RCA Microprocessor Products

RCA LSI Microprocessor

The RCA Microprocessor (COSMAC) is a COS/MOS 8-bit, register-oriented central processing unit designed for use as a general-purpose computing element. COSMAC is implemented as two LSI devices, one in a 40-pin package and the other in a 28-pin package.

COSMAC is ideally suited for a wide variety of commercial, industrial, and government applications. The architecture has been designed with a total microcomputer system in mind, so that systems with maximum flexibility and minimum cost can be realized.

The CMOS technology used in COSMAC provides a high noise immunity so that the processor can operate in electrically hostile environments. The processor can be powered by unregulated power supplies over a wide operating-voltage range. It has a separate internal voltage supply, so that it can operate at maximum speed while interfacing to various external circuit technologies, including TTL. Only a single-phase system clock is required, and the processor power consumption is very low. Furthermore, COSMAC is completely static, so that its system clock can be controlled to interface with very slow memories or I/0 devices. It is capable of operating over the full −55°C to +125°C temperature range.

COSMAC provides a set of simple, easy-to-use, general-purpose instructions. One can learn to design programs for COSMAC-based products with minimum effort. Unlimited subroutine nesting is possible. The instruction set facilitates the use of interpretive macro-instructions.The on-chip scratchpad of sixteen general-purpose 16-bit registers may be used to provide multiple program counters, data pointers, and data storage. Three specific registers are treated by the hardware as implicit built-in DMA address pointer, program counter for interrupt servicing, and interrupt/subroutine stack pointer, respectively. A simple one-step program loading facility is provided on the chip. Fig. 1 shows the internal structure of COSMAC.

The COSMAC I/0 interface was designed to provide direct control of I/0 devices so that over-all system complexity and cost can be reduced. Flexible, open-ended I/0 instructions allow unlimited device attachment. The hardware I/0 interface is capable of supporting devices operating in polled, interrupt driven, and Direct Memory Access modes. Fig. 2 illustrates the general form of a system incorporating the COSMAC Microprocessor.

Design aids such as a Microprocessor hardware support kit (COSMAC Microkit) and a program development system with manuals and software are provided to help system designers in the development of Microprocessor-based products.

The following list summarizes the advanced features and operating characteristics of the COSMAC Microprocessor:

- **8-bit parallel organization**
- **8-bit bidirectional common bus for input/output and memory**
- **static COS/MOS circuitry**
- **low power consumption**
- **single voltage option**
- **4-12 Vdc operating range**
- **TTL compatibility**
- **high noise immunity**
- **single-phase clock**
- **single-pulse clear**
- **built-in program load mode**
- **standard RAM/ROM compatibility**
- **separate memory address lines**
- **direct memory addressing up to 65,536 bytes**
- **program interrupt mode**
- **program controlled interrupt mask (enable/disable)**
- **self-contained DMA channel (cycle stealing mode)**
- **flexible programmed I/0 mode**
- **four I/0 flag inputs directly testable by branch instructions**
- **separate 4-bit I/0 control code**
- **two I/0 sync pulses**
- **one-byte instruction format with two machine cycles for each instruction**
- **59 easy-to-use instructions**
- **multiple program counters**
- **multiple data registers**
- **multiple address registers**
- **add, subtract, shift, and logical operations**
- **immediate address mode**
- **indirect pointer address mode**
- **flexible subroutine nesting procedures**
- **branch and link capability**
- **same fetch and execute cycles for all instructions**

2M1122A
Printed in U.S.A. 3/75

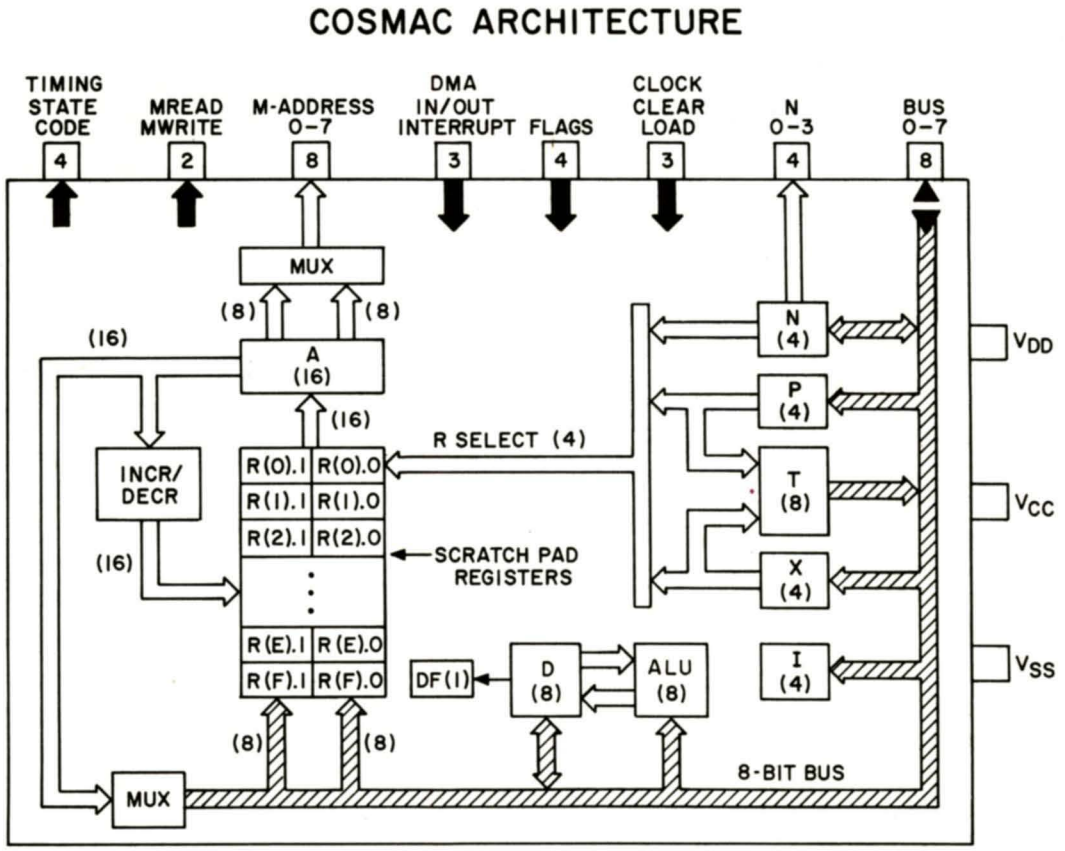


Fig. 1 – COSMAC Architecture.

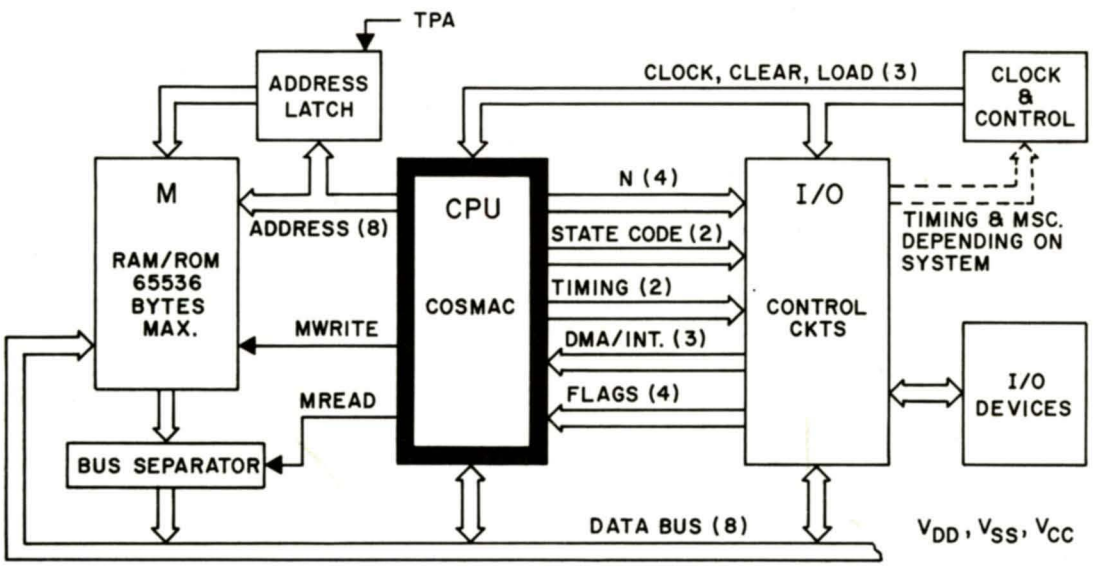
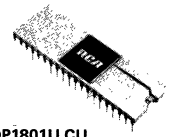


Fig. 2 – System Block Diagram.



CDP1801U,CU
40-Lead Dual-In-Line Ceramic (D)
H-1847

CDP1801R,CR
28-Lead Dual-In-Line Ceramic (D)
H-1848

RCA CDP1801, CDP1801C Microprocessor (COSMAC)

CDP1801U, CDP1801CU Microprocessor Control IC
CDP1801R, CDP1801CR 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**
- **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**
- **57 easy-to-use instructions**
- **16 x 16 matrix of registers for use as multiple program counters, data pointers, or data registers**

The RCA-CDP1801 and CDP1801C Microprocessors (COSMAC) are LSI COS/MOS, 8-bit register-oriented central-processing units (CPU) designed for use as general-purpose computing or control elements in a wide range of stored-program systems or products. The CDP1801 and CDP1801C each comprise two units; the control unit designated CDP-1801U, CDP1801CU and the register unit designated CDP1801R, CDP1801CR.

The CDP1801 is functionally identical to the CDP1801C. The CDP1801 has an operating voltage range of 3 to 12 volts; the CDP1801C, an operating voltage range of 4 to 6 volts.

These microprocessors include all of the circuits required for fetching, interpreting, and executing instructions which have been stored in standard types of memories. Extensive input/output (I/O) control features are also provided to facilitate system design.

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

The CDP1801U and CDP1801CU control units are supplied in a 40-lead hermetic ceramic dual-in-line package (D suffix) and in chip form (H suffix). The CDP1801R and CDP1801CR register units are supplied in a 28-lead hermetic ceramic dual-in-line package (D suffix) and in chip form (H suffix). For ordering information, see dimensional outline page.

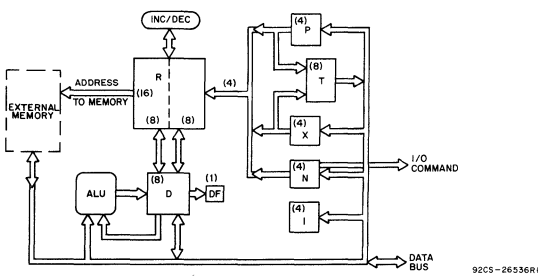


Fig. 1—CDP1801, CDP1801C Microprocessor data flow chart.

Printed in USA/9-75

MAXIMUM RATINGS, *Absolute-Maximum Values*

Storage-Temperature Range (T_{stg})
..... −65 to +150°C
Operating-Temperature Range (T_A)
..... −55 to +125°C
DC Supply-Voltage Range (V_{CC}, V_{DD})
(All voltage values referenced to V_{SS} terminal)
$V_{CC} \leqslant V_{DD}$:
CDP1801..... 0.5 to +15 V
CDP1801C ..... 0.5 to +7 V
Power Dissipation Per Package (P_D):
For T_A = −55 to +100°C
..... 500 mW
For T_A = +100 to +125°C
..... Derate Linearly to 200 mW
Device Dissipation Per Output Transistor:
For T_A = −55°C to +125°C ..... 100 mW
Input Voltage Range, All Inputs
..... −0.5 to V_{DD} +0.5 V
Lead Temperature (During Soldering):
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm)
from case for 10 s max. .....+265°C

OPERATING CONDITIONS AT T_A = 25°C Unless Otherwise Specified

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

CHARACTERISTIC	CONDITIONS		LIMITS				UNITS
	V_{CC}[1] (V)	V_{DD} (V)	CDP1801 Min.	CDP1801 Max.	CDP1801C Min.	CDP1801C Max.	
Supply-Voltage Range (At T_A = Full Package–Temperature Range)	–	–	3	12	4	6	V
Recommended Input Voltage	–	–	V_{SS}	V_{CC}	V_{SS}	V_{CC}	V
Clock Input Rise or Fall Time, t_r,t_f	3 - 15	3 - 15	–	15	–	15	μs
Instruction Time (See Fig. 4)	5	5	16	–	16	–	μs
	5 - 10	10	6	–	–	–	
Clock Input Frequency, f_{CL}	5	5	DC	1	DC	1	MHz
	5 - 10	10	DC	3	–	–	
Clock Pulse Width, t_{WL}, t_{WH}	5	5	500	–	500	–	ns
	5 - 10	10	160	–	–	–	
Clear Pulse Width	5	5	500	–	500	–	ns
	5 - 10	10	160	–	–	–	
Data Hold Time, t_{DH}	5	5	0	–	0	–	ns
	5 - 10	10	0	–	–	–	

Notes:

1. $V_{CC} \leqslant V_{DD}$. For CD1801C $V_{DD} = V_{CC} = 5$ volts.
2. Because a large number of nodes may be switching simultaneously, a 0.1 μF by-pass capacitor is recommended in the power supply.
3. In order to maintain proper circuit operation, the CDP1801 intra-unit wiring capacitance should be less than 25 pF.

ELECTRICAL CHARACTERISTICS at $T_A = 25^oC$

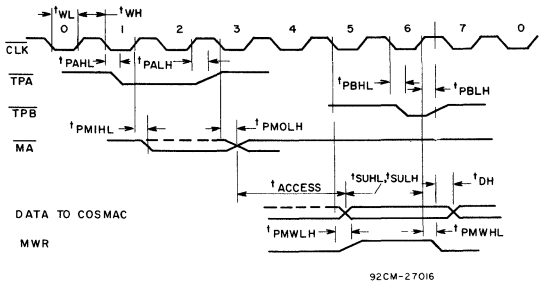
CHARACTERISTIC	CONDITIONS V_O (V)	CONDITIONS $V_{CC}=V_{DD}$ (V)	CDP1801U, CDP1801R LIMITS Min.	CDP1801U, CDP1801R LIMITS Typ.	CDP1801U, CDP1801R LIMITS Max.	CDP1801CU, CDP1801CR LIMITS Min.	CDP1801CU, CDP1801CR LIMITS Typ.	CDP1801CU, CDP1801CR LIMITS Max.	UNITS
Static									
Quiescent Device Current, I_L (See Figs. 11, 12)	–	5	–	0.01	0.1	–	0.01	0.5	mA
	–	10	–	0.01	0.5	–	–	–	
	–	15	–	0.05	1.0	–	–	–	
Output Voltage: Low-Level, V_{OL}	–	5	–	0	0.05	–	0	0.05	V
	–	10	–	0	0.05	–	–	–	
High-Level, V_{OH}	–	5	4.95	5	–	4.95	5	–	V
	–	10	9.95	10	–	–	–	–	
Noise Immunity: Inputs Low, V_{NL}	0.5	5	1.5	2.25	–	1.5	2.25	–	V
	–1	10	3.0	3.45	–	–	–	–	
Inputs High, V_{NH}	4.5	5	1.5	2.25	–	1.5	2.25	–	V
	9	10	3.0	3.45	–	–	–	–	
Noise Margin: Inputs Low, V_{NML}	1	5	1	–	–	1	–	–	V
	1	10	1	–	–	–	–	–	
Inputs High, V_{NMH}	4	5	1	–	–	1	–	–	V
	9	10	1	–	–	–	–	–	
Output Drive Current●: N-Channel (Sink), I_DN (See Figs. 7, 8)	0.4	5	1.6	3.2	–	1.6	3.2	–	mA
	0.5	10	3.6	7.2	–	–	–	–	
P-Channel (Source), I_DP (See Figs. 5, 6)	2.5	5	–0.8	–1.6	–	–0.8	–1.6	–	mA
	4.6	5	0	0	–	0	0	–	
	9.5	10	–0.45	–0.9	–	–	–	–	
Input Leakage Current (Any Input), I_{IL}, I_{IH} (See Figs. 13, 14)	–	5	–	–	–	–	$\pm10^{-5}$	±1	μA
	–	15	–	$\pm10^{-5}$	±1	–	–	–	
Dynamic at $T_A = 25^oC$, Input $t_r, t_f = 20$ ns, $C_L = 50$ pF									
Propagation Delay Times: (See Fig. 2) Clock to TPA t_{PAHL}	–	5	–	685	–	–	685	–	ns
	–	10	–	240	–	–	–	–	
t_{PALH}	–	5	–	720	–	–	720	–	ns
	–	10	–	250	–	–	–	–	
Clock to TPB t_{PBHL}	–	5	–	430	–	–	430	–	ns
	–	10	–	190	–	–	–	–	
t_{PBLH}	–	5	–	650	–	–	650	–	ns
	–	10	–	230	–	–	–	–	
Clock-to-Memory Address MA8 to MA15 t_{PMIHL}	–	5	–	900	–	–	900	–	ns
	–	10	–	340	–	–	–	–	
MA0 to MA7 t_{PMOLH}	–	5	–	870	–	–	870	–	ns
	–	10	–	300	–	–	–	–	

● Values are for signal lines going to external connections.

ELECTRICAL CHARACTERISTICS at $T_A = 25^oC$ (CONT'D)

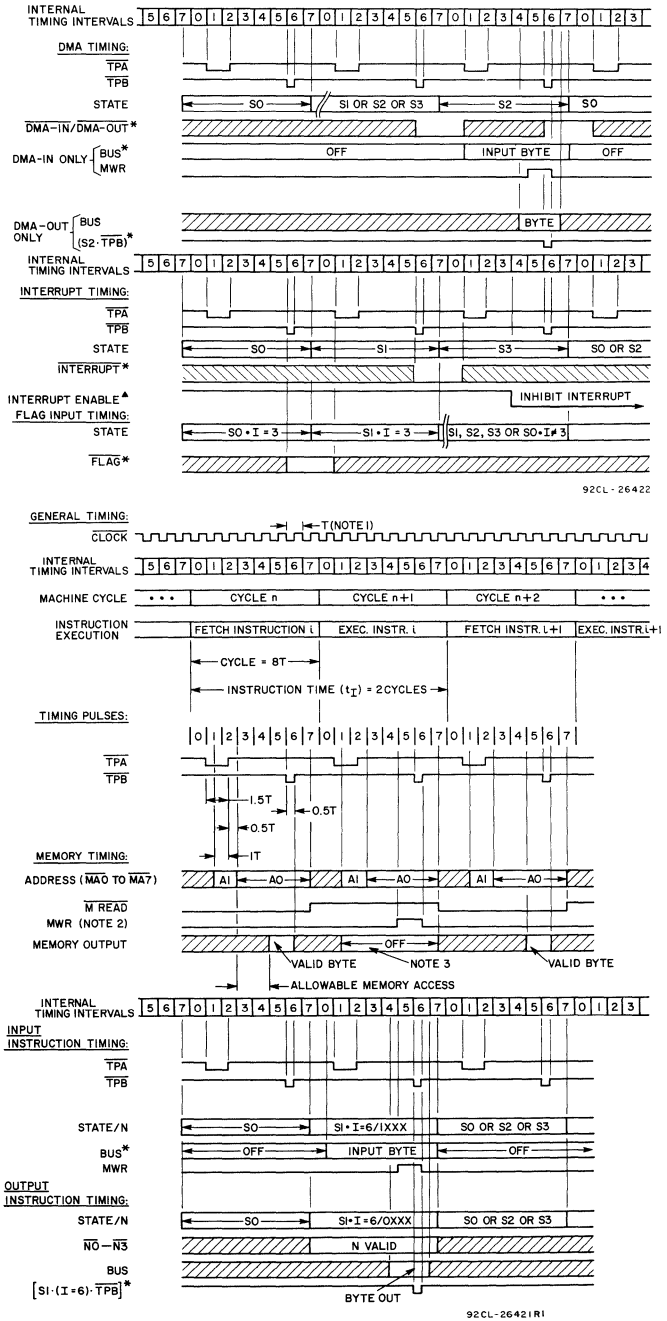
CHARACTERISTIC	CONDITIONS V_O (V)	CONDITIONS $V_{CC}=V_{DD}$ (V)	CDP1801U, CDP1801R LIMITS Min.	CDP1801U, CDP1801R LIMITS Typ.	CDP1801U, CDP1801R LIMITS Max.	CDP1801CU, CDP1801CR LIMITS Min.	CDP1801CU, CDP1801CR LIMITS Typ.	CDP1801CU, CDP1801CR LIMITS Max.	UNITS
Dynamic at $T_A = 25^oC$, Input $t_r, t_f = 20$ ns, $C_L = 50$ pF									
Data Setup Time: (See Fig. 2) t_{SULH}	–	5	–	750	–	–	750	–	ns
t_{SUHL}	–	10	–	450	–	–	–	–	
Clock-to-Memory Write Time (See Fig. 2) t_{PMWHL}	–	5	–	780	–	–	780	–	ns
t_{PMWLH}	–	10	–	270	–	–	–	–	
Device Dissipation (Total, Both Units), P_D OP CODE = 00 f_{CL} = 1.0 MHz	–	5	–	3	–	–	3	–	mW
f_{CL} = 3.0 MHz	–	10	–	32	–	–	–	–	
Transition Time●: (See Figs. 9, 10) t_{TLH}	–	5	–	170	–	–	170	–	ns
	–	10	–	100	–	–	–	–	
t_{THL}	–	5	–	30	–	–	30	–	ns
	–	10	–	20	–	–	–	–	
Input Capacitance, C_I	Any Input		–	5	–	–	5	–	pF

● Values are for signal lines going to external connections.



Notes: 1. This timing diagram is used to show signal relationships only and does not represent any specific machine cycle.
2. All measurements are referenced to 50% point of the waveforms.

Fig. 2—Timing waveforms.



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.

* = SIGNAL GENERATED BY USER
▲ = INTERNAL TO COSMAC

Fig. 3—CDP1801 and CDP1801C Microprocessor timing diagram.

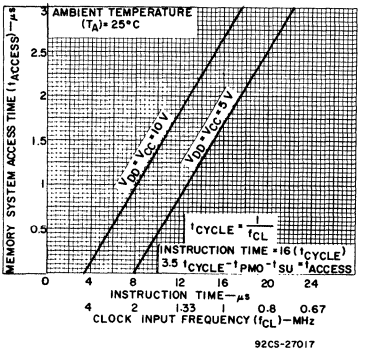


Fig. 4—Typical instruction time vs. memory system access time.

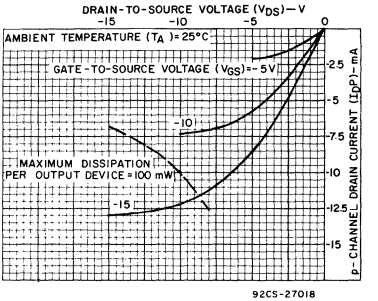


Fig. 5—Typical output-P-channel drain characteristics.

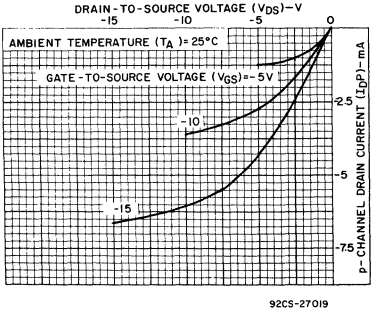


Fig. 6—Minimum output-P-channel drain characteristics.

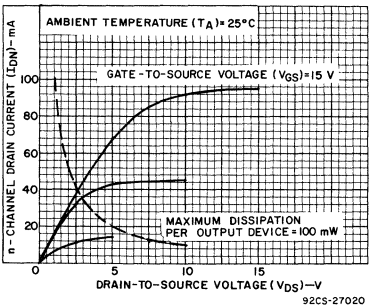


Fig. 7—Typical output-N-channel drain characteristics.

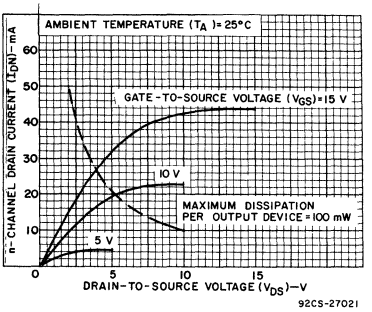


Fig. 8—Minimum output-N-channel drain characteristics.

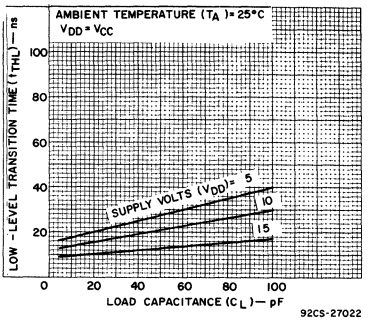


Fig. 9—Typical high-to-low level transition time vs. load capacitance.

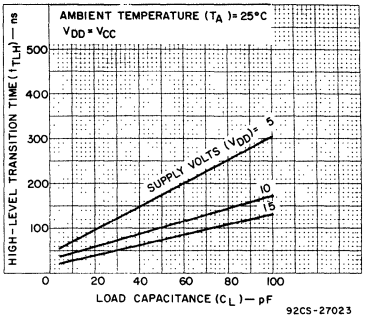


Fig. 10—Typical low-to-high level transition time vs. load capacitance.

TEST CIRCUITS

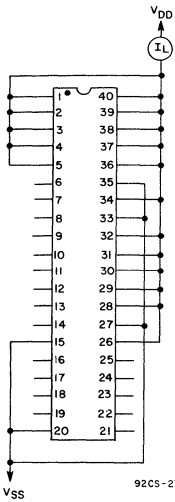


Fig. 11—CDP1801U, CDP1801CU quiescent device current.

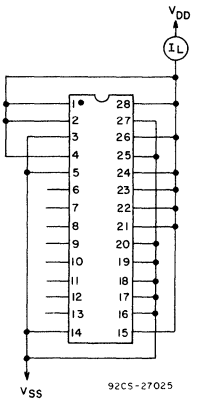


Fig. 12—CDP1801R, CDP1801CR quiescent device current.

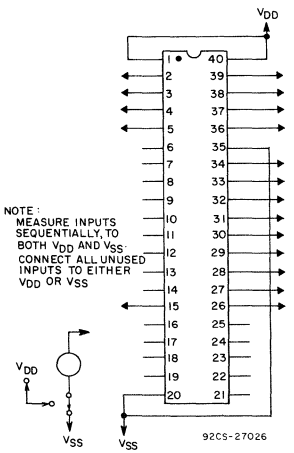


Fig. 13—CDP1801U, CDP1801CU input leakage current.

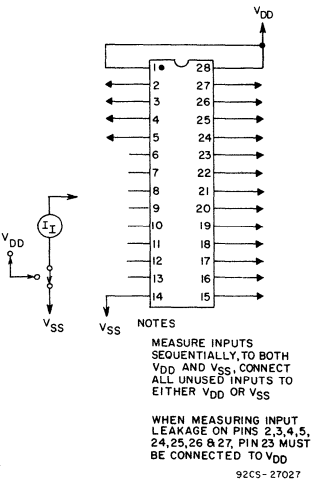


Fig. 14—CDP1801R, CDP1801CR input leakage current.

The following is a brief description of the COSMAC microprocessor. For more detailed information see the following publication: "User Manual for the COSMAC Microprocessor"—MPM-101. A Hardware Kit, CDP-18S001 including the CDP1801 Microprocessor and support logic is also available.

The RCA Microprocessor (COSMAC) is implemented by two COS/MOS chips. The RCA CDP1801U and CDP1801CU, in a 40-lead dual-in-line ceramic package, each contain the arithmetic logic unit (ALU), control logic, and various working registers. The RCA CDP-1801R and CDP1801CR, in a 28-lead dual-in-line ceramic package, each contain 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. Also, the input clock may be stopped indefinitely without loss of information.

Architecture

The COSMAC block diagram is shown in Fig. 15. 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 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.

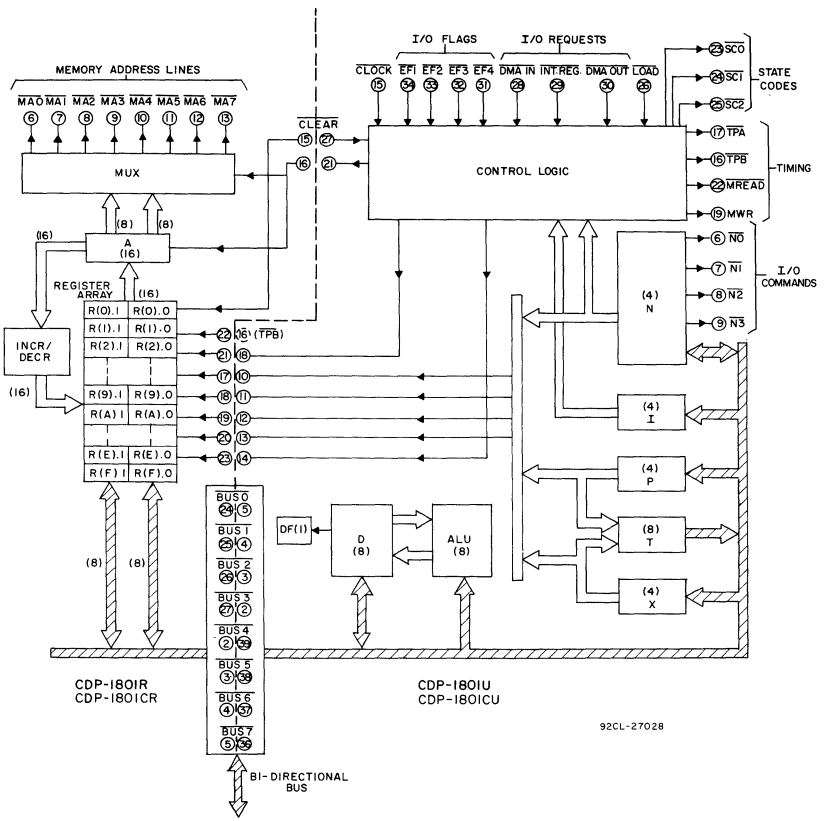


Fig. 15—CDP1801, CDP1801C Microprocessor block diagram.

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) in certain ALU or I/O operations.

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 scratchpad 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 61 through 67;
3. input instructions 69 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(0) is always used as the data pointer during the DMA operation. The data is read from or written into the memory location pointed to by the R(0) register, whether the request is for "in" or "out". At the end of the transfer, R(0) 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 or 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 scratchpad registers in R to be used to hold general data.

Interrupt Servicing

Register R(1) is always used as the program counter whenever interrupt servicing is initiated. When an interrupt request comes in and the interrupt is allowed by the designer

Interrupt Servicing – Cont'd

(again, nothing takes place until the end of the current execute machine cycle of the instruction is completed), the contents of the X and P registers are stored in the temporary register T, and X and P are set to new values; hex digit 2 in X and hex digit 1 in P. 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 X and P 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	Holds Low-order Instr. Digit
I	4 Bits	Holds High-order Instr. Digit
T	8 Bits	Holds old X, P after Interrupt (X is high byte)
IE	1 Bit	Interrupt Enable

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

DMA Action: Finish executing current instruction; R(0) points to memory area for data transfer; data is loaded into or read out of memory; and increment R(0).

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.

Slash (/) mark in Operations Column indicates else or otherwise.

Table I — Instruction Summary

Register Operations

(Code (Note 1), Assembler Mnemonic (Note 2), 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)) ∨D→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)) ∨D→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

*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.

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))

Note 1: The use of non-specified machine codes is not recommended.

Input-Output Byte Transfer

I	N	(Note 3)		
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	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

Note 2: 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", MPM-102, for details.

Note 3: When executing any of the 69 to 6F instructions, the contents of the D register may be altered.

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.

SIGNAL DESCRIPTION

Signal		Function
$\overline{\text{CLEAR}}$	–	A single negative 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 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)	–	A positive pulse appearing in a memory-write cycle, after the address lines have settled down.
$\overline{\text{M READ}}$ (Read Level)	–	A low level on $\overline{\text{M READ}}$ indicates a memory read cycle. It can be used to control three-state outputs from the addressed memory which may have a common data input and output bus. If a memory does not have a three-state high-impedance output, $\overline{\text{M READ}}$ is useful for driving memory/bus separator gates.

▲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. It is recommended that MEM.LOC.0000 not be used by the program.

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

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

$\overline{N0}$ to $\overline{N3}$ (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). N bits are set at the end of every S0 cycle.

$\overline{EF1}$ to $\overline{EF4}$ (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). The flag(s) are set at the end of every S0 cycle.

$\overline{INTERRUPT}$, $\overline{DMA\text{-}IN}$, $\overline{DMA\text{-}OUT}$ (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.

$\overline{SC0}$, $\overline{SC1}$, $\overline{SC2}$ (3 State Code Lines) – These three lines indicate to the I/O controllers that the CPU is: 1)processing a DMA request, 2)acknowledging an interrupt request, 3)fetching an instruction, 4)executing an I/O instruction, or 5)all other instruction executions. The levels of state code are tabulated below.

State Type	State Code Lines $\overline{SC2}$	$\overline{SC1}$	$\overline{SC0}$
S2 (DMA)	H	L	H
S3 (Interrupt)	H	L	L
S0 (Fetch)	L	H	H
S1 • [I=6] (I/O Instruction Execute)	H	H	L
S1 • [I≠6] (All other instructions)	H	H	H

All these states last one machine cycle. They may be assumed valid at TPA. H = V_{CC}, L = V_{SS}.

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

$\overline{LOAD}$ – A low level that holds the CPU in IDLE 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}.**

COSMAC Microprocessor State Transitions

Fig. 16 shows the CDP1801 and CDP1801C Microprocessor state transitions. Each machine cycle requires the same period of time—8 clock cycles. The execution of each COSMAC instruction requires two machine cycles, S0 followed by S1. S2 is the response to a DMA request and S3 is the interrupt response discussed in the preceding text.

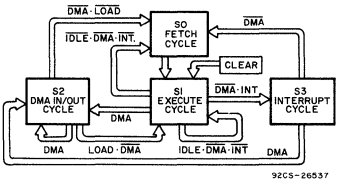


Fig. 16—CDP1801, CDP1801C Microprocessor state transitions.

OPERATING AND HANDLING CONSIDERATIONS FOR CDP1801 MICROPROCESSOR

1. Handling

All inputs and outputs of this device have a network for electrostatic protection during handling. Recommended handling practices for COS/MOS devices are described in ICAN-6000 "Handling and Operating Considerations for MOS Integrated Circuits", available on request from RCA Solid State Division, Box 3200, Somerville, N.J. 08876.

2. Operating

Operating Voltage

During operation near the maximum supply voltage limit, care should be taken to avoid or suppress power supply turn-on and turn-off transients, power supply ripple, or ground noise; any of these conditions must not cause $V_{DD}-V_{SS}$ to exceed the absolute maximum rating. V_{CC} must be less than equal to V_{DD}. Power supplies should be sequenced to insure compliance.

Input Signals

To prevent damage to the input protection circuit, input signals should never be greater than V_{DD} nor less than V_{SS}. Input currents must not exceed 10 mA even when the power supply is off.

Unused Inputs

A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

Output Short Circuits

Shorting of outputs to V_{DD} or V_{SS} may damage COS/MOS devices by exceeding the maximum device dissipation.

OPERATING AND HANDLING CONSIDERATIONS FOR CDP1801UH AND CDP1801RH CHIPS

Mounting Considerations

All COS/MOS chips are non-gold backed and require the use of epoxy mounting. DuPont No. 5504A conductive silver paste or equivalent recommended. In any case the manufacturer's recommendations for storage and use should be followed. If DuPont No. 5504A paste is used, the bond should be cured at temperatures between 185°C and 200° for 75 minutes.

In COS/MOS circuits P-channel substrates are connected to V_{DD}, therefore, when chips are mounted and a conductive paste is used care must be taken to keep the active substrate isolated from ground or other circuit elements.

Packing, Shipping, and Storage Criteria

Solid-state chips, unlike packaged devices, are non-hermetic devices, normally fragile and small in physical size, and therefore, require special handling considerations as follows:

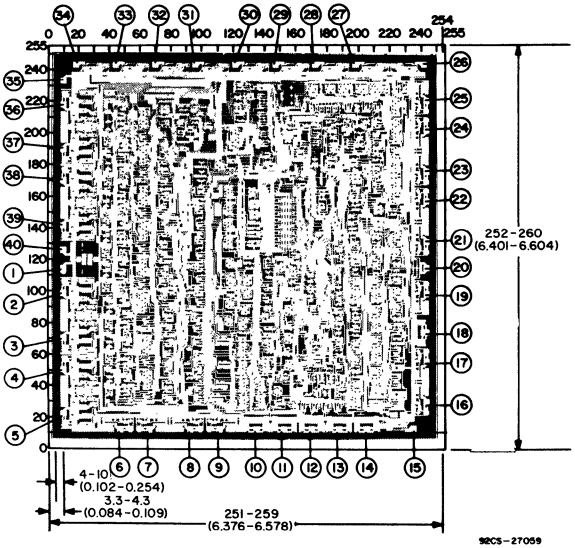
1. Chips must be stored under proper conditions to insure that they are not subjected to a moist and/or contaminated atmosphere that could alter their electrical, physical, or mechanical characteristics. After the shipping container is opened, the chip must be stored under the following conditions:
   A. Storage temperature, 40°C max.
   B. Relative humidity, 50% max.
   C. Clean, dust-free environment.
2. The user must exercise proper care when handling chips to prevent even the slightest physical damage to the chip.
3. During mounting and lead bonding of chips the user must use proper assembly techniques to obtain proper electrical, thermal, and mechanical performance.
4. After the chip has been mounted and bonded, any necessary procedure must be followed by the user to insure that these non-hermetic chips are not subjected to moist or contaminated atmosphere which might cause the development of electrical conductive paths across the relatively small insulating surfaces. In addition, proper consideration must be given to the protection of these devices from other harmful environments which could conceivably adversely affect their proper performance.

These unmounted and unencapsulated chips are tested electrically and visually inspected to meet RCA's specifications when they are shipped by RCA. Written notification of non-conformance to such specifications must be made to RCA within 90 days of the date of the shipment by RCA. After shipment from RCA, RCA assumes no responsibility for chips that have been subjected to further

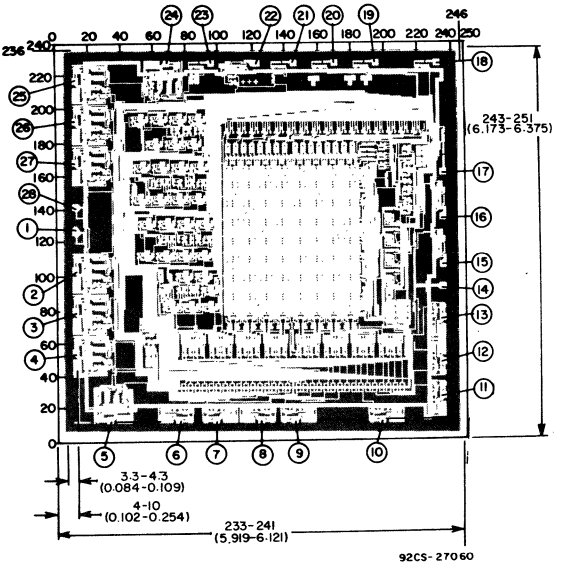
OPERATING AND HANDLING CONSIDERATIONS FOR CDP1801UH, CDP1801RH CHIPS – CONT'D

processing, such as, but not limited to, lead bonding or chip mounting operations. RCA reserves the right to change the chip design and processing without notification.

For additional recommended handling practices for COS/MOS chips, refer to ICAN-6000 "Handling and Operating Considerations for MOS Integrated Circuits".

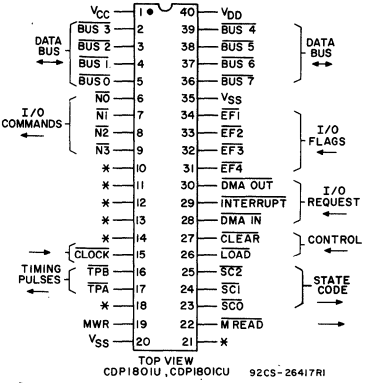


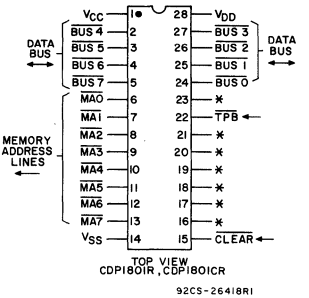
Dimensions and Pad Layout for CDP1801UH Chip



Dimensions and Pad Layout for CDP1801RH Chip

Dimensions in parentheses are in millimeters and are derived from the basic inch dimensions as indicated. Grid graduations are in mils (10^{-3} inch).





Package Interconnections

Pin Terminals To:						* 10	* 11	* 12	* 13	* 14		* 18	* 21					
CDP1801U,CU	1	2	3	4	5	10	11	12	13	14	16	18	21	27	36	37	38	39	40
CDP1801R,CR	1	27	26	25	24	17	18	19	20	23	22	21	16	15	5	4	3	2	28
						* 17	* 18	* 19	* 20	* 23		* 21	* 16						

* 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-shifing capability. All outputs have the same drive capability shether they have three-state outputs or not.
4. For the CDP1801C, V_{CC} must be connected to V_{DD}.

Fig. 17—Terminal assignment diagrams.

DIMENSIONAL OUTLINES

CDP1801R, CDP1801CR 28-Lead Ceramic

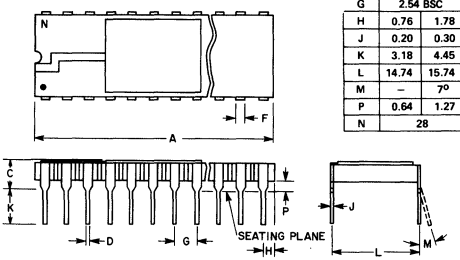
DIM.	MILLIMETERS MIN.	MILLIMETERS MAX.	INCHES MIN.	INCHES MAX.
A	35.06	36.06	1.380	1.420
C	2.16	3.68	0.085	0.145
D	0.43	0.56	0.017	0.023
F	1.27 REF.		0.050 REF.	
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	3.18	4.45	0.125	0.175
L	14.74	15.74	0.580	0.620
M	–	7°	–	7°
P	0.64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

92CM-26419

CDP1801U, CDP1801CU 40-Lead Ceramic

DIM.	MILLIMETERS MIN.	MILLIMETERS MAX.	INCHES MIN.	INCHES 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	

92CM-27029



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).

ORDERING INFORMATION

The CDP1801 and CDP1801C are 2-package microprocessors consisting of a control unit, CDP1801U or CDP1801CU, and a register unit, CDP1801R or CDP1801CR.

When ordering both the control unit and the register unit, **request the CDP1801 or the CDP1801C.** When ordering either the control unit or the register unit, add the appropriate suffix letter to the type number as follows:

Control Unit: CDP1801**U**, CDP1801**CU**
Register Unit: CDP1801**R**, CDP1801**CR**

When ordering a chip, add the suffix letter H to the specific type number as follows:

Control Unit Chip: CDP1801**UH**
Register Unit Chip: CDP1801**RH**

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.

COSMAC Software Development Package

THE ASSEMBLER

The RCA symbolic assembly program (assembler) is a computer program that assists the designer in preparation of machine code programs for the RCA COSMAC microprocessor. The assembler allows the system designer to write his program (source program) in easy-to-remember symbolics. The assembler converts these English-like mnemonics into machine-langauge binary words (object program). It also does the tedious "book-keeping", keeping track of cross references within the program, and facilitates the combining of subprograms to form larger programs.

The assembler operates on a "one-for-one" basis, in that each phrase of a statement in the source program translates directly into a specific machine-language byte in the object program. The designer thus has complete control over the object program generated, but with maximum convenience and readability.

Using the RCA assembler offers the following advantages:

1. Fewer mistakes – because mnemonics are provided for each instruction, and because writing within a simple syntax reduces errors.
2. Easy program revision – because memory locations and register names are designer-assigned mnemonics, which do not change when new instructions are inserted or old ones deleted.
3. Easy understandability, both by the designer and later by others – because of the mnemonics used and because of the opportunity to add documentation comments at will. The assembler will ignore these comments in the process of assembling the source program.

The RCA assembly program provides two levels of sophistication. At the first level, each instruction is referred to by name, and each statement defines a single instruction. The mnemonics of this level are similar to other conventional assemblers, and are the easiest for programming novices to learn.

The second level of the language provides a variety of shorthand symbolics, some Fortran-like features, and greater flexibility.

The assembler is written in standard Fortran IV, to make it easy to install on a variety of computers.

THE SIMULATOR/DEBUGGER

The COSMAC simulator/debugger enables the designer to check out his program on timesharing systems, using a variety of powerful debugging tools. Fig. 1 shows the inter-relationship of the assembler and simulator/debugger in the RCA software support system.

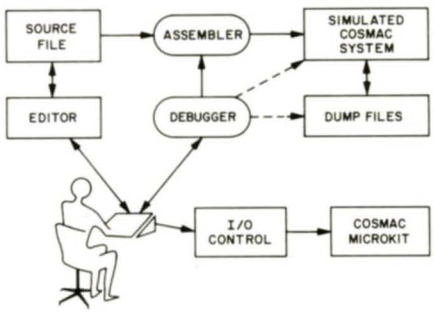


Fig. 1 – RCA Software Support System.

To use the support system, the program designer works at a terminal to prepare his source program, using the editor, and then calls on the assembler to assemble the program. After assembly, the prepared code is entered into the memory of the simulated COSMAC system. During simulation the user can stop the program at specified points or by specified events. He can ask for the contents of registers and memory locations using his original assembly mnemonics, and change the values at will. He can ask for traces –– blow-by-blow descriptions of program execution. He can ask the simulator to monitor specified locations and stop when they are read, written to, or executed from. At any time he can store away the entire state of the simulated machine for later retrieval –– a particularly useful tool during debugging sessions, or to avoid the cost of reassembly. When satisfied, the user loads the program directly into his microprocessor hardware through his terminal, which acts as an input device to the microprocessor. Final verification of the program can now be done in real time on the Microprocessor Hardware Support Kit (COSMAC Microkit).

(Over)

2M1122B
Printed in U.S.A. 3/75

SOFTWARE SUPPORT SYSTEM AVAILABILITY

The software support system is available in two forms. It has been installed on the General Electric Information Services International Network for use by timesharing customers. It is also available from RCA as a Fortran IV tape for installation on any appropriate interactive computer. In this case, it is supplied as a 9-track, 800 bpi IBM compatible-tape, with a detailed installation manual.

RCA Microprocessor Hardware Support Kit

The RCA Microprocessor Hardware Support Kit (COSMAC Microkit) is a prototyping system for the development of systems based on the RCA COSMAC Microprocessor. A Teletype*, or a terminal with a TTL bit-serial interface, can be attached to this system. With such a terminal, it is an elementary but complete computer system. The user is expected to provide something additional to achieve his goals —— software, I/0 device controller, and perhaps additional memory.

In the basic configuration, the Microkit provides the following:

1. 19" rack-mountable card nest, with printed-circuit backplane
2. Self-contained power supply
3. Front panel with basic controls
4. Eleven 4.5" x 3.0" 44-pin PC cards:

   CPU card
   Clock and Control
   Bus Separato (2)
   Address Latch Card
   512-byte RAM Cards (2)
   512-byte PROM Card
   I/0 Decoder Card
   Terminal Card (for Teletype*, or Execuport**, etc.)
   Byte I/0 Card

The 512-byte PROM card contains the Utility Program, which performs commonly required functions: program loading, memory dump, modification of memory locations, paper tape punch, saving of registers, and start of program execution at a given location.

The Microkit has been designed to allow the user to add memory cards and device electronics for configuring the system to his particular requirements. A total of nine spare memory PC card positions are prebussed for memory expansion and a total of fourteen spare I/0 PC card positions (one prewired) are made available for user device control electronics. Small 44-pin PC cards, 4.5" x 3" are used in the basic Microkit card-set for modularity and flexibility in building new systems. Larger cards up to 4.5" x 6.5", can be used in the enclosure, if desired. The power supply is capable of providing up to 2 amperes at 5 volts, for additional cards.

Users may write and debug programs with the aid of software packages available on timesharing systems. The object code for a program may be automatically loaded into the Microkit RAM for further debugging.

The Microkit is designed for easy reconfigurability and hardware extension, as well as for easy interaction with a timesharing system in software development.

The Microkit is provided with a detailed manual, which carefully describes the hardware and the Utility Program, and discusses generally the problems of designing a Microprocessor-based system. Numerous examples are provided.

A block diagram of the Microkit is shown in Fig. 2. Each block represents one card.

A photograph of the Microkit is shown in Fig. 1.

* Registered trademark of Teletype Corporation

** Registered trademark of Computer Transceiver Systems Inc.

Fig. 1 – COSMAC Microkit.

2M1122C
Printed in U.S.A. 3/75

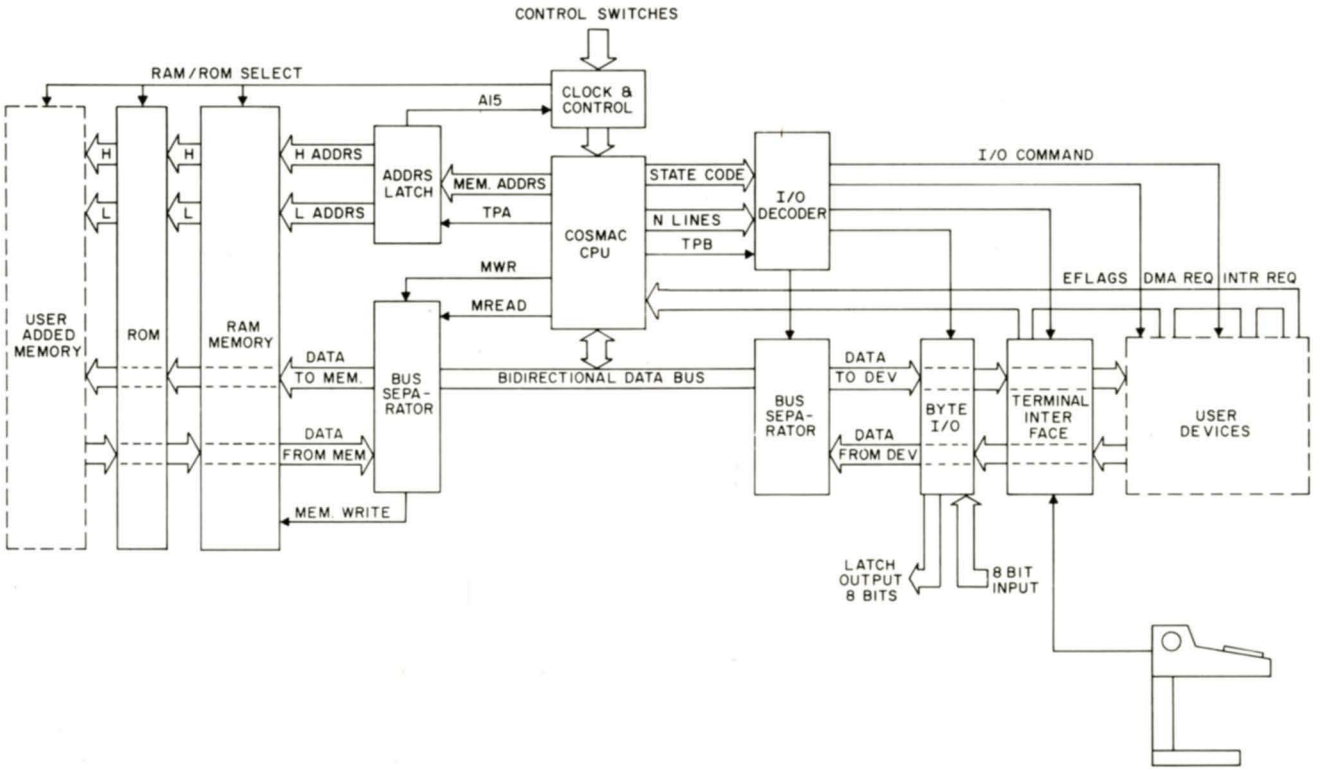


Fig. 2 – Microkit Block Diagram.

RCA Microprocessor Products

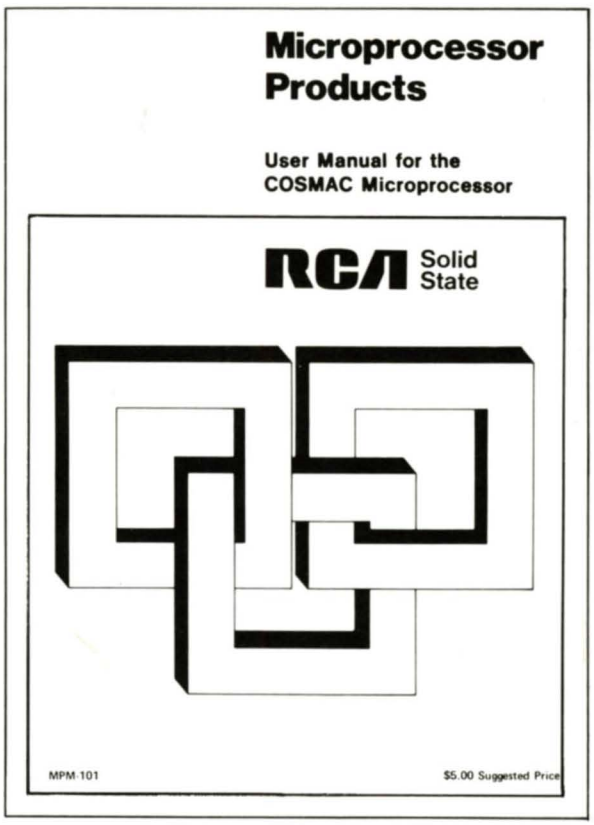
User Manual For The COSMAC Microprocessor (MPM-101)

The RCA Microprocessor Manual provides a detailed guide to the COSMAC Microprocessor. It is written for the electrical engineer with no assumptions of familiarity with computers. It describes the microprocessor architecture and its set of simple, easy-to-use instructions. Examples are given to illustrate the operation of each instruction.

For system designers, this manual illustrates practical methods of adding external memory and control circuits. Since the processor is capable of supporting Input/Output (I/0) devices in polled, interrupt-driven, and Direct-Memory-Access modes, detailed examples are provided for the I/0 instructions and the use of the I/0 interface lines. The latter include Direct-Memory-Access and interrupt inputs, external flag inputs, command lines, processor state indicators, external timing pulses, etc.

This manual also gives a description of machine-code programming methods. Detailed examples are provided. Potential programming errors are discussed. Programming techniques regarding interrupt response, long branch, subroutine linkage and nesting, and various programming techniques, are also described.

This basic manual is intended to help the design engineer understand the COSMAC Microprocessor and aid him in developing simpler and more powerful products based on Microprocessors.



2M1122D
Printed in U.S.A. 3/75

RCA Microprocessor Products

Program Development Guide For The COSMAC Microprocessor (MPM-102)

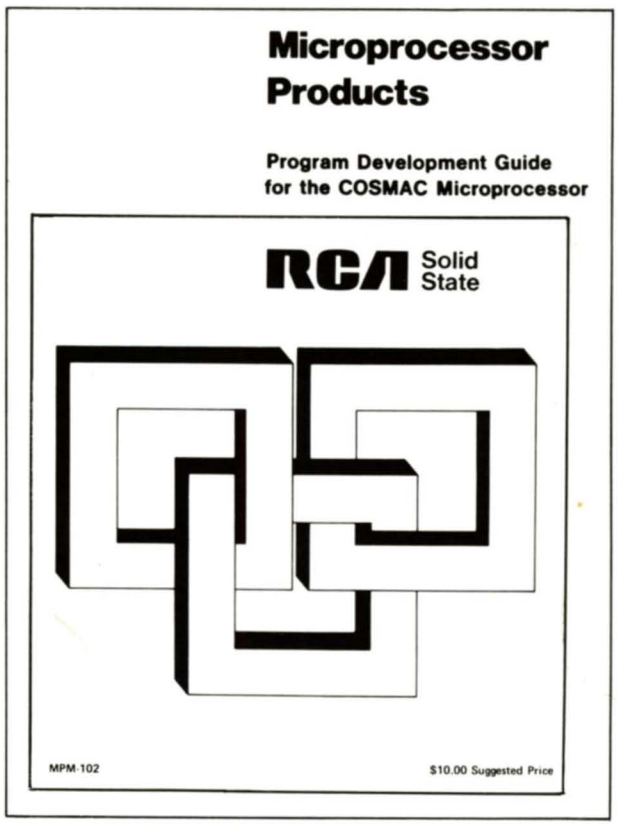
The Program Development Guide is a comprehensive manual to be used with the COSMAC Software Development Package for designing application programs. First, it reviews the architecture of the COSMAC Microprocessor so that program designers can become familiar with its register set and instruction repertoire, and with other functions such as reset, load, start, DMA, and interrupt. The manual then describes the COSMAC Software Development Package (CSDP). CSDP is an interactive program which is available via nationwide timesharing services or can be installed on a customer's own interactive computer system.

The COSMAC assembly language provides the programmer with a means of writing and modifying programs using convenient mnemonics. It has been designed to provide the designer with complete control over the object program generated, but with maximum convenience and readability. The format is free form; no column lineup is needed. Blanks can be used at will to improve readability. Comments can be inserted on any line. The program is self documenting and multiple instructions per line are permitted.

At the first level (Level I), each instruction is referred to by name, and each statement defines a single instruction. The mnemonics of this level are similar to those of other conventional assemblers, and are the easiest for programming novices to learn.

The second level (Level II) provides a variety of shorthand symbolics, some Fortran-like features, and greater flexibility. It allows additional mnemonics for some instructions, a wider use of symbolic names than Level I, the use of expressions to specify locations and values, etc. Level II is thus a "higher-level" assembly language.

Several useful programming techniques, common programming bugs, and sample programs are discussed.



2M1122E
Printed in U.S.A. 3/75