integral circuit pair, form a microprocessor (COSMAC) that is an LSI COS/MOS, 8-bit, register-oriented central processing unit (CPU) designed for use as a general-purpose computing or control element. The COSMAC architecture was designed with emphasis on the total microcomputer system as an integral entity so that systems having maximum flexibility and minimum cost can be realized.

Applications:

Point-of-sale terminals
Automotive controls
Entertainment/Educational devices
Communication controllers
Small business machines
Intelligent computer peripherals
Process monitors and controllers
Industrial controllers
Smart instruments
Traffic controls
Military/Aerospace applications

Printed in USA/2-75

MAXIMUM RATINGS, *Absolute-Maximum Values:*

STORAGE-TEMPERATURE RANGE	−65 to +150°C
OPERATING-TEMPERATURE RANGE	−55 to +125°C
DC SUPPLY-VOLTAGE RANGE	
V_{DD}▲ ($V_{CC} \leqslant V_{DD}$)	−0.5 to +15 V
DEVICE DISSIPATION (PER PACKAGE)	200 mW
LEAD TEMPERATURE (DURING SOLDERING):	
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 seconds max.	265°C

▲All voltage values are referenced to V_{SS} terminal.

OPERATING CONDITIONS AT T_A = 25°C

For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges.

Characteristic	V_{DD}	Min.	Typ.	Max.	Units
Supply Voltage Range	–	4	10	12	V
Maximum Input Voltage Swing	–	V_{SS}	–	V_{DD}	V
Recommended Input Voltage Swing	–	V_{SS}	–	V_{CC}	
Clock Input Rise or Fall Time	4-12 V	–	–	15	μs
Instruction Time (Using memory having 1-μs access time)	10 V V_{CC}=5 V	–	8	–	μs
Clock Input Frequency	10 V	DC	2	–	MHz

Notes:
1. $V_{CC} \leqslant V_{DD}$
2. Because a large number of nodes may be switching simultaneously, a 0.1 μF by-pass capacitor is recommended in the power supply.

ELECTRICAL CHARACTERISTICS @ T_A = 25°C; V_{SS} = Gnd; t_r,t_f = 20 ns

CHARACTERISTIC	TEST CONDITIONS	V_{CC} V	V_{DD} V	TYPICAL VALUES	UNITS
STATIC					
Quiescent Device Current, I_L		5	5	1	mA
		10	10	2	
Output Voltage: Low Level, V_{OL}	Driving COS/MOS Loads	–	–	0	V
High Level, V_{OH}		–	–	V_{CC}	
Output Drive Current: N-Channel (Sink), I_{DN}	V_O=0.4 V	5	10	5	mA
P-Channel (Source), I_{DP}	V_O=4.6 V	5	10	1.5	
	V_O=2.5 V	5	5	5	
		5	10	15	
Noise Immunity (Any Input) V_{NL}	V_O=0.8 V	5	5	2.25	V
V_{NH}	V_O=4.2 V	5	5	2.25	
DYNAMIC					
Average Device Dissipation (Total Both Units), P_d	f_{CL}=2 MHz C_L=50 pF	5	10	60	mW
Average Device Current (I_{dd} + I_{cc})		5	10	6.5	mA
Settling Time, t_s	Fig. 1	5	5	1500	ns
		5	10	750	
Transition Time: t_{THL}	C_L=50 pF	5	10	30	ns
t_{TLH}		5	10	80	
Average Input Capacitance	Any Input			5	pF

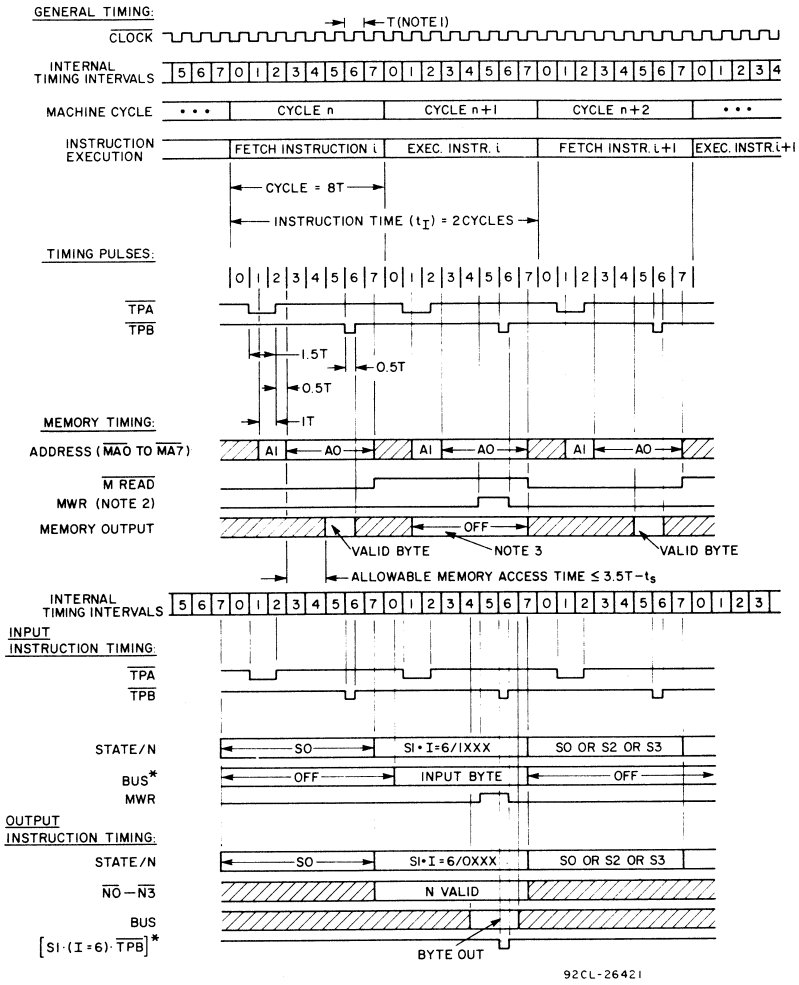
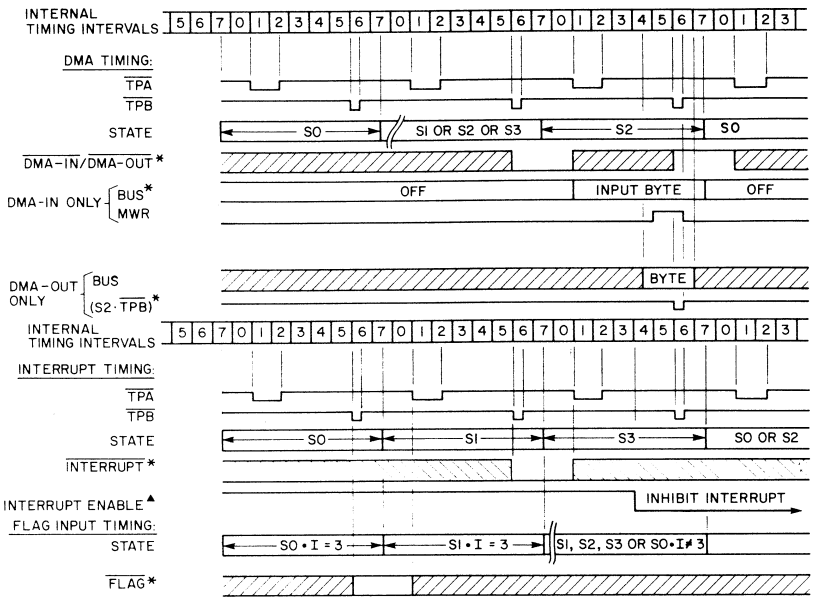


Fig. 1—COSMAC Timing.

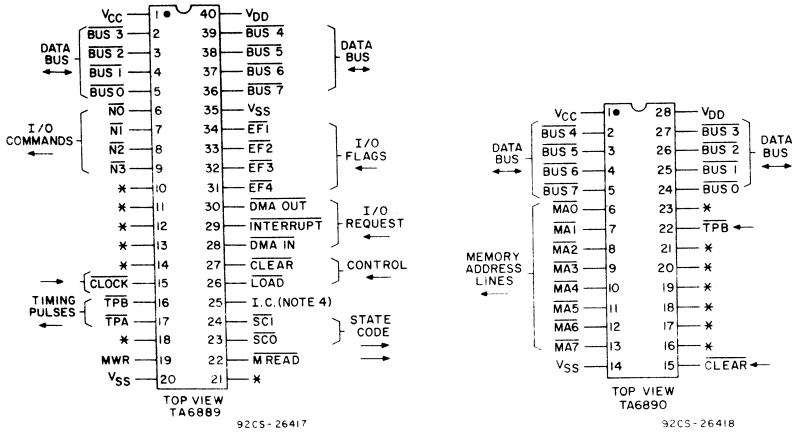


* = SIGNAL GENERATED BY USER
▲ = INTERNAL TO COSMAC

NOTES:
1. MINIMUM T DETERMINED BY V_{DD} -- NO MAXIMUM T
2. MEMORY WRITE PULSE WIDTH (MWR) ≈ 1.5 T
3. MEMORY OUTPUT "OFF" INDICATES HIGH-IMPEDANCE CONDITION.
4. SHADING INDICATES "DON'T CARE" OR INTERNAL DELAYS DEPENDING ON V_{DD} AND THE CLOCK SPEED.

92CL-26422

Fig. 1—COSMAC Timing (Cont'd).



Package Interconnections

						*	*	*	*	*		*	*					
TA6889 Pin No.	1	2	3	4	5	10	11	12	13	14	16	18	21	27	36	37	38	39	40
TA6890 Pin No.	1	27	26	25	24	17	18	19	20	23	22	21	16	15	5	4	3	2	28
						*	*	*	*	*		*	*						

* These pins are for interchip connections only.

Notes:

1. Any unused input pins should be connected to V_{DD} or V_{CC}.
2. The Data Bus lines are bi-directional and have three-state outputs. They may be individually connected to V_{CC} through external pull-up resistors (22 kΩ recommended) to prevent floating inputs.
3. All inputs have the same noise immunity and level-shifting capability. All outputs have the same drive capability whether they have three-state outputs or not.
4. Pin 25 of TA6889 is used for an internal connection—**do not use**.

Fig. 2—Terminal Assignment Diagrams.

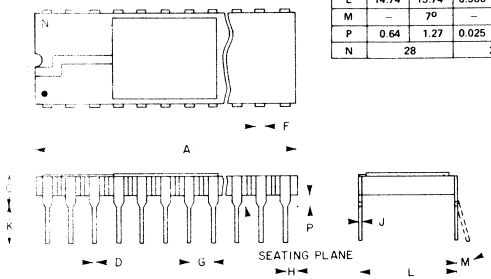
DIMENSIONAL OUTLINES

TA6890
28-Lead Ceramic

DIM.	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	35.06	36.06	1.380	1.420
C	2.16	3.68	0.085	0.145
D	0.43	0.56	0.017	0.023
F	1.27 REF.		0.050 REF.	
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	3.18	4.45	0.125	0.175
L	14.74	15.74	0.580	0.620
M	–	7°	–	7°
P	0.64	1.27	0.025	0.050
N	28		28	

TA6889
40-Lead Ceramic

DIM.	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	50.30	51.30	1.980	2.020
C	2.42	3.93	0.095	0.155
D	0.43	0.56	0.017	0.023
F	1.27 REF.		0.050 REF.	
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	3.18	4.45	0.125	0.175
L	14.74	15.74	0.580	0.620
M	–	7°	–	7°
P	0.64	1.27	0.025	0.050
N	40		40	



NOTES:
1. Leads within 0.13 mm (0.005) radius of true position at maximum material condition.
2. Dimension "L" to center of leads when formed parallel.
3. When this device is supplied solder-dipped, the maximum lead thickness (narrow portion) will not exceed 0.013 in. (0.33 mm).

The following is a brief description of the COSMAC microprocessor. For more detailed information see the following two publications:

1. "User Manual for the COSMAC Microprocessor" – MPM-101
2. "Program Development Guide for the COSMAC Microprocessor" – MPM-102

The RCA Microprocessor (COSMAC) is implemented by two COS/MOS chips. The Dev. No. TA6889, in a 40-lead dual-in-line ceramic package, contains the arithmetic logic unit (ALU), control logic, and various working registers. The Dev. No. TA6890, in a 28-lead dual-in-line ceramic package, contains the multi-purpose 16 x 16 register array, a buffer register, associated controls, and an increment/decrement circuit associated with the register array.

COSMAC is a static system; therefore, the clock input frequency can be chosen to interface with memories or I/O devices having speeds that vary over a wide range.

Architecture

The COSMAC block diagram is shown in Fig. 3. The principal feature of this system is a register array (R) consisting of sixteen, 16-bit scratchpad registers. Individual registers in the array (R) are designated (selected) by a 4-bit binary code from one of the 4-bit registers labeled N, P, and X. The contents of any register, through the 16-bit buffer register A, can be directed to any one of the following three paths:

1. the external memory (multiplexed, higher-order byte first, on to 8 memory address lines);
2. the D register (either of the two bytes can be gated to D);
3. the increment/decrement circuit where it is increased or decreased by one and stored back in the selected 16-bit register.

The three paths, depending on the nature of the instruction, may operate independently or in various combinations in the same machine cycle.

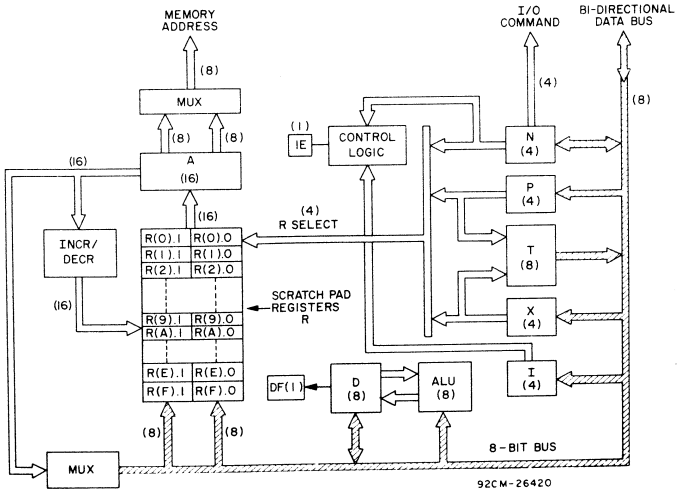


Fig. 3–COSMAC Block Diagram.

Every COSMAC instruction consists of two 8-clock-pulse machine cycles. The first cycle is the fetch cycle, and the second is the execute cycle. During the fetch cycle the four bits in the P designator select one of the 16 registers R(P) as the current program counter. This selected R(P) contains the address of the memory location from which the instruction is to be fetched. When the instruction is read out from the memory, the higher-order 4 bits of the instruction byte are loaded into the I register and the lower-order 4 bits are fed to the N register. The content of the program counter is automatically incremented by one so that R(P) is now "pointing" to the next byte in the memory.

The X designator selects one of the 16 registers R(X) to "point" to the memory for an operand (or data).

The N designator can perform the following five functions depending on the type of instruction fetched:

1. designate one of the 16 registers in R to be acted upon during register operations;
2. indicate to the I/O devices a command code or device-selection code for peripherals;
3. indicate the specific operation to be executed during the ALU instructions, types of tests to be performed during the Branch instructions, or operating modes of interrupt handling instructions;
4. indicate the value to be loaded into P to designate a new register to be used as the program counter R(P);
5. indicate the value to be loaded into X to designate a new register to be used as data pointer R(X).

The registers in R can be assigned by a programmer in three different ways: as program counters, as data pointers, or as scratch-pad locations (data registers) to hold two bytes of data.

Program Counters

Any register can be the main program counter; the address of the selected register is held in the P designator. The other registers in R can be used as subroutine program counters. By a single instruction the contents of the P register can be changed to effect a "call" to a subroutine. When interrupts are being serviced, register R(1) is used as the program counter for the interrupt servicing routine. At all other times the register designated as program counter is at the discretion of the user.

Data Pointers

The registers in R may be used as data pointers to indicate where the data (operand) is located in the memory. The register designated by X [i.e., R(X)] points to the operand for the following instructions (see Table I):

1. ALU operations F0 through F7;
2. output instructions 60 through 67;
3. input instructions 68 through 6F.

The register designated by N [i.e., R(N)] points to the operand for the "load D from memory" instruction 4N and the "store D" instruction 5N. The register designated by P (i.e., the program counter) is used as the data pointer for ALU instructions F8 through FF. During these instruction executions the operation is referred to as "data immediate".

Another important use of R as a data pointer supports the built-in Direct-Memory-Access (DMA) function. When a DMA-in or DMA-out request is received, one machine cycle is "stolen". This operation occurs at the end of the execute machine cycle in the current instruction. Register R(O) is always used as the data pointer during the DMA operation (hardwired). The data is read from or written into the memory location (depending on out or in request) pointed to by the R(O) register. At the end of the transfer, R(O) is incremented by one so that the processor is ready to act upon the next DMA byte transfer request. This feature in the COSMAC architecture saves a substantial amount of logic when fast exchanges of blocks of data are required, such as with magnetic discs of during CRT-display-refresh cycles.

A program load facility, using this DMA channel, is provided to enable users to load programs into the memory. This facility provides a simple, one-step means for initially entering programs into the microprocessor system and eliminates the requirement for specialized "bootstrap" ROM's.

Data Registers

When registers in R are used to store bytes of data, four instructions are provided which allow D to receive from or write into either the higher-order- or lower-order-byte portions of the register designated by N. By this mechanism (together with loading by data immediate) program pointer and data pointer designations are initiated. Also, this technique allows scratch-pad registers in R to be used to hold general data. Some simple controllers can be designed which use no RAM (random access memory)—the 32-byte capacity of R suffices for program counting, data pointing, and data storing.

Interrupt Servicing

Register R(1) is always used as the program counter whenever interrupt servicing is effected. When an interrupt request comes in and the interrupt is allowed by the designer (again, nothing takes place until the end of the current execute machine cycle of the instruction is completed), the contents of the P and X registers are stored in the temporary register T, and P and X are set to new values; hex digit 1 in P and hex digit 2 in X. Interrupt enable is automatically deactivated to inhibit further interruptions. The interrupt routine is now in control; the contents of T are saved by means of a single instruction (78) in the memory location pointed to by R(X), where X = 2. At the conclusion of the interrupt, the routine restores the pre-interrupted values of P and X with a single instruction. The interrupt-enable flip-flop can be activated to permit further interrupts or can be disabled to prevent them.

COSMAC Register Summary

D	8 Bits	Data Register (Accumulator)
DF	1 Bit	Data Flag (ALU Carry)
R	16 Bits	1 of 16 Scratchpad Registers
P	4 Bits	Designates which register is Program Counter
X	4 Bits	Designates which register is Data Pointer
N	4 Bits	Low-order Instruction Digit
I	4 Bits	High-order Instruction Digit
T	8 Bits	Holds old X, P after Interrupt
IE	1 Bit	Interrupt Enable

Interrupt Action: X and P are stored in T after executing current instruction; designator P is set to 1; designator X is set to 2; interrupt enable is reset to 0 (inhibit); and the interrupt request is serviced.

DMA Action: Finish executing current instruction; R(O) points to memory area for data transfer; data is loaded into or read out of memory; and increment R(O).

Note: In the event of concurrent DMA and INTERRUPT requests, DMA has priority.

Instruction Set

The COSMAC instruction summary is given in Table I. Hexadecimal notation is used to refer to the 4-bit binary codes. Many of the instructions have been discussed in the Architecture section. Symbols used are:

R(W): Register designated by W, where W = N, or X, or P
R(W).0: Lower-order byte of R(W)
R(W).1: Higher-order byte of R(W)

Operation Notation

M(R(N)) → D;R(N) + 1

This notation means: The memory byte pointed to by R(N) is loaded into D, and R(N) is incremented by 1.

Table I — Instruction Summary

Register Operations

(Code; Assembler Mnemonic (Note); Name; Operation)

I	N			
1	N	INC	INCREMENT	R(N)+1
2	N	DEC	DECREMENT	R(N)−1
8	N	GLO	GET LO	R(N).0→D
9	N	GHI	GET HI	R(N).1→D
A	N	PLO	PUT LO	D→R(N).0
B	N	PHI	PUT HI	D→R(N).1

N=0,1,2, . . .,9,A,B, . . .,E,F (Hexadecimal Notation)

Memory Reference

I	N			
4	N	LDA	LOAD ADV	M(R(N))→D;R(N)+1
5	N	STR	STORE	D→M(R(N))

ALU Operations

I	N			
F	0	LDX	LOAD BY X	M(R(X))→D
F	1	OR	OR	M(R(X)) vD→D
F	2	AND	AND	M(R(X))·D→D
F	3	XOR	EXCL.OR	M(R(X))⊕ D→D
* F	4	ADD	ADD	M(R(X))+D→D;C→DF
* F	5	SD	SUBTRACT D	M(R(X))−D→D;C→DF
* F	6	SHR	SHIFT RIGHT	SHIFT D RIGHT; LSB→DF;0→MSB
* F	7	SM	SUBTRACT M	D−M(R(X))→D;C→DF
F	8	LDI	LOAD IMM	M(R(P))→D;R(P)+1
F	9	ORI	OR IMM	M(R(P)) vD→D;R(P)+1
F	A	ANI	AND IMM	M(R(P))·D→P;R(P)+1
F	B	XRI	EXCL.OR IMM	M(R(P))⊕ D→D; R(P)+1
* F	C	ADI	ADD IMM	M(R(P))+D→D; C→DF;R(P)+1
* F	D	SDI	SUBT D IMM	M(R(P))−D→D; C→DF;R(P)+1
* F	F	SMI	SUBT M IMM	D−M(R(P))→D; C→DF;R(P)+1

Branching

I	N			
3	0	BR	UNCOND.BR.	M(R(P))→R(P).0
3	2	BZ	BR.IF D=00	M(R(P))→R(P).0 IF D=00/R(P)+1
3	3	BDF	BR.IF DF=1	M(R(P))→R(P).0 IF DF=1/R(P)+1
3	4	B1	BR.IF EF1=1	M(R(P))→R(P).0 IF EF1=1/R(P)+1
3	5	B2	BR.IF EF2=1	M(R(P))→R(P).0 IF EF2=1/R(P)+1
3	6	B3	BR.IF EF3=1	M(R(P))→R(P).0 IF EF3=1/R(P)+1
3	7	B4	BR.IF EF4=1	M(R(P))→R(P).0 IF EF4=1/R(P)+1
3	8	SKP	SKIP	R(P)+1
3	A	BNZ	BR.IF D≠00	M(R(P))→R(P).0 IF D≠00/R(P)+1
3	B	BNF	BR.IF DF=0	M(R(P))→R(P).0 IF DF=0/R(P)+1
3	C	BN1	BR.IF EF1=0	M(R(P))→R(P).0 IF EF1=0/R(P)+1
3	D	BN2	BR.IF EF2=0	M(R(P))→R(P).0 IF EF2=0/R(P)+1
3	E	BN3	BR.IF EF3=0	M(R(P))→R(P).0 IF EF3=0/R(P)+1
3	F	BN4	BR.IF EF4=0	M(R(P))→R(P).0 IF EF4=0/R(P)+1

Control

I	N			
0	0	IDL	IDLE	WAIT FOR INTERRUPT/ DMA-IN/ DMA-OUT
D	N	SEP	SET P	N→P
E	N	SEX	SET X	N→X
7	0	RET	RETURN	M(R(X))→ X, P; R(X)+1;1→IE
7	1	DIS	DISABLE	M(R(X))→X, P; R(X)+1;0→IE
7	8	SAV	SAVE	T→M(R(X))

Input-Output Byte Transfer

I	N			
6	0	OUT 0	OUTPUT 0	M(R(X))→BUS; R(X)+1;N=0
6	1	OUT 1	OUTPUT 1	M(R(X))→BUS; R(X)+1;N=1
6	2	OUT 2	OUTPUT 2	M(R(X))→BUS; R(X)+1;N=2
6	3	OUT 3	OUTPUT 3	M(R(X))→BUS; R(X)+1;N=3
6	4	OUT 4	OUTPUT 4	M(R(X))→BUS; R(X)+1;N=4
6	5	OUT 5	OUTPUT 5	M(R(X))→BUS; R(X)+1;N=5
6	6	OUT 6	OUTPUT 6	M(R(X))→BUS; R(X)+1;N=6
6	7	OUT 7	OUTPUT 7	M(R(X))→BUS; R(X)+1;N=7
6	8	INP 0	INPUT 0	BUS→M(R(X)); N=8
6	9	INP 1	INPUT 1	BUS→M(R(X)); N=9
6	A	INP 2	INPUT 2	BUS→M(R(X)); N=A
6	B	INP 3	INPUT 3	BUS→M(R(X)); N=B
6	C	INP 4	INPUT 4	BUS→M(R(X)); N=C
6	D	INP 5	INPUT 5	BUS→M(R(X)); N=D
6	E	INP 6	INPUT 6	BUS→M(R(X)); N=E
6	F	INP 7	INPUT 7	BUS→M(R(X)); N=F

*These are the only operations that modify DF. DF is set or reset by an ALU carry during add or subtract. Subtraction is by 2's complement: A−B = A+$\overline{B}$+1.

Note: This type of abbreviated nomenclature is used when programs are designed with the aid of the COSMAC Assembler Simulator/Debugger System,which is available on commercial timesharing systems. Refer to "Program Development Guide for the COSMAC Microprocessor" for details.

Test and Branch

The Test and Branch instructions can branch unconditionally, test for D=0 or D=1, test for DF=0 or DF=1, or can test the status of the four I/O flags. A "successful" branch loads the byte following the instruction into the lower-order byte position of the current program counter, effecting a branch within the current 256-byte "page" of memory. If the test to branch is not successful, the next instruction in sequence is executed.

Memory and Input/Output Device Interface

The COSMAC CPU, by providing a synchronous interface to the external controllers for I/O devices and memories, minimizes the cost of interface controllers. The I/O interface is capable of supporting devices operating in polled, interrupt-driven, or direct-memory-access modes.

$\overline{\text{CLEAR}}$ — A single short pulse is required. A momentary low on this line places COSMAC in a repeating IDLE cycle▲ with P = 0, R(O) = 0000 and IE = 1 (interrupt request allowed).

$\overline{\text{CLOCK}}$ — Single-phase clock. A typical clock frequency is 2 MHz at V_{DD} = 10 V. The clock is counted down internally to 8 clock pulses per machine cycle.

$\overline{\text{MA0}}$ to $\overline{\text{MA7}}$ (8 Memory Address Lines) — The most significant 8 bits of the memory address is multiplexed out first on these lines and held in a latch in the memory system that is set by $\overline{\text{TPA}}$. The 8 least significant bits are then multiplexed out on the same lines. The memory system always sees a 16-bit address within one memory-addressing cycle.

MWR (Write Pulse) — This pulse appears late in a memory-write cycle, after the address lines have settled down.

$\overline{\text{M READ}}$ (Read Level) — If a memory does not have a 3-state high-impedance output, $\overline{\text{M READ}}$ is useful for driving memory/bus separator gates; otherwise it is used to control 3-state outputs from the addressed

▲A repeating IDLE cycle represents an instruction halt. The processor will remain in this halt state until an I/O Request ($\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, or $\overline{\text{DMA-OUT}}$) is activated. When this request occurs, the IDLE cycle is terminated and the I/O request is serviced, and then the normal program is resumed. If a DMA request is used to bring the processor out of IDLE, it will increment the contents of R(O) by 1. The first instruction will, therefore, be fetched from memory location 0001 and not 0000. Thus, program execution begins at location 0001 with R(O) as the program counter.

memory. A low on $\overline{\text{M READ}}$ indicates a read cycle; the low $\overline{\text{M READ}}$ line enables the memory-output-bus gates during the read cycle (see Fig. 1, Timing Diagram).

$\overline{\text{BUS 0}}$ to $\overline{\text{BUS 7}}$ (Data Bus) – 8-bit bidirectional data bus lines. These lines are used for transferring data between the memory, the microprocessor, and I/O devices.

$\overline{\text{N0}}$ to $\overline{\text{N3}}$ (I/O Command) – Issued by an I/O instruction. They are interpreted by I/O control logic to move data between the memory and the I/O interface (discussed in the Architecture section). These lines can be used to issue command codes or device selection codes to the I/O devices (independently or combined with the memory byte on the data bus when an I/O instruction is executed).

$\overline{\text{EF0}}$ to $\overline{\text{EF3}}$ (4 Flags) – These levels enable the I/O controllers to transfer status information to the processor. These levels can be tested by the conditional branch instructions. They can be used in conjunction with the INTERRUPT request line to establish interrupt priorities. These flags can also be used by I/O devices to "call the attention" of the processor, in which case the program must routinely test the status of these flag(s).

$\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, $\overline{\text{DMA-OUT}}$ (3 I/O requests) – These requests were discussed in the Architecture section. They are sampled by COSMAC in the interval between the leading edge of TPB and the leading edge of TPA. The DMA request has a higher priority than the INTERRUPT request.

$\overline{\text{SC0}}$, $\overline{\text{SC1}}$ (2 State Code Lines) – These two lines indicate to the I/O controllers that the CPU is: 1) executing an I/O instruction, 2) acknowledging an interrupt request, 3) processing a DMA request, or 4) none of the above states. The levels of state code are tabulated below.

State Type	State Code Lines $\overline{\text{SC1}}$	$\overline{\text{SC0}}$
S2 (DMA)	L	H
S3 (Interrupt)	L	L
S0 (Fetch)	H	H
S1 • [I=6] (I/O Instruction Execute)	H	L
S1 • [I≠6] (All other instructions)	H	H

All these states last one machine cycle. They may be assumed valid at $\overline{\text{TPA}}$.

$\overline{\text{TPA}}$, $\overline{\text{TPB}}$ (2 Timing Pulses) – Both occur once in each machine cycle ($\overline{\text{TPB}}$ follows $\overline{\text{TPA}}$). They are used by I/O controllers to interpret codes and to time interaction with the data bus. The trailing edge of $\overline{\text{TPA}}$ is used by the memory system to latch the higher-order byte of the 16-bit memory address.

$\overline{\text{LOAD}}$ – A level that holds the CPU in IDLE-DMA mode and allows an I/O device to load the memory without the need for a "bootstrap" loader. It modifies the IDLE condition so that DMA operation does not force execution of the next instruction.

V_{DD}, V_{SS}, V_{CC} (Power Levels) – The internal voltage supply V_{DD} is isolated from the Input/Output voltage supply V_{CC} so that the processor may operate at maximum speed while interfacing with various external circuit technologies, including T^2L at 5 volts. **V_{CC} must be less than or equal to V_{DD}.**

When incorporating RCA Solid State Devices in equipment, it is recommended that the designer refer to "Operating Considerations for RCA Solid State Devices", Form No. 1CE-402, available on request from RCA Solid State Division, Box 3200, Somerville, N.J. 08876.