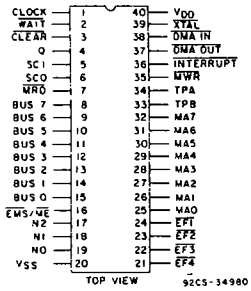
T-49-19-08

# CDP1804AC

**TERMINAL ASSIGNMENT**




# CMOS 8-Bit Microcomputer With On-Chip RAM, ROM, and Counter/Timer

**Performance Features:**

- *Instruction time of 3.2 µs, -40 to +85° C*
- *123 instructions-upwards sofware compatible with CDP1802, CDP1805A, and CDP1806A*
- *BCD arithmetic instructions*
- *Low-power IDLE mode*
- *Pin compatible with CDP1802, CDP1805A, and CDP1806A except for terminal 16 (terminal 18 for chip-carrier package)*
- *64K-byte memory address capability*
- *2 K bytes of on-chip ROM*
- *64 bytes of on-chip RAM*
- *16 x 16 matrix of on-board registers*
- *On-chip crystal or RC controlled oscillator*
- *8-bit Counter/Timer*

The RCA-CDP1804AC is a functional and performance enhancement of the CDP1802, CDP1805A, and CDP1806A CMOS 8-bit register-oriented microprocessor series and is designed for use in a wide variety of general-purpose applications.

The CDP1804AC hardware enhancements include a 2K-byte ROM, a 64-byte RAM, and a 8-bit presettable down counter. The Counter/Timer, which generates an internal interrupt request, can be programmed for use in time-base, event-counting, and pulse-duration measurement applications. The Counter/Timer underflow output can also be directed to the Q output terminal.

The CDP1805AC and CDP1806AC which are identical to the CDP1804AC, except for the on-chip memory, should be used for CDP1804AC development purposes.

The CDP1804AC software enhancements include 32 more instructions than the CDP1802. The 32 additional software instructions include subroutine call and return capability, enhanced data transfer manipulation, counter/timer control, improved interrupt handling, single-instruction loop counting, and BCD arithmetic.

Upwards software and hardware compatibility are maintained when substituting a CDP1804AC for other CDP1800-series microprocessors. Pinout is identical except for the replacement of $V_{CC}$ with $\overline{EMS}/\overline{ME}$.

The CDP1804AC has an operating voltage range of 4 V to 6.5 V and is supplied in a 40-lead hermetic dual-in-line ceramic package (D suffix), in a 40-lead dual-in-line plastic package (E suffix), and in a 44-lead plastic chip-carrier package (Q suffix).

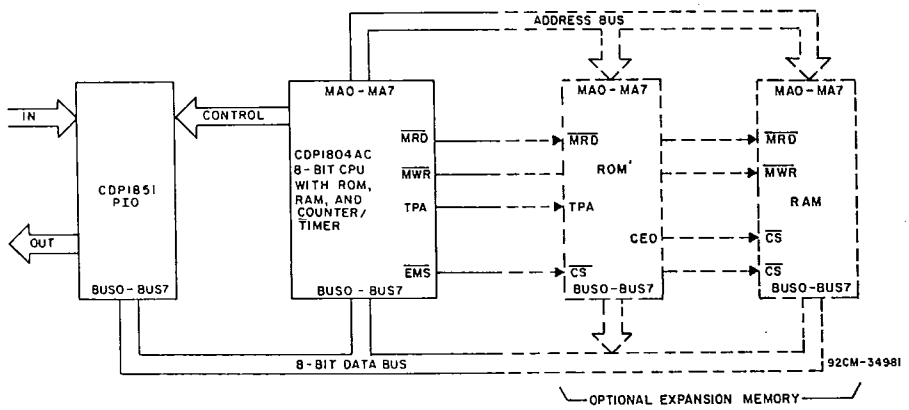



*Fig. 1 - Typical CDP1804AC microprocessor system.*

File Number **1371**

# CDP1804AC

T-49-19-08

**MAXIMUM RATINGS,** *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, ($V_{DD}$):
(Voltage referenced to $V_{SS}$ Terminal) ..... -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 +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 (All Package Types) ..... 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

2

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

**RECOMMENDED 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 | CONDITION | LIMITS | | UNITS |
|---|---|---|---|---|
| | | CDP1804ACD CDP1804ACE | | |
| | $V_{DD}$ (V) | MIN. | MAX. | |
| DC Operating Voltage Range | — | 4 | 6.5 | V |
| Input Voltage Range | — | $V_{SS}$ | $V_{DD}$ | |
| Minimum Instruction Time* ($f_{CL}$=5 MHz) | 5 | 3.2 | — | μs |
| Maximum DMA Transfer Rate | 5 | — | 0.625 | Mbytes/s |
| Maximum Clock Input Frequency, Load Capacitance (CL) = 50 pF | 5 | DC | 5 | MHz |
| Maximum External Counter/Timer Clock Input Frequency to $\overline{EF1}$, $\overline{EF2}$ $t_{CLX}$ | 5 | DC | 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.

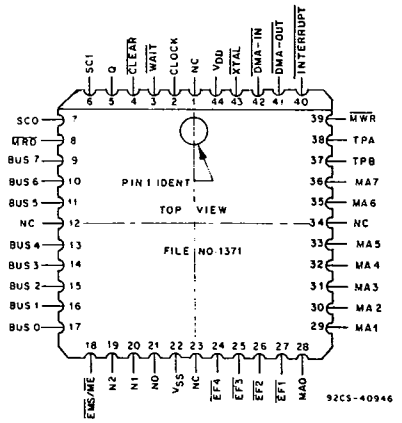
# CDP1804AC

**STATIC ELECTRICAL CHARACTERISTICS at $T_A$ = -40 to +85° C, $V_{DD}$ ± 5%, Except as noted**

| CHARACTERISTIC | | CONDITIONS | | | LIMITS | | | UNITS |
|---|---|---|---|---|---|---|---|---|
| | | | | | CDP1804ACD, CDP1804ACE | | | |
| | | $V_O$ (V) | $V_{IN}$ (V) | $V_{DD}$ (V) | Min. | Typ.* | Max. | |
| Quiescent Device Current | $I_{DD}$ | — | 0, 5 | 5 | — | 50 | 200 | μA |
| Output Low Drive (Sink) Current (Except $\overline{XTAL}$) | $I_{OL}$ | 0.4 | 0, 5 | 5 | 1.6 | 4 | — | mA |
| $\overline{XTAL}$ Output | $I_{OL}$ | 0.4 | 5 | 5 | 0.2 | 0.4 | — | mA |
| Output High Drive (Source) Current (Except $\overline{XTAL}$) | $I_{OH}$ | 4.6 | 0, 5 | 5 | -1.6 | -4 | — | mA |
| $\overline{XTAL}$ | $I_{OH}$ | 4.6 | 0 | 5 | -0.1 | -0.2 | — | mA |
| 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 | — | V |
| Input Low Voltage (BUS 0 — BUS 7, $\overline{EMS}/\overline{ME}$) | $V_{IL}$ | 0.5, 4.5 | — | 5 | — | — | 1.5 | V |
| Input High Voltage (BUS 0 — BUS 7, $\overline{EMS}/\overline{ME}$) | $V_{IH}$ | 0.5, 4.5 | — | 5 | 3.5 | — | — | V |
| Schmitt Trigger Input Voltage (Except BUS 0 — BUS 7, $\overline{EMS}/\overline{ME}$) | | | | | | | | V |
| Positive Trigger Threshold | $V_P$ | | | | 2.2 | 2.9 | 3.6 | V |
| Negative Trigger Threshold | $V_N$ | 0.5, 4.5 | — | 5 | 0.9 | 1.9 | 2.8 | V |
| Hysteresis | $V_H$ | | | | 0.3 | 0.9 | 1.6 | V |
| Input Leakage Current | $I_{IN}$ | — | 0, 5 | 5 | — | ±0.1 | ±5 | μA |
| 3-State Output Leakage Current | $I_{OUT}$ | 0, 5 | 0, 5 | 5 | — | ±0.2 | ±5 | μA |
| Input Capacitance | $C_{IN}$ | — | — | — | — | 5 | 7.5 | pF |
| Output Capacitance | $C_{OUT}$ | — | — | — | — | 10 | 15 | pF |
| Total Power Dissipation$^\Delta$ Run | | — | — | 5 | — | 35 | 50 | mW |
| Idle "00" at M(0000) | | — | — | 5 | — | 12 | 18 | mW |
| Minimum Data Retention Voltage | $V_{DR}$ | $V_{DD} = V_{DR}$ | | | — | 2 | 2.4 | V |
| Data Retention Current | $I_{DR}$ | $V_{DD}$ = 2.4 | | | — | 25 | 100 | μA |

*Typical values are for $T_A$ = 25° C and nominal $V_{DD}$. $^\Delta$External Clock: f=5 MHz, $t_r$,$t_f$=10 ns. $C_L$=50 pF.

**TERMINAL ASSIGNMENT**




**44-Lead Plastic Chip-Carrier Package (Q Suffix)**

T-49-19-08

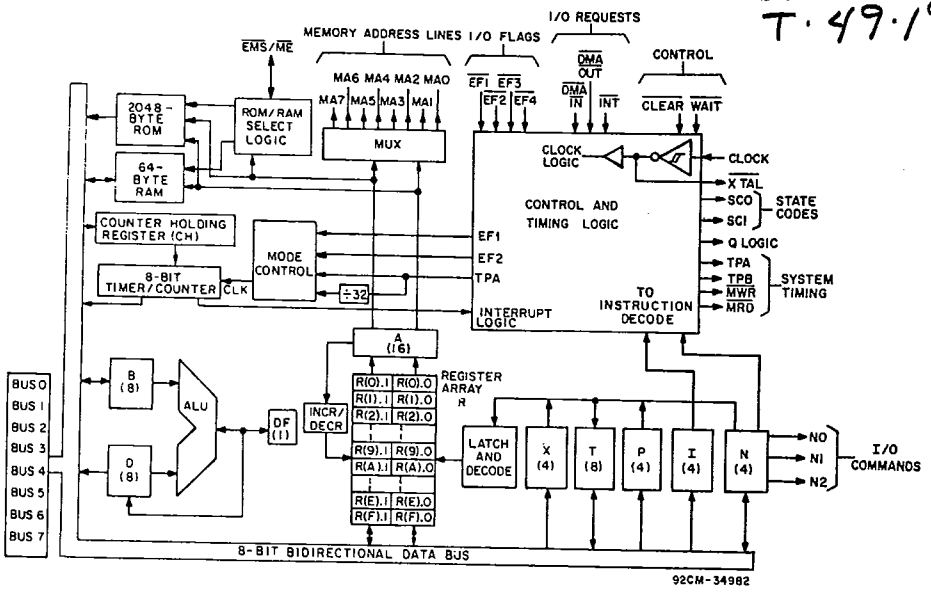



*Fig. 2 - Block diagram for CDP1804AC.*

## Enhanced 1804AC Operation

### ROM/RAM

The 2K-byte ROM is mask-programmable and mask-selectable in any 2K block of the available 64K address space in the RUN (ROM/RAM) mode. (The procedure is detailed in the Mask-Programming section at the end of the data sheet.)

The 64-byte RAM is mask-selectable in any 64-byte block of memory in the RUN (ROM/RAM) mode. It may also be externally selected via the $\overline{ME}$ input in the RUN (RAM only) mode.

The $\overline{EMS}/\overline{ME}$ pin serves a dual function. In the RUN (ROM/RAM) mode, $\overline{EMS}$ acts as an active low output to indicate when the internal ROM or RAM is not selected. This provides a convenient chip-select signal for any optional expansion memory devices and a stable-address latch signal for synchronous RAMs. In the RUN (RAM only) mode, $\overline{ME}$ acts as an active low input and is used to select the internal RAM, which is not mask-selected in this mode. Decoding is performed externally and the RAM may reside in any 64-byte block.

### Timing

Timing for the CDP1804AC is the same as the CDP1802 microprocessor series, with the following exceptions:

- 4.5 clock cycles are provided for memory access instead of 5.
- Q changes 1/2 clock cycle earlier during the SEQ and REQ instructions.
- Flag lines ($\overline{EF1}$-$\overline{EF4}$) are sampled at the end of the S0 cycle instead of at the beginning of the S1 cycle.
- Pause can only occur on the low-to-high transition of either TPA or TPB, instead of any negative clock transition.

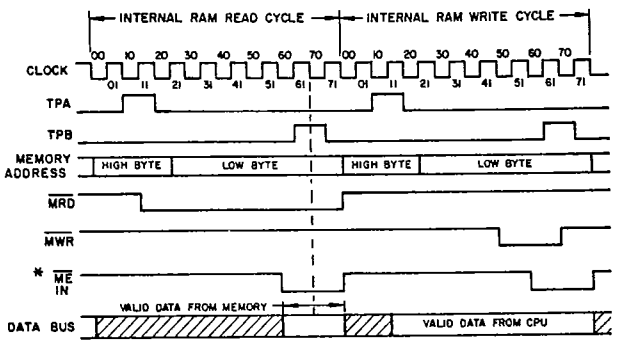
### Special Features

Schmitt triggers are provided on all inputs, except $\overline{EMS}/\overline{ME}$, and BUS 0 - BUS 7, for maximum immunity from noise and slow signal transitions. A Schmitt trigger in the oscillator section allows operation with an RC or crystal.

The CDP1802 series LOAD mode is not retained. This mode ($\overline{WAIT}$, $\overline{CLEAR}$=0) is the RUN (ROM/RAM) mode on the CDP1804AC.

A low power mode is provided, which is initiated via the IDLE instruction. In this mode all external signals, except the oscillator, are stopped on the low-to-high transition of TPB. All outputs remain in their previous states, $\overline{MRD}$ is set to a logic "1", and the data bus floats. The IDLE mode is exited by a DMA or INT condition. The INT includes both external interrupts and interrupts generated by the Counter/Timer. The only restrictions are that the Timer mode, which uses the TPA ÷ 32 clock source, and the underflow condition of the Pulse Width Measurement modes are not available to exit the IDLE mode.

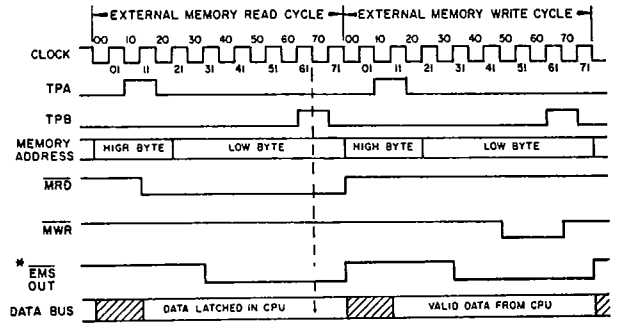
### TIMING WAVEFORMS FOR POSSIBLE OPERATING MODES




*NOTE FOR RUN (RAM ONLY) MODE:
ME HAS A MINIMUM SETUP AND HOLD TIME WITH RESPECT TO THE BEGINNING OF CLOCK 70. FOR A MEMORY READ OPERATION, RAM DATA WILL APPEAR ON THE DATA BUS DURING THE TIME ME IS ACTIVE AFTER CLOCK 31. THE TIME SHOWN CAN BE LONGER, IF FOR INSTANCE, A DMA OUT OPERATION IS PERFORMED ON INTERNAL RAM DATA, TO ALLOW DATA ENOUGH TIME TO BE LATCHED INTO AN EXTERNAL DEVICE. THE INTERNAL RAM IS AUTOMATICALLY DESELECTED AT THE END OF CLOCK 71, INDEPENDENT OF ME.
NOTE FOR RUN (ROM/RAM) MODE:
INTERNAL MEMORY DATA WILL APPEAR ON THE DATA BUS AFTER CLOCK PULSE 31.

92CS-34983

*Fig. 3 - Internal memory operation timing waveforms for CDP1804AC.*




*FOR RUN (ROM/RAM) MODE ONLY.
NOTE: FOR THE RUN (RAM ONLY) MODE ME MUST BE HIGH DURING EXTERNAL MEMORY ACCESSES.

92CS-34984

*Fig. 4 - External memory operation timing waveforms for CDP1804AC.*

### 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. 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:

MRD = $V_{DD}$: Input data from I/O to CPU and Memory
MRD = $V_{SS}$: Output data from Memory to I/O



2581 B-14

**$\overline{EF1}$ 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. The flag(s) are sampled at the end of every S0 cycle. $\overline{EF1}$ and $\overline{EF2}$ are also used for event counting and pulse-width measurement in conjunction with the Counter/Timer.

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

$\overline{DMA\text{-}IN}$ and $\overline{DMA\text{-}OUT}$ are sampled during TPB every S1, S2, and S3 cycle. $\overline{INTERRUPT}$ is sampled during TPB every S1 and S2 cycle.

**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 (MIE) 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 R(0) is incremented.

Note: In the event of concurrent DMA and INTERRUPT requests, DMA-IN has priority followed by DMA-OUT and then Interrupt. (The Interrupt request is not internally latched and must be held true after DMA).

**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.

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

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

**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 multiplexed 16-bit memory address.

**MA0 to MA7 (8 Memory Address Lines):**

In each cycle, the higher-order byte of a 16-bit 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 clock after the termination of TPA.

**$\overline{MWR}$ (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 and to indicate the direction of data transfer during and I/O instruction.

**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. The Q-line can also be controlled by the Counter/Timer underflow via the Enable Toggle Q instruction. The Enable Toggle Q 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 command is cleared by a LOAD COUNTER (LDC) instruction with the Counter/Timer stopped, a CPU reset, or a BRANCH COUNTER INTERRUPT (BCI) instruction with the counter interrupt flip-flop set.

**CLOCK:**

Input for externally generated single-phase clock. The maximum clock frequency is 5 MHZ at $V_{DD}$ = 5 V. 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, if the on-chip oscillator is utilized.

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

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

| $\overline{CLEAR}$ | $\overline{WAIT}$ | MODE |
|---|---|---|
| L | L | RUN (ROM/RAM) |
| L | H | RESET |
| H | L | PAUSE |
| H | H | RUN (RAM ONLY) |

**$\overline{ME}$ (Memory Enable) RUN (RAM ONLY) Mode**

This active low input is used to select or deselect the internal RAM. It must be active prior to clock 70 for an internal RAM access to take place. Internal RAM data will appear on the data bus during the time that $\overline{ME}$ is active (after clock 31). Thus, if this data is to be latched into an external device (i.e., during an OUTPUT instruction or DMA-OUT cycle), $\overline{ME}$ should be wide enough to provide enough time for valid data to be latched. The internal RAM is automatically deselected after clock 71. $\overline{ME}$ is ineffective when $\overline{MRD} \cdot \overline{MWR} = 1$.

In the RUN (RAM ONLY) mode the internal RAM is not internally mask-decoded. Decoding of the starting address is performed externally, and may reside in any 64-byte block of memory.

# CDP1804AC

**$\overline{EMS}$ (External Memory Select) RUN (ROM/RAM) Mode**

This active low output 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.

Note that in the RUN (ROM/RAM) mode data from the internal ROM or RAM, when selected, will appear on the data bus after clock 31.

**$V_{DD}$, $V_{SS}$, (Power Levels):**

$V_{SS}$ 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 $V_{DD}$.

## ARCHITECTURE

Fig. 2 shows a block diagram of the CDP1804AC. 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 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
4. to any other 16-bit scratch-pad register in the array.

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

Most 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 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
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
2. output instructions
3. input instructions
4. register to memory transfer
5. memory to register transfer
6. interrupt and subroutine handling.

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, and the RLDI instruction 68CN. 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 CDP1804AC 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, 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 new RLDI, RLXA, RSXD, and RNX instructions also allow loading, storing, and exchanging the full 16-bit contents of the R registers without affecting the D register. The new DBNZ instruction allows decrementing and branching-on-not-zero of any 16-bit R register also without affecting the D register.

### 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 underflow output of the Counter/Timer. The output of Q is also available as a microprocessor output.

### 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) |
| Q | 1 Bit | Output Flip-Flop |
| CNTR | 8-Bits | Counter/Timer |
| CH | 8 Bits | Holds Counter Jam Value |
| MIE | 1 Bit | Master Interrupt Enable |
| CIE | 1 Bit | Counter Interrupt Enable |
| XIE | 1 Bit | External Interrupt Enable |
| CIL | 1 Bit | Counter Interrupt Latch |

### 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. Master Interrupt Enable is automatically deactivated to inhibit further interrupts. The user's interrupt routine is now in control; the contents of T may be saved by means of a single SAV instruction (78) in the memory location pointed to by R(X) or the contents of T, D, and DF may be saved using a single DSAV instruction (6876). At the conclusion of the interrupt, the user's routine may restore the pre-interrupted value of X and P with either a RET instruction (70) which permits further interrupts, or a DIS instruction (71), which disables further interrupts.

T-49-19-08

### Interrupt Generation and Arbitration (See Fig. 5)

Interrupt requests can be generated from the following sources:

1. Externally through the interrupt input (Request not latched)
2. Internally due to Counter/Timer response (Request is latched)
   a. On the transition from count $(01)_{16}$ to its next value (counter underflow)
   b. On the ↓ transition of $\overline{EF1}$ in pulse measurement mode 1
   c. On the ↓ transition of $\overline{EF2}$ in pulse measurement mode 2

For an interrupt to be serviced by the CPU, the appropriate Interrupt Enable flip-flops must be set. Thus, the External Interrupt Enable flip-flop must be set to service an external interrupt request, and the Counter Interrupt Enable flip-flop must be set to service an Internal Counter/Timer interrupt request. In addition, the Master Interrupt Enable flip-flop (as used in the CDP1802A) must be set to service either type of request. All 3 flip-flops are initially enabled with the application of a hardware reset, and, can be selectively enabled or disabled with software: CIE, CID instructions for the CIE flip-flop; XIE, XID instructions for the XIE flip-flop; RET, DIS instructions for the MIE flip-flop.

Short branch instructions on Counter Interrupt (BCI) and External Interrupt (BXI) can be placed in the user's interrupt service routine to provide a means of identifying and prioritizing the interrupt source. Note, however, that since the External Interrupt request is not latched, it must remain active until the short branch is executed if this priority arbitration scheme is used.

Interrupt requests can also be polled if automatic interrupt service is not desired (MIE=0). With the Counter Interrupt and External Interrupt short branch instructions, the branch will be taken if an interrupt request is pending, regardless of the state of any of the 3 Interrupt Enable flip-flops. The latched counter interrupt request signal will be reset when the branch is taken, when the CPU is reset, or with a LDC instruction with the Counter stopped. Note that exiting a counter-initiated interrupt routine without resetting the counter interrupt latch will result in immediately re-entering the interrupt routine.

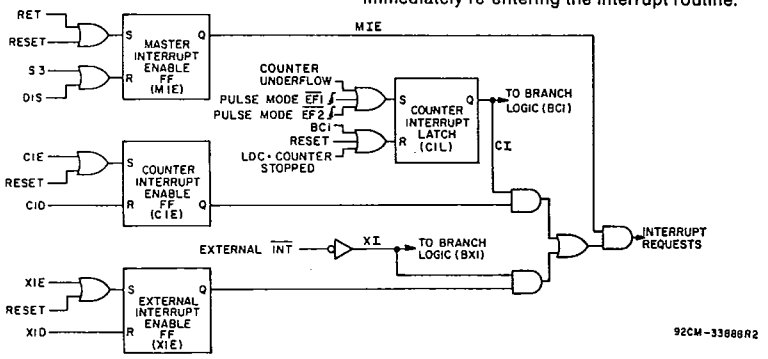



*Fig. 5 - Interrupt logic-control diagram for CDP1804AC.*





2584 C-03

# CDP1804AC

**Counter/Timer and Controls (See Fig. 6)**

This logic consists of a presettable 8-bit down-counter (Modulo 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 Counter Interrupt Latch. It will continue decrementing on subsequent counts. If the counter is preset to $(00)_{16}$ a full 256 counts will occur.

During a Load Counter instruction (LDC) if the counter was stopped with a STPC instruction, the counter and its holding register (CH) are loaded with the value in the D register and any previous counter interrupt is cleared. If the LDC is executed when the counter is running, the contents of the D register are loaded into the holding register (CH) only and any previous counter interrupt is not cleared. (LDC resets the Counter Interrupt Latch only when the counter is stopped). After counting down to $(01)_{16}$ the next count will load the new initial value into the counter, set the Counter Interrupt Latch, and operation will continue.

The Counter/Timer 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. The divide-by-32 prescaler is reset when the counter is in a mode other than the Timer mode, system reset, or stopped by a STPC.
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 (gate input) is low. On the transition of $\overline{EF1}$ to the positive state, the count is stopped, the mode is cleared, and the interrupt request latched. If the counter underflows while the input is low, interrupt will also be set, but counting will continue.
5. Pulse Duration Measurement 2: Operation is identical to Pulse Duration Measurement 1, except $\overline{EF2}$ is used as the gate input.

The modes can be changed without affecting the stored count.

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) instruction clears the counter mode and stops counting. The STPC instruction should be executed prior to a GEC instruction, if the counter is in the Event Counter Mode 1 or 2.

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 other modes, the instruction should be used only after the mode has been cleared by a Stop Counter instruction.

The Enable Toggle Q instruction (ETQ) 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 output changes state. This action is independent of the counter mode and the Interrupt Enable flip-flops. The Enable Toggle Q condition is cleared by an LDC with the Counter/Timer stopped; system Reset, or a BCI with CI=1. Note that SEQ and REQ instructions are independent of ETQ.—they can Set or Reset Q while the counter is running.

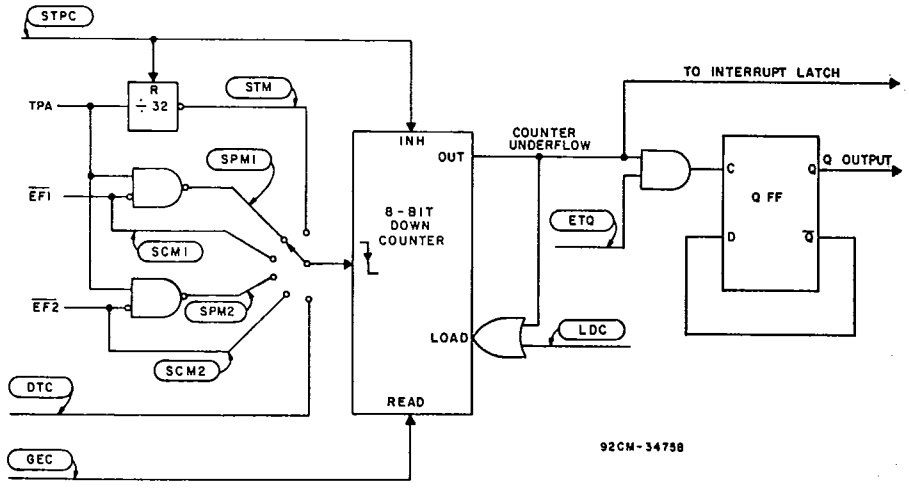



*Fig. 6 - Counter/Timer diagram for CDP1804AC.*

T-49-19-08

### On-Chip Clock (See Figs. 7, 8 and 9)

Clock circuits may use either an external crystal or an RC network.

A typical crystal oscillator circuit is shown in Fig. 7. The crystal is connected between terminals 1 and 39 (CLOCK and $\overline{XTAL}$) in parallel with a resistance. RF (1 megohm typ.). Frequency trimming capacitors, $C_{IN}$ and $C_{OUT}$, may be required at terminals 1 and 39. For additional information on crystal oscillators, see ICAN-6565.

Because of the Schmitt Trigger input, an RC oscillator can be used as shown in Fig. 8. The frequency is approximately 1/RC (See Fig. 9).

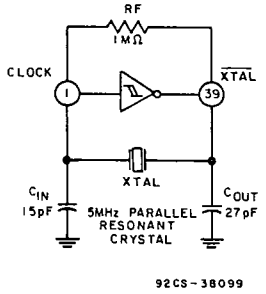



*Fig. 7 - Typical 5-MHz crystal oscillator.*

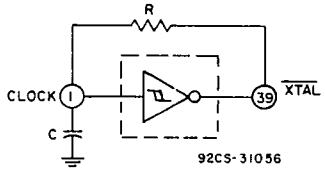



*Fig. 8 - RC network for oscillator.*

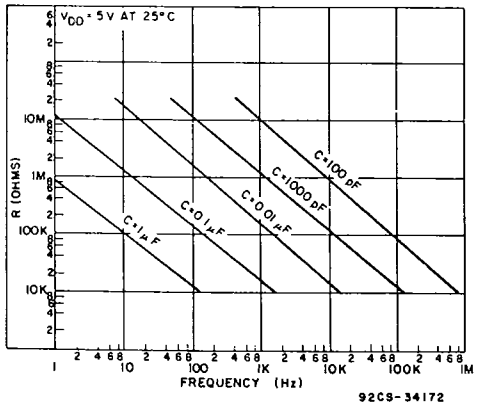



*Fig. 9 - Nominal component values as a function of frequency for the RC oscillator.*

### CONTROL MODES

| $\overline{CLEAR}$ | $\overline{WAIT}$ | MODE |
|---|---|---|
| L | L | RUN (ROM/RAM) |
| L | H | RESET |
| H | L | PAUSE |
| H | H | RUN (RAM ONLY) |

The function of the modes are defined as follows:

### RESET

The levels of the CDP1804A external signal lines will asynchronously be forced by RESET to the following states:

| | | |
|---|---|---|
| Q=0 | SC1,SC0=0, 1 | BUS 0-7=0 |
| $\overline{EMS}$/$\overline{ME}$=INPUT | (EXECUTE) | MA0-7=R0.1 |
| $\overline{MRD}$=1 | N0, N1, N2=0, 0, 0 | TPA=0 |
| TPB=0 | $\overline{MWR}$=1 | |

**Internal changes caused by RESET are:**
I, N instruction register is cleared to 00. XIE and CIE are set to allow interrupts following initialize. CIL is cleared (any pending counter interrupt is cleared), counter is stopped, the counter mode is cleared, and ETQ is disabled.

### Initialization Cycle

The first machine cycle following termination of RESET is an initialization cycle which requires 9 clock pulses. During this cycle the CPU remains in S1 and the following additional changes occur:

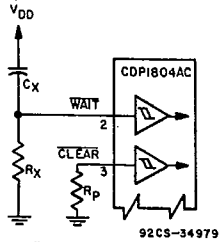
I → MIE
X, P → T (The old value of X, P will be put into T. This only has meaning following an orderly Reset with power applied).
X, P, R0 → 0 (X, P and R0 are cleared).

Interrupt and DMA servicing is suppressed during the initialization cycle. The next cycle is an S0 or an S2 but never an S1 or S3. The use of a 71 instruction followed by 00 at memory locations 0000 and 0001, may be used to reset MIE so as to preclude interrupts until ready for them.

**Reset and Initialize do not affect:**
**D (Accumulator)**
**DF**
**R1, R2, R3, R4, R5, R6, R7, R8, R9, RA, RB, RC, RD, RE, RF**
**CH (Counter Holding Register)**
**Counter (the counter is stopped but the value is unaffected)**

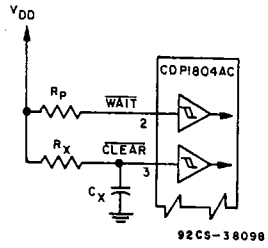
### Power-up Reset/Run Circuits

Power-up Reset/Run (ROM/RAM) and Reset/Run (RAM only) can be realized with the circuits shown in Fig. 10 and 11.




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

*Fig. 10 - Reset/Run (ROM/RAM) diagram.*




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

*Fig. 11 - Reset/Run (RAM only) diagram.*

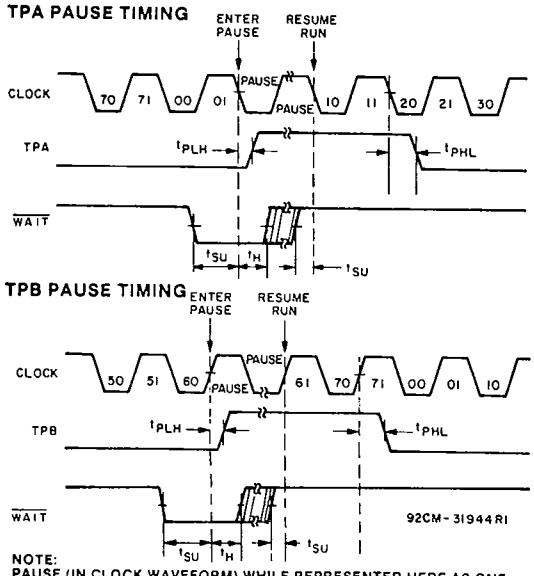
### PAUSE

Pause is a low power mode which stops the internal CPU timing generator and freezes the state of the processor. The CPU may be held in the Pause mode indefinitely. Hardware pause can occur at two points in a machine cycle, on the low to high transition of either TPA or TPB. A TPB pause can also be initiated by software with the execution of an IDLE instruction. In the pause mode, the oscillator continues to run but subsequent clock transitions are ignored. TPA and TPB remain at their previous state (see Fig. 12).

Pause is entered from RUN (RAM only) by dropping $\overline{WAIT}$ low, and from RUN (ROM/RAM) by raising $\overline{CLEAR}$ high. Appropriate setup and hold times must be met.

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

Hardware-initiated pause is exited to RUN (RAM only) by raising the $\overline{Wait}$ line, and the RUN (ROM/RAM) by lowering $\overline{CLEAR}$. Pause entered with an IDLE instruction requires $\overline{DMA}$, $\overline{INTERRUPT}$ or RESET to resume execution.




NOTE:
PAUSE (IN CLOCK WAVEFORM) WHILE REPRESENTED HERE AS ONE CLOCK CYCLE IN DURATION, COULD BE INFINITELY LONG.

*Fig. 12 - Pause mode timing waveforms.*

### 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 next high-to-low clock transition, while if paused at TPB, it will resume on the next low-to-high clock transition. (See Fig. 12). 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.

### SCHMITT TRIGGER INPUTS

All inputs except BUS 0 — BUS 7 and $\overline{ME}$ contain a Schmitt Trigger circuit, which is especially useful on the $\overline{CLEAR}$ input as a power-up RESET (See Fig. 10 and 11) and the CLOCK input (See Fig. 7 and 8).

### STATE TRANSITIONS

The CDP1804AC state transitions are shown in Fig. 13. Each machine cycle requires the same period of time, 8 clock pulses, except the initialization cycle (INT) which requires 9 clock pulses. Reset is asynchronous and can be forced at any time.

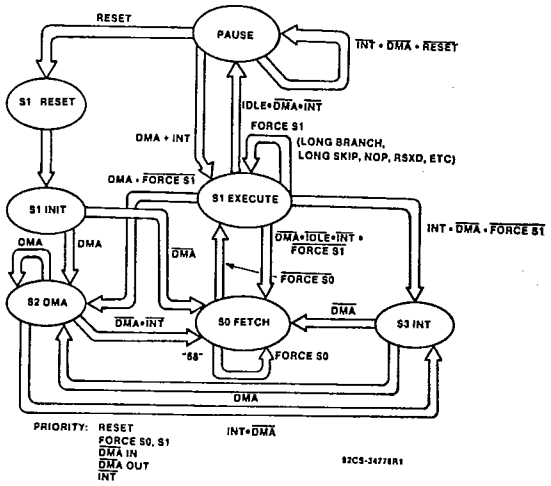



*Fig. 13 - State transition diagram.*

## INSTRUCTION SET

The CDP1804AC 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**

| INSTRUCTION | NO. OF MACHINE CYCLES | MNEMONIC | OP CODE | OPERATION |
|---|---|---|---|---|
| MEMORY REFERENCE | | | | |
| LOAD IMMEDIATE | 2 | LDI | F8 | M(R(P))→D; R(P)+1→R(P) |
| REGISTER LOAD IMMEDIATE | 5 | RLDI | 68CN■ | M(R(P))→R(N).1; M(R(P))+1→R(N).0; R(P)+2→R(P) |
| LOAD VIA N | 2 | LDN | 0N | M(R(N))→D; FOR N NOT 0 |
| LOAD ADVANCE | 2 | LDA | 4N | M(R(N))→D; R(N)+1→R(N) |
| LOAD VIA X | 2 | LDX | F0 | M(R(X))→D |
| LOAD VIA X AND ADVANCE | 2 | LDXA | 72 | M(R(X))→D; R(X)+1→R(X) |
| REGISTER LOAD VIA X AND ADVANCE | 5 | RLXA | 686N■ | M(R(X))→R(N).1; M(R(X)+1)→R(N).0; R(X))+2→R(X) |
| STORE VIA N | 2 | STR | 5N | D→M(RN)) |
| STORE VIA X AND DECREMENT | 2 | STXD | 73 | D→M(R(X)); R(X)-1→R(X) |
| REGISTER STORE VIA X AND DECREMENT | 5 | RSXD | 68AN■ | R(N).0→M(R(X)); R(N).1→M(R(X)-1); R(X)-2→R(X) |
| REGISTER OPERATIONS | | | | |
| INCREMENT REG N | 2 | INC | 1N | R(N)+1→R(N) |
| DECREMENT REG N | 2 | DEC | 2N | R(N)-1→R(N) |
| DECREMENT REG N AND LONG BRANCH IF NOT EQUAL 0 | 5 | DBNZ | 682N | R(N)-1→R(N); IF R(N) NOT 0, M(R(P))→R(P).1, M(R(P)+1)→R(P).0, ELSE R(P)+2→R(P) |
| INCREMENT REG X | 2 | IRX | 60 | R(X)+1→R(X) |
| GET LOW REG N | 2 | GLO | 8N | R(N).0→D |
| PUT LOW REG N | 2 | PLO | AN | D→R(N).0 |
| GET HIGH REG N | 2 | GHI | 9N | R(N).1→D |
| PUT HIGH REG N | 2 | PHI | BN | D→R(N).1 |
| REGISTER N TO REGISTER X COPY | 4 | RNX | 68BN■ | R(N)→R(X) |
| LOGIC OPERATIONS (Note 5) | | | | |
| OR | 2 | OR | F1 | M(R(X)) OR D→D |
| OR IMMEDIATE | 2 | ORI | F9 | M(R(P)) OR D→D; R(P)+1→R(P) |
| EXCLUSIVE OR | 2 | XOR | F3 | M(R(X)) XOR D→D |
| EXCLUSIVE OR IMMEDIATE | 2 | XRI | FB | M(R(P)) XOR D→D; R(P)+1→R(P) |
| AND | 2 | AND | F2 | M(R(X)) AND D→D |
| AND IMMEDIATE | 2 | ANI | FA | M(R(P)) AND D→D; R(P)+1→R(P) |
| SHIFT RIGHT | 2 | SHR | F6 | SHIFT D RIGHT, LSB(D)→DF, 0→MSB(D) |
| SHIFT RIGHT WITH CARRY | 2 | SHRC | 76▲ | SHIFT D RIGHT, LSB(D)→DF, DF→MSB(D) |
| RING SHIFT RIGHT | 2 | RSHR | | |
| SHIFT LEFT | 2 | SHL | FE | SHIFT D LEFT, MSB(D)→DF, 0→LSB(D) |

■Previous contents of T register are destroyed during instruction execution.

▲This instruction is associated with more than one mnemonic. Each mnemonic is individually listed.

Table I — INSTRUCTION SUMMARY (Cont'd)

| INSTRUCTION | NO. OF MACHINE CYCLES | MNEMONIC | OP CODE | OPERATION |
|---|---|---|---|---|
| LOGIC OPERATIONS (Note 5) (Cont'd) | | | | |
| SHIFT LEFT WITH CARRY | 2 | SHLC | 7E▲ | SHIFT D LEFT, MSB(D)→DF, |
| RING SHIFT LEFT | 2 | RSHL | | DF→LSB(D) |
| ARITHMETIC OPERATIONS (Note 5) | | | | |
| ADD | 2 | ADD | F4 | M(R(X))+D→DF, D |
| DECIMAL ADD | 4 | DADD | 68F4 | M(R(X))+D→DF, D<br>DECIMAL ADJUST→DF, D |
| ADD IMMEDIATE | 2 | ADI | FC | M(R(P))+D→DF, D; R(P)+1→R(P) |
| DECIMAL ADD IMMEDIATE | 4 | DADI | 68FC | M(R(P))+D→DF,D<br>R(P)+1→R(P)<br>DECIMAL ADJUST→DF, D |
| ADD WITH CARRY | 2 | ADC | 74 | M(R(X))+D+DF→DF, D |
| DECIMAL ADD WITH CARRY | 4 | DADC | 6874 | M(R(X))+D+DF→DF, D<br>DECIMAL ADJUST→DF, D |
| ADD WITH CARRY, IMMEDIATE | 2 | ADCI | 7C | M(R(P))+D+DF→DF, D<br>R(P)+1→R(P) |
| DECIMAL ADD WITH CARRY, IMMEDIATE | 4 | DACI | 687C | M(R(P))+D+DF→DF, D<br>R(P)+1→R(P)<br>DECIMAL ADJUST→DF, D |
| SUBTRACT D | 2 | SD | F5 | M(R(X))-D→DF, D |
| SUBTRACT D IMMEDIATE | 2 | SDI | FD | M(R(P))-D→DF, D;<br>R(P)+1→R(P) |
| SUBTRACT D WITH BORROW | 2 | SDB | 75 | M(R(X))-D-(NOT DF)→DF, D |
| SUBTRACT D WITH BORROW, IMMEDIATE | 2 | SDBI | 7D | M(R(P))-D-(NOT DF)→DF, D;<br>R(P)+1→R(P) |
| SUBTRACT MEMORY | 2 | SM | F7 | D-M(R(X))→DF, D |
| DECIMAL SUBTRACT MEMORY | 4 | DSM | 68F7 | D-M(R(X))→DF, D<br>DECIMAL ADJUST→DF, D |
| SUBTRACT MEMORY IMMEDIATE | 2 | SMI | FF | D-M(R(P))→DF, D;<br>R(P)+1→R(P) |
| DECIMAL SUBTRACT MEMORY, IMMEDIATE | 4 | DSMI | 68FF | D-M(R(P))→DF, D<br>R(P)+1→R(P)<br>DECIMAL ADJUST→DF, D |
| SUBTRACT MEMORY WITH BORROW | 2 | SMB | 77 | D-M(R(X))-(NOT DF)→DF, D |
| DECIMAL SUBTRACT MEMORY WITH BORROW | 4 | DSMB | 6877 | D-M(R(X))-(NOT DF)→DF, D<br>DECIMAL ADJUST→DF, D |
| SUBTRACT MEMORY WITH BORROW, IMMEDIATE | 2 | SMBI | 7F | D-M(R(P))-(NOT DF)→DF, D<br>R(P)+1→R(P) |
| DECIMAL SUBTRACT MEMORY WITH BORROW, IMMEDIATE | 4 | DSBI | 687F | D-M(R(P))-(NOT DF)→DF, D<br>R(P)+1→R(P)<br>DECIMAL ADJUST→DF, D |
| BRANCH INSTRUCTIONS — SHORT BRANCH | | | | |
| SHORT BRANCH | 2 | BR | 30 | M(R(P))→R(P).0 |
| NO SHORT BRANCH (SEE SKP) | 2 | NBR | 38▲ | R(P)+1→R(P) |
| SHORT BRANCH IF D = 0 | 2 | BZ | 32 | IF D = 0, M(R(P))→R(P).0<br>ELSE R(P)+1→R(P) |
| SHORT BRANCH IF D NOT 0 | 2 | BNZ | 3A | IF D NOT 0, M(R(P))→R(P).0<br>ELSE R(P)+1→R(P) |

▲This instruction is associated with more than one mnemonic. Each mnemonic is individually listed.

T-49-19-08

Table I — INSTRUCTION SUMMARY (Cont'd)

| INSTRUCTION | NO. OF MACHINE CYCLES | MNEMONIC | OP CODE | OPERATION |
|---|---|---|---|---|
| BRANCH INSTRUCTIONS — SHORT BRANCH (Cont'd) | | | | |
| SHORT BRANCH IF DF = 1 | 2 | BDF | 33▲ | IF DF = 1, M(R(P))→R(P).0 |
| SHORT BRANCH IF POS OR ZERO | 2 | BPZ | | ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EQUAL OR GREATER | 2 | BGE | | |
| SHORT BRANCH IF DF = 0 | 2 | BNF | 3B▲ | IF D = 0, M(R(P))→R(P).0 |
| SHORT BRANCH IF MINUS | 2 | BM | | ELSE R(P)+1→R(P) |
| SHORT BRANCH IF LESS | 2 | BL | | |
| SHORT BRANCH IF Q = 1 | 2 | BQ | 31 | IF Q = 1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF Q = 0 | 2 | BNQ | 39 | IF Q = 0, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF1 = 1 ($\overline{EF1}$ = $V_{SS}$) | 2 | B1 | 34 | IF EF1 = 1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF1 = 0 ($\overline{EF1}$ = $V_{DD}$) | 2 | BN1 | 3C | IF EF1 = 0, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF2 = 1 ($\overline{EF2}$ = $V_{SS}$) | 2 | B2 | 35 | IF EF2 = 1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF2 = 0 ($\overline{EF2}$ = $V_{DD}$) | 2 | BN2 | 3D | IF EF2 = 0, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF3 = 1 ($\overline{EF3}$ = $V_{SS}$) | 2 | B3 | 36 | IF EF3 = 1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF3 = 0 ($\overline{EF3}$ = $V_{DD}$) | 2 | BN3 | 3E | IF EF3 = 0, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF4 = 1 ($\overline{EF4}$ = $V_{SS}$) | 2 | B4 | 37 | IF EF4 =1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH IF EF4 = 0 ($\overline{EF4}$ = $V_{DD}$) | 2 | BN4 | 3F | IF EF4 = 0, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| SHORT BRANCH ON COUNTER INTERRUPT | 3 | BCI | 683E* | IF CI=1, M(R(P))→R(P).0; 0→CI ELSE R(P)+1→R(P) |
| SHORT BRANCH ON EXTERNAL INTERRUPT | 3 | BXI | 683F | IF XI=1, M(R(P))→R(P).0 ELSE R(P)+1→R(P) |
| BRANCH INSTRUCTIONS — LONG BRANCH | | | | |
| LONG BRANCH | 3 | LBR | C0 | M(R(P))→R(P).1, M(R(P)+1)→R(P).0 |
| NO LONG BRANCH (SEE LSKP) | 3 | NLBR | C8▲ | R(P)+2→R(P) |
| LONG BRANCH IF D = 0 | 3 | 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 | 3 | 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 | 3 | 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 | 3 | 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 | 3 | 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 | 3 | LBNQ | C9 | IF Q = 0, M(R(P))→R(P).1 M(R(P)+1)→R(P).0 ELSE R(P)+2→R(P) |

▲This instruction is associated with more than one mnemonic. Each mnemonic is individually listed.

*ETQ cleared by LDC with the Counter/Timer stopped, reset of CPU, or BCI • (CI=1).
CI = Counter Interrupt, XI = External Interrupt.





2590 C-09

# CDP1804AC

Table I — INSTRUCTION SUMMARY (Cont'd)

| INSTRUCTION | NO. OF MACHINE CYCLES | MNEMONIC | OP CODE | OPERATION |
|---|---|---|---|---|
| SKIP INSTRUCTIONS | | | | |
| SHORT SKIP (SEE NBR) | 2 | SKP | 38▲ | R(P)+1→R(P) |
| LONG SKIP (SEE NLBR) | 3 | LSKP | C8▲ | R(P)+→R(P) |
| LONG SKIP IF D = 0 | 3 | LSZ | CE | IF D = 0, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF D NOT 0 | 3 | LSNZ | C6 | IF D NOT 0, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF DF = 1 | 3 | LSDF | CF | IF DF = 1, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF DF = 0 | 3 | LSNF | C7 | IF DF = 0, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF Q = 1 | 3 | LSQ | CD | IF Q = 1, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF Q = 0 | 3 | LSNQ | C5 | IF Q = 0, R(P)+2→R(P) ELSE CONTINUE |
| LONG SKIP IF MIE = 1 | 3 | LSIE | CC | IF MIE = 1, R(P)+2→R(P) ELSE CONTINUE |
| CONTROL INSTRUCTIONS | | | | |
| IDLE | 2 | IDL | 00# | STOP ON TPB; WAIT FOR DMA OR INTERRUPT; · BUS FLOATS |
| NO OPERATION | 3 | NOP | C4 | CONTINUE |
| SET P | 2 | SEP | DN | N→P |
| SET X | 2 | SEX | EN | N→X |
| SET Q | 2 | SEQ | 7B | 1→Q |
| RESET Q | 2 | REQ | 7A | 0→Q |
| PUSH X, P TO STACK | 2 | MARK | 79 | (X, P)→T; (X, P)→M(R(2)) THEN P→X; R(2)−1→R(2) |
| TIMER/COUNTER INSTRUCTIONS | | | | |
| LOAD COUNTER | 3 | LDC | 6806• | CNTR STOPPED; D→CH, CNTR; 0=CI.CNTR RUNNING; D→CH |
| GET COUNTER | 3 | GEC | 6808 | CNTR→D |
| STOP COUNTER | 3 | STPC | 6800 | STOP CNTR CLOCK; 0→÷32 PRESCALER |
| DECREMENT TIMER/COUNTER | 3 | DTC | 6801 | CNTR-1→CNTR |
| SET TIMER MODE AND START | 3 | STM | 6807 | TPA÷32→ CNTR |
| SET COUNTER MODE 1 AND START | 3 | SCM1 | 6805 | $\overline{EF1}$→ CNTR CLOCK |
| SET COUNTER MODE 2 AND START | 3 | SCM2 | 6803 | EF2→ CNTR CLOCK |
| SET PULSE WIDTH MODE 1 AND START | 3 | SPM1 | 6804 | TPA.$\overline{EF1}$→ CNTR CLOCK; $\overline{EF1}$ ↓ STOPS COUNT |
| SET PULSE WIDTH MODE 2 AND START | 3 | SPM2 | 6802 | TPA.EF2→ CNTR CLOCK; $\overline{EF2}$ ↓ STOPS COUNT |
| ENABLE TOGGLE Q | 3 | ETQ | 6809• | IF CNTR = 01 • NEXT CNTR CLOCK ↓ : $\overline{Q}$→Q |

▲This instruction is associated with more than one mnemonic. Each mnemonic is individually listed.

#An IDLE instruction initiates an S1 cycle. All external signals, except the oscillator, are stopped on the low-to-high transition of TPB. All outputs remain in their previous states, $\overline{MRD}$, $\overline{MWR}$, $\overline{EMS}$ are set to a logic '1' and the data bus floats. The processor will continue to IDLE until an I/O request (INTERRUPT, DMA-IN, or DMA-OUT) is activated. When the request is acknowledged, the IDLE cycle is terminated and the I/O request is serviced, and the normal operation is resumed. (To respond to an INTERRUPT during an IDLE, MIE and either CIE or XIE must be enabled).

• ETQ cleared by LDC with the Counter/Timer stopped, reset of CPU or BCI · (CI = 1).

CI = Counter Interrupt, XI = External Interrupt.

T-49-19-08

Table I — INSTRUCTION SUMMARY (Cont'd)

| INSTRUCTION | NO. OF MACHINE CYCLES | MNEMONIC | OP CODE | OPERATION |
|---|---|---|---|---|
| INTERRUPT CONTROL | | | | |
| EXTERNAL INTERRUPT ENABLE | 3 | XIE | 680A | 1→XIE |
| EXTERNAL INTERRUPT DISABLE | 3 | XID | 680B | 0→XIE |
| COUNTER INTERRUPT ENABLE | 3 | CIE | 680C | 1→CIE |
| COUNTER INTERRUPT DISABLE | 3 | CID | 680D | 0→CIE |
| RETURN | 2 | RET | 70 | M(R(X))→X, P; R(X)+1→R(X); 1→MIE |
| DISABLE | 2 | DIS | 71 | M(R(X)→X, P; R(X)+1→R(X); 0→MIE |
| SAVE | 2 | SAV | 78 | T→M(R(X)) |
| SAVE T, D, DF | 6 | DSAV | 6876■ | R(X)-1→R(X), T→M(R(X)), R(X)-1→R(X), D→M (R(X)), R(X)-1→R(X), SHIFT D RIGHT WITH CARRY, D→M(R(X)) |
| INPUT-OUTPUT BYTE TRANSFER | | | | |
| OUTPUT 1 | 2 | OUT 1 | 61 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 1 |
| OUTPUT 2 | 2 | OUT 2 | 62 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 2 |
| OUTPUT 3 | 2 | OUT 3 | 63 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 3 |
| OUTPUT 4 | 2 | OUT 4 | 64 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 4 |
| OUTPUT 5 | 2 | OUT 5 | 65 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 5 |
| OUTPUT 6 | 2 | OUT 6 | 66 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 6 |
| OUTPUT 7 | 2 | OUT 7 | 67 | M(R(X))→BUS; R(X)+1→R(X); N LINES = 7 |
| INPUT 1 | 2 | INP 1 | 69 | BUS→M(R(X)); BUS→D; N LINES = 1 |
| INPUT 2 | 2 | INP 2 | 6A | BUS→M(R(X)); BUS→D; N LINES = 2 |
| INPUT 3 | 2 | INP 3 | 6B | BUS→M(R(X)); BUS→D; N LINES = 3 |
| INPUT 4 | 2 | INP 4 | 6C | BUS→M(R(X)); BUS→D; N LINES = 4 |
| INPUT 5 | 2 | INP 5 | 6D | BUS→M(R(X)); BUS→D; N LINES = 5 |
| INPUT 6 | 2 | INP 6 | 6E | BUS→M(R(X)); BUS→D; N LINES = 6 |
| INPUT 7 | 2 | INP 7 | 6F | BUS→M(R(X)); BUS→D; N LINES = 7 |
| CALL AND RETURN | | | | |
| STANDARD CALL | 10 | 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 | 8 | 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) |

■Previous contents of T register are destroyed during instruction execution.

**NOTES FOR TABLE I**

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 counter or external interrupts (BCI, BXI)

   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 MIE=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. Instruction 6800 through 68FF take a minimum of 3 machine cycles and up to a maximum of 10 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.

5. Arithmetic Operations:
   The arithmetic and shift operations are the only instructions that can alter the content of DF. The syntax '(NOT DF)' denotes the subtraction of the borrow.
   Binary Operations:
   After an ADD instruction —
   DF=1 denotes a carry has occurred. Result is greater than $FF_{16}$.
   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 in two's complement form.

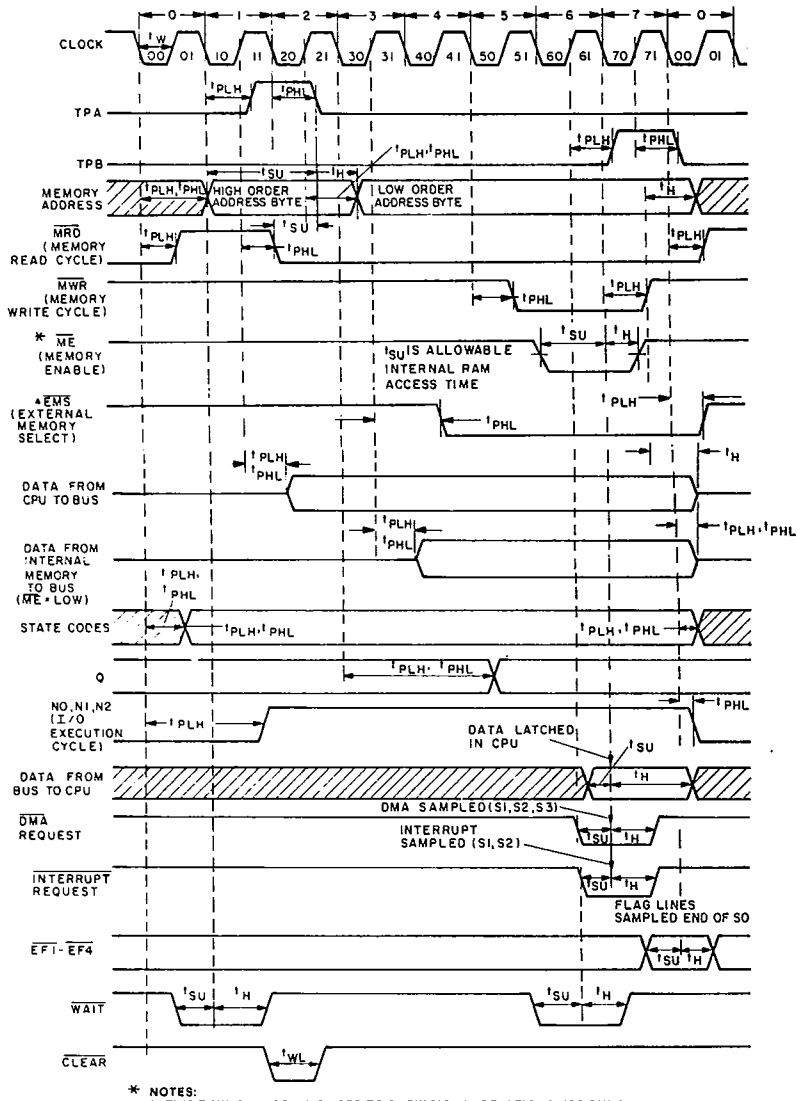
   Binary Coded Decimal Operations:
   After a BCD ADD instruction —
   DF=1 denotes a carry has occurred. Result is greater than $99_{10}$.
   DF=0 denotes a carry has not occurred.
   After a BCD SUBTRACT instruction —
   DF=1 denotes no borrow. D is a true positive decimal number.

   (Example)
   ```
    99      D
   -88      M(R(X))
   ---
    11      D      DF=1
   ```

   DF=0 denotes a borrow. D is in ten's complement form.

   (Example)
   ```
    88      D
   -99      M(R(X))
   ---
    89      D      DF=0
   ```

   89 is the ten's complement of 11, which is the correct answer (with a minus value denoted by DF=0).




* 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 ARE AS INDICATED "DON'T CARE" OR UNDEFINED STATE. MULTIPLE TRANSITIONS MAY OCCUR DURING THIS PERIOD.

92CL-34986RI

*FOR THE RUN (RAM ONLY) MODE ONLY.
▲ FOR THE RUN (RAM/ROM) MODE ONLY.

*Fig. 14 - Dynamic timing waveforms for CDP1804AC.*

# CDP1804AC

**DYNAMIC ELECTRICAL CHARACTERISTICS at $T_A$ = -40 to +85°C; $C_L$ = 50 pF; Input $t_r$, $t_f$ = 10ns; Input Pulse Levels = 0.1 V to $V_{DD}$ -0.1 V; $V_{DD}$ = 5 V, ±5%.**

| CHARACTERISTIC | | LIMITS CDP1804AC Typ.* | LIMITS CDP1804AC Max. | UNITS |
|---|---|---|---|---|
| Propagation Delay Times: | | | | |
| Clock to TPA, TPB | $t_{PLH}$, $t_{PHL}$ | 150 | 275 | ns |
| Clock-to-Memory High-Address Byte | $t_{PLH}$, $t_{PHL}$ | 325 | 550 | |
| Clock-to-Memory Low-Address Byte | $t_{PLH}$, $t_{PHL}$ | 275 | 450 | |
| Clock to $\overline{MRD}$ | $t_{PLH}$, $t_{PHL}$ | 200 | 325 | |
| Clock to $\overline{MWR}$ | $t_{PLH}$, $t_{PHL}$ | 150 | 275 | |
| Clock to (CPU DATA to BUS) | $t_{PLH}$, $t_{PHL}$ | 375 | 625 | |
| Clock to State Code | $t_{PLH}$, $t_{PHL}$ | 225 | 400 | |
| Clock to Q | $t_{PLH}$, $t_{PHL}$ | 250 | 425 | |
| Clock to N | $t_{PLH}$, $t_{PHL}$ | 250 | 425 | |
| Clock to Internal RAM Data to BUS | $t_{PLH}$, $t_{PHL}$ | 420 | 650 | |
| Clock to $\overline{EMS}$ | $t_{PLH}$, $t_{PHL}$ | 275 | 450 | |
| Minimum Set Up and Hold Times:■ | | | | |
| Data Bus Input Set-Up | $t_{SU}$ | -100 | 0 | ns |
| Data Bus Input Hold | $t_H$ | 125 | 225 | |
| $\overline{DMA}$ Set-Up | $t_{SU}$ | -75 | 0 | |
| $\overline{DMA}$ Hold | $t_H$ | 100 | 175 | |
| $\overline{ME}$ Set-Up | $t_{SU}$ | 125 | 320 | |
| $\overline{ME}$ Hold | $t_H$ | 0 | 50 | |
| Interrupt Set-Up | $t_{SU}$ | -100 | 0 | |
| Interrupt Hold | $t_H$ | 100 | 175 | |
| $\overline{WAIT}$ Set-Up | $t_{SU}$ | 20 | 50 | |
| $\overline{EF1\text{-}4}$ Set-Up | $t_{SU}$ | -125 | 0 | |
| $\overline{EF1\text{-}4}$ Hold | $t_H$ | 175 | 300 | |
| Minimum Pulse Width Times:■ | | | | |
| $\overline{CLEAR}$ Pulse Width | $t_{WL}$ | 100 | 175 | ns |
| $\overline{CLOCK}$ Pulse Width | $t_{WL}$ | 75 | 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.

**TIMING SPECIFICATIONS as a function of T ($T = 1/f_{CLOCK}$) at $T_A$ = -40 to +85°C, $V_{DD}$ = 5 V, ±5%.**

| CHARACTERISTIC | | LIMITS CDP1804AC Min. | LIMITS CDP1804AC Typ.• | UNITS |
|---|---|---|---|---|
| High-Order Memory-Address Byte Set-Up to TPA ↘ Time | $t_{SU}$ | 2T-275 | 2T-175 | ns |
| $\overline{MRD}$ to TPA ↘ Time | $t_{SU}$ | T/2-100 | T/2-75 | |
| High-Order Memory-Address Byte Hold After TPA Time | $t_H$ | T/2+75 | T/2+100 | |
| Low-Order Memory-Address Byte Hold After WR Time | $t_H$ | T+180 | T+240 | |
| CPU Data to Bus Hold After WR Time | $t_H$ | T+110 | T+150 | |
| Required Memory Access Time Address to Data | $t_{ACC}$ | 4.5T-440 | 4.5T-330 | |

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

T-49-19-08

**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 |
|---|---|---|---|---|---|---|---|---|---|
| S1 | RESET | | | 0→Q,I,N, COUNTER PRESCALER, CIL; 1→CIE, XIE | 00 | UNDEFINED | 1 | 1 | 0 |
| | INITIALIZE NOT PROGRAMMER ACCESSIBLE | | | X, P→T THEN 0→X, P; 1→MIE, 0000→R0 | 00▲ | UNDEFINED | 1 | 1 | 0 |
| S0 | | FETCH | | MRP→I, N; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| S1 | 0 | 0 | IDL | STOP AT TPB WAIT FOR DMA OR INT | HIGH Z | R0 | 1 | 1 | 0 |
| | 0 | 1-F | LDN | MRN→D | MRN | RN | 0 | 1 | 0 |
| | 1 | 0-F | INC | RN+1→RN | HIGH Z | RN | 1 | 1 | 0 |
| | 2 | 0-F | DEC | RN-1→RN | HIGH Z | RN | 1 | 1 | 0 |
| | 3 | 0-F | SHORT BRANCH | TAKEN: MRP→RP.0<br>NOT TAKEN: RP+1→RP | MRP | RP | 0 | 1 | 0 |
| | 4 | 0-F | LDA | MRN→D; RN+1→RN | MRN | RN | 0 | 1 | 0 |
| | 5 | 0-F | STR | D→MRN | D | RN | 1 | 0 | 0 |
| | 6 | 0 | IRX | RX+1→RX | MRX | RX | 1 | 1 | 0 |
| | 6 | 1 | OUT 1 | MRX→BUS; RX+1→RX | MRX | RX | 0 | 1 | 1 |
| | | 2 | OUT 2 | | | | | | 2 |
| | | 3 | OUT 3 | | | | | | 3 |
| | | 4 | OUT 4 | | | | | | 4 |
| | | 5 | OUT 5 | | | | | | 5 |
| | | 6 | OUT 6 | | | | | | 6 |
| | | 7 | OUT 7 | | | | | | 7 |
| | 6 | 9 | INP 1 | BUS→MRX, D | DATA FROM I/O DEVICE | RX | 1 | 0 | 1 |
| | | 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→MIE | MRX | RX | 0 | 1 | 0 |
| | | 1 | DIS | MRX→X,P; RX+1→RX 0→MIE | MRX | RX | 0 | 1 | 0 |
| | | 2 | LDXA | MRX→D; RX+1→RX | MRX | RX | 0 | 1 | 0 |
| | | 3 | STXD | D→MRX; RX-1→RX | D | RX | 1 | 0 | 0 |
| | | 4 | ADC | MRX+D+DF→DF, D | MRX | RX | 0 | 1 | 0 |
| | | 5 | SDB | MRX- D-DFN→DF, D | MRX | RX | 0 | 1 | 0 |
| | | 6 | SHRC | LSB(D)→DF; DF→MSB(D) | HIGH Z | RX | 1 | 1 | 0 |
| | | 7 | SMB | D- MRX- DFN→DF, D | MRX | RX | 0 | 1 | 0 |
| | | 8 | SAV | T→MRX | T | RX | 1 | 0 | 0 |
| | | 9 | MARK | X,P→T, MR2; P→X R2-1→R2 | T | R2 | 1 | 0 | 0 |
| | | A | REQ | 0→Q | HIGH Z | RP | 1 | 1 | 0 |
| | | B | SEQ | 1→Q | HIGH Z | RP | 1 | 1 | 0 |
| | | C | ADCI | MRP+D+DF→DF, D; RP+1 | MRP | RP | 0 | 1 | 0 |
| | | D | SDBI | MRP-D-DFN→DF, D; RP+1 | MRP | RP | 0 | 1 | 0 |
| | | E | SHLC | MSB(D)→DF; DF→LSB(D) | HIGH Z | RP | 1 | 1 | 0 |
| | | F | SMBI | D-MRP-DFN→DF, D; RP+1 | MRP | RP | 0 | 1 | 0 |
| | 8 | 0-F | GLO | RN.0→D | RN.0 | RN | 1 | 1 | 0 |
| | 9 | 0-F | GHI | RN.1→D | RN.1 | RN | 1 | 1 | 0 |
| | A | 0-F | PLO | D→RN.0 | D | RN | 1 | 1 | 0 |
| | B | 0-F | PHI | D→RN.1 | D | RN | 1 | 1 | 0 |

▲ = Data bus floats for first 2-1/2 clocks of the 9 clock initialization cycle; all zeros for remainder of cycle.



2596 D-01

**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 |
|---|---|---|---|---|---|---|---|---|---|
| S1#1 | C | 0-3, 8-B | LONG BRANCH | TAKEN: MRP→B; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | TAKEN:B→RP.1;MRP→RP.0 | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1#1 | | | | NOT TAKEN RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | NOT TAKEN: RP+1→RP | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1#1 | | 5 | LONG SKIP | TAKEN: RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | 6 | | TAKEN: RP+1→RP | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1#1 | | 7<br>C<br>D | | NOT TAKEN: NO OPERATION | MRP | RP | 0 | 1 | 0 |
| #2 | | E<br>F | | NOT TAKEN: NO OPERATION | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1#1 | | 4 | NOP | NO OPERATION | MRP | RP | 0 | 1 | 0 |
| #2 | | | | NO OPERATION | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1 | D | 0-F | SEP | N→P | NN | RN | 1 | 1 | 0 |
| | E | 0-F | SEX | N→X | NN | RN | 1 | 1 | 0 |
| | F | 0 | LDX | MRX→D | MRX | RX | 0 | 1 | 0 |
| | | 1 | OR | MRX OR D→D | MRX | RX | 0 | 1 | 0 |
| | | 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) | HIGH Z | RX | 1 | 1 | 0 |
| | | 8 | LDI | MRP→D; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| | | 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) | HIGH Z | RP | 1 | 1 | 0 |
| S2 | DMA IN | | | BUS→MR0; R0+1→R0 | DATA FROM I/O DEVICE | R0 | 1 | 0 | 0 |
| | DMA OUT | | | MR0→BUS; R0+1→R0 | MR0 | R0 | 0 | 1 | 0 |
| S3 | INTERRUPT | | | X,P→T; 0→MIE 1→P; 2→X | HIGH Z | RN | 1 | 1 | 0 |

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 |
|---|---|---|---|---|---|---|---|---|---|
| THE FOLLOWING ARE ALL LINKED INSTRUCTIONS "68" PRECEDES ALL THE OP CODES, SO THERE IS A DOUBLE FETCH | | | | | | | | | |
| S1 | 0 | 0 | STPC | STOP COUNTER CLOCK; 0→÷32 PRESCALER | HIGH Z | R0 | 1 | 1 | 0 |
| | | 1 | DTC | CNTR-1→CNTR | HIGH Z | R1 | 1 | 1 | 0 |
| | | 2 | SPM2 | CNTR-1 ON EF2 AND TPA | HIGH Z | R2 | 1 | 1 | 0 |
| | | 3 | SCM2 | CNTR-1 ON EF2 0 TO 1 | HIGH Z | R3 | 1 | 1 | 0 |
| | | 4 | SPM1 | CNTR-1 ON EF1 AND TPA | HIGH Z | R4 | 1 | 1 | 0 |
| | | 5 | SCM1 | CNTR-1 ON EF1 0 TO 1 | HIGH Z | R5 | 1 | 1 | 0 |
| | | 6 | LDC | CNTR STOPPED: D→CH, CNTR; 0→CI<br>CNTR RUNNING: D→CH | D | R6 | 1 | 1 | 0 |
| | | 7 | STM | CNTR-1 ON TPA÷32 | HIGH Z | R7 | 1 | 1 | 0 |
| | | 8 | GEC | CNTR→D | CNTR | R8 | 1 | 1 | 0 |
| | | 9 | ETQ | IF CNTR THRU 0: $\overline{Q}$→Q | HIGH Z | R9 | 1 | 1 | 0 |
| | | A | XIE | 1→XIE | HIGH Z | RA | 1 | 1 | 0 |
| | | B | XID | 0→XIE | HIGH Z | RB | 1 | 1 | 0 |
| | | C | CIE | 1→CIE | HIGH Z | RC | 1 | 1 | 0 |
| | | D | CID | 0→CIE | HIGH Z | RD | 1 | 1 | 0 |
| S1#1 | 2 | 0-F | DBNZ | RN-1→RN | HIGH Z | RN | 1 | 1 | 0 |
| #2 | | | | MRP→B; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #3 | | | | TAKEN: B→RP.1, MRP→RP.0<br>NOT TAKEN: RP+1→RP | M(RP+1) | RP+1 | 0 | 1 | 0 |
| S1 | 3 | E | BCI | TAKEN: MRP→RP.0; 0→CI<br>NOT TAKEN: RP+1→RP | MRP | RP | 0 | 1 | 0 |
| | | F | BXI | TAKEN: MRP→RP.0<br>NOT TAKEN: RP+1→RP | MRP | RP | 0 | 1 | 0 |
| S1#1 | 6 | 0-F | RLXA | MRX→B, RX+1→RX | MRX | RX | 0 | 1 | 0 |
| #2 | | | | B→T; MRX→B; RX+1→RX | M(RX+1) | RX+1 | 0 | 1 | 0 |
| #3 | | | | B, T→RN.0, RN.1 | HIGH Z | RN | 1 | 1 | 0 |
| S1#1 | 7 | 4 | DADC | MRX+D+DF→DF, D | MRX | RX | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP | 1 | 1 | 1 |
| S1#1 | 7 | 6 | DSAV | RX-1→RX | HIGH Z | RX | 1 | 1 | 0 |
| #2 | | | | T→MRX; RX-1→RX | T | RX-1 | 1 | 0 | 0 |
| #3 | | | | D→MRX; RX-1→RX<br>SHIFT D RIGHT WITH CARRY | D | RX-2 | 1 | 0 | 0 |
| #4 | | | | D→MRX | D | RX-3 | 1 | 0 | 0 |
| S1#1 | 7 | 7 | DSMB | D-MRX-(NOT DF)→DF, D | MRX | RX | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP | 1 | 1 | 0 |
| S1#1 | 7 | C | DACI | MRP+D+DF→DF, D; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP+1 | 1 | 1 | 0 |
| S1#1 | 7 | F | DSBI | D-MRP-(NOT DF)→DF, D; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP+1 | 1 | 1 | 0 |
| S1#1 | 8 | 0-F | SCAL | RN.0, RN.1→T, B | HIGH Z | RN | 1 | 1 | 0 |
| #2 | | | | T→MRX; RX-1→RX | RN.0 | RX | 1 | 0 | 0 |
| #3 | | | | B→MRX, RX-1→RX | RN.1 | RX-1 | 1 | 0 | 0 |
| #4 | | | | RP.0, RP.1→T, B | HIGH Z | RP | 1 | 1 | 0 |
| #5 | | | | B, T→RN.1, RN.0 | HIGH Z | RN | 1 | 1 | 0 |
| #6 | | | | MRN→B; RN+1→RN | MRP | RP | 0 | 1 | 0 |
| #7 | | | | B→T; MRN→B; RN+1→RN | M(RP+1) | RP+1 | 0 | 1 | 0 |
| #8 | | | | B, T→RP.0, RP.1 | HIGH Z | RP | 1 | 1 | 0 |

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 |
|---|---|---|---|---|---|---|---|---|---|
| THE FOLLOWING ARE ALL LINKED INSTRUCTIONS "68" PRECEDES ALL THE OP CODES, SO THERE IS A DOUBLE FETCH | | | | | | | | | |
| S1#1 | 9 | 0-F | SRET | RN.0, RN.1→T, B | HIGH Z | RN | 1 | 1 | 0 |
| #2 | | | | RX+1→RX | HIGH Z | RX | 1 | 1 | 0 |
| #3 | | | | B, T→RP.1, RP.0 | HIGH Z | RP | 1 | 1 | 0 |
| #4 | | | | MRX→B; RX+1→RX | M(RX+1) | RX+1 | 0 | 1 | 0 |
| #5 | | | | B→T; MRX→B | M(RX+1) | RX+2 | 0 | 1 | 0 |
| #6 | | | | B, T→RN.0, RN.1 | HIGH Z | RN | 1 | 1 | 0 |
| S1#1 | A | 0-F | RSXD | RN.0, RN.1→T, B | HIGH Z | RN | 1 | 1 | 0 |
| #2 | | | | T→MRX; RX-1→RX | RN.0 | RX | 1 | 0 | 0 |
| #3 | | | | B→MRX; RX-1→RX | RN.1 | RX-1 | 1 | 0 | 0 |
| S1#1 | B | 0-F | RNX | RN.0, RN.1→T, B | HIGH Z | RN | 1 | 1 | 0 |
| #2 | | | | B, T→RX.1, RX.0 | HIGH Z | RX | 1 | 1 | 0 |
| S1#1 | C | 0-F | RLDI | MRP→B; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | B→T; MRP→B; RP+1→RP | M(RP+1) | RP+1 | 0 | 1 | 0 |
| #3 | | | | B, T→RN.0, RN.1; RP+1→RP | HIGH Z | RN | 1 | 1 | 0 |
| S1#1 | F | 4 | DADD | MRX+D→DF, D | MRX | RX | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP | 1 | 1 | 0 |
| S1#1 | F | 7 | DSM | D-MRX→DF, D | MRX | RX | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP | 1 | 1 | 0 |
| S1#1 | F | C | DADI | MRP+D→DF, D; RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP+1 | 1 | 1 | 0 |
| S1#1 | F | F | DSMI | D-MRP→DF, D RP+1→RP | MRP | RP | 0 | 1 | 0 |
| #2 | | | | DECIMAL ADJUST→DF, D | HIGH Z | RP+1 | 1 | 1 | 0 |

# CDP1804AC

T-49-19-08

## Instruction Summary

| | N | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0 | IDL | LDN | | | | | | | | | | | | | | |
| 1 | INC | | | | | | | | | | | | | | | |
| 2 | DEC | | | | | | | | | | | | | | | |
| 3 | BR | BQ | BZ | BDF | B1 | B2 | B3 | B4 | SKP | BNQ | BNZ | BNF | BN1 | BN2 | BN3 | BN4 |
| 4 | LDA | | | | | | | | | | | | | | | |
| 5 | STR | | | | | | | | | | | | | | | |
| 6 | IRX | OUT | | | | | | | * | INP | | | | | | |
| 7 | RET | DIS | LDXA | STXD | ADC | SDB | SHRC | SMB | SAV | MARK | REQ | SEQ | ADCI | SDBI | SHLC | SMBI |
| 8 | GLO | | | | | | | | | | | | | | | |
| 9 | GHI | | | | | | | | | | | | | | | |
| A | PLO | | | | | | | | | | | | | | | |
| B | PHI | | | | | | | | | | | | | | | |
| C | LBR | LBQ | LBZ | LBDF | NOP | LSNQ | LSNZ | LSNF | LSKP | LBNQ | LBNZ | LBNF | LSIE | LSQ | LSZ | LSDF |
| D | SEP | | | | | | | | | | | | | | | |
| E | SEX | | | | | | | | | | | | | | | |
| F | LDX | OR | AND | XOR | ADD | SD | SHR | SM | LDI | ORI | ANI | XRI | ADI | SDI | SHL | SMI |
| | '68' LINKED OPCODES (DOUBLE FETCH) | | | | | | | | | | | | | | | |
| 0 | STPC | DTC | SPM2 | SCM2 | SPM1 | SCM1 | LDC | STM | GEC | ETQ | XIE | XID | CIE | CID | — | — |
| 2 | DBNZ | | | | | | | | | | | | | | | |
| 3 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | BCI | BXI |
| 6 | RLXA | | | | | | | | | | | | | | | |
| 7 | — | — | — | — | DADC | — | DSAV | DSMB | — | — | — | — | DACI | — | — | DSBI |
| 8 | SCAL | | | | | | | | | | | | | | | |
| 9 | SRET | | | | | | | | | | | | | | | |
| A | RSXD | | | | | | | | | | | | | | | |
| B | RNX | | | | | | | | | | | | | | | |
| C | RLDI | | | | | | | | | | | | | | | |
| F | — | — | — | — | DADD | — | — | DSM | — | — | — | — | DADI | — | — | DSMI |

* '68' IS USED AS A LINKING OPCODE FOR THE DOUBLE FETCH INSTRUCTIONS.

# CDP1804AC

### CDP1804AC Mask-Programming

The ROM pattern for the CDP1804AC may be submitted on a suitable media, such as floppy diskette, ROM or EPROM.

In addition to specifying the 2K-byte ROM pattern, the address space for the ROM and RAM must also be defined. The locations of ROM and RAM in the CDP1804AC are determined by AND-gate decoders which decode the upper memory addresses and are programmable at the time of ROM pattern masking during device fabrication. The logical values of the decoder inputs are selectable as 1 or P (positive), 0 or N (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 of the 32 available 2K-byte blocks within the 65,536 locations of memory. Similarly, the RAM has a 10-bit decoder and can be selected at one or more of the available 64-byte blocks. If the RAM is located within the ROM space, only the RAM will be enabled at the locations where both are mapped. The RAM may also be selectively disabled.

### Programming Options

#### Address Options

The logic levels of high-order address bits are mask programmable in the CDP1804AC. The high (1), low (0), or "don't care" (X) logic status of the high-order address bits is dependent upon the desired starting address of the 2K-byte ROM block and the 64-byte RAM block. The desired logic levels for the high-order address bits (A15 through A6) can be selected by use of the ROM order sheet.

Multiple mapping can be achieved by choosing X (don't care) for one or more of the high-order address lines; this choice will cause the ROM or RAM block to appear in more than one location in the 64K memory space. The RAM may also be disabled completely in the RUN (ROM/RAM) mode by programming the RAM enable bit to an N.