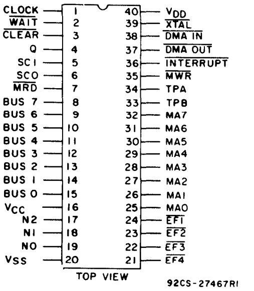
CDP1802BC



TERMINAL ASSIGNMENT

CMOS 8-Bit Microprocessor

Features:

- *Minimum instruction fetch-execute time of 3.2 μs (maximum clock frequency = 5 MHz) at V_{DD} = 5 V*
- *Any combination of standard RAM and ROM up to 65,536 bytes*
- *Operates with slow memories, up to 775 ns access time at f_{CL} = 5 MHz*
- *8-bit parallel organization with bidirectional data bus and multiplexed address bus*
- *16 x 16 matrix of registers for use as multiple program counters, data pointers, or data registers*
- *On-chip DMA, interrupt, and flag inputs*
- *Programmable single-bit output port*
- *91 easy-to-use instructions*

The RCA-CDP1802BC LSI CMOS 8-bit register-oriented central-processing unit (CPU) is designed for use as a general-purpose computing or control element in a wide range of stored-program systems or products.

The CDP1802BC includes 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 1800 series architecture is 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 1800 series CPU also provides a synchronous interface to memories and external controllers for I/O devices, and minimizes the cost of interface controllers. Further, the I/O interface is capable of supporting devices operating in polled, interrupt-driven, or direct memory-access modes.

The CDP1802BC has a recommended operating voltage range of 4 to 6.5 volts. These types are supplied in 40-lead dual-in-line side-brazed ceramic packages (D suffix), 40-lead dual-in-line plastic packages (E suffix), and 44-lead plastic chip-carrier (PCC) packages (Q suffix).

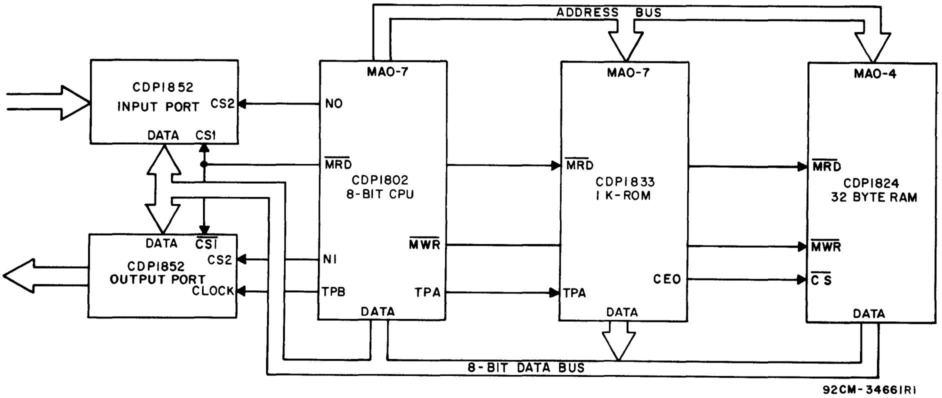


Fig 1 - Typical CDP1802BC small microprocessor system.

CDP1802BC

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD}):
(All voltages referenced to V_{SS} terminal)
CDP1802BC .... −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
For T_A = -40° C to +85° C (PACKAGE TYPE Q) * .... 500 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
FOR T_A=FULL PACKAGE-TEMPERATURE RANGE .... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D .... −55 to +125°C
PACKAGE TYPE E and Q .... −40 to +85°C
STORAGE TEMPERATURE RANGE (T_{stg}) .... −65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16±1/32 in. (1.59±0 79 mm) from case for 10 s max. .... +265°C

* Printed-circuit board mount: 57 mm x 57 mm minimum area x 1.6 mm thick G10 epoxy glass, or equivalent.

OPERATING CONDITIONS at T_A=-40° C to +85° C

For maximum reliability, 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)	CDP1802BC Min.	CDP1802BC Max.	
DC Operating Voltage Range	—	—	4.0	6.5	V
Input Voltage Range	—	—	V_{SS}	V_{DD}	
Maximum Clock Input Rise or Fall Time, t_r,t_f	4 to 6.5	4 to 6.5	—	1	μs
Minimum Instruction Time[2]	5	5	3.2	—	
Maximum DMA Transfer Rate	5	5	—	667	KBytes/s
Maximum Clock Input Frequency, f_{CL} Load Capacitance (C_L)=50 pF	5	5	DC	5	MHz

[1] V_{CC} must never exceed V_{DD}.
[2] Equals 2 machine cycles—one Fetch and one Execute operation for all instructions except Long Branch and Long Skip, which require 3 machine cycles—one Fetch and two Execute operations.

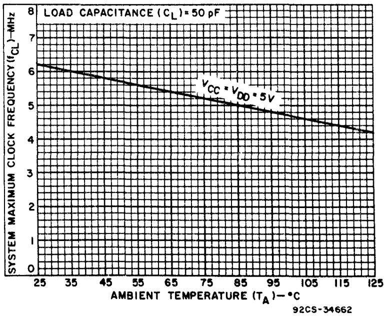


Fig. 2 - Typical maximum clock frequency as a function of temperature.

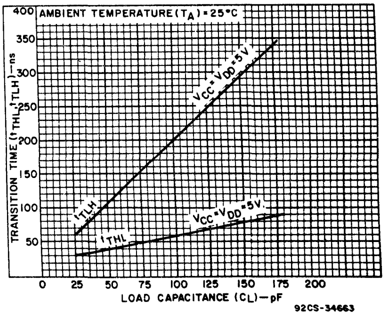
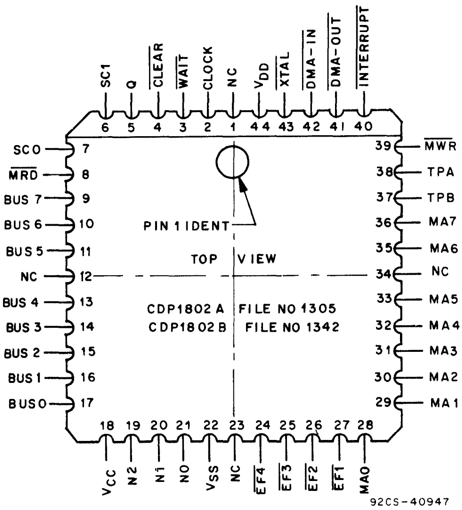


Fig. 3 - Typical transition time vs. load capacitance.

TERMINAL ASSIGNMENT



Plastic Chip-Carrier (PCC) Package

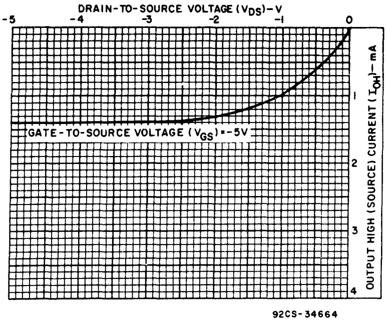


Fig. 4 - Minimum output high (source) current characteristics.

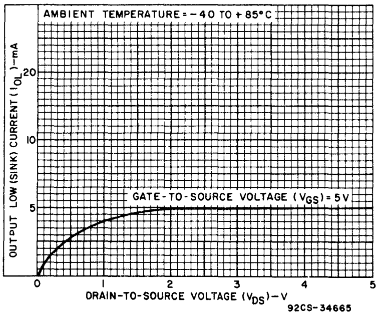


Fig. 5 - Minimum output low (sink) current characteristics.

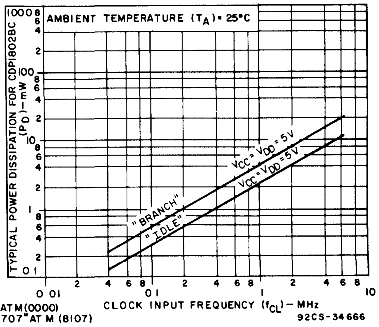
CDP1802BC

STATIC ELECTRICAL CHARACTERISTICS at T_A=-40 to +85° C, except as noted.

CHARACTERISTIC		CONDITIONS			LIMITS			UNITS
		V_{OUT} (V)	V_{IN} (V)	V_{CC}, V_{DD} (V)	CDP1802BC			
					Min.	Typ.*	Max.	
Quiescent Device Current	I_{DD}	—	—	5	—	1	200	µA
Output Low Drive (Sink) Current (Except $\overline{XTAL}$)	I_{OL}	0.4	0,5	5	1.1	2.2	—	mA
$\overline{XTAL}$		0.4	5	5	170	350	—	µA
Output High Drive (Source) Current (Except $\overline{XTAL}$)	I_{OH}	4.6	0,5	5	−0.27	−0.55	—	mA
$\overline{XTAL}$		4.6	0	5	−125	−250	—	µA
Output Voltage Low-Level	V_{OL}	—	0,5	5	—	0	0.1	V
Output Voltage High Level	V_{OH}	—	0,5	5	4.9	5	—	
Input Low Voltage	V_{IL}	0.5,4.5	—	5	—	—	1.5	
Input High Voltage	V_{IH}	0.5,4.5	—	5	3.5	—	—	
$\overline{CLEAR}$ Input Voltage Schmitt Hysteresis	V_H	—	—	5	0.4	0.5	—	
Input Leakage Current	I_{IN}	Any Input	0,5	5	—	±10^{-4}	±1	µA
3-State Output Leakage Current	I_{OUT}	0,5	0,5	5	—	±10^{-4}	±1	
Total Power Dissipation, f=5 MHz$^{\Delta}$		—	—	5	—	15	30	mW
Minimum Data Retention Voltage	V_{DR}	V_{DD}=V_{DR}			—	2	2.4	V
Data Retention Current	I_{DR}	V_{DD}=2.4 V			—	0.5	—	µA
Input Capacitance	C_{IN}				—	5	7.5	pF
Output Capacitance	C_{OUT}				—	10	15	

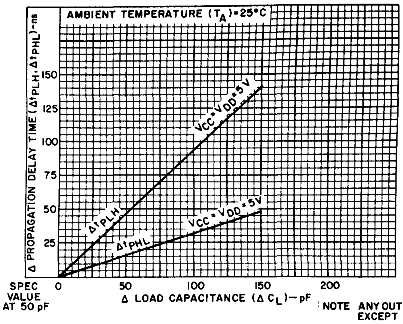
*Typical values are for T_A=25° C and nominal V_{DD}.

$^{\Delta}$Idle "00" at M(0000), C_L=50 pF



NOTES
IDLE = "00" AT M(0000)
BRANCH = "3707" AT M (8107)
C_L = 50 pF

Fig. 6 - Typical power dissipation as a function of clock frequency for BRANCH instruction and IDLE instruction.



NOTE ANY OUTPUT EXCEPT $\overline{XTAL}$

Fig. 7 - Typical change in propagation delay as a function of a change in load capacitance.

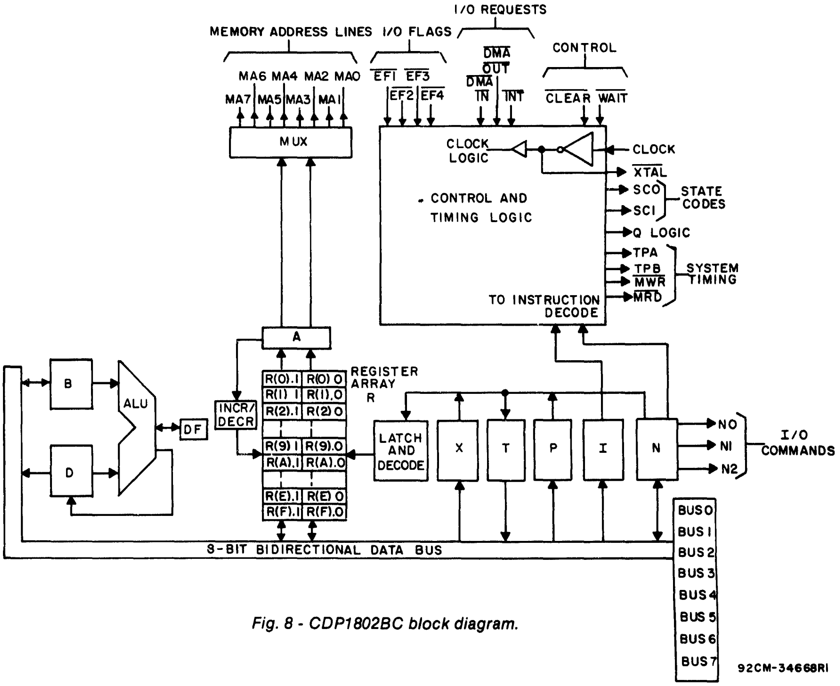


Fig. 8 - CDP1802BC block diagram.

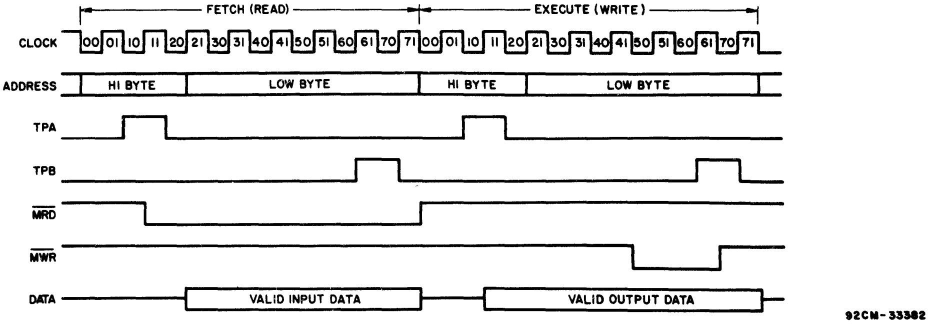


Fig. 9 - Basic dc timing waveforms, one instruction cycle.

SIGNAL DESCRIPTIONS

BUS 0 to 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.

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 MRD signal.

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

$\overline{\text{EF1}}$ to $\overline{\text{EF4}}$ (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.

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

These inputs are sampled by the CDP1802BC during the interval between the leading edge of TPB and the leading edge of TPA.

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_{CC}, 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. TPA is suppressed in IDLE when the CPU is in the load mode.

MA0 to MA7 (8 Memory Address Lines):

In each cycle, the higher-order byte of a 16-bit CPU 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 after the termination of TPA. Latching of all 8 higher-order address bits would permit a memory system of 64K bytes.

$\overline{\text{MWR}}$ (Write Pulse):

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

$\overline{\text{MRD}}$ (Read Level):

A low level on $\overline{\text{MRD}}$ 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, $\overline{\text{MRD}}$ 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 5 MHz at V_{CC}= V_{DD}= 5 volts. The clock is counted down internally to 8 clock pulses per machine cycle.

$\overline{\text{XTAL}}$:

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

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

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

$\overline{\text{CLEAR}}$	$\overline{\text{WAIT}}$	MODE
L	L	LOAD
L	H	RESET
H	L	PAUSE
H	H	RUN

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 peripheral devices operating at lower voltage. V_{CC} must be less than or equal to V_{DD}. All outputs swing from V_{SS} to V_{CC}. The recommended input voltage swing is V_{SS} to V_{CC}.

ARCHITECTURE

The CPU block diagram is shown in Fig. 8. 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 two exceptions, CPU instructions consist of two 8-clock-pulse machine cycles. The first cycle is the fetch cycle, and the second—and third 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 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);
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 1800-series 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 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.

The Q Flip Flop

An internal flip flop, Q, can be set or reset by instruction and can be sensed by conditional branch instructions. 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 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-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.

CDP1802BC

CPU Register Summary

D	8 Bits	Data Register (Accumulator)
DF	1 Bit	Data Flag (ALU Carry)
B	8 Bits	Auxiliary Holding Register
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 nibble)
IE	1 Bit	Interrupt Enable
Q	1 Bit	Output Flip Flop

CDP1802 Control Modes

The $\overline{\text{WAIT}}$ and $\overline{\text{CLEAR}}$ lines provide four control modes as listed in the following truth table:

$\overline{\text{CLEAR}}$	$\overline{\text{WAIT}}$	MODE
L	L	LOAD
L	H	RESET
H	L	PAUSE
H	H	RUN

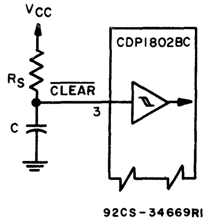
The function of the modes are defined as follows:

Load

Holds the CPU in the IDLE execution state 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-IN operation does not force execution of the next instruction.

Reset

Registers I, N, Q are reset, IE is 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 and registers X, P, and R(0) are reset. Interrupt and DMA servicing are suppressed during the initialization cycle. The next cycle is an S0, S1, or an S2 but never an 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. Powerup reset can be realized by connecting an RC network directly to the CLEAR pin, since it has a Schmitt-triggered input, see Fig. 10.



The RC time constant should be greater than the oscillator start-up time (typically 20 ms).

Fig. 10 - Reset diagram.

Pause

Stops the internal CPU timing generator on the first negative high-to-low transition of the input clock. The oscillator continues to operate, but subsequent clock transitions are ignored.

Run

May be initiated from the Pause or Reset mode functions. If initiated from Pause, the CPU resumes operation on the first negative high-to-low transition of the input clock. 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.

RUN-MODE STATE TRANSITIONS

The CDP1802BC CPU state transitions when in the RUN and RESET modes are shown in Fig. 11. Each machine cycle requires the same period of time, 8 clock pulses, except the initialization cycle, which requires 9 clock pulses. The execution of an instruction requires either two or three machine cycles, S0 followed by a single S1 cycle or two S1 cycles. S2 is the response to a DMA request and S3 is the interrupt response. Table II shows the conditions on Data Bus and Memory-Address lines during all machine states.

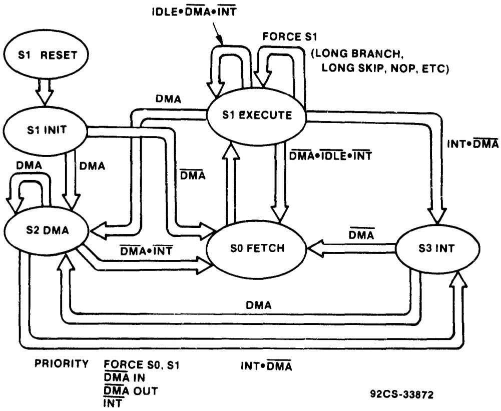

IDLE•$\overline{\text{DMA}}$•$\overline{\text{INT}}$
FORCE S1
(LONG BRANCH, LONG SKIP, NOP, ETC)
S1 RESET
DMA
S1 EXECUTE
S1 INIT
INT•$\overline{\text{DMA}}$
$\overline{\text{DMA}}$•$\overline{\text{IDLE}}$•$\overline{\text{INT}}$
DMA
DMA
$\overline{\text{DMA}}$
S2 DMA
$\overline{\text{DMA}}$
S0 FETCH
S3 INT
$\overline{\text{DMA}}$•$\overline{\text{INT}}$
DMA
PRIORITY
FORCE S0, S1
DMA IN
DMA OUT
INT
INT•$\overline{\text{DMA}}$
92CS-33872

Fig. 11 - State transition diagram.

INSTRUCTION SET

The CPU 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 (See Notes following table, pp. 11 and 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; (RN)+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, O→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		

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 ($\overline{\text{EF1}}=V_{SS}$)	B1	34	IF EF1=1, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF1=0 ($\overline{\text{EF1}}=V_{CC}$)	BN1	3C	IF EF1=0, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF2=1 ($\overline{\text{EF2}}=V_{SS}$)	B2	35	IF EF2=1, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF2=0 ($\overline{\text{EF2}}=V_{CC}$)	BN2	3D	IF EF2=0, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF3=1 ($\overline{\text{EF3}}=V_{SS}$)	B3	36	IF EF3=1, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF3=0 ($\overline{\text{EF3}}=V_{CC}$)	BN3	3E	IF EF3=0, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

TABLE I — INSTRUCTION SUMMARY (Cont'd)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
BRANCH INSTRUCTIONS—SHORT BRANCH			

SHORT BRANCH IF EF4=1
($\overline{\text{EF4}}=V_{SS}$)
	B4	37	IF EF4=1, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

SHORT BRANCH IF EF4=0
($\overline{\text{EF4}}=V_{CC}$)
	BN4	3F	IF EF4=0, M(R(P))→R(P).0
			ELSE R(P)+1→R(P)

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	CB	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

CDP1802BC

TABLE I — INSTRUCTION SUMMARY (Cont'd)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
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
SAVE	SAV	78	T→M(R(X))
PUSH X,P TO STACK	MARK	79	(X,P)→T; (X,P)→M(R(2)) THEN P→X; R(2)−1→R(2)
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
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

ƒTHE 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

§THIS INSTRUCTION IS ASSOCIATED WITH MORE THAN ONE MNEMONIC EACH MNEMONIC IS INDIVIDUALLY LISTED

#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.

Notes for TABLE I

1. Long-Branch, Long-Skip and No Op instructions are the only instructions that 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).

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A=-40 to +85° C, C_L=50 pF, V_{DD}±5%, except as noted.

CHARACTERISTIC		V_{CC} (V)	V_{DD} (V)	LIMITS Typ.•	LIMITS Max.	UNITS
Propagation Delay Times.						ns
Clock to TPA, TPB	t_{PLH}, t_{PHL}	5	5	200	300	
Clock-to-Memory High-Address Byte	t_{PLH}, t_{PHL}	5	5	475	525	
Clock-to-Memory Low-Address Byte Valid	t_{PLH}, t_{PHL}	5	5	175	250	
Clock to $\overline{MRD}$	t_{PLH}, t_{PHL}	5	5	175	275	
Clock to $\overline{MWR}$	t_{PLH}, t_{PHL}	5	5	175	225	
Clock to (CPU DATA to BUS) Valid	t_{PLH}, t_{PHL}	5	5	250	375	
Clock to State Code	t_{PLH}, t_{PHL}	5	5	250	400	
Clock to Q	t_{PLH}, t_{PHL}	5	5	200	300	
Clock to N (0-2)	t_{PLH}, t_{PHL}	5	5	275	350	
Minimum Setup and Hold Times·						
Data Bus Input Setup	t_{SU}	5	5	−20	0	
Data Bus Input Hold	t_H■	5	5	125	150	
$\overline{DMA}$ Setup	t_{SU}	5	5	0	30	
$\overline{DMA}$ Hold	t_H■	5	5	100	150	
Interrupt Setup	t_{SU}	5	5	−75	0	
Interrupt Hold	t_H■	5	5	75	125	
$\overline{WAIT}$ Setup	t_{SU}	5	5	20	40	
$\overline{EF1-4}$ Setup	t_{SU}	5	5	−30	0	
$\overline{EF1-4}$ Hold	t_H■	5	5	100	150	
Minimum Pulse Width Times.						
$\overline{CLEAR}$ Pulse Width	t_{WL}■	5	5	100	150	
$\overline{CLOCK}$ Pulse Width	t_{WL}	5	5	90	100	

•Typical values are for T_A=25°C and nominal V_{DD}

■Maximum limits of minimum characteristics are the values above which all devices function

Notes for TABLE I (Continued)

2. The short-branch instructions are two bytes long. The first byte specifies the condition to be tested, and the second 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

   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.

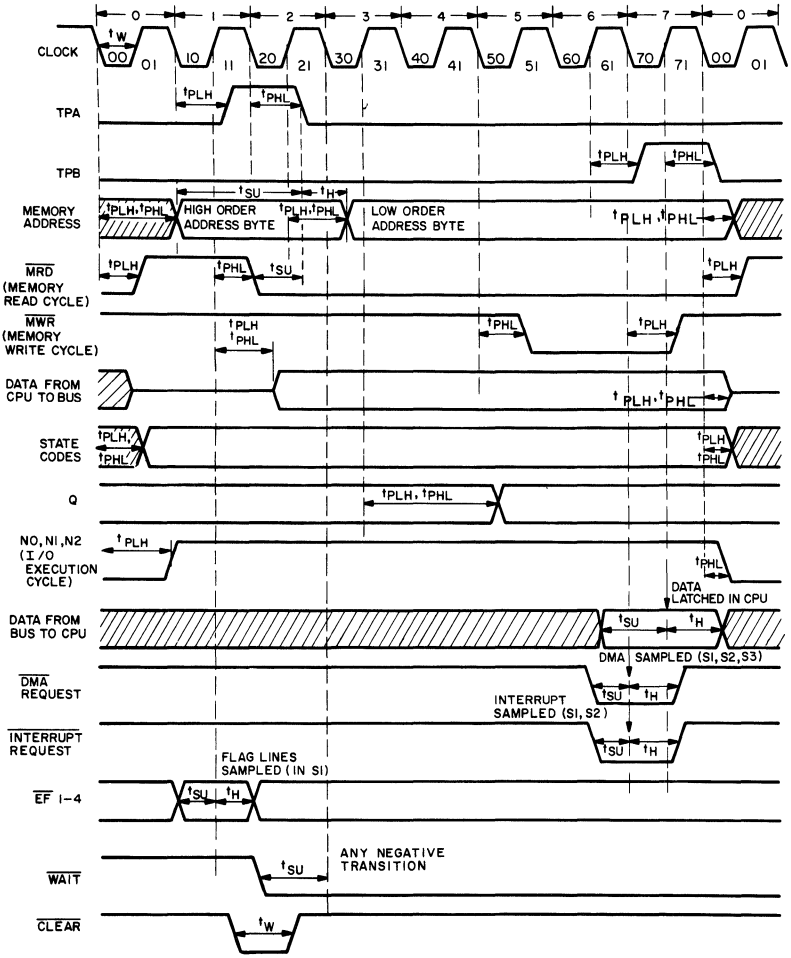
   The Long-Skip instructions take three cycles to complete (1 fetch +2 execute).

   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.

CDP1802BC



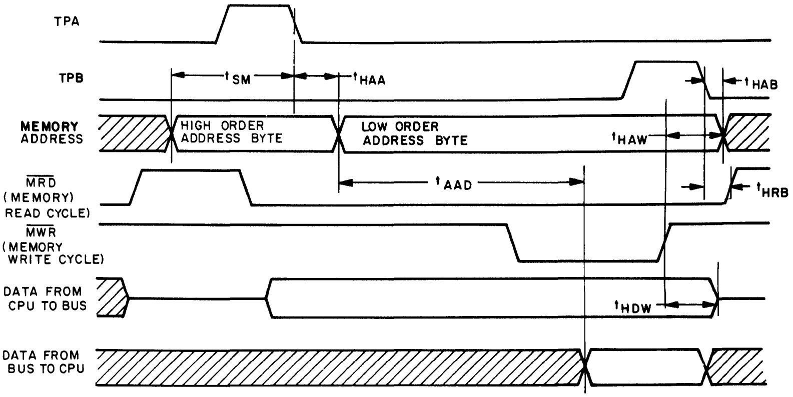
NOTES:

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. 12 - Timing waveforms.



92CM-34922

Fig. 13 - Clock frequency dependent relative timing waveforms.

TIMING SPECIFICATIONS as a function of T(T=1/f_{CLOCK}) at T_A=-40 to +85° C

CHARACTERISTIC		V_{CC} (V)	V_{DD} (V)	LIMITS		UNITS
				Min.	Typ.•	
High-Order Memory-Address Byte Set Up to TPA ↘ Time	t_{SM}	5	5	2T-325	2T-275	ns
High-Order Memory-Address Byte Hold after TPA Time	t_{HAA}	5	5	T/2-25	T/2-15	
Low-Order Memory-Address Byte Hold after WR Time	t_{HAW}	5	5	T-30	T+0	
CPU Data to Bus Hold after WR Time	t_{HDW}	5	5	T-175	T-125	
Low-Order Memory-Address Byte Hold after TPB Time	t_{HAB}	5	5	T/2+0	T/2+100	
$\overline{\text{MRD}}$ Hold to TPB Time	t_{HRB}	5	5	T/2-25	T/2+0	
Required Memory Access Time Address to Data	t_{AAD}	5	5	5T-225	5T-175	
$\overline{\text{MRD}}$ to TPA (↘)	t_{SU}	5	5	T/2-20	T/2-15	

•Typical values are for T_A=25° C and nominal V_{DD}

CDP1802BC

TABLE II. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES

STATE	I	N	MNEMONIC	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{MRD}$	$\overline{MWR}$	N LINES	NOTES[G]
S1	RESET			0→I,N,Q,X,P; 1→IE	00	XXXX	1	1	0	A
S1	INITIALIZE NOT PROGRAMMER ACCESSIBLE			0000→R	00	XXXX	1	1	0	B
S0	FETCH			MRP→I, N; RP+1→RP	MRP	RP	0	1	0	C
S1	0	0	IDL	IDLE	MR0	R0	0	1	0	D,3
	0	1-F	LDN	MRN→D	MRN	RN	0	1	0	3
	1	0-F	INC	RN+1→RN	FLOAT	RN	1	1	0	1
	2	0-F	DEC	RN−1→RN	FLOAT	RN	1	1	0	1

	3	0-F	SHORT
			BRANCH
				TAKEN;
				MRP→RP.0
				NOT TAKEN;
				RP+1→RP
					MRP	RP	0	1	0	3

	4	0-F	LDA	MRN→D; RN+1→RN	MRN	RN	0	1	0	3
	5	0-F	STR	D→MRN	D	RN	1	0	0	2
	6	0	IRX	RX+1→RX	MRX	RX	0	1	0	2
	6	1	OUT 1	MRX→BUS; RX+1→RX	MRX	RX	0	1	1	6
		2	OUT 2						2	
		3	OUT 3						3	
		4	OUT 4						4	
		5	OUT 5						5	
		6	OUT 6						6	
		7	OUT 7						7	
		9	INP 1	BUS→MRX,D	DATA FROM I/O DEVICE	RX	1	0	1	5
		A	INP 2						2	
		B	INP 3						3	
		C	INP 4						4	
		D	INP 5						5	
		E	INP 6						6	
		F	INP 7						7	
	7	0	RET	MRX→(X,P); RX+1→RX; 1→IE	MRX	RX	0	1	0	3
		1	DIS	MRX→(X,P); RX+1→RX; 0→IE	MRX	RX	0	1	0	3
		2	LDXA	MRX→D; RX+1→RX	MRX	RX	0	1	0	3
		3	STXD	D→MRX; RX−1→RX	D	RX	1	0	0	2
		4	ADC	MRX+D+ DF→DF,D	MRX	RX	0	1	0	3

TABLE II. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES (CONT'D)

STATE	I	N	MNEMONIC	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{MRD}$	$\overline{MWR}$	N LINES	NOTES[G]
S1	7	5	SDB	MRX–D–DFN→DF,D	MRX	RX	0	1	0	3
		6	SHRC	LSB(D)→DF; DF→MSB(D)	FLOAT	RX	1	1	0	1
		7	SMB	D–MRX–DFN→DF,D	MRX	RX	0	1	0	3
		8	SAV	T→MRX	T	RX	1	0	0	2
		9	MARK	(X,P)→T, MR2; P→X; R2–1→R2	T	R2	1	0	0	2
		A	REQ	0→Q	FLOAT	RP	1	1	0	1
		B	SEQ	1→Q	FLOAT	RP	1	1	0	1
		C	ADCI	MRP+D+DF→DF,D; RP+1	MRP	RP	0	1	0	3
		D	SDBI	MRP–D–DFN→DF,D; RP+1	MRP	RP	0	1	0	3
		E	SHLC	MSB(D)→DF, DF→LSB(D)	FLOAT	RP	1	1	0	1
		F	SMBI	D–MRP–DFN→DF,D; RP+1	MRP	RP	0	1	0	3
	8	0-F	GLO	RN.0→D	RN.0	RN	1	1	0	1
	9	0-F	GHI	RN 1→D	RN 1	RN	1	1	0	1
	A	0-F	PLO	D→RN.0	D	RN	1	1	0	1
	B	0-F	PHI	D→RN.1	D	RN	1	1	0	1
S1#1	C	0-3, 8-B	LONG BRANCH	TAKEN MRP→B; RP+1→RP	MRP	RP	0	1	0	4
#2				TAKEN: B→RP 1; MRP→RP.0	M(RP+1)	RP+1	0	1	0	4
S1#1				NOT TAKEN: RP+1→RP	MRP	RP	0	1	0	4
#2				NOT TAKEN: RP+1→RP	M(RP+1)	RP+1	0	1	0	4
S1#1		5 6 7 C D E F	LONG SKIP	TAKEN RP+1→RP	MRP	RP	0	1	0	4
#2				TAKEN RP+1→RP	M(RP+1)	RP+1	0	1	0	4
S1#1				NOT TAKEN: NO OPERATION	MRP	RP	0	1	0	4
#2				NOT TAKEN· NO OPERATION	MRP	RP	0	1	0	4
S1#1		4	NOP	NO OPERATION	MRP	RP	0	1	0	4
#2				NO OPERATION	MRP	RP	0	1	0	4

CDP1802BC

TABLE II. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES (CONT'D)

STATE	I	N	MNEMONIC	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{\text{MRD}}$	$\overline{\text{MWR}}$	N LINES	NOTES[G]
S1	D	0-F	SEP	N→P	NN	RN	1	1	0	1
	E	0-F	SEX	N→X	NN	RN	1	1	0	1
	F	0	LDX	MRX→D	MRX	RX	0	1	0	3
		1	OR	MRX OR D→D	MRX	RX	0	1	0	3
		2	AND	MRX AND D→D						
		3	XOR	MRX XOR D→D						
		4	ADD	MRX+D→DF,D						
		5	SD	MRX−D→DF,D						
		7	SM	D−MRX→DF,D						
		6	SHR	LSB(D)→DF; 0→MSB(D)	FLOAT	RX	1	1	0	1
		8	LDI	MRP→D; RP+1→RP	MRP	RP	0	1	0	3
		9	ORI	MRP OR D→D; RP+1→RP						
		A	ANI	MRP AND D→D; RP+1→RP						
		B	XRI	MRP XOR D→D; RP+1→RP						
		C	ADI	MRP+D→DF,D; RP+1→RP						
		D	SDI	MRP−D→DF,D; RP+1→RP						
		F	SMI	D−MRP→DF,D; RP+1→RP						
		E	SHL	MSB(D)→DF; 0→LSB(D)	FLOAT	RP	1	1	0	1
S2	DMA IN			BUS→MR0; R0+1→R0	DATA FROM I/O DEVICE	R0	1	0	0	F, 7
	DMA OUT			MR0→BUS; R0+1→R0	MR0	R0	0	1	0	F, 8
S3	INTERRUPT			X,P→T, 0→IE 1→P; 2→X	FLOAT	RN	1	1	0	9
S1	LOAD			IDLE ($\overline{\text{CLEAR}}$, $\overline{\text{WAIT}}$=0)	M(R0−1)	R0−1	0	1	0	E,3

NOTES:

A. IE=1, TPA, TPB suppressed, state=S1.

B. BUS=0 for entire cycle.

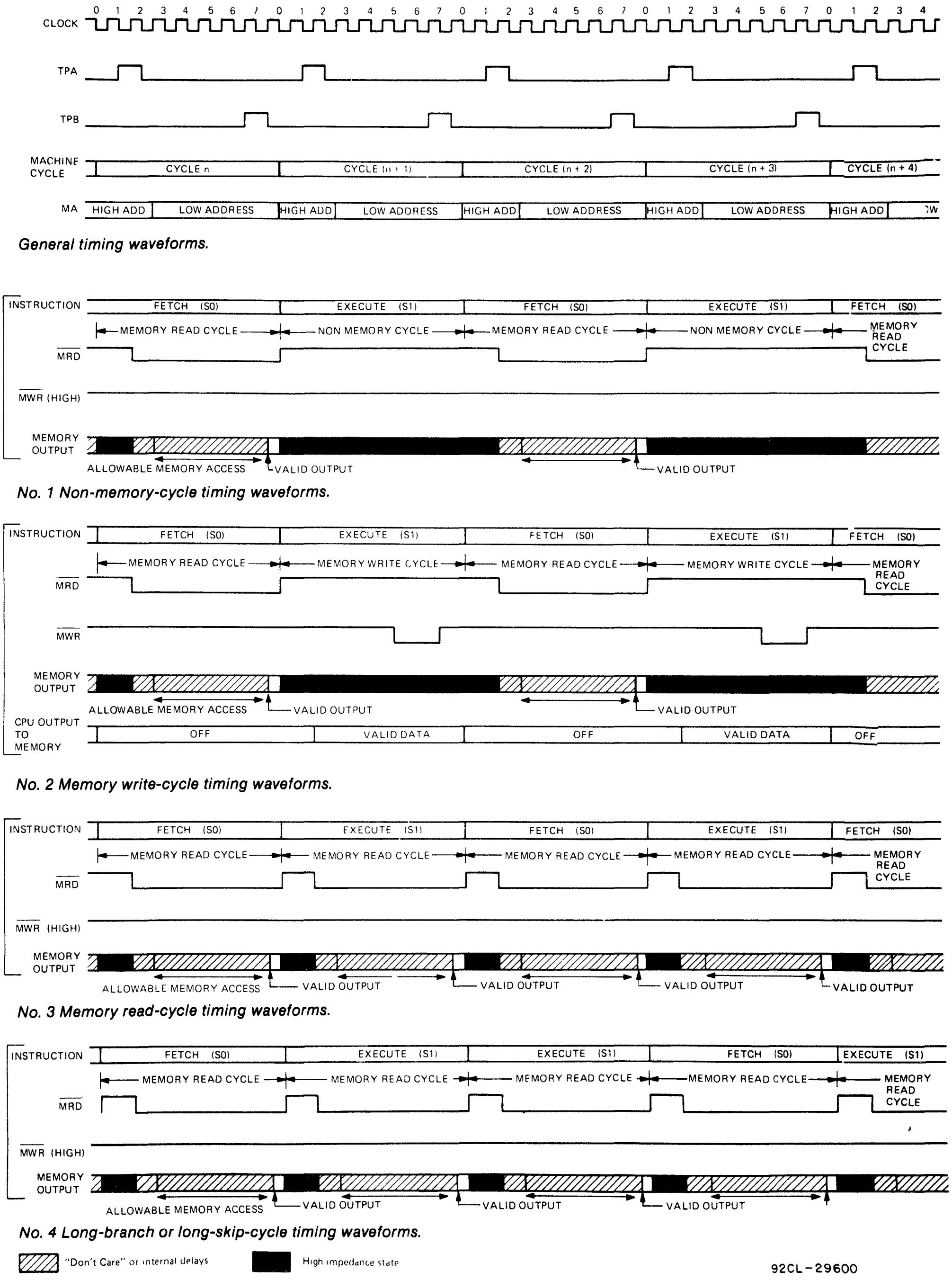
C. Next state always S1.

D. Wait for DMA or INTERRUPT.

E. Suppress TPA, wait for DMA.

F. IN REQUEST has priority over OUT REQUEST.

G. Number refers to machine cycle. See Fig. 14 timing waveforms for machine cycles 1 through 9.



General timing waveforms.

No. 1 Non-memory-cycle timing waveforms.

No. 2 Memory write-cycle timing waveforms.

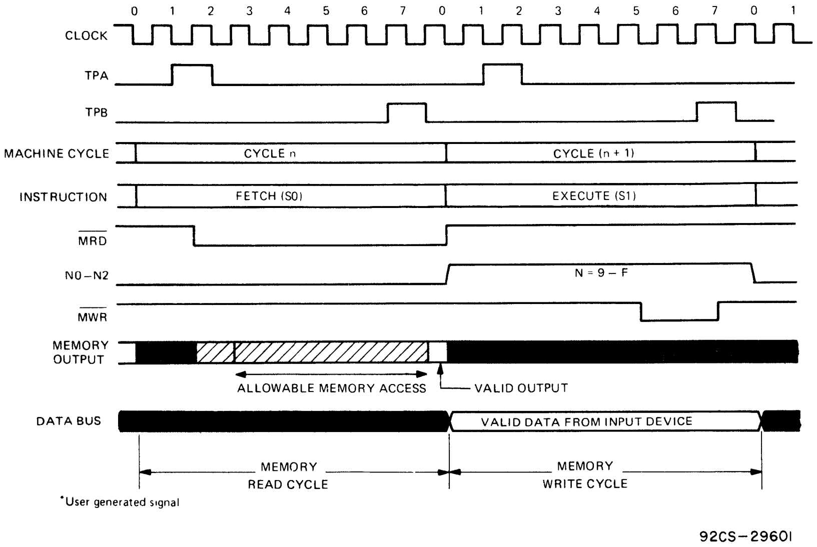
No. 3 Memory read-cycle timing waveforms.

No. 4 Long-branch or long-skip-cycle timing waveforms.

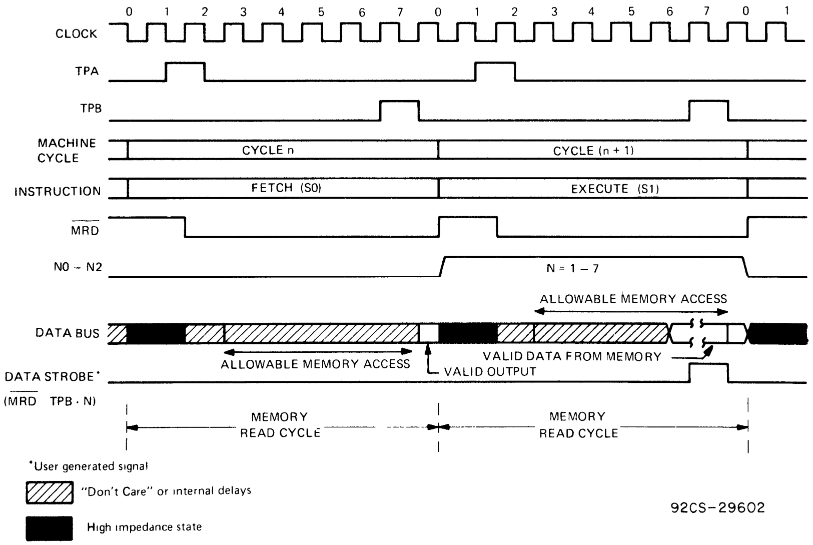
▨ "Don't Care" or internal delays ■ High impedance state

Fig. 14 - Machine-cycle timing waveforms (propagation delays not shown).

CDP1802BC

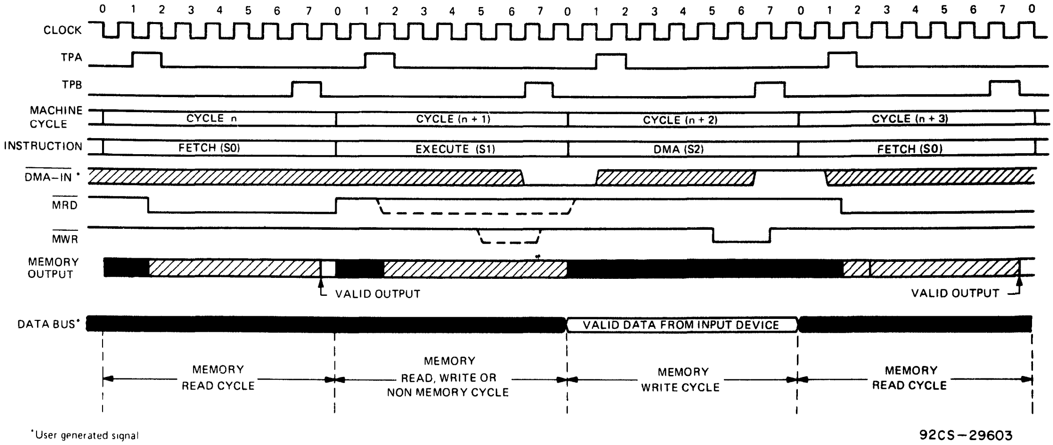


No. 5 Input-cycle timing waveforms.

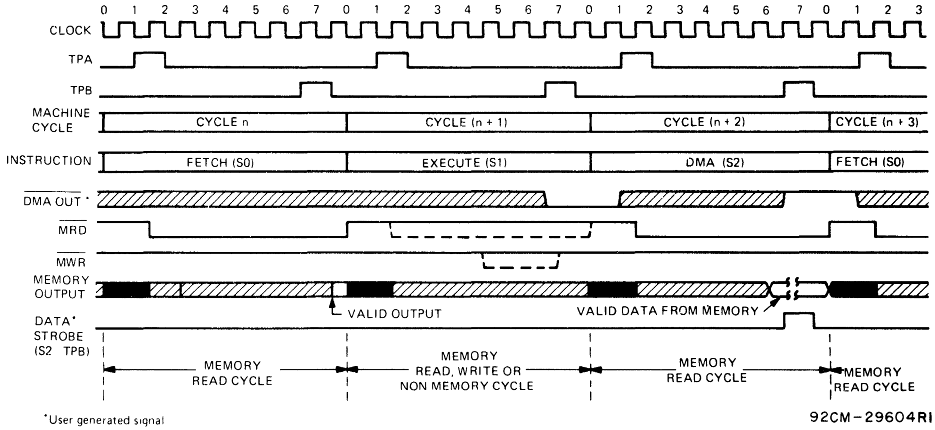


No. 6 Output-cycle timing waveforms.

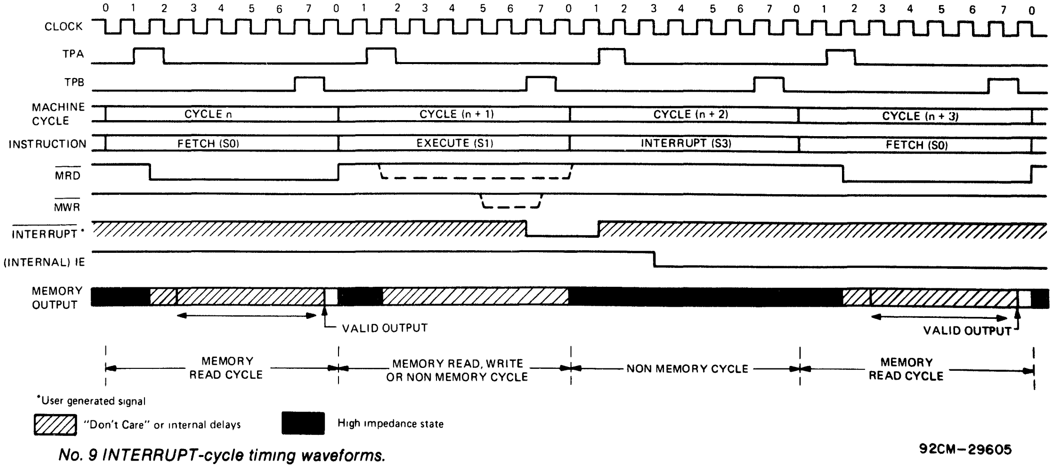
Fig. 14 - Machine-cycle timing waveforms (propagation delays not shown). Continued.



No. 7 DMA-IN-cycle timing waveforms.



No. 8 DMA-OUT-cycle timing waveforms.



No. 9 INTERRUPT-cycle timing waveforms.

Fig. 14 - Machine-cycle timing waveforms (propagation delays not shown). Continued.