# Silicon Gate MOS 8080A

## SINGLE CHIP 8-BIT N-CHANNEL MICROPROCESSOR

*The 8080A is functionally and electrically compatible with the Intel® 8080.*

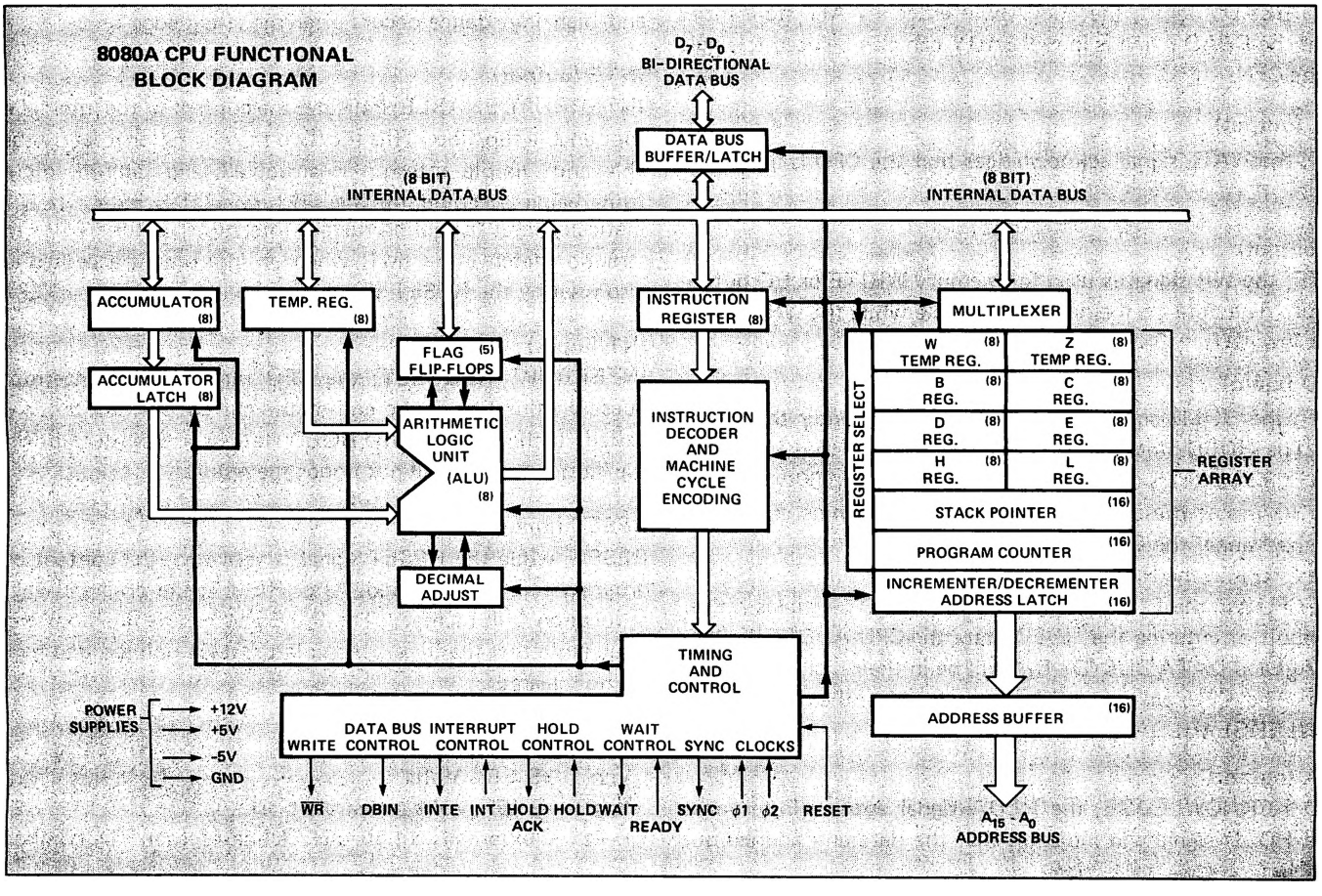
- **TTL Drive Capability**
- **2 μs Instruction Cycle**
- **Powerful Problem Solving Instruction Set**
- **Six General Purpose Registers and an Accumulator**
- **Sixteen Bit Program Counter for Directly Addressing up to 64K Bytes of Memory**
- **Sixteen Bit Stack Pointer and Stack Manipulation Instructions for Rapid Switching of the Program Environment**
- **Decimal, Binary and Double Precision Arithmetic**
- **Ability to Provide Priority Vectored Interrupts**
- **512 Directly Addressed I/O Ports**

The Intel® 8080A is a complete 8-bit parallel central processing unit (CPU). It is fabricated on a single LSI chip using Intel's n-channel silicon gate MOS process. This offers the user a high performance solution to control and processing applications.

The 8080A contains six 8-bit general purpose working registers and an accumulator. The six general purpose registers may be addressed individually or in pairs providing both single and double precision operators. Arithmetic and logical instructions set or reset four testable flags. A fifth flag provides decimal arithmetic operation.

The 8080A has an external stack feature wherein any portion of memory may be used as a last in/first out stack to store/retrieve the contents of the accumulator, flags, program counter and all of the six general purpose registers. The sixteen bit stack pointer controls the addressing of this external stack. This stack gives the 8080A the ability to easily handle multiple level priority interrupts by rapidly storing and restoring processor status. It also provides almost unlimited subroutine nesting.

This microprocessor has been designed to simplify systems design. Separate 16-line address and 8-line bi-directional data busses are used to facilitate easy interface to memory and I/O. Signals to control the interface to memory and I/O are provided directly by the 8080A. Ultimate control of the address and data busses resides with the HOLD signal. It provides the ability to suspend processor operation and force the address and data busses into a high impedance state. This permits OR-tying these busses with other controlling devices for (DMA) direct memory access or multi-processor operation.



## 8080A FUNCTIONAL PIN DEFINITION

The following describes the function of all of the 8080A I/O pins. Several of the descriptions refer to internal timing periods.

**$A_{15}$-$A_0$ (output three-state)**
ADDRESS BUS; the address bus provides the address to memory (up to 64K 8-bit words) or denotes the I/O device number for up to 256 input and 256 output devices. $A_0$ is the least significant address bit.

**$D_7$-$D_0$ (input/output three-state)**
DATA BUS; the data bus provides bi-directional communication between the CPU, memory, and I/O devices for instructions and data transfers. Also, during the first clock cycle of each machine cycle, the 8080A outputs a status word on the data bus that describes the current machine cycle. $D_0$ is the least significant bit.

**SYNC (output)**
SYNCHRONIZING SIGNAL; the SYNC pin provides a signal to indicate the beginning of each machine cycle.

**DBIN (output)**
DATA BUS IN; the DBIN signal indicates to external circuits that the data bus is in the input mode. This signal should be used to enable the gating of data onto the 8080A data bus from memory or I/O.

**READY (input)**
READY; the READY signal indicates to the 8080A that valid memory or input data is available on the 8080A data bus. This signal is used to synchronize the CPU with slower memory or I/O devices. If after sending an address out the 8080A does not receive a READY input, the 8080A will enter a WAIT state for as long as the READY line is low. READY can also be used to single step the CPU.

**WAIT (output)**
WAIT; the WAIT signal acknowledges that the CPU is in a WAIT state.

**$\overline{WR}$ (output)**
WRITE; the $\overline{WR}$ signal is used for memory WRITE or I/O output control. The data on the data bus is stable while the $\overline{WR}$ signal is active low ($\overline{WR}$ = 0).

**HOLD (input)**
HOLD; the HOLD signal requests the CPU to enter the HOLD state. The HOLD state allows an external device to gain control of the 8080A address and data bus as soon as the 8080A has completed its use of these buses for the current machine cycle. It is recognized under the following conditions:

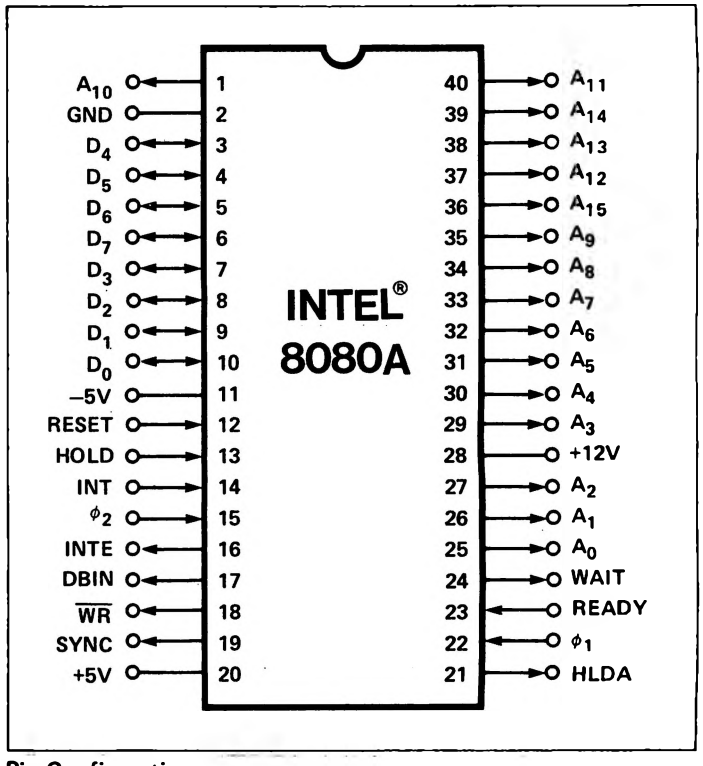
- the CPU is in the HALT state.
- the CPU is in the T2 or TW state and the READY signal is active.

As a result of entering the HOLD state the CPU ADDRESS BUS ($A_{15}$-$A_0$) and DATA BUS ($D_7$-$D_0$) will be in their high impedance state. The CPU acknowledges its state with the HOLD ACKNOWLEDGE (HLDA) pin.

**HLDA (output)**
HOLD ACKNOWLEDGE; the HLDA signal appears in response to the HOLD signal and indicates that the data and address bus will go to the high impedance state. The HLDA signal begins at:

- T3 for READ memory or input.
- The Clock Period following T3 for WRITE memory or OUTPUT operation.

In either case, the HLDA signal appears after the rising edge of $\phi_1$ and high impedance occurs after the rising edge of $\phi_2$.



Pin Configuration

**INTE (output)**
INTERRUPT ENABLE; indicates the content of the internal interrupt enable flip/flop. This flip/flop may be set or reset by the Enable and Disable Interrupt instructions and inhibits interrupts from being accepted by the CPU when it is reset. It is automatically reset (disabling further interrupts) at time T1 of the instruction fetch cycle (M1) when an interrupt is accepted and is also reset by the RESET signal.

**INT (input)**
INTERRUPT REQUEST; the CPU recognizes an interrupt request on this line at the end of the current instruction or while halted. If the CPU is in the HOLD state or if the Interrupt Enable flip/flop is reset it will not honor the request.

**RESET (input)[1]**
RESET; while the RESET signal is activated, the content of the program counter is cleared. After RESET, the program will start at location 0 in memory. The INTE and HLDA flip/flops are also reset. Note that the flags, accumulator, stack pointer, and registers are not cleared.

| | |
|---|---|
| $V_{SS}$ | Ground Reference. |
| $V_{DD}$ | +12 ± 5% Volts. |
| $V_{CC}$ | +5 ± 5% Volts. |
| $V_{BB}$ | -5 ±5% Volts (substrate bias). |
| $\phi_1$, $\phi_2$ | 2 externally supplied clock phases. (non TTL compatible) |

## ABSOLUTE MAXIMUM RATINGS*

| | |
|---|---|
| Temperature Under Bias | 0°C to +70° C |
| Storage Temperature | -65°C to +150°C |
| All Input or Output Voltages With Respect to $V_{BB}$ | -0.3V to +20V |
| $V_{CC}$, $V_{DD}$ and $V_{SS}$ With Respect to $V_{BB}$ | -0.3V to +20V |
| Power Dissipation | 1.5W |

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## D.C. CHARACTERISTICS

$T_A$ = 0°C to 70°C, $V_{DD}$ = +12V ± 5%, $V_{CC}$ = +5V ± 5%, $V_{BB}$ = -5V ± 5%, $V_{SS}$ = 0V, Unless Otherwise Noted.

| Symbol | Parameter | Min. | Typ. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|---|
| $V_{ILC}$ | Clock Input Low Voltage | $V_{SS}$-1 | | $V_{SS}$+0.8 | V | |
| $V_{IHC}$ | Clock Input High Voltage | 9.0 | | $V_{DD}$+1 | V | |
| $V_{IL}$ | Input Low Voltage | $V_{SS}$-1 | | $V_{SS}$+0.8 | V | |
| $V_{IH}$ | Input High Voltage | 3.3 | | $V_{CC}$+1 | V | |
| $V_{OL}$ | Output Low Voltage | | | 0.45 | V | $I_{OL}$ = 1.9mA on all outputs, |
| $V_{OH}$ | Output High Voltage | 3.7 | | | V | $I_{OH}$ = -150μA. |
| $I_{DD\,(AV)}$ | Avg. Power Supply Current ($V_{DD}$) | | 40 | 70 | mA | Operation |
| $I_{CC\,(AV)}$ | Avg. Power Supply Current ($V_{CC}$) | | 60 | 80 | mA | $T_{CY}$ = .48 μsec |
| $I_{BB\,(AV)}$ | Avg. Power Supply Current ($V_{BB}$) | | .01 | 1 | mA | |
| $I_{IL}$ | Input Leakage | | | ±10 | μA | $V_{SS} \leqslant V_{IN} \leqslant V_{CC}$ |
| $I_{CL}$ | Clock Leakage | | | ±10 | μA | $V_{SS} \leqslant V_{CLOCK} \leqslant V_{DD}$ |
| $I_{DL}$ [2] | Data Bus Leakage in Input Mode | | | -100<br>-2.0 | μA<br>mA | $V_{SS} \leqslant V_{IN} \leqslant V_{SS}$+0.8V<br>$V_{SS}$+0.8V $\leqslant V_{IN} \leqslant V_{CC}$ |
| $I_{FL}$ | Address and Data Bus Leakage During HOLD | | | +10<br>-100 | μA | $V_{ADDR/DATA} = V_{CC}$<br>$V_{ADDR/DATA} = V_{SS}$ + 0.45V |

## CAPACITANCE

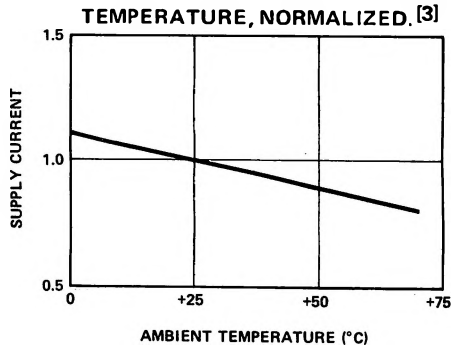
$T_A$ = 25°C $V_{CC} = V_{DD} = V_{SS}$ = 0V, $V_{BB}$ = -5V

| Symbol | Parameter | Typ. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $C_\phi$ | Clock Capacitance | 17 | 25 | pf | $f_c$ = 1 MHz |
| $C_{IN}$ | Input Capacitance | 6 | 10 | pf | Unmeasured Pins |
| $C_{OUT}$ | Output Capacitance | 10 | 20 | pf | Returned to $V_{SS}$ |

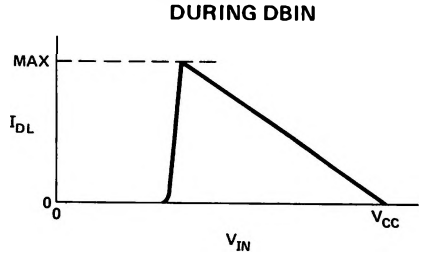
NOTES:

1. The RESET signal must be active for a minimum of 3 clock cycles.
2. When DBIN is high and $V_{IN} > V_{IH}$ an internal active pull up will be switched onto the Data Bus.
3. $\Delta I_{supply} / \Delta T_A$ = -0.45%/°C.

TYPICAL SUPPLY CURRENT VS. TEMPERATURE, NORMALIZED.[3]




DATA BUS CHARACTERISTIC DURING DBIN



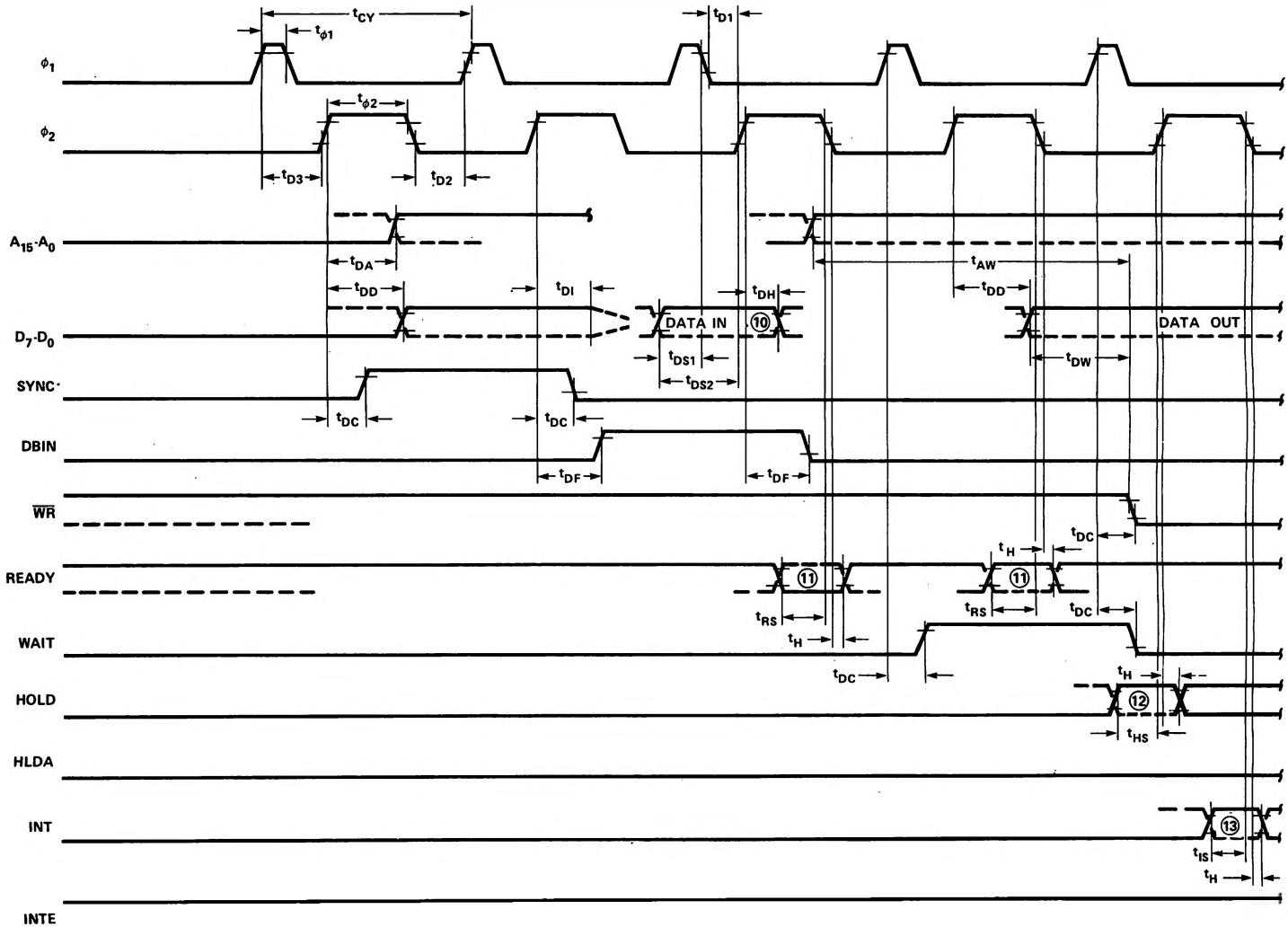
## A.C. CHARACTERISTICS

$T_A$ = 0°C to 70°C, $V_{DD}$ = +12V ± 5%, $V_{CC}$ = +5V ± 5%, $V_{BB}$ = -5V ± 5%, $V_{SS}$ = 0V, Unless Otherwise Noted

| Symbol | Parameter | Min. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $t_{CY}$[3] | Clock Period | 0.48 | 2.0 | μsec | |
| $t_r$, $t_f$ | Clock Rise and Fall Time | 0 | 50 | nsec | |
| $t_{\phi 1}$ | $\phi_1$ Pulse Width | 60 | | nsec | |
| $t_{\phi 2}$ | $\phi_2$ Pulse Width | 220 | | nsec | |
| $t_{D1}$ | Delay $\phi_1$ to $\phi_2$ | 0 | | nsec | |
| $t_{D2}$ | Delay $\phi_2$ to $\phi_1$ | 70 | | nsec | |
| $t_{D3}$ | Delay $\phi_1$ to $\phi_2$ Leading Edges | 80 | | nsec | |
| $t_{DA}$ [2] | Address Output Delay From $\phi_2$ | | 200 | nsec | $C_L$ = 100pf |
| $t_{DD}$ [2] | Data Output Delay From $\phi_2$ | | 220 | nsec | |
| $t_{DC}$ [2] | Signal Output Delay From $\phi_1$, or $\phi_2$ (SYNC, $\overline{WR}$, WAIT, HLDA) | | 120 | nsec | $C_L$ = 50pf |
| $t_{DF}$ [2] | DBIN Delay From $\phi_2$ | 25 | 140 | nsec | |
| $t_{DI}$[1] | Delay for Input Bus to Enter Input Mode | | $t_{DF}$ | nsec | |
| $t_{DS1}$ | Data Setup Time During $\phi_1$ and DBIN | 30 | | nsec | |

## TIMING WAVEFORMS [14]

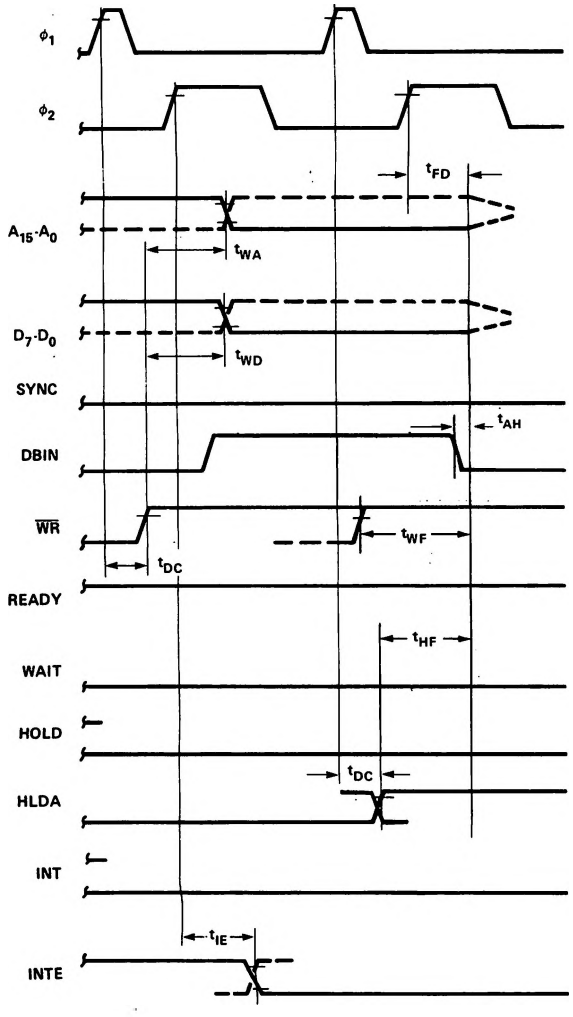
(Note: Timing measurements are made at the following reference voltages: CLOCK "1" = 8.0V "0" = 1.0V; INPUTS "1" = 3.3V, "0" = 0.8V; OUTPUTS "1" = 2.0V, "0" = 0.8V.)



## A.C. CHARACTERISTICS (Continued)

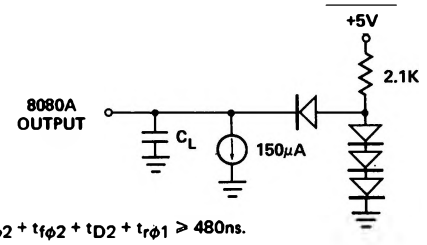
$T_A$ = 0°C to 70°C, $V_{DD}$ = +12V ± 5%, $V_{CC}$ = +5V ± 5%, $V_{BB}$ = -5V ± 5%, $V_{SS}$ = 0V, Unless Otherwise Noted

| Symbol | Parameter | Min. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $t_{DS2}$ | Data Setup Time to $\phi_2$ During DBIN | 150 | | nsec | |
| $t_{DH}$ [1] | Data Hold Time From $\phi_2$ During DBIN | [1] | | nsec | |
| $t_{IE}$ [2] | INTE Output Delay From $\phi_2$ | | 200 | nsec | $C_L$ = 50pf |
| $t_{RS}$ | READY Setup Time During $\phi_2$ | 120 | | nsec | |
| $t_{HS}$ | HOLD Setup Time to $\phi_2$ | 140 | | nsec | |
| $t_{IS}$ | INT Setup Time During $\phi_2$ (During $\phi_1$ in Halt Mode) | 120 | | nsec | |
| $t_H$ | Hold Time From $\phi_2$ (READY, INT, HOLD) | 0 | | nsec | |
| $t_{FD}$ | Delay to Float During Hold (Address and Data Bus) | | 120 | nsec | |
| $t_{AW}$ [2] | Address Stable Prior to $\overline{WR}$ | [5] | | nsec | $C_L$=100pf: Address, Data<br>$C_L$=50pf: $\overline{WR}$, HLDA, DBIN |
| $t_{DW}$ [2] | Output Data Stable Prior to $\overline{WR}$ | [6] | | nsec | |
| $t_{WD}$ [2] | Output Data Stable From $\overline{WR}$ | [7] | | nsec | |
| $t_{WA}$ [2] | Address Stable From $\overline{WR}$ | [7] | | nsec | |
| $t_{HF}$ [2] | HLDA to Float Delay | [8] | | nsec | |
| $t_{WF}$ [2] | $\overline{WR}$ to Float Delay | [9] | | nsec | |
| $t_{AH}$ [2] | Address Hold Time After DBIN During HLDA | -20 | | nsec | |



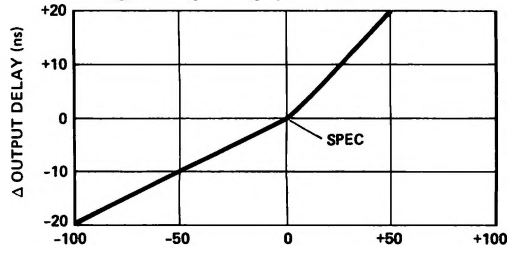

NOTES:

1. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured. $t_{DH}$ = 50 ns or $t_{DF}$, whichever is less.
2. Load Circuit.




3. $t_{CY} = t_{D3} + t_{r\phi2} + t_{\phi2} + t_{f\phi2} + t_{D2} + t_{r\phi1} \geq 480$ns.

**TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE**




4. The following are relevant when interfacing the 8080A to devices having $V_{IH}$ = 3.3V:
   a) Maximum output rise time from .8V to 3.3V = 100ns @ $C_L$ = SPEC.
   b) Output delay when measured to 3.0V = SPEC +60ns @ $C_L$ = SPEC.
   c) If $C_L \neq$ SPEC, add .6ns/pF if $C_L > C_{SPEC}$, subtract .3ns/pF (from modified delay) if $C_L < C_{SPEC}$.
5. $t_{AW} = 2\,t_{CY} - t_{D3} - t_{r\phi2} - 140$nsec.
6. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi2} - 170$nsec.
7. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi2} + 10$ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.
8. $t_{HF} = t_{D3} + t_{r\phi2} - 50$ns.
9. $t_{WF} = t_{D3} + t_{r\phi2} - 10$ns
10. Data in must be stable for this period during DBIN · $T_3$. Both $t_{DS1}$ and $t_{DS2}$ must be satisfied.
11. Ready signal must be stable for this period during $T_2$ or $T_W$. (Must be externally synchronized.)
12. Hold signal must be stable for this period during $T_2$ or $T_W$ when entering hold mode, and during $T_3$, $T_4$, $T_5$ and $T_{WH}$ when in hold mode. (External synchronization is not required.)
13. Interrupt signal must be stable during this period of the last clock cycle of any instruction in order to be recognized on the following instruction. (External synchronization is not required.)
14. This timing diagram shows timing relationships only; it does not represent any specific machine cycle.

## INSTRUCTION SET

The accumulator group instructions include arithmetic and logical operators with direct, indirect, and immediate addressing modes.

Move, load, and store instruction groups provide the ability to move either 8 or 16 bits of data between memory, the six working registers and the accumulator using direct, indirect, and immediate addressing modes.

The ability to branch to different portions of the program is provided with jump, jump conditional, and computed jumps. Also the ability to call to and return from subroutines is provided both conditionally and unconditionally. The RESTART (or single byte call instruction) is useful for interrupt vector operation.

Double precision operators such as stack manipulation and double add instructions extend both the arithmetic and interrupt handling capability of the 8080A. The ability to increment and decrement memory, the six general registers and the accumulator is provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer. Further capability is provided by the ability to rotate the accumulator left or right through or around the carry bit.

Input and output may be accomplished using memory addresses as I/O ports or the directly addressed I/O provided for in the 8080A instruction set.

The following special instruction group completes the 8080A instruction set: the NOP instruction, HALT to stop processor execution and the DAA instructions provide decimal arithmetic capability. STC allows the carry flag to be directly set, and the CMC instruction allows it to be complemented. CMA complements the contents of the accumulator and XCHG exchanges the contents of two 16-bit register pairs directly.

### Data and Instruction Formats

Data in the 8080A is stored in the form of 8-bit binary integers. All data transfers to the system data bus will be in the same format.

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|

DATA WORD

The program instructions may be one, two, or three bytes in length. Multiple byte instructions must be stored in successive words in program memory. The instruction formats then depend on the particular operation executed.

One Byte Instructions

TYPICAL INSTRUCTIONS

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | OP CODE |
|---|---|---|---|---|---|---|---|---|

Register to register, memory reference, arithmetic or logical, rotate, return, push, pop, enable or disable Interrupt instructions

Two Byte Instructions

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | OP CODE |
|---|---|---|---|---|---|---|---|---|
| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | OPERAND |

Immediate mode or I/O instructions

Three Byte Instructions

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | OP CODE |
|---|---|---|---|---|---|---|---|---|
| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | LOW ADDRESS OR OPERAND 1 |
| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | HIGH ADDRESS OR OPERAND 2 |

Jump, call or direct load and store instructions

For the 8080A a logic "1" is defined as a high level and a logic "0" is defined as a low level.

## INSTRUCTION SET

### Summary of Processor Instructions

| Mnemonic | Description | Instruction Code[1] $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | Clock[2] Cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| $MOV_{r1,r2}$ | Move register to register | 0 | 1 | D | D | D | S | S | S | 5 |
| MOV M, r | Move register to memory | 0 | 1 | 1 | 1 | 0 | S | S | S | 7 |
| MOV r, M | Move memory to register | 0 | 1 | D | D | D | 1 | 1 | 0 | 7 |
| HLT | Halt | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 7 |
| MVI r | Move immediate register | 0 | 0 | D | D | D | 1 | 1 | 0 | 7 |
| MVI M | Move immediate memory | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 10 |
| INR r | Increment register | 0 | 0 | D | D | D | 1 | 0 | 0 | 5 |
| DCR r | Decrement register | 0 | 0 | D | D | D | 1 | 0 | 1 | 5 |
| INR M | Increment memory | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 10 |
| DCR M | Decrement memory | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 10 |
| ADD r | Add register to A | 1 | 0 | 0 | 0 | 0 | S | S | S | 4 |
| ADC r | Add register to A with carry | 1 | 0 | 0 | 0 | 1 | S | S | S | 4 |
| SUB r | Subtract register from A | 1 | 0 | 0 | 1 | 0 | S | S | S | 4 |
| SBB r | Subtract register from A with borrow | 1 | 0 | 0 | 1 | 1 | S | S | S | 4 |
| ANA r | And register with A | 1 | 0 | 1 | 0 | 0 | S | S | S | 4 |
| XRA r | Exclusive Or register with A | 1 | 0 | 1 | 0 | 1 | S | S | S | 4 |
| ORA r | Or register with A | 1 | 0 | 1 | 1 | 0 | S | S | S | 4 |
| CMP r | Compare register with A | 1 | 0 | 1 | 1 | 1 | S | S | S | 4 |
| ADD M | Add memory to A | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 7 |
| ADC M | Add memory to A with carry | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 7 |
| SUB M | Subtract memory from A | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 7 |
| SBB M | Subtract memory from A with borrow | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 7 |
| ANA M | And memory with A | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 7 |
| XRA M | Exclusive Or memory with A | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 7 |
| ORA M | Or memory with A | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 7 |
| CMP M | Compare memory with A | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 7 |
| ADI | Add immediate to A | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 7 |
| ACI | Add immediate to A with carry | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 7 |
| SUI | Subtract immediate from A | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 7 |
| SBI | Subtract immediate from A with borrow | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 7 |
| ANI | And immediate with A | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 7 |
| XRI | Exclusive Or immediate with A | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 7 |
| ORI | Or immediate with A | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 7 |
| CPI | Compare immediate with A | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 7 |
| RLC | Rotate A left | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 4 |
| RRC | Rotate A right | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 4 |
| RAL | Rotate A left through carry | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 4 |
| RAR | Rotate A right through carry | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 4 |
| JMP | Jump unconditional | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 10 |
| JC | Jump on carry | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 10 |
| JNC | Jump on no carry | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 10 |
| JZ | Jump on zero | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 10 |
| JNZ | Jump on no zero | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 10 |
| JP | Jump on positive | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 10 |
| JM | Jump on minus | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 10 |
| JPE | Jump on parity even | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 10 |
| JPO | Jump on parity odd | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 10 |
| CALL | Call unconditional | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 17 |
| CC | Call on carry | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 11/17 |
| CNC | Call on no carry | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 11/17 |
| CZ | Call on zero | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 11/17 |
| CNZ | Call on no zero | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 11/17 |
| CP | Call on positive | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 11/17 |
| CM | Call on minus | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 11/17 |
| CPE | Call on parity even | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 11/17 |
| CPO | Call on parity odd | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 11/17 |
| RET | Return | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 10 |
| RC | Return on carry | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 5/11 |
| RNC | Return on no carry | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 5/11 |
| RZ | Return on zero | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 5/11 |
| RNZ | Return on no zero | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 5/11 |
| RP | Return on positive | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 5/11 |
| RM | Return on minus | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 5/11 |
| RPE | Return on parity even | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 5/11 |
| RPO | Return on parity odd | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 5/11 |
| RST | Restart | 1 | 1 | A | A | A | 1 | 1 | 1 | 11 |
| IN | Input | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 10 |
| OUT | Output | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 10 |
| LXI B | Load immediate register Pair B & C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 |
| LXI D | Load immediate register Pair D & E | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 10 |
| LXI H | Load immediate register Pair H & L | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 10 |
| LXI SP | Load immediate stack pointer | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 10 |
| PUSH B | Push register Pair B & C on stack | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 11 |
| PUSH D | Push register Pair D & E on stack | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 11 |
| PUSH H | Push register Pair H & L on stack | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 11 |
| PUSH PSW | Push A and Flags on stack | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 11 |
| POP B | Pop register pair B & C off stack | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 10 |
| POP D | Pop register pair D & E off stack | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 10 |
| POP H | Pop register pair H & L off stack | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 10 |
| POP PSW | Pop A and Flags off stack | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 10 |
| STA | Store A direct | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 13 |
| LDA | Load A direct | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 13 |
| XCHG | Exchange D & E, H & L Registers | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 4 |
| XTHL | Exchange top of stack, H & L | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 18 |
| SPHL | H & L to stack pointer | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 5 |
| PCHL | H & L to program counter | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 5 |
| DAD B | Add B & C to H & L | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 10 |
| DAD D | Add D & E to H & L | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 10 |
| DAD H | Add H & L to H & L | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 10 |
| DAD SP | Add stack pointer to H & L | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 10 |
| STAX B | Store A indirect | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 7 |
| STAX D | Store A indirect | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 7 |
| LDAX B | Load A indirect | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 7 |
| LDAX D | Load A indirect | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 7 |
| INX B | Increment B & C registers | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 5 |
| INX D | Increment D & E registers | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 5 |
| INX H | Increment H & L registers | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 5 |
| INX SP | Increment stack pointer | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 5 |
| DCX B | Decrement B & C | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 5 |
| DCX D | Decrement D & E | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| DCX H | Decrement H & L | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 5 |
| DCX SP | Decrement stack pointer | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 5 |
| CMA | Complement A | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 4 |
| STC | Set carry | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 4 |
| CMC | Complement carry | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 4 |
| DAA | Decimal adjust A | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 |
| SHLD | Store H & L direct | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 16 |
| LHLD | Load H & L direct | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 16 |
| EI | Enable Interrupts | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 4 |
| DI | Disable interrupt | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 4 |
| NOP | No-operation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |

NOTES: 1. DDD or SSS — 000 B — 001 C — 010 D — 011 E — 100 H — 101 L — 110 Memory — 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.