intel®

# 8080A/8080A-1/8080A-2
# 8-BIT N-CHANNEL MICROPROCESSOR

- **TTL Drive Capability**
- **2 μs (−1:1.3 μs, −2:1.5 μs) Instruction Cycle**
- **Powerful Problem Solving Instruction Set**
- **6 General Purpose Registers and an Accumulator**
- **16-Bit Program Counter for Directly Addressing up to 64K Bytes of Memory**
- **16-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**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 40-Lead Cerdip and Plastic Packages**
  (See Packaging Spec. Order #231369)

1

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 6 8-bit general purpose working registers and an accumulator. The 6 general purpose registers may be addressed individually or in pairs providing both single and double precision operators. Arithmetic and logical instructions set or reset 4 testable flags. A fifth flag provides decimal arithmetic operation.

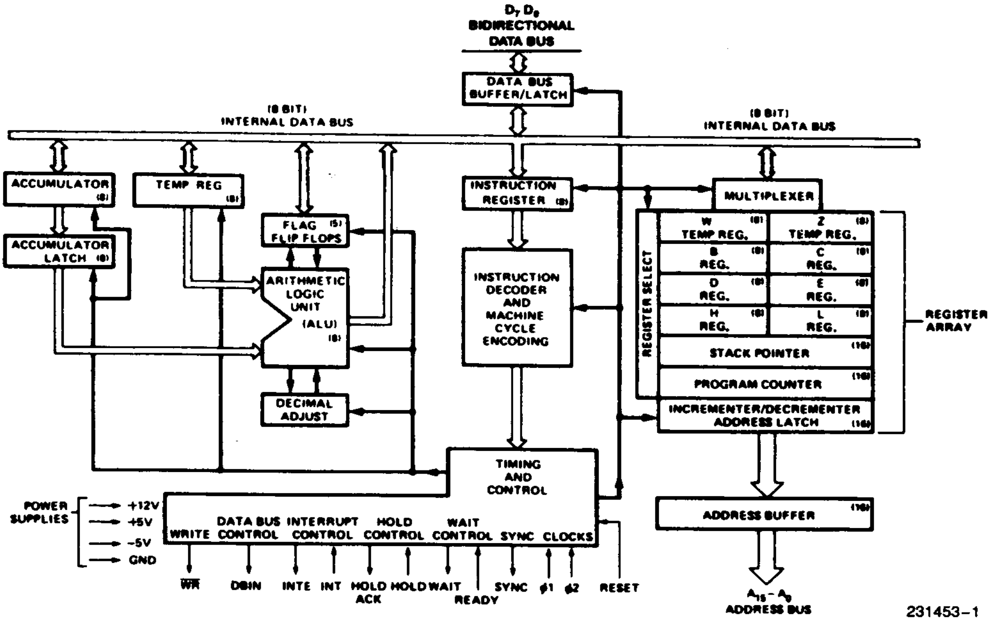
The 8080A has an external stack feature wherein any portion of memory may be used as a last in/first out stack to store/retrieve the contents of the accumulator, flags, program counter, and all of the 6 general purpose registers. The 16-bit stack pointer controls the addressing of this external stack. 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 bidirectional 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.

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

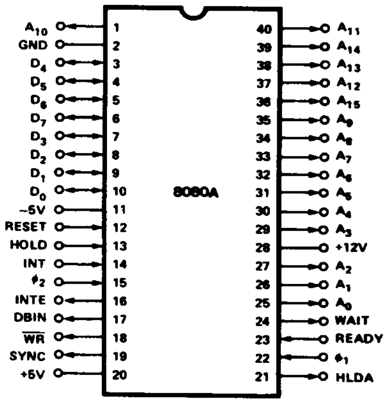


November 1986
Order Number: 231453-001




231453-1

**Figure 1. Block Diagram**




231453-2

**Figure 2. Pin Configuration**

**Table 1. Pin Description**

| Symbol | Type | Name and Function |
|---|---|---|
| $A_{15}$–$A_0$ | O | **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. |
| $D_7$–$D_0$ | I/O | **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 | O | **SYNCHRONIZING SIGNAL:** The SYNC pin provides a signal to indicate the beginning of each machine cycle. |
| DBIN | O | **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 | I | **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 | O | **WAIT:** The WAIT signal acknowledges that the CPU is in a WAIT state. |
| $\overline{WR}$ | O | **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 | I | **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 busses for the current machine cycle. It is recognized under the following conditions:<br>• the CPU is in the HALT state.<br>• 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 | O | **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:<br>• T3 for READ memory or input.<br>• The Clock Period following T3 for WRITE memory or OUTPUT operation.<br>In either case, the HLDA signal appears after the rising edge of $\phi_2$. |
| INTE | O | **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 | I | **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[1] | I | **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. |
| $V_{DD}$ | | **POWER:** +12 ±5% V. |
| $V_{CC}$ | | **POWER:** +5 ±5% V. |
| $V_{BB}$ | | **POWER:** −5 ±5% V. |
| $\phi_1$, $\phi_2$ | | **CLOCK PHASES:** 2 externally supplied clock phases. (non TTL compatible) |

**NOTE:**
1. The RESET signal must be active for a minimum of 3 clock cycles.

## 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

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING:* *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

## D.C. CHARACTERISTICS

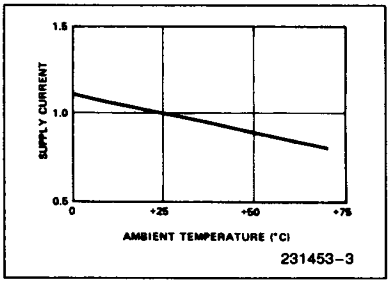
$T_A$ = 0°C to 70°C, $V_{DD}$ = +12V ±5%, $V_{CC}$ = +5V ±5%, $V_{BB}$ = −5V ±5%, $V_{SS}$ = 0V; unless otherwise noted

| 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 | $I_{OL}$ = 1.9 mA on All Outputs, $I_{OH}$ = −150 μA. |
| $V_{OH}$ | Output High Voltage | 3.7 | | | V | |
| $I_{DD\,(AV)}$ | Avg. Power Supply Current ($V_{DD}$) | | 40 | 70 | mA | Operation $T_{CY}$ = 0.48 μs |
| $I_{CC\,(AV)}$ | Avg. Power Supply Current ($V_{CC}$) | | 60 | 80 | mA | |
| $I_{BB\,(AV)}$ | Avg. Power Supply Current ($V_{BB}$) | | 0.01 | 1 | mA | |
| $I_{IL}$ | Input Leakage | | | ±10 | μA | $V_{SS} \le V_{IN} \le V_{CC}$ |
| $I_{CL}$ | Clock Leakage | | | ±10 | μA | $V_{SS} \le V_{CLOCK} \le V_{DD}$ |
| $I_{DL}$ | Data Bus Leakage in Input Mode | | | −100<br>−2.0 | μA<br>mA | $V_{SS} \le V_{IN} \le V_{SS} + 0.8V$<br>$V_{SS} + 0.8V \le V_{IN} \le V_{CC}$ |
| $I_{FL}$ | 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 $V_{SS}$ |



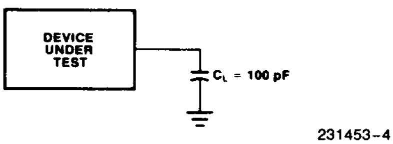

**Typical Supply Current vs Temperature, Normalized**
**$\Delta I\ Supply/\Delta T_A$ = −0.45%/°C**

## A.C. CHARACTERISTICS (8080A)

$T_A$ = 0°C to 70°C, $V_{DD}$ = +12V ±5%, $V_{CC}$ = +5V ±5%, $V_{BB}$ = −5V ±5%, $V_{SS}$ = 0V; unless otherwise noted

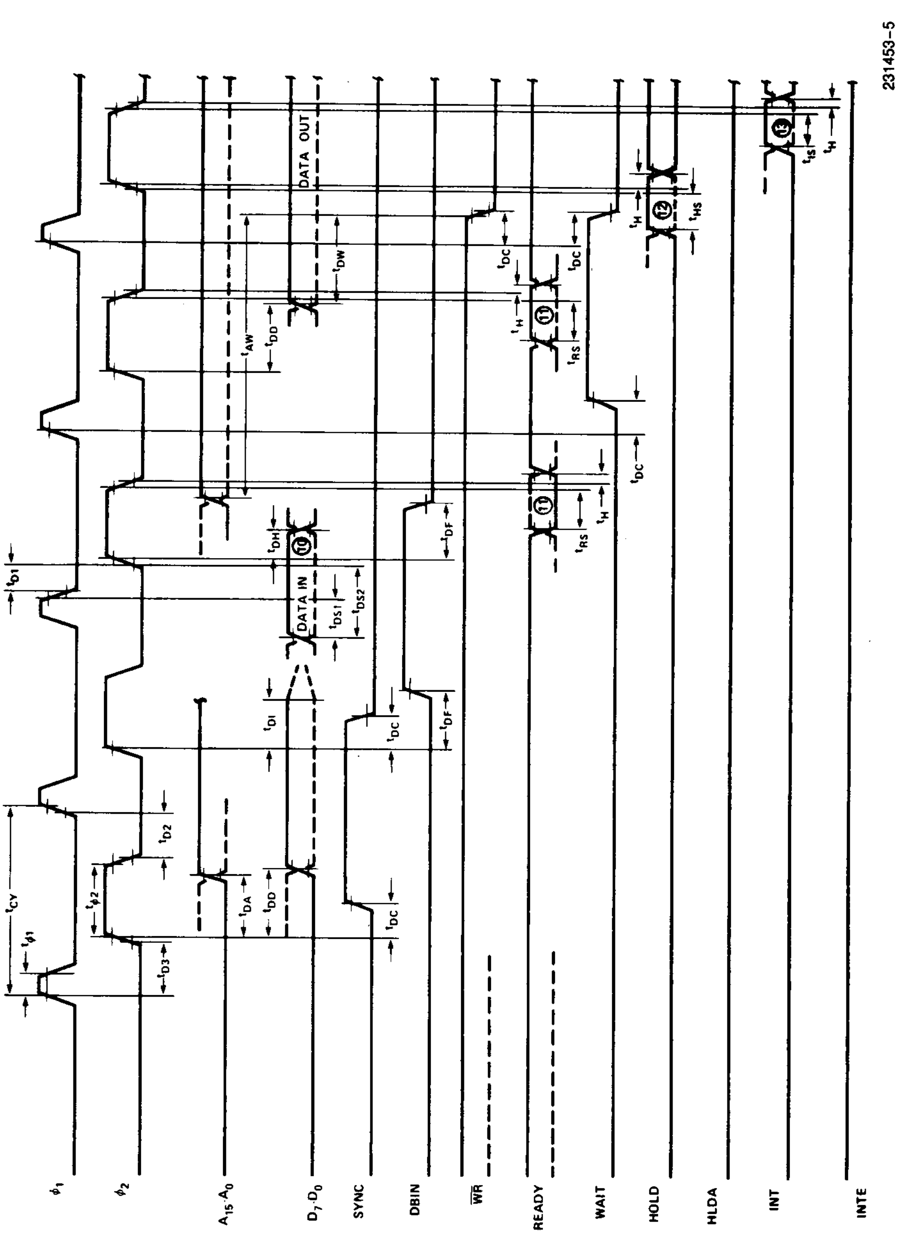
| Symbol | Parameter | Min | Max | −1 Min | −1 Max | −2 Min | −2 Max | Unit | Test Condition |
|---|---|---|---|---|---|---|---|---|---|
| $t_{CY}$(3) | Clock Period | 0.48 | 2.0 | 0.32 | 2.0 | 0.38 | 2.0 | μs | |
| $t_r$, $t_f$ | Clock Rise and Fall Time | 0 | 50 | 0 | 25 | 0 | 50 | ns | |
| $t_{\phi1}$ | $\phi_1$ Pulse Width | 60 | | 50 | | 60 | | ns | |
| $t_{\phi2}$ | $\phi_2$ Pulse Width | 220 | | 145 | | 175 | | ns | |
| $t_{D1}$ | Delay $\phi_1$ to $\phi_2$ | 0 | | 0 | | 0 | | ns | |
| $t_{D2}$ | Delay $\phi_1$ to $\phi_2$ | 70 | | 60 | | 70 | | ns | |
| $t_{D3}$ | Delay $\phi_1$ to $\phi_2$ Leading Edges | 80 | | 60 | | 70 | ns | | |
| $t_{DA}$ | Address Output Delay From $\phi_2$ | | 200 | | 150 | | 175 | ns | $C_L$ = 100 pF |
| $t_{DD}$ | Data Output Delay From $\phi_2$ | | 200 | | 180 | | 200 | ns | |
| $t_{DC}$ | Signal Output Delay From $\phi_1$ or $\phi_2$ (SYNC, WR, WAIT, HLDA) | | 120 | | 110 | | 120 | ns | $C_L$ = 50 pF |
| $t_{DF}$ | DBIN Delay From $\phi_2$ | 25 | 140 | 25 | 130 | 25 | 140 | ns | |
| $t_{DI}$(1) | Delay for Input Bus to Enter Input Mode | | $t_{DF}$ | | $t_{DF}$ | | $t_{DF}$ | ns | |
| $t_{DS1}$ | Data Setup Time During $\phi_1$ and DBIN | 30 | | 10 | | 20 | | ns | |
| $t_{DS2}$ | Data Setup Time to $\phi_2$ During DBIN | 150 | | 120 | | 130 | | ns | |
| $t_{DH}$(1) | Data Hold Time From $\phi_2$ and DBIN | (1) | | (1) | | (1) | | ns | |
| $t_{IE}$ | INTE Output Delay From $\phi_2$ | | 200 | | 200 | | 200 | ns | $C_L$ = 50 pF |
| $t_{RS}$ | READY Setup Time During $\phi_2$ | 120 | | 90 | | 90 | | ns | |
| $t_{HS}$ | HOLD Setup Time During $\phi_2$ | 140 | | 120 | | 120 | | ns | |
| $t_{IS}$ | INT Setup Time During $\phi_2$ | 120 | | 100 | | 100 | | ns | |
| $t_H$ | Hold Time From $\phi_2$ (READY, INT, HOLD) | 0 | | 0 | | 0 | | ns | |
| $t_{FD}$ | Delay to Float During Hold (Address and Data Bus) | | 120 | | 120 | | 120 | ns | |
| $t_{AW}$ | Address Stable Prior to WR | (5) | | (5) | | (5) | | ns | |
| $t_{DW}$ | Output Data Stable Prior to WR | (6) | | (6) | | (6) | | ns | |
| $t_{WD}$ | Output Data Stable From WR | (7) | | (7) | | (7) | | ns | |
| $t_{WA}$ | Address Stable From WR | (7) | | (7) | | (7) | | ns | |
| $t_{HF}$ | HLDA to Float Delay | (8) | | (8) | | (8) | | ns | |
| $t_{WF}$ | WR to Float Delay | (9) | | (9) | | (9) | | ns | |
| $t_{AH}$ | Address Hold Time After DBIN During HLDA | −20 | | −20 | | −20 | | ns | |

## A.C. TESTING LOAD CIRCUIT



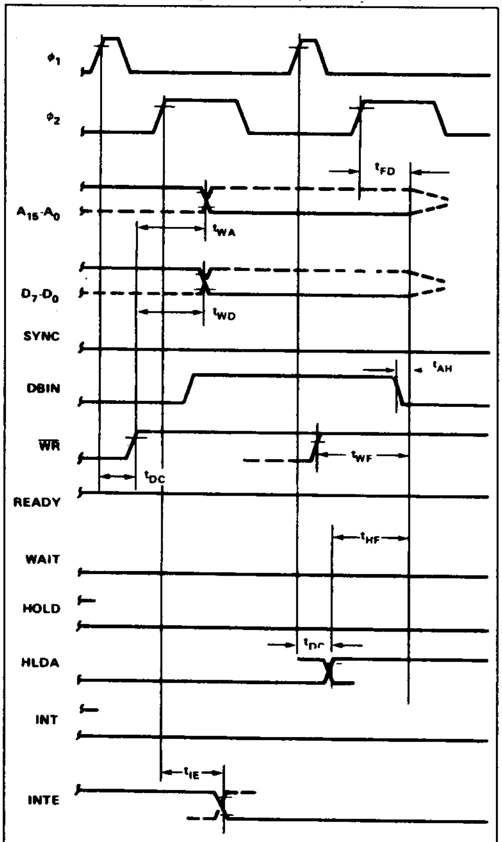

$C_L$ = 100 pF
$C_L$ Includes Jig Capacitance

## WAVEFORMS




**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

## WAVEFORMS (Continued)



231453-6

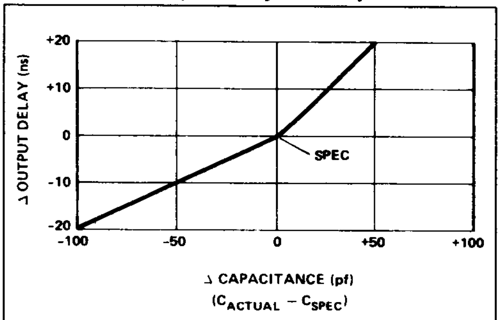
### NOTES:

(Parenthesis gives −1, −2 specifications, respectively.)

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. $t_{CY} = t_{D3} + t_{r\phi2} + t_{\phi2} + t_{f\phi2} + t_{D2} + t_{r\phi1} \geq$ 480 ns (−1:320 ns, − 2:380 ns).

**Typical Δ Output Delay vs Δ Capacitance**



231453-7

3. The following are relevant when interfacing the 8080A to devices having $V_{IH}$ = 3.3V:
   a) Maximum output rise time from 0.8V to 3.3V = 100 ns @ $C_L$ = SPEC.
   b) Output delay when measured to 3.0V = SPEC +60 ns @ $C_L$ = SPEC.
   c) If $C_L$ = SPEC, add 0.6 ns/pF if $C_L > C_{SPEC}$, subtract 0.3 ns/pF (from modified delay) if $C_L < C_{SPEC}$.

4. $t_{AW} = 2\ t_{CY} - t_{D3} - t_{r\phi2}$ − 140 ns (−1:110 ns, − 2:130 ns).

5. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi2}$ − 170 ns (−1:150 ns, − 2:170 ns).

6. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi2}$ + 10 ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.

7. $t_{HF} = t_{D3} + t_{r\phi2}$ −50 ns.

8. $t_{WF} = t_{D3} + t_{r\phi2}$ − 10 ns.

9. Data in must be stable for this period during DBIN $T_3$. Both $t_{DS1}$ and $t_{DS2}$ must be satisfied.

10. Ready signal must be stable for this period during $T_2$ or $T_W$. (Must be externally synchronized.)

11. 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.)

12. 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.)

13. 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 they 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.

| Format | Byte | TYPICAL INSTRUCTIONS |
|---|---|---|
| **One Byte 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 | Immediate mode or I/O instructions |
| $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$ | OPERAND | |
| **Three Byte Instructions** | | |
| $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$ | OP CODE | Jump, call or direct load and store instructions |
| $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$ | 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 | |

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

## Table 2. Instruction Set Summary

| Mnemonic* | Instruction Code (1) $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | Operations Description | Clock Cycles (2) |
|---|---|---|---|
| **MOVE, LOAD, AND STORE** | | | |
| MOVr1,r2 | 0 1 D D D S S S | Move register to register | 5 |
| MOV M,r | 0 1 1 1 0 S S S | Move register to memory | 7 |
| MOV r,M | 0 1 D D D 1 1 0 | Move memory to register | 7 |
| MVI r | 0 0 D D D 1 1 0 | Move immediate register | 7 |
| MVI M | 0 0 1 1 0 1 1 0 | Move immediate memory | 10 |
| LXI B | 0 0 0 0 0 0 0 1 | Load immediate register Pair B & C | 10 |
| LXI D | 0 0 0 1 0 0 0 1 | Load immediate register Pair D & E | 10 |
| LXI H | 0 0 1 0 0 0 0 1 | Load immediate register Pair H & L | 10 |
| STAX B | 0 0 0 0 0 0 1 0 | Store A indirect | 7 |
| STAX D | 0 0 0 1 0 0 1 0 | Store A indirect | 7 |
| LDAX B | 0 0 0 0 1 0 1 0 | Load A indirect | 7 |
| LDAX D | 0 0 0 1 1 0 1 0 | Load A indirect | 7 |
| STA | 0 0 1 1 0 0 1 0 | Store A direct | 13 |
| LDA | 0 0 1 1 1 0 1 0 | Load A direct | 13 |
| SHLD | 0 0 1 0 0 0 1 0 | Store H & L direct | 16 |
| LHLD | 0 0 1 0 1 0 1 0 | Load H & L direct | 16 |
| XCHG | 1 1 1 0 1 0 1 1 | Exchange D & E, H & L Registers | 4 |
| **STACK OPS** | | | |
| PUSH B | 1 1 0 0 0 1 0 1 | Push register Pair B & C on stack | 11 |
| PUSH D | 1 1 0 1 0 1 0 1 | Push register Pair D & E on stack | 11 |
| PUSH H | 1 1 1 0 0 1 0 1 | Push register Pair H & L on stack | 11 |
| PUSH PSW | 1 1 1 1 0 1 0 1 | Push A and Flags on stack | 11 |
| POP B | 1 1 0 0 0 0 0 1 | Pop register Pair B & C off stack | 10 |
| POP D | 1 1 0 1 0 0 0 1 | Pop register Pair D & E off stack | 10 |
| POP H | 1 1 1 0 0 0 0 1 | Pop register Pair H & L off stack | 10 |
| POP PSW | 1 1 1 1 0 0 0 1 | Pop A and Flags off stack | 10 |
| XTHL | 1 1 1 0 0 0 1 1 | Exchange top of stack, H & L | 18 |
| SPHL | 1 1 1 1 1 0 0 1 | H & L to stack pointer | 5 |
| LXI SP | 0 0 1 1 0 0 0 1 | Load immediate stack pointer | 10 |
| INX SP | 0 0 1 1 0 0 1 1 | Increment stack pointer | 5 |
| DCX SP | 0 0 1 1 1 0 1 1 | Decrement stack pointer | 5 |
| **JUMP** | | | |
| JMP | 1 1 0 0 0 0 1 1 | Jump unconditional | 10 |
| JC | 1 1 0 1 1 0 1 0 | Jump on carry | 10 |
| JNC | 1 1 0 1 0 0 1 0 | Jump on no carry | 10 |
| JZ | 1 1 0 0 1 0 1 0 | Jump on zero | 10 |
| JNZ | 1 1 0 0 0 0 1 0 | Jump on no zero | 10 |
| JP | 1 1 1 1 0 0 1 0 | Jump on positive | 10 |
| JM | 1 1 1 1 1 0 1 0 | Jump on minus | 10 |
| JPE | 1 1 1 0 1 0 1 0 | Jump on parity even | 10 |
| JPO | 1 1 1 0 0 0 1 0 | Jump on parity odd | 10 |
| PCHL | 1 1 1 0 1 0 0 1 | H & L to program counter | 5 |
| **CALL** | | | |
| CALL | 1 1 0 0 1 1 0 1 | Call unconditional | 17 |
| CC | 1 1 0 1 1 1 0 0 | Call on carry | 11/17 |
| CNC | 1 1 0 1 0 1 0 0 | Call on no carry | 11/17 |
| CZ | 1 1 0 0 1 1 0 0 | Call on zero | 11/17 |
| CNZ | 1 1 0 0 0 1 0 0 | Call on no zero | 11/17 |
| CP | 1 1 1 1 0 1 0 0 | Call on positive | 11/17 |
| CM | 1 1 1 1 1 1 0 0 | Call on minus | 11/17 |
| CPE | 1 1 1 0 1 1 0 0 | Call on parity even | 11/17 |
| CPO | 1 1 1 0 0 1 0 0 | Call on parity odd | 11/17 |
| **RETURN** | | | |
| RET | 1 1 0 0 1 0 0 1 | Return | 10 |
| RC | 1 1 0 1 1 0 0 0 | Return on carry | 5/11 |
| RNC | 1 1 0 1 0 0 0 0 | Return on no carry | 5/11 |
| RZ | 1 1 0 0 1 0 0 0 | Return on zero | 5/11 |
| RNZ | 1 1 0 0 0 0 0 0 | Return on no zero | 5/11 |
| RP | 1 1 1 1 0 0 0 0 | Return on positive | 5/11 |
| RM | 1 1 1 1 1 0 0 0 | Return on minus | 5/11 |
| RPE | 1 1 1 0 1 0 0 0 | Return on parity even | 5/11 |
| RPO | 1 1 1 0 0 0 0 0 | Return on parity odd | 5/11 |
| **RESTART** | | | |
| RST | 1 1 A A A 1 1 1 | Restart | 11 |
| **INCREMENT AND DECREMENT** | | | |
| INR r | 0 0 D D D 1 0 0 | Increment register | 5 |
| DCR r | 0 0 D D D 1 0 1 | Decrement register | 5 |
| INR M | 0 0 1 1 0 1 0 0 | Increment memory | 10 |
| DCR M | 0 0 1 1 0 1 0 1 | Decrement memory | 10 |
| INX B | 0 0 0 0 0 0 1 1 | Increment B & C registers | 5 |
| INX D | 0 0 0 1 0 0 1 1 | Increment D & E registers | 5 |
| INX H | 0 0 1 0 0 0 1 1 | Increment H & L registers | 5 |
| DCX B | 0 0 0 0 1 0 1 1 | Decrement B & C | 5 |
| DCX D | 0 0 0 1 1 0 1 1 | Decrement D & E | 5 |
| DCX H | 0 0 1 0 1 0 1 1 | Decrement H & L | 5 |
| **ADD** | | | |
| ADD r | 1 0 0 0 0 S S S | Add register to A | 4 |
| ADC r | 1 0 0 0 1 S S S | Add register to A with carry | 4 |
| ADD M | 1 0 0 0 0 1 1 0 | Add memory to A | 7 |
| ADC M | 1 0 0 0 1 1 1 0 | Add memory to A with carry | 7 |
| ADI | 1 1 0 0 0 1 1 0 | Add immediate to A | 7 |
| ACI | 1 1 0 0 1 1 1 0 | Add immediate to A with carry | 7 |
| DAD B | 0 0 0 0 1 0 0 1 | Add B & C to H & L | 10 |
| DAD D | 0 0 0 1 1 0 0 1 | Add D & E to H & L | 10 |
| DAD H | 0 0 1 0 1 0 0 1 | Add H & L to H & L | 10 |
| DAD SP | 0 0 1 1 1 0 0 1 | Add stack pointer to H & L | 10 |

**Table 2. Instruction Set Summary** (Continued)

| Mnemonic* | Instruction Code (1) $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | Operations Description | Clock Cycles (2) |
|---|---|---|---|
| **SUBTRACT** | | | |
| SUB r | 1 0 0 1 0 S S S | Subtract register from A | 4 |
| SBB r | 1 0 0 1 1 S S S | Subtract register from A with borrow | 4 |
| SUB M | 1 0 0 1 0 1 1 0 | Subtract memory from A | 7 |
| SBB M | 1 0 0 1 1 1 1 0 | Subtract memory from A with borrow | 7 |
| SUI | 1 1 0 1 0 1 1 0 | Subtract immediate from A | 7 |
| SBI | 1 1 0 1 1 1 1 0 | Subtract immediate from A with borrow | 7 |
| **LOGICAL** | | | |
| ANA r | 1 0 1 0 0 S S S | And register with A | 4 |
| XRA r | 1 0 1 0 1 S S S | Exclusive or register with A | 4 |
| ORA r | 1 0 1 1 0 S S S | Or register with A | 4 |
| CMP r | 1 0 1 1 1 S S S | Compare register with A | 4 |
| ANA M | 1 0 1 0 0 1 1 0 | And memory with A | 7 |
| XRA M | 1 0 1 0 1 1 1 0 | Exclusive Or memory with A | 7 |
| ORA M | 1 0 1 1 0 1 1 0 | Or memory with A | 7 |
| CMP M | 1 0 1 1 1 1 1 0 | Compare memory with A | 7 |
| ANI | 1 1 1 0 0 1 1 0 | And immediate with A | 7 |
| XRI | 1 1 1 0 1 1 1 0 | Exclusive Or immediate with A | 7 |
| ORI | 1 1 1 1 0 1 1 0 | Or immediate with A | 7 |
| CPI | 1 1 1 1 1 1 1 0 | Compare immediate with A | 7 |

| Mnemonic* | Instruction Code (1) $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | Operations Description | Clock Cycles (2) |
|---|---|---|---|
| **ROTATE** | | | |
| RLC | 0 0 0 0 0 1 1 1 | Rotate A left | 4 |
| RRC | 0 0 0 0 1 1 1 1 | Rotate A right | 4 |
| RAL | 0 0 0 1 0 1 1 1 | Rotate A left through carry | 4 |
| RAR | 0 0 0 1 1 1 1 1 | Rotate A right through carry | 4 |
| **SPECIALS** | | | |
| CMA | 0 0 1 0 1 1 1 1 | Complement A | 4 |
| STC | 0 0 1 1 0 1 1 1 | Set carry | 4 |
| CMC | 0 0 1 1 1 1 1 1 | Complement carry | 4 |
| DAA | 0 0 1 0 0 1 1 1 | Decimal adjust A | 4 |
| **INPUT/OUTPUT** | | | |
| IN | 1 1 0 1 1 0 1 1 | Input | 10 |
| OUT | 1 1 0 1 0 0 1 1 | Output | 10 |
| **CONTROL** | | | |
| EI | 1 1 1 1 1 0 1 1 | Enable Interrupts | 4 |
| DI | 1 1 1 1 0 0 1 1 | Disable Interrupt | 4 |
| NOP | 0 0 0 0 0 0 0 0 | No-operation | 4 |
| HLT | 0 1 1 1 0 1 1 0 | Halt | 7 |

**NOTES:**
1. DDD or SSS: B = 000, C = 001, D = 010, E = 011, H = 100, L = 101, Memory = 110, A = 111.
2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.