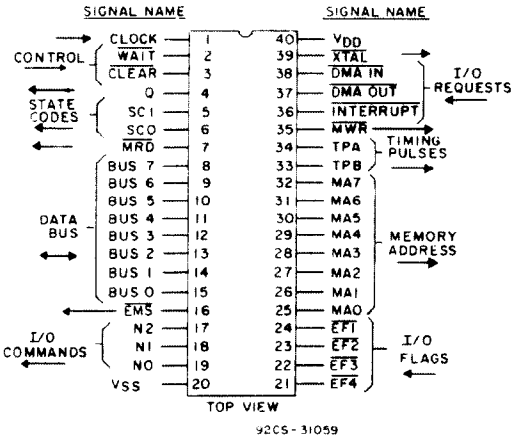
CDP1804, CDP1804C Types
Objective Data

COSMAC Microcomputer

Features:

- **Typical instruction fetch-execute time of 2.5 or 3.75 μs at V_{DD} = 5 V**
- **Static CMOS circuitry — no minimum clock frequency**
- **Single voltage supply**
- **High noise immunity, wide operating voltage range**
- **Low power**
- **Operating temperature range:**
  −55 to +125°C (CDP1804D, CDP1804CD)
  −40 to +85°C (CDP1804E, CDP1804CE)
- **8-Bit parallel organization**
- **Pinout same as CDP1802 except for V_{CC} terminal**
- **Contains ROM, RAM, I/O, and Timer/Counter**
- **Expandable in memory and I/O**
- **On-chip oscillator controlled by crystal or RC network; single phase clock**
- **Upwards software compatible with CDP1802**
- **113 easy-to-use instructions**
- **16 x 16 matrix of registers for use as multiple program counters, data pointers or data registers**
- **Test mode allows prototyping with external memory**



TERMINAL ASSIGNMENT

The RCA-CDP1804 and CDP1804C are LSI CMOS 8-bit register-oriented microcomputers designed for use in a wide variety of general-purpose computing and control applications. They can provide an effective cost and minimum chip-count solution for many low-level applications.

The CDP1804 contains a 2048-byte mask-programmable ROM, 64 bytes of executable RAM, and an 8-bit presettable down-counter with a conditional ÷ 32 prescaler, in addition to the standard architecture of the CDP1802 microprocessor. This includes a bank of 16 registers of 16 bits each that can be used as multiple program counters, pointers to data in memory, or as data storage registers and a complement of Input/Output lines including 3 "N-bit" selection lines, four flag-input lines, one serial-output line and the DMA and Interrupt Request inputs. Upwards software compatibility is maintained since the CDP1804 contains the entire instruction set of the CDP1802 plus 22 added instructions. The terminal arrangements of the two devices are identical except that the V_{CC} terminal is eliminated, and an External Memory Select ($\overline{EMS}$) terminal is substituted. This output is at a low level whenever external memory is being addressed, allowing simple add-on external memory expansion. The starting address for the internal ROM is programmable in increments of 2K bytes. The starting address for the RAM is programmable in increments of 64 bytes. The CDP1804 performs the same functions as the CDP1802, except that the Load Mode of the CDP1802 is replaced with a Test Mode in the CDP1804. The Test Mode disables both internal ROM and RAM, and enables external memory. The new instructions are all 3 or more-cycle instructions. Timing for the CDP1804 is the same as the CDP1802, except 4.5 clock pulses are provided for memory access, Q changes 1/2 cycle earlier during SEQ and REQ instructions, and flag lines are sampled at the end of the S0 machine cycle.

The counter/timer generates an interrupt at count '0' and then is automatically re-initialized. It has several modes of operation, all under software control:

- Event counter — counter decremented by EF1 or EF2.
- Timer — counter decremented by TPA divided by 32.
- Pulse duration measurements — TPA clocks to counter are gated by EF1 or EF2 lines.

The CDP1804 and CDP1804C are functionally identical. They differ in that the CDP1804 has an operating voltage range of 4 to 10.5 volts and the CDP1804C has an operating voltage range of 4 to 6.5 volts. Both are supplied in 40-lead hermetic dual-in-line ceramic packages (D suffix) and in 40-lead dual-in-line plastic packages (E suffix).

CDP1804, CDP1804C Types

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD})
(Voltage referenced to V_{SS} Terminal)
CDP1804 . . . . . −0.5 to +11 V
CDP1804C . . . . . −0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS . . . . . −0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT . . . . . ±10 mA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A = −40 to +60°C (PACKAGE TYPE E) . . . . . 500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E) . . . . . Derate Linearly at 12 mW/°C to 200 mW
For T_A = −55 to +100°C (PACKAGE TYPE D) . . . . . 500 mW
For T_A = +100 to +125°C (PACKAGE TYPE D) . . . . Derate Linearly at 12 mW/°C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
FOR T_A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types) . . . . . 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D . . . . . −55 to +125°C
PACKAGE TYPE E . . . . . −40 to +85°C
STORAGE TEMPERATURE RANGE (T_{stg}) . . . . . −65 to +150°C
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 = −40 to +85°C

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

CHARACTERISTIC	CONDITIONS		LIMITS[1]				UNITS
	f_{CL} (MHz)	V_{DD} (V)	CDP1804D CDP1804E		CDP1804CD CDP1804CE		
			Min.	Max.	Min.	Max.	
Supply-Voltage Range	–	–	4	10.5	4	6.5	V
Input Voltage Range	–	–	V_{SS}	V_{DD}	V_{SS}	V_{DD}	V
Clock Input Rise or Fall Time, t_rCL, t_fCL	–	4 – 6.5	–	1	–	1	μs
	–	4 – 10.5	–	1	–	–	
Instruction Time[2]	4	5	4		4		μs
	8	10	2		–		
DMA Transfer Rate	–	5	–	2	–	2	M Bytes/sec
	–	10	–	4	–	–	
Clock Input Frequency, f_{CL}[3]	–	5	DC	4	DC	4	MHz
	–	10	DC	8	–	–	

NOTES:
1. Typical values for Instruction Time.
2. Equals 2 machine cycles – one Fetch and one Execute operation for all instructions except Long Branch, Long Skip, NOP, and "68" family instructions, which are more than two cycles.
3. Load Capacitance (C_L) = 50 pF.

DYNAMIC POWER DISSIPATION at T_A=25°C

ALL TYPES
8 mW Typ. at V_{DD} = 5 V, f_{CL} = 4 MHz

CDP1804D, CDP1804E
80 mW Typ. at V_{DD} = 10 V, f_{CL} = 8 MHz

DATA RETENTION VOLTAGE

ALL TYPES
2 V Min. at T_A = 25°C
2.5 V Min. at T_A = −40 to +85°C

CDP1804, CDP1804C Types

STATIC ELECTRICAL CHARACTERISTICS at T_A = −40 to +85°C, Except as noted

CHARACTERISTIC	CONDITIONS			LIMITS						UNITS
				CDP1804D CDP1804E			CDP1804CD CDP1804CE			
	V_O (V)	V_{IN} (V)	V_{DD} (V)	Min.	Typ.*	Max.	Min.	Typ.*	Max.	
Quiescent Device Current, I_L	–	0,5	5	–	10	–	–	10	–	μA
	–	0,10	10	–	100	–	–	–	–	
Output Low Drive (Sink) Current, I_{OL} (Except $\overline{XTAL}$)	0.4	0,5	5	1.68	4	–	1.68	4	–	mA
	0.5	0,10	10	4.4	10.4	–	–	–	–	
$\overline{XTAL}$ Output I_{OL}	0.4	5	5	76	170	–	76	170	–	μA
Output High Drive (Source) Current I_{OH} (Except $\overline{XTAL}$)	4.6	0,5	5	−1.68	−4	–	−1.68	−4	–	mA
	9.5	0,10	10	−4.4	−10.4	–	–	–	–	
$\overline{XTAL}$ Output I_{OH}	4.6	0	5	−38	−85	–	−38	−85	–	μA
Output Voltage Low-Level V_{OL}	–	0,5	5	–	0	0.05	–	0	0.05	V
	–	0,10	10	–	0	0.05	–	–	–	
Output Voltage High Level, V_{OH}	–	0,5	5	4.95	5	–	4.95	5	–	
	–	0,10	10	9.95	10	–	–	–	–	
Input Low Voltage V_{IL}	0.5,4.5	–	5	–	–	1.5	–	–	1.5	V
	1,9	–	10	–	–	3	–	–	–	
Input High Voltage V_{IH}	0.5,4.5	–	5	3.5	–	–	3.5	–	–	
	1,9	–	10	7	–	–	–	–	–	
Input Leakage Current I_{IN}	Any Input	0,5	5	–	±0.1	±1	–	±0.1	±1	μA
		0,10	10	–	±0.1	±1	–	–	–	
3-State Output Leakage Current I_{OUT}	0,5	0,5	5	–	±0.2	±2	–	±0.2	±2	μA
	0,10	0,10	10	–	±0.2	±2	–	–	–	

* Typical values are for T_A = 25°C.

ARCHITECTURE

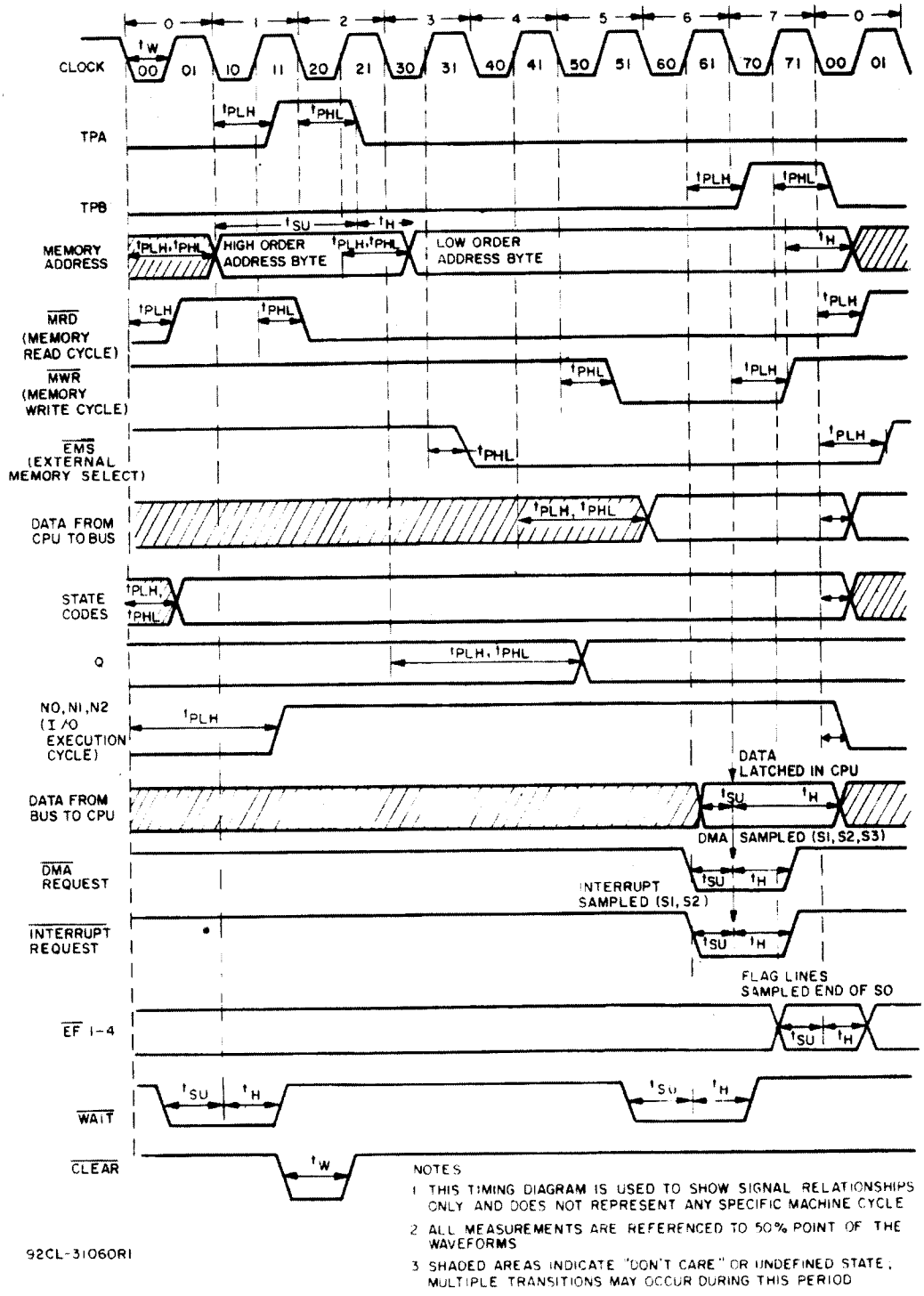
The CDP1804 block diagram is shown in Fig. 2. The principal feature of this system is a register array (R) consisting of sixteen 16-bit scratchpad registers. Individual registers in the array (R) are designated (selected) by a 4-bit binary code from one of the 4-bit registers labeled N, P, and X. The 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.

With few exceptions, COSMAC instructions consist of two 8-clock-pulse machine cycles. The first cycle is the fetch cycle, and the second—and more if necessary—are execute cycles. During the fetch cycle the four bits in the P designator select one of the 16 registers R(P) as the current program counter. The selected register 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 into the N register. The content of the program counter is automatically incremented by one

CDP1804, CDP1804C Types



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
3. SHADED AREAS INDICATE "DON'T CARE" OR UNDEFINED STATE, MULTIPLE TRANSITIONS MAY OCCUR DURING THIS PERIOD

Fig. 1 – Timing waveforms.

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 the specific operation required in a class of miscellaneous instructions (70-73 and 78-7B);
4. indicate the value to be loaded into P to designate a new register to be used as the program counter R(P);

CDP1804, CDP1804C Types

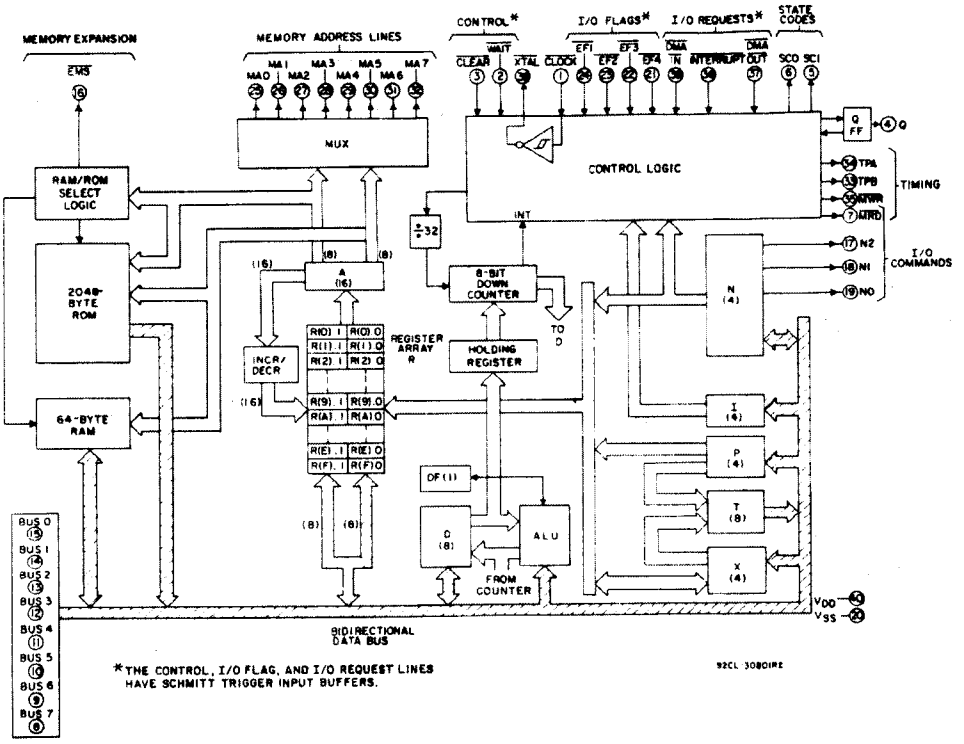


Fig. 2 – Block diagram for CDP1804.

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. 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 user's interrupt servicing routine. After reset, and during a DMA operation, R (0) is used as the program counter. 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 a location in memory. The register designated by X (i.e., R(X)) points to memory for the following instructions (see Table I):

1. ALU operations F1-F5, F7, 74, 75, 77;
2. output instructions 61 through 67;
3. input instructions 69 through 6F;
4. certain miscellaneous instructions—70-73, 78, 60, F0.

The register designated by N (i.e., R(N)) points to memory for the "load D from memory" instructions 0N and 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-FD, FF, 7C, 7D, 7F. 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 (DMA-Out) or written into (DMA-In) the memory location pointed to by the R(0) register. 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.

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

CDP1804, CDP1804C Types

mechanism (together with loading by data immediate) program pointer and data pointer designations are initialized. Also, this technique allows scratchpad registers in R to be used to hold general data. By employing increment or decrement instructions, such registers may be used as loop counters.

and the interrupt is allowed by the program (again, nothing takes place until the completion of the current instruction), 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 de-

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 nybble)
IE	1 Bit	Interrupt Enable
Q	1 Bit	Output Flip Flop

The Q Flip Flop

An internal flip flop, Q, can be set or reset by instruction and can be sensed by conditional branch instructions. It can also be driven by the output of the counter. The output of Q is also available as a microprocessor output.

Interrupt Servicing

Register R(1) is always used as the program counter whenever interrupt servicing is initiated. When an interrupt request occurs activated to inhibit further interruptions. The user's interrupt routine is now in control; the contents of T may be saved by means of a single instruction (78) in the memory location pointed to by R (X). At the conclusion of the interrupt, the user's routine may restore the pre-interrupted value of X and P with a single instruction (70 or 71). The interrupt-enable flip-flop can be activated to permit further interrupts or can be disabled to prevent them.

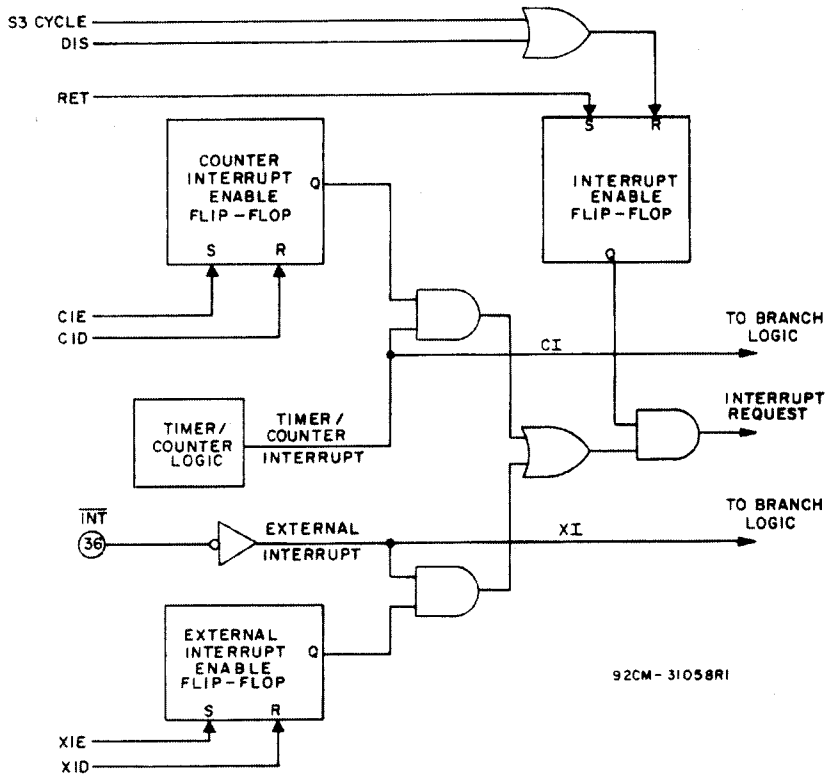


Fig. 3 – CDP1804 interrupt logic control diagram.

Interrupts can be generated either externally through the Interrupt input, or internally via the counter on the transition from count $(01)_{16}$ to its next value. Short branch instructions on Counter Interrupt and External Interrupt provide a means of identifying the source of the interrupt. The branch will be taken if an interrupt request is pending, regardless of the state of the interrupt enable flip-flop. The counter interrupt request flip-flop will be reset if its branch is taken. Interrupt logic control is shown in Fig. 3.

Counter and Controls

This logic consists of a presettable 8-bit down-counter (Module N type), and a conditional divide-by 32 prescaler. After counting down to $(01)_{16}$ the counter returns to its initial value at the next count and sets the timer/counter interrupt. It will continue decrementing on subsequent counts. If the counter is preset to $(00)_{16}$ a full 256 counts will occur.

During a load instruction to the counter, the counter and its buffer register are loaded, the prescaler reset, the mode reset and any previous interrupts cleared. Read operations do not affect the counter.

The counter has the following five programmable modes:

1. Event counter 1: input to counter is connected to the $\overline{EF1}$ terminal. The high-to-low transition decrements the counter.
2. Event counter 2: Input to counter is connected to the $\overline{EF2}$ terminal. The high-to-low transition decrements the counter.
3. Timer: Input to counter is from the divide-by-32 prescaler clocked by TPA. The prescaler is decremented on the low-to-high transition of TPA.
4. Pulse duration measurement 1: Input to counter connected to TPA. Each low-to-high transition of TPA decrements the counter if the input signal at $\overline{EF1}$ terminal is low. On the transition of this signal to the positive state, the count is stopped, the mode is cleared, and the interrupt request latched. If the counter decrements to "one" while the input is low, interrupt will also be set, but the counter will continue.
5. Pulse duration measurement 2: Input to counter connected to TPA. Each low-to-high transition of TPA decrements the counter if the input signal at $\overline{EF2}$ terminal is low. On the transition of this signal to the positive state, the count is stopped, the mode is cleared, and the interrupt request latched. If the counter decrements to "one" while the input is low, interrupt will also be set, but the counter will continue.

Those modes which use $\overline{EF1}$ and $\overline{EF2}$ terminals as inputs do not exclude testing these flags for branch instructions.

The stop counter (STPC) command clears the counter mode and stops counting.

In addition to the five programmable modes, the decrement counter instruction (DTC) enables the user to count in software. In order to avoid conflict with counting done in the event counter mode, the instruction should be used only after the mode has been cleared by a stop counter instruction.

The toggle Q enable command connects the Q-line flip-flop to the output of the counter, such that each time the counter decrements from 01 to its next value, the Q line changes state. This action is independent of the counter mode and the Interrupt Enable flip-flops.

Memory

The locations of ROM and RAM in the CDP1804 are determined by AND-gate decoders which decode the upper memory addresses and are programmable at the time of ROM pattern masking. The logical values of the decoder inputs are selectable as 1 (positive), 0 (negative), or X (don't care). A 5-bit decoder is used for the ROM selection, so the ROM can be placed at one or more 2K blocks in memory. Similarly, the RAM has a 10-bit decoder and can be selected at one or more of the 1024, 64-byte spaces within the 65,536 locations of memory. If the RAM is located within the ROM space, the RAM will be enabled at the locations where both are mapped.

An External Memory Select output ($\overline{EMS}$) is provided for memory expansion. This output is low whenever the external memory is being addressed, i.e. during a memory reference machine cycle in which the address does not correspond to internal ROM or RAM. Both internal ROM and RAM are disabled when the Test Mode is activated ($\overline{CLEAR}$ = $\overline{WAIT}$ = low) forcing all instructions to be fetched and executed from external memory. In this case, $\overline{EMS}$ is low for any internal or external memory reference machine cycle. $\overline{EMS}$ is low only when an external memory reference operation is in progress, and the addresses are stable.

CDP1804, CDP1804C Types

INSTRUCTION SET

The CDP1804 instruction summary is given in Table I. Hexadecimal notation is used to refer to the 4-bit binary codes.

In all registers bits are numbered from the least significant bit (LSB) to the most significant bit (MSB) starting with 0.

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 → R(N)

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
(For Notes, see page 12)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
MEMORY REFERENCE			
LOAD VIA N	LDN	0N	M(R(N))→D; FOR N NOT 0
LOAD ADVANCE	LDA	4N	M(R(N))→D; R(N) +1 → R(N)
LOAD VIA X	LDX	F0	M(R(X))→D
LOAD VIA X AND ADVANCE	LDXA	72	M(R(X))→D; R(X) + 1 → R(X)
LOAD IMMEDIATE	LDI	F8	M(R(P))→D; R(P) +1 → R(P)
STORE VIA N	STR	5N	D→M(R(N))
STORE VIA X AND DECREMENT	STXD	73	D→M(R(X)); R(X) –1 → R(X)
REGISTER OPERATIONS			
INCREMENT REG N	INC	1N	R(N) +1 → R(N)
DECREMENT REG N	DEC	2N	R(N) –1 → R(N)
INCREMENT REG X	IRX	60	R(X) +1 → R(X)
GET LOW REG N	GLO	8N	R(N).0→D
PUT LOW REG N	PLO	AN	D→R(N).0
GET HIGH REG N	GHI	9N	R(N).1→D
PUT HIGH REG N	PHI	BN	D→R(N).1
LOGIC OPERATIONS◆◆			
OR	OR	F1	M(R(X)) OR D→D
OR IMMEDIATE	ORI	F9	M(R(P)) OR D→D; R(P) +1→R(P)
EXCLUSIVE OR	XOR	F3	M(R(X)) XOR D→D
EXCLUSIVE OR IMMEDIATE	XRI	FB	M(R(P)) XOR D→D; R(P) +1→R(P)
AND	AND	F2	M(R(X)) AND D→D
AND IMMEDIATE	ANI	FA	M(R(P)) AND D→D; R(P) +1→R(P)
SHIFT RIGHT	SHR	F6	SHIFT D RIGHT, LSB(D)→DF, 0→MSB(D)
SHIFT RIGHT WITH CARRY	SHRC	76◆	SHIFT D RIGHT, LSB(D)→DF, DF→MSB(D)
RING SHIFT RIGHT	RSHR		
SHIFT LEFT	SHL	FE	SHIFT D LEFT, MSB(D)→DF, 0→LSB(D)
SHIFT LEFT WITH CARRY	SHLC	7E◆	SHIFT D LEFT, MSB(D)→DF, DF→LSB(D)
RING SHIFT LEFT	RSHL		
REGISTER LOAD VIA X AND ADVANCE	RLXA	686N	M(R(X))→R(N).1; M(R(X) +1)→R(N).0; R(X) +2→R(X)
REGISTER LOAD IMMEDIATE	RLDI	68CN	M(R(P))→R(N).1; M(R(P) +1)→R(N).0; R(P) +2→R(P)
REGISTER STORE VIA X AND DECREMENT	RSXD	68AN	R(N).0→M(R(X)); R(N).1→M(R(X)-1); R(X) –2→R(X)
REGISTER N TO REGISTER X COPY	RNX	68BN	R(N)→R(X)

CDP1804, CDP1804C Types

TABLE I – INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
ARITHMETIC OPERATIONS◆◆			
ADD	ADD	F4	M(R(X)) +D→DF, D
ADD IMMEDIATE	ADI	FC	M(R(P)) +D→DF, D; R(P) +1→R(P)
ADD WITH CARRY	ADC	74	M(R(X)) +D +DF→DF, D
ADD WITH CARRY, IMMEDIATE	ADCI	7C	M(R(P)) +D +DF→DF, D R(P) +1→R(P)
SUBTRACT D	SD	F5	M(R(X))−D→DF, D
SUBTRACT D IMMEDIATE	SDI	FD	M(R(P))−D→DF, D; R(P) +1→R(P)
SUBTRACT D WITH BORROW	SDB	75	M(R(X))−D−(NOT DF)→DF, D
SUBTRACT D WITH BORROW, IMMEDIATE	SDBI	7D	M(R(P))−D−(NOT DF)→DF, D; R(P) +1→R(P)
SUBTRACT MEMORY	SM	F7	D−M(R(X))→DF, D
SUBTRACT MEMORY IMMEDIATE	SMI	FF	D−M(R(P))→DF, D; R(P) +1→R(P)
SUBTRACT MEMORY WITH BORROW	SMB	77	D−M(R(X))−(NOT DF)→DF, D
SUBTRACT MEMORY WITH BORROW, IMMEDIATE	SMBI	7F	D−M(R(P))−(NOT DF)→DF, D R(P) +1→R(P)
BRANCH INSTRUCTIONS–SHORT BRANCH			
SHORT BRANCH	BR	30	M(R(P))→R(P).0
NO SHORT BRANCH (SEE SKP)	NBR	38◆	R(P) +1→R(P)
SHORT BRANCH IF D=0	BZ	32	IF D=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF D NOT 0	BNZ	3A	IF D NOT 0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF DF=1	BDF	33◆	IF DF=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF POS OR ZERO	BPZ		
SHORT BRANCH IF EQUAL OR GREATER	BGE		
SHORT BRANCH IF DF=0	BNF	3B◆	IF DF=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF MINUS	BM		
SHORT BRANCH IF LESS	BL		
SHORT BRANCH IF Q=1	BQ	31	IF Q=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF Q=0	BNQ	39	IF Q=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF1=1 ($\overline{EF1} = V_{SS}$)	B1	34	IF EF1=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF1=0 ($\overline{EF1} = V_{DD}$)	BN1	3C	IF EF1=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF2=1 ($\overline{EF2} = V_{SS}$)	B2	35	IF EF2=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF2=0 ($\overline{EF2} = V_{DD}$)	BN2	3D	IF EF2=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF3=1 ($\overline{EF3} = V_{SS}$)	B3	36	IF EF3=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF3=0 ($\overline{EF3} = V_{DD}$)	BN3	3E	IF EF3=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF4=1 ($\overline{EF4} = V_{SS}$)	B4	37	IF EF4=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH IF EF4=0 ($\overline{EF4} = V_{DD}$)	BN4	3F	IF EF4=0, M(R(P))→R(P).0 ELSE R(P) +1→R(P)
SHORT BRANCH ON COUNTER INTERRUPT	BCI	683E	IF CI=1, M(R(P))→R(P).0; 0→CI ELSE R(P) +1→R(P)
SHORT BRANCH ON EXTERNAL INTERRUPT	BXI	683F	IF XI=1, M(R(P))→R(P).0 ELSE R(P) +1→R(P)

◆NOTE: THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC. EACH MNEMONIC IS INDIVIDUALLY LISTED.

◆◆NOTE: THE ARITHMETIC OPERATIONS AND THE SHIFT INSTRUCTIONS ARE THE ONLY INSTRUCTIONS THAT CAN ALTER THE DF.

AFTER AN ADD INSTRUCTION:
- DF = 1 DENOTES A CARRY HAS OCCURRED
- DF = 0 DENOTES A CARRY HAS NOT OCCURRED

AFTER A SUBTRACT INSTRUCTION:
- DF = 1 DENOTES NO BORROW. D IS A TRUE POSITIVE NUMBER
- DF = 0 DENOTES A BORROW. D IS TWO'S COMPLEMENT

THE SYNTAX "−(NOT DF)" DENOTES THE SUBTRACTION OF THE BORROW

CDP1804, CDP1804C Types

TABLE I – INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
BRANCH INSTRUCTIONS–LONG BRANCH			
LONG BRANCH	LBR	C0	M(R(P))→R(P).1 M(R(P) +1)→R(P).0
NO LONG BRANCH (SEE LSKP)	NLBR	C8◆	R(P) +2→R(P)
LONG BRANCH IF D=0	LBZ	C2	IF D=0, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
LONG BRANCH IF D NOT 0	LBNZ	CA	IF D NOT 0, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
LONG BRANCH IF DF=1	LBDF	C3	IF DF=1, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
LONG BRANCH IF DF=0	LBNF	C8	IF DF=0, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
LONG BRANCH IF Q=1	LBQ	C1	IF Q=1, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
LONG BRANCH IF Q=0	LBNQ	C9	IF Q=0, M(R(P))→R(P).1 M(R(P) +1)→R(P).0 ELSE R(P) +2→R(P)
SKIP INSTRUCTIONS			
SHORT SKIP (SEE NBR)	SKP	38◆	R(P) +1→R(P)
LONG SKIP (SEE NLBR)	LSKP	C8◆	R(P) +2→R(P)
LONG SKIP IF D=0	LSZ	CE	IF D=0, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF D NOT 0	LSNZ	C6	IF D NOT 0, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF DF=1	LSDF	CF	IF DF=1, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF DF=0	LSNF	C7	IF DF=0, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF Q=1	LSQ	CD	IF Q=1, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF Q=0	LSNQ	C5	IF Q=0, R(P) +2→R(P) ELSE CONTINUE
LONG SKIP IF IE=1	LSIE	CC	IF IE=1, R(P) +2→R(P) ELSE CONTINUE
CONTROL INSTRUCTIONS			
IDLE	IDL	00=	WAIT FOR DMA OR INTERRUPT; M(R(0))→BUS
NO OPERATION	NOP	C4	CONTINUE
SET P	SEP	DN	N→P
SET X	SEX	EN	N→X
SET Q	SEQ	7B	1→Q
RESET Q	REQ	7A	0→Q
PUSH X,P TO STACK	MARK	79	(X,P)→T; (X,P)→M(R(2)) THEN P→X; R(2)–1→R(2)
ENABLE TOGGLE Q	ETQ	6809■	COUNTER = 01 AND COUNTER CLOCK ↓ ⇒ $\overline{Q}$→Q

■ETQ cleared by LDC, reset of CPU, or BCI · (CI = 1).

=An idle instruction initiates a repeating S1 cycle. The processor will continue to idle until an I/O request ($\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, or $\overline{\text{DMA-OUT}}$) is activated. When the request is acknowledged, the IDLE cycle is terminated and the I/O request is serviced, and then normal operation is resumed.

◆NOTE THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC EACH MNEMONIC IS INDIVIDUALLY LISTED.

CDP1804, CDP1804C Types

TABLE I – INSTRUCTION SUMMARY (CONT'D)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
COUNTER INSTRUCTIONS			
LOAD COUNTER	LDC	6806	D→COUNTER
GET COUNTER	GEC	6808	COUNTER→D
STOP COUNTER	STPC	6800	0→COUNTER CLOCK
DECREMENT COUNTER	DTC	6801	COUNTER – 1→COUNTER
SET TIMER MODE AND START	STM	6807	TPA÷32→COUNTER CLOCK
SET COUNTER MODE 1 AND START	SCM1	6805	EF1→COUNTER CLOCK
SET COUNTER MODE 2 AND START	SCM2	6803	EF2→COUNTER CLOCK
SET PULSE WIDTH MODE 1, START	SPM1	6804	TPA·EF1→COUNTER CLOCK; $\overline{EF1}$ ⌐ STOPS COUNT
SET PULSE WIDTH MODE 2, START	SPM2	6802	TPA·EF2→COUNTER CLOCK; $\overline{EF2}$ ⌐ STOPS COUNT
INTERRUPT CONTROL			
EXTERNAL INTERRUPT ENABLE	XIE	680A	1→XIE
EXTERNAL INTERRUPT DISABLE	XID	680B	0→XIE
COUNTER INTERRUPT ENABLE	CIE	680C	1→CIE
COUNTER INTERRUPT DISABLE	CID	680D	0→CIE
RETURN	RET	70	M(R(X))→X,P; R(X)+1→R(X);1→IE
DISABLE	DIS	71	M(R(X))→X,P; R(X)+1→R(X);0→IE
SAVE	SAV	78	T→M(R(X))
INPUT–OUTPUT BYTE TRANSFER			
OUTPUT 1	OUT 1	61	M(R(X))→BUS;R(X)+1→R(X);N LINES=1
OUTPUT 2	OUT 2	62	M(R(X))→BUS;R(X)+1→R(X);N LINES=2
OUTPUT 3	OUT 3	63	M(R(X))→BUS;R(X)+1→R(X);N LINES=3
OUTPUT 4	OUT 4	64	M(R(X))→BUS;R(X)+1→R(X);N LINES=4
OUTPUT 5	OUT 5	65	M(R(X))→BUS;R(X)+1→R(X);N LINES=5
OUTPUT 6	OUT 6	66	M(R(X))→BUS;R(X)+1→R(X);N LINES=6
OUTPUT 7	OUT 7	67	M(R(X))→BUS;R(X)+1→R(X);N LINES=7
INPUT 1	INP 1	69	BUS→M(R(X)); BUS→D; N LINES = 1
INPUT 2	INP 2	6A	BUS→M(R(X)); BUS→D; N LINES = 2
INPUT 3	INP 3	6B	BUS→M(R(X)); BUS→D; N LINES = 3
INPUT 4	INP 4	6C	BUS→M(R(X)); BUS→D; N LINES = 4
INPUT 5	INP 5	6D	BUS→M(R(X)); BUS→D; N LINES = 5
INPUT 6	INP 6	6E	BUS→M(R(X)); BUS→D; N LINES = 6
INPUT 7	INP 7	6F	BUS→M(R(X)); BUS→D; N LINES = 7
CALL AND RETURN			
STANDARD CALL	SCAL	688N	R(N).0→M(R(X)); R(N).1→M(R(X)–1); R(X)–2→R(X); R(P)→ R(N); THEN M(R(N))→ R(P).1; M(R(N)+1)→R(P).0; R(N)+2→R(N)
STANDARD RETURN	SRET	689N	R(N)→R(P); M(R(X)+1) → R(N).1; M(R(X)+2)→ R(N).0; R(X)+2→R(X)

CDP1804, CDP1804C Types

1. Long-Branch, Long-Skip and No Op instructions require three cycles to complete (1 fetch + 2 execute).

   Long-Branch instructions are three bytes long. The first byte specifies the condition to be tested; and the second and third byte, the branching address.

   The long-branch instructions can:

   a) Branch unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) effect an unconditional no branch

   If the tested condition is met, then branching takes place; the branching address bytes are loaded in the high-and-low-order bytes of the current program counter, respectively. This operation effects a branch to any memory location.

   If the tested condition is not met, the branching address bytes are skipped over, and the next instruction in sequence is fetched and executed. This operation is taken for the case of unconditional no branch (NLBR).

2. The short-branch instructions are two or three bytes long. The first byte specifies the condition to be tested, and the second specifies the branching address, except for the branches on interrupt. For those, the first two bytes specify the condition to be tested and the third byte specifies the branching address.

   The short-branch instruction can:

   a) Branch unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) Test the status (1 or 0) of the four EF flags
   f) Effect an unconditional no branch
   g) Test for interrupts

   If the tested condition is met, then branching takes place; the branching address byte is loaded into the low-order byte position of the current program counter. This effects a branch within the current 256-byte page of the memory, i.e., the page which holds the branching address. If the tested condition is not met, the branching address byte is skipped over, and the next instruction in sequence is fetched and executed. This same action is taken in the case of unconditional no branch (NBR).

3. The skip instructions are one byte long. There is one Unconditional Short-Skip (SKP) and eight Long-Skip instructions.

   The Unconditional Short-Skip instruction takes 2 cycles to complete (1 fetch + 1 execute). Its action is to skip over the byte following it. Then the next instruction in sequence is fetched and executed. This SKP instruction is identical to the unconditional no-branch instruction (NBR) except that the skipped-over byte is not considered part of the program.

   The Long-Skip instructions take three cycles to complete (1 fetch + 2 execute).

   They can:

   a) Skip unconditionally
   b) Test for D=0 or D≠0
   c) Test for DF=0 or DF=1
   d) Test for Q=0 or Q=1
   e) Test for IE=1

   If the tested condition is met, then Long Skip takes place; the current program counter is incremented twice. Thus two bytes are skipped over and the next instruction in sequence is fetched and executed. If the tested condition is not met, then no action is taken. Execution is continued by fetching the next instruction in sequence.

4. Instructions 6800 through 680D, 683E, and 683F take 3 machine cycles; 68BN takes 4 machine cycles; 686N, 68AN, and 68CN take 5 machine cycles; 688N takes 10 machine cycles; and 689N takes 8 machine cycles. In all cases, the first two cycles are fetches and subsequent cycles are executes. The first byte (68) of these two byte op codes is used to generate the second fetch, the second byte is then interpreted differently than the same code without the 68 prefix. DMA and INT requests are not serviced until the end of the last execute cycle.

CDP1804, CDP1804C Types

SIGNAL DESCRIPTIONS

BUS 0 to BUS 7 (Data Bus)

8-bit bi-directional DATA BUS lines. These lines are used for transferring data between the memory, the microprocessor, and I/O devices.

N0 to N2 (I/O Lines)

Activated by an I/O instruction to signal the I/O control logic of a data transfer between memory and I/O interface. 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 being executed). The N bits are low at all times except when an I/O instruction is being executed. During this time their state is the same as the corresponding bits in the N register.

The direction of data flow is defined in the I/O instruction by bit N3 (internally) and is indicated by the level of the $\overline{MRD}$ signal.

$\overline{MRD} = V_{DD}$: Data from I/O to CPU and Memory

$\overline{MRD} = V_{SS}$: Data from Memory to I/O

$\overline{EF1}$ to $\overline{EF4}$ (4 Flags)

These inputs enable the I/O controllers to transfer status information to the processor. The 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 sampled at the beginning of every S1 cycle.

$\overline{INTERRUPT}$, $\overline{DMA\text{-}IN}$, $\overline{DMA\text{-}OUT}$ (3 I/O Requests)

These inputs are sampled by the CDP1804 during the interval in the middle of TPB.

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. The interrupt action requires one machine cycle (S3).

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-IN has priority followed by DMA-OUT and then INTERRUPT.

SC0, SC1, (2 State Code Lines)

These outputs indicate that the CPU is: 1 fetching an instruction, or 2) executing an instruction, or 3) processing a DMA request, or 4) acknowledging an interrupt request. The levels of state code are tabulated below. All states are valid at TPA.

$H = V_{DD}$, $L = V_{SS}$.

State Type	State Code Lines	
	SC1	SC0
S0 (Fetch)	L	L
S1 (Execute)	L	H
S2 (DMA)	H	L
S3 (Interrupt)	H	H

TPA, TPB (2 Timing Pulses)

Positive 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.

MA0 to MA7 (8 Memory Address Lines)

The higher-order byte of a 16-bit COSMAC memory address appears on the memory address lines MA0-7 first. Those bits required by the memory system can be strobed into external address latches by timing pulse TPA. The low-order byte of the 16-bit address appears on the address lines, 1/2 cycle after the termination of TPA. Latching of all 8 higher-order address bits would permit a memory system of 64K bytes.

CDP1804, CDP1804C Types

$\overline{MWR}$ (Write Pulse)

A negative pulse appearing in a memory-write cycle, after the address lines have stabilized.

$\overline{MRD}$ (Read Level)

A low level on $\overline{MRD}$ 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, $\overline{MRD}$ is useful for driving memory/bus separator gates. It is also used to indicate the direction of data transfer during an I/O instruction. For additional information see Table I.

Q

Single bit output from the CPU which can be set or reset under program control. During SEQ or REQ instruction execution, Q is set or reset between the trailing edge of TPA and the leading edge of TPB.

CLOCK

Input for externally generated single-phase clock. A typical clock frequency is 4 MHz at V_{DD} = 5 volts.

The clock is counted down internally to 8 clock pulses per machine cycle.

$\overline{XTAL}$

Connection to be used with clock input terminal, for an external crystal, or RC network if the on-chip oscillator is utilized. The crystal is connected between terminals 1 and 39 (CLOCK and $\overline{XTAL}$) in parallel with a resistance (10 megohms typ.). Frequency trimming capacitors may be required at terminals 1 and 39. For additional information on crystal oscillators, see ICAN-6565. A typical RC oscillator is shown in Fig. 4. The frequency is approximately $0.25/RC$.

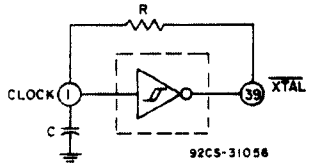


Fig. 4 – RC network for oscillator

$\overline{WAIT}$, $\overline{CLEAR}$ (2 Control Lines)

Provide four control modes as listed in the following truth table:

$\overline{CLEAR}$	$\overline{WAIT}$	MODE
L	L	Test
L	H	Reset
H	L	Pause
H	H	Run

The function of the modes are defined as follows:

Test

Disables internal ROM and RAM, so that all memory references are to external memory. This mode is equivalent to the RUN mode in all other respects.

Reset

Registers I, N, Q, and prescaler are reset. IE, XIE, and CIE are set and 0's (V_{SS}) are placed on the data bus. TPA and TPB are suppressed while reset is held and the CPU is placed in S1. The first machine cycle after termination of reset is an initialization cycle which requires 9 clock pulses. During this cycle the CPU remains in S1, X, P→T, and then registers X, P, and R(0) are reset. Interrupt and DMA servicing are suppressed during the initialization cycle. The next cycle is an S0 or an S2 but never an S1 or S3. With the use of a 71 instruction followed by 00 at memory locations 0000 and 0001, this feature may be used to reset IE, so as to preclude interrupts until ready for them. Power-up reset-run can be realized by connecting an RC network to $\overline{CLEAR}$.

CDP1804, CDP1804C Types

Pause

Stops the internal CPU timing generator, freezing the state of the processor. Pause can occur at two points in a machine cycle, on the low-to-high transition of either TPA or TPB. The oscillator continues to run but subsequent clock transitions are ignored (see Fig. 5).

If Pause is entered while in the event counter mode, the appropriate E Flag transitions will continue to decrement the counter.

Run

May be initiated from the Pause or Reset mode functions. If initiated from Pause, the CPU resumes operation at the point it left off. If paused at TPA, it will resume on the first high-to-low clock transition, while if paused at TPB, it will resume on the first low-to-high clock transition (see Fig. 5). When initiated from the Reset operation, the first machine cycle following Reset is always the initialization cycle. The initialization cycle is then followed by a DMA (S2) cycle or fetch (S0) from location 0000 in memory.

$\overline{EMS}$ — This signal line is used for external memory expansion. It is low when external memory is being addressed and high at all other times. It is initiated 1.5 clock periods after TPA (at which time all addresses are stable) and terminates at the end of the cycle. Use of $\overline{EMS}$ for memory selection allows 3.5 clock cycles for data access.

V_{DD}, V_{SS} — V_{SS} is the most negative supply voltage terminal and is normally connected to ground. V_{DD} is the positive supply voltage terminal. All outputs swing from V_{SS} to V_{DD}. The recommended input voltage swing is from V_{SS} to V_{DD}.

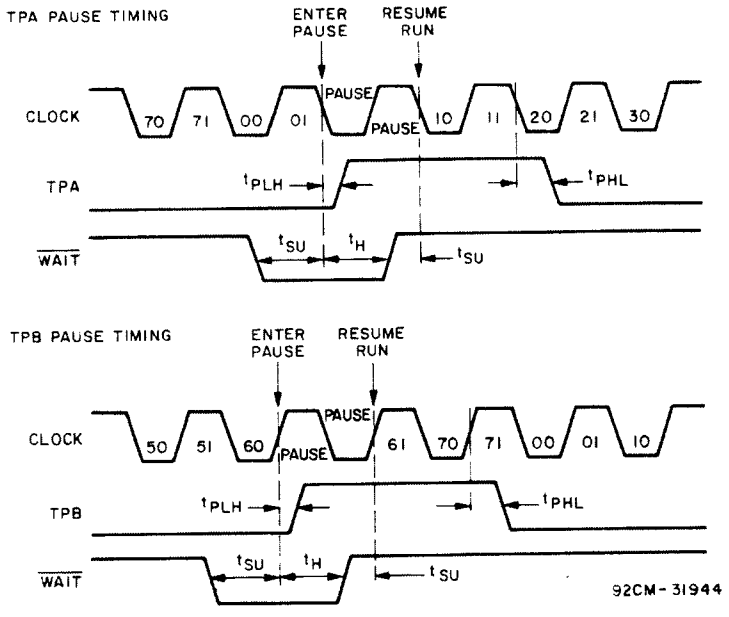


Fig. 5 — Pause mode timing diagrams.

CDP1804, CDP1804C Types

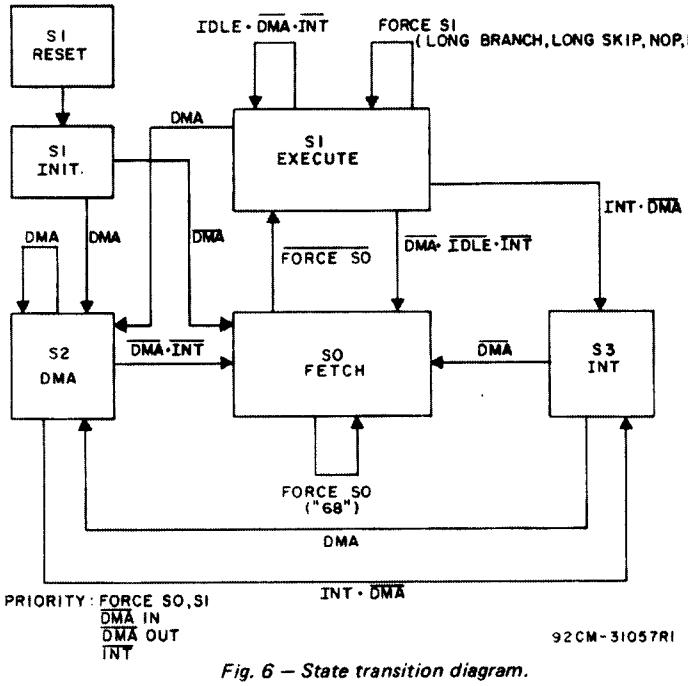


Fig. 6 — State transition diagram.

STATE TRANSITIONS

The CDP1804 state transitions are shown in Fig. 6. Each machine cycle requires the same period of time, 8 clock pulses, except the initialization cycle (INIT) which requires 9 clock pulses. The execution of an instruction requires either 2 or 3 machine cycles. S2 is the response to a DMA request and S3 is the interrupt response.