National Semiconductor

MARCH 1977
Pub. No. 420305226-001C

S-61 RES ORIG
003639
T-3639
NSC

INS8080A 8-Bit N-Channel Microprocessor

general description

The INS8080A is an 8-bit microprocessor housed in a standard, 40-pin dual-in-line package. The chip, which is fabricated using N-channel silicon gate MOS technology, functions as the central processing unit (CPU) in National Semiconductor's N8080 microcomputer family.

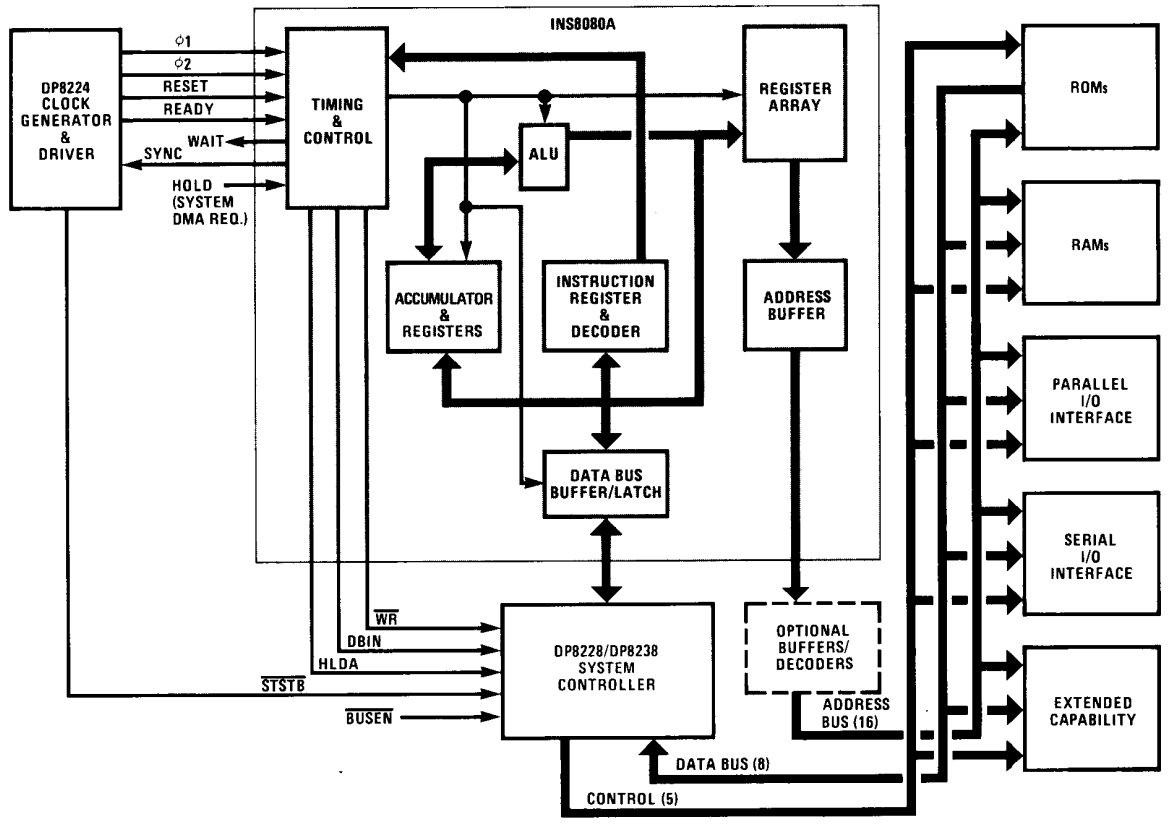
The INS8080A has a 16-bit address bus that is capable of addressing up to 65k bytes of memory and up to 256 input and 256 output devices. Data is routed to and from the INS8080A on a separate bidirectional 8-bit bus. This data bus is also TRI-STATE®, making direct memory addressing (DMA) and multiprocessing applications possible. The INS8080A directly provides signals to control the interface to memory and I/O ports. All buses, including control, are TTL compatible.

An asynchronous interrupt capability is included in the INS8080A to allow external signals to change the instruction sequence. The interrupting device may vector the program to a particular service routine location (or some other direct function) by specifying an interrupt instruction to be executed.

features

- 2μs Instruction Cycle
- Variable Length Instructions
- General Purpose Registers – Six plus an Accumulator
- Direct Addressing up to 65k Bytes
- Variable Length Stack Accessed by 16-bit Stack Pointer
- Addresses 256 Input and 256 Output Ports
- Provisions for Vectored Interrupts
- TRI-STATE® Bus for DMA and Multiprocessing Capability
- TRI-STATE TTL Drive Capabilities for Address and Data Buses
- Decimal Arithmetic Capability
- Multiple Addressing Modes
  - Direct
  - Register
  - Register Indirect
  - Immediate
- Direct Plug-in Replacement for Intel 8080A

N8080A microcomputer family block diagram



 DA-B20M37R2/Printed in U.S.A.

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

Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under dc electrical characteristics.

dc electrical 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.

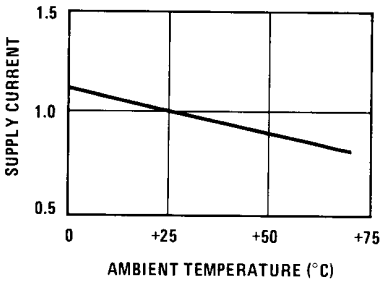
Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	$V_{SS}-1$		$V_{SS}+0.8$	V	
V_{IHC}	Clock Input High Voltage	9.0		$V_{DD}+1$	V	
V_{IL}	Input Low Voltage	$V_{SS}-1$		$V_{SS}+0.8$	V	
V_{IH}	Input High Voltage	3.3		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 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} = 0.48μs
$I_{BB\ (AV)}$	Avg. Power Supply Current (V_{BB})		0.01	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	μA	$V_{SS} \leqslant V_{IN} \leqslant V_{SS} + 0.8V$
				-2.0	mA	$V_{SS} + 0.8V \leqslant V_{IN} \leqslant V_{CC}$
I_{FL}	Address and Data Bus Leakage			+10	μA	$V_{ADDR/DATA} = V_{CC}$
	During HOLD			-100	μA	$V_{ADDR/DATA} = V_{SS} + 0.45V$

capacitance

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

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

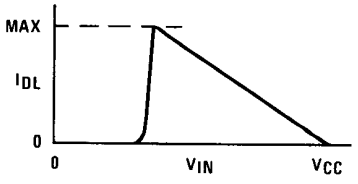
TYPICAL SUPPLY CURRENT VS. TEMPERATURE, NORMALIZED[3]



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 pullup will be switched onto the Data Bus.
3. ΔI supply / ΔT_A = -0.45%/°C.

DATA BUS CHARACTERISTIC DURING DBIN

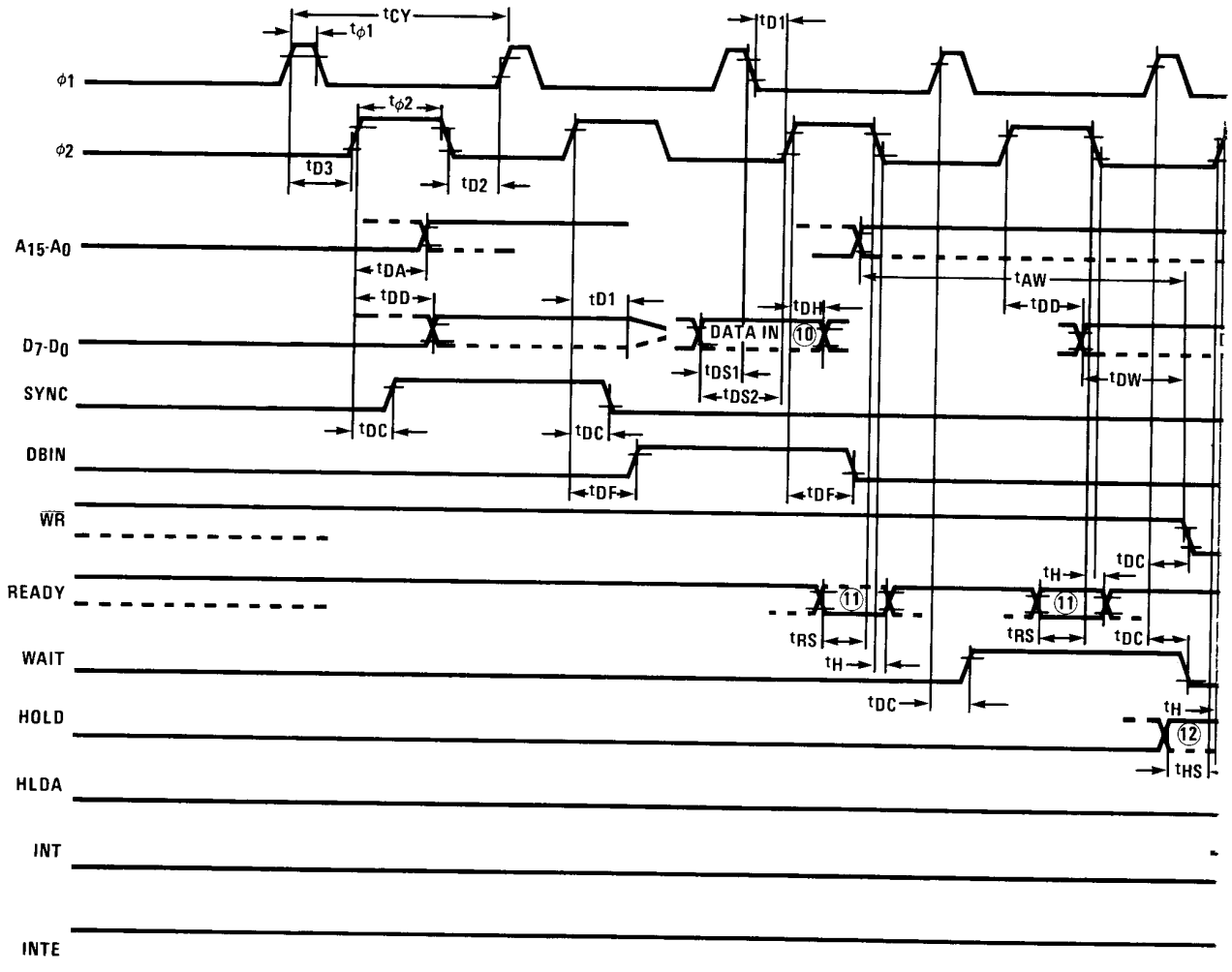


ac electrical characteristics

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{CY} [3]	Clock Period	0.48	2.0	μs	
t_r, t_f	Clock Rise and Fall Time	0	50	ns	
$t_{\phi 1}$	ϕ_1 Pulse Width	60		ns	
$t_{\phi 2}$	ϕ_2 Pulse Width	220		ns	
t_{D1}	Delay ϕ_1 to ϕ_2	0		ns	
t_{D2}	Delay ϕ_2 to ϕ_1	70		ns	
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges	80		ns	
t_{DA} [2]	Address Output Delay from ϕ_2		200	ns	C_L = 100pF
t_{DD} [2]	Data Output Delay from ϕ_2		220	ns	C_L = 100pF
t_{DC} [2]	Signal Output Delay from ϕ_1 or ϕ_2 (SYNC, $\overline{WR}$, WAIT, HLDA)		120	ns	C_L = 50pF
t_{DF} [2]	DBIN Delay from ϕ_2	25	140	ns	C_L = 50pF
t_{DI} [1]	Delay for Input Bus to Enter Input Mode		t_{DF}	ns	
t_{DS1}	Data Setup Time During ϕ_1 and DBIN	30		ns	

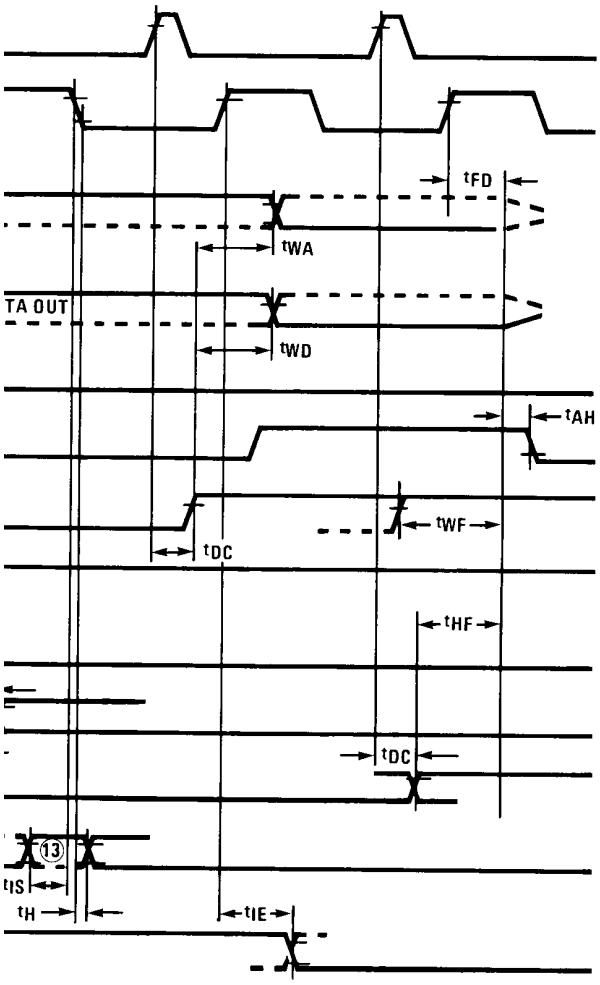
timing waveforms [14]

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.



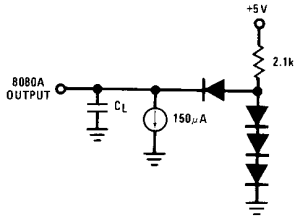
ac electrical characteristics (cont'd.)

Symbol	Parameter	Min.	Max.	Unit	Test Condition
t_{DS2}	Data Setup Time to ϕ_2 During DBIN	150		ns	
t_{DH} [1]	Data Hold Time from ϕ_2 During DBIN	1		ns	
t_{IE} [2]	INTE Output Delay from ϕ_2		200	ns	C_L = 50pF
t_{RS}	READY Setup Time During ϕ_2	120		ns	
t_{HS}	HOLD Setup Time to ϕ_2	140		ns	
t_{IS}	INT Setup Time During ϕ_2 (During ϕ_1 in Halt Mode)	120		ns	
t_H	Hold Time from ϕ_2 (READY, INT, HOLD)	0		ns	
t_{FD}	Delay to Float During Hold (Address and Data Bus)		120	ns	
t_{AW} [2]	Address Stable Prior to $\overline{WR}$	5		ns	C_L = 100pF : Address, Data C_L = 50pF: $\overline{WR}$, HLDA, DBIN
t_{DW} [2]	Output Data Stable Prior to $\overline{WR}$	6		ns	
t_{WD} [2]	Output Data Stable from $\overline{WR}$	7		ns	
t_{WA} [2]	Address Stable from $\overline{WR}$	7		ns	
t_{HF} [2]	HLDA to Float Delay	8		ns	
t_{WF} [2]	$\overline{WR}$ to Float Delay	9		ns	
t_{AH} [2]	Address Hold Time After DBIN During HLDA	-20		ns	

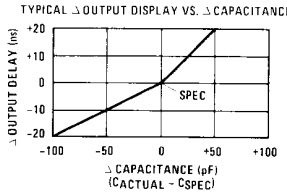


Notes:

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

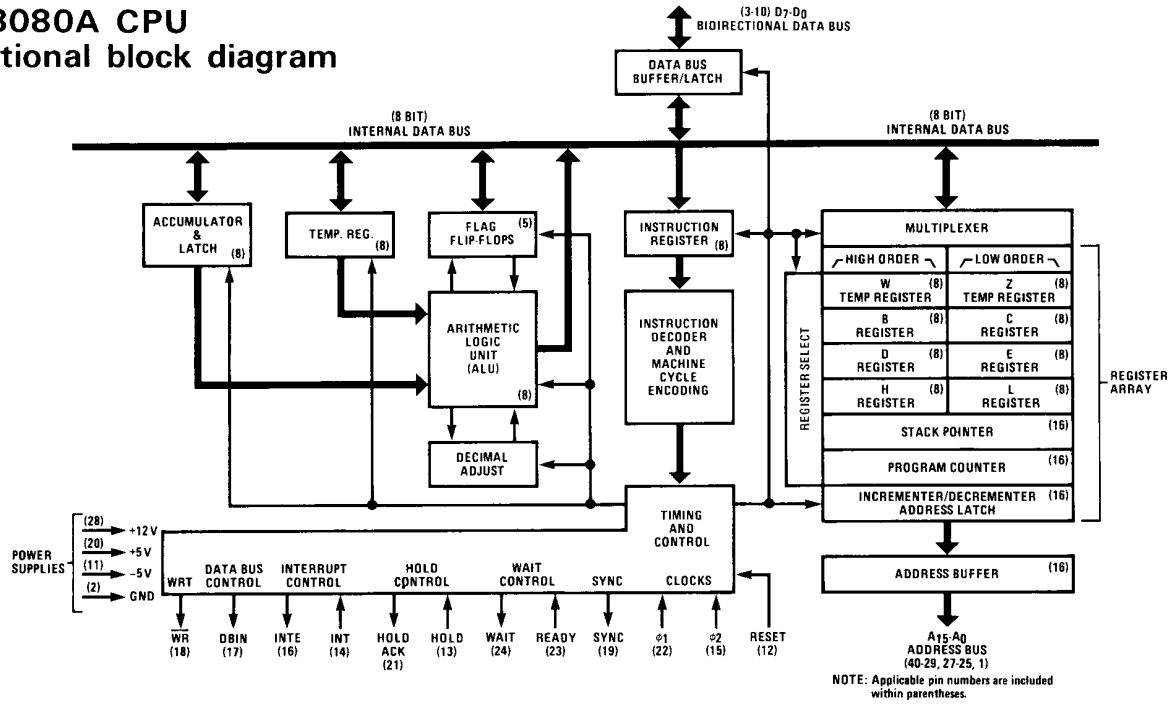


3. $t_{CY} = t_{D3} + t_{r\phi2} + t_{\phi2} + t_{D2} + t_{f\phi2} + t_{r\phi1} \geqslant 480$ns.



4. The following are relevant when interfacing the INS8080A to devices having V_{IH} = 3.3V:
   a) Maximum output rise time from 0.8V 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 0.6ns/pF if $C_L > C_{SPEC}$, subtract 0.3ns/pF (from modified delay) if $C_L < C_{SPEC}$.
5. $t_{AW} = 2t_{CY} - t_{D3} - t_{r\phi2} - 140$ns.
6. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi2} - 170$ns.
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.

INS8080A CPU functional block diagram



INS8080A functional pin definition

The following describes the function of all of the INS8080A input/output pins. Some of these descriptions reference internal timing periods.

INPUT SIGNALS

Ready: When high (logic 1), indicates that valid memory or input data are available to the CPU on the INS8080A data bus. The READY signal is used to synchronize the CPU with slower memory or input/output devices. If the INS8080A does not receive a high READY input after sending out an address to memory or an input/output device, the INS8080A enters a WAIT mode for as long as the READY input remains low (logic 0). The CPU may also be single stepped by the use of the READY signal.

Hold: When high, requests that the CPU enter the HOLD mode. When the CPU is in the HOLD mode, the CPU address and data buses both will be in the high-impedance state. The HOLD mode allows an external device to gain control of the INS8080A address and data buses immediately following the completion of the current machine cycle by the CPU. The CPU acknowledges the HOLD mode via the HOLD ACKNOWLEDGE (HLDA) output line. The HOLD request is recognized under the following conditions:

- The CPU is in the HALT mode.
- The READY signal is active and the CPU is in the t_2 or t_W state.

Interrupt (INT) Request: When high, the CPU recognizes an interrupt request on this line after completing the current instruction or while in the HALT mode. An interrupt request is not honored if the CPU is in the HOLD mode (HLDA = logic 1) or the Interrupt Enable Flip-flop is reset (INTE = logic 0).

Reset: When activated (high) for a minimum of three clock periods, the content of the Program Counter is cleared and the Interrupt Enable and Hold Acknowledge Flip-flops are reset. Following a RESET, program execution starts at memory location 0. It should be noted that the status flags, accumulator, stack pointer, and registers are not cleared during the RESET sequence.

ϕ_1 and ϕ_2 Clocks: Two non-TTL compatible clock phases which provide nonoverlapping timing references for internal storage elements and logic circuits of the CPU.

+12 Volts: V_{DD} Supply.

+5 Volts: V_{CC} Supply.

-5 Volts: V_{BB} Supply.

Ground: V_{SS} (0 volt) reference.

OUTPUT SIGNALS

Synchronizing (SYNC) Signal: When activated (high), the beginning of a new machine cycle is indicated and the status word is outputted on the Data Bus.

Address (A_{15} – A_0) Bus: This bus comprises sixteen TRI-STATE output lines. The bus provides the address to memory (up to 65k bytes) or denotes the input/output device number for up to 256 input and 256 output peripherals.

Wait: When high, acknowledges that the CPU is in the WAIT mode.

Write ($\overline{WR}$): When low, the data on the data bus are stable for WRITE memory or output operation.

Hold Acknowledge (HLDA): Goes high in response to a logic 1 on the HOLD line and indicates that the data and address bus will go to the high-impedance state. The HLDA begins at one of the following times:

- The t_3 state of a READ memory input operation.
- The clock period following the t_3 state of a WRITE memory output operation.

In both cases, the HLDA signal starts after the rising edge of the ϕ_1 clock, and high impedance occurs after the rising edge of the ϕ_2 clock.

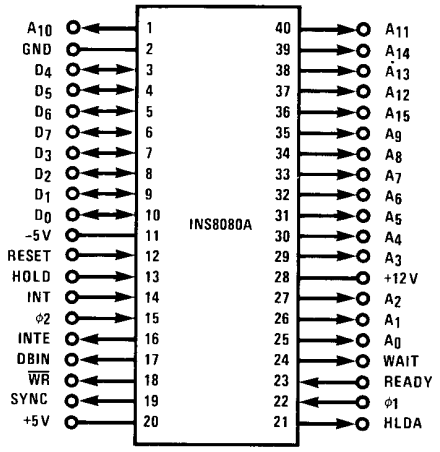
Interrupt Enable (INTE): Indicates the content of the internal Interrupt Enable Flip-flop. The Enable and Disable Interrupt (EI and DI) Instructions cause the Interrupt Enable Flip-flop to be set and reset, respectively. When the flip-flop is reset (INTE = logic 0), it inhibits interrupts from being accepted by the CPU. In addition, the Interrupt Enable Flip-flop is automatically reset (thereby disabling further interrupts) at the t_1 state of the instruction fetch cycle, when an interrupt is accepted; it is also reset by the RESET Signal.

Data Bus In (DBIN): When high, indicates to external circuits that the data bus is in the input mode. The DBIN Signal should be used to gate data from memory or an I/O device onto the Data Bus.

INPUT/OUTPUT SIGNALS

Data ($D_7 - D_0$) Bus: This bus comprises eight TRI-STATE input/output lines. The bus provides bidirectional communication between the CPU, memory, and input/output devices for instructions and data transfers. A status word (which describes the current machine cycle) is also outputted on the data bus during the first state of each machine cycle (SYNC = logic 1).

pin configuration



8080A status

Instructions for the 8080A require from one to five machine cycles for complete execution. The 8080A sends out 8 bits of status information on the data bus at the beginning of each machine cycle (during SYNC time). The following table defines the status information.

Status Information Definition

Symbols	Data Bus Bit	Definition
INTA*	D_0	Acknowledge signal for INTERRUPT request. Signal should be used to gate a restart instruction onto the data bus when DBIN is active.
$\overline{WO}$	D_1	Indicates that the operation in the current machine cycle will be a WRITE memory or OUTPUT function (WO = 0). Otherwise, a READ memory or INPUT operation will be executed.
STACK	D_2	Indicates that the address bus holds the pushdown stack address from the Stack Pointer.
HLTA	D_3	Acknowledge signal for HALT Instruction.
OUT	D_4	Indicates that the address bus contains the address of an output device and the data bus will contain the output data when WR is active.
M_1	D_5	Provides a signal to indicate that the CPU is in the fetch cycle for the first byte of an instuction.
INP*	D_6	Indicates that the address bus contains the address of an input device and the input data should be placed on the data bus when DBIN is active.
MEMR*	D_7	Designates that the data bus will be used for memory read data.

*These three status bits can be used to control the flow of data onto the INS8080A data bus.

Status Word Chart

Machine Cycle	Type	Data Bus Bit							
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Instruction Fetch	1	1	0	1	0	0	0	1	0
Memory Read	2	1	0	0	0	0	0	1	0
Memory Write	3	0	0	0	0	0	0	0	0
Stack Read	4	1	0	0	0	0	1	1	0
Stack Write	5	0	0	0	0	0	1	0	0
Input Read	6	0	1	0	0	0	0	1	0
Output Write	7	0	0	0	1	0	0	0	0
Interrupt Acknowledge	8	0	0	1	0	0	0	1	1
Halt Acknowledge	9	1	0	0	0	1	0	1	0
Interrupt Acknowledge While Halt	10	0	0	1	0	1	0	1	1

instruction set

Mnemonic		Description	Operation	Op Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	No. of Bytes	No. of Machine (M) Cycles	No. of States (T)	Condition Flags S	Z	AC	P	CY
DATA TRANSFER GROUP																			
LDA		Load Accumulator Direct	(A) ← ((byte 3)(byte 2))	0	0	1	1	1	0	1	0	3	4	13					
LDAX	B	Load Accumulator Indirect	(A) ← ((B)(C))	0	0	0	0	1	0	1	0	1	2	7					
LDAX	D	Load Accumulator Indirect	(A) ← ((D)(E))	0	0	0	1	1	0	1	0	1	2	7					
LHLD		Load H and L Direct	(L) ← ((byte 3)(byte 2)) (H) ← ((byte 3)(byte 2) + 1)	0	0	1	0	1	0	1	0	3	5	16					
LXI	B	Load Immediate, Registers B and C	(B) ← (byte 3) (C) ← (byte 2)	0	0	0	0	0	0	0	1	3	3	10					
LXI	D	Load Immediate, Registers D and E	(D) ← (byte 3) (E) ← (byte 2)	0	0	0	1	0	0	0	1	3	3	10					
LXI	H	Load Immediate, Registers H and L	(H) ← (byte 3) (L) ← (byte 2)	0	0	1	0	0	0	0	1	3	3	10					
LXI	SP	Load Immediate, Stack Pointer	(SPH) ← (byte 3) (SPL) ← (byte 2)	0	0	1	1	0	0	0	1	3	3	10					
MOV	M, r	Move to Memory	((H)(L)) ← (r)	0	1	1	1	0	*S*	*S*	*S*	1	2	7	(Flags Not Affected)				
MOV	r, M	Move from Memory	(r) ← ((H)(L))	0	1	*D*	*D*	*D*	1	1	0	1	2	7					
MOV	r1, r2	Move Registers	(r1) ← (r2)	0	1	*D*	*D*	*D*	*S*	*S*	*S*	1	1	5					
MVI	M	Move to Memory Immediate	((H)(L)) ← (byte 2)	0	0	1	1	0	1	1	0	2	3	10					
MVI	r	Move Immediate	(r) ← (byte 2)	0	0	*D*	*D*	*D*	1	1	0	2	2	7					
SHLD		Store H and L Direct	((byte 3)(byte 2)) ← (L) ((byte 3)(byte 2) + 1) ← (H)	0	0	1	0	0	0	1	0	3	5	16					
STA		Store Accumulator Direct	((byte 3)(byte 2)) ← (A)	0	0	1	1	0	0	1	0	3	4	13					
STAX	B	Store Accumulator Indirect	((B)(C)) ← (A)	0	0	0	0	0	0	1	0	1	2	7					
STAX	D	Store Accumulator Indirect	((D)(E)) ← (A)	0	0	0	1	0	0	1	0	1	2	7					
XCHG		Exchange H and L with D and E	(H) ↔ (D) (L) ↔ (E)	1	1	1	0	1	0	1	1	1	1	4					
ARITHMETIC GROUP																			
ACI		Add Immediate with Carry	(A) ← (A) + (byte 2) + (CY)	1	1	0	0	1	1	1	0	2	2	7	↕	↕	↕	↕	↕
ADC	M	Add Memory with Carry	(A) ← (A) + ((H)(L)) + (CY)	1	0	0	0	1	1	1	0	1	2	7	↕	↕	↕	↕	↕
ADC	r	Add Register with Carry	(A) ← (A) + (r) + (CY)	1	0	0	0	1	*S*	*S*	*S*	1	1	4	↕	↕	↕	↕	↕
ADD	M	Add Memory	(A) ← (A) + ((H)(L))	1	0	0	0	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
ADD	r	Add Register	(A) ← (A) + (r)	1	0	0	0	0	*S*	*S*	*S*	1	1	4	↕	↕	↕	↕	↕
ADI		Add Immediate	(A) ← (A) + (byte 2)	1	1	0	0	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
DAA		Decimal Adjust Accumulator	8-bit number in Accumulator is converted to two 4-bit BCD digits	0	0	1	0	0	1	1	1	1	1	4	↕	↕	↕	↕	↕
DAD	B	Add B and C to H and L	(H)(L) ← (H)(L) + (B)(C)	0	0	0	0	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	D	Add D and E to H and L	(H)(L) ← (H)(L) + (D)(E)	0	0	0	1	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	H	Add H and L to H and L	(H)(L) ← (H)(L) + (H)(L)	0	0	1	0	1	0	0	1	1	3	10	·	·	·	·	↕
DAD	SP	Add Stack Pointer to H and L	(H)(L) ← (H)(L) + (SP)	0	0	1	1	1	0	0	1	1	3	10	·	·	·	·	↕
DCR	M	Decrement Memory	((H)(L)) ← ((H)(L)) - 1	0	0	1	1	0	1	0	1	1	3	10	↕	↕	↕	↕	·
DCR	r	Decrement Register	(r) ← (r) - 1	0	0	*D*	*D*	*D*	1	0	1	1	1	5	↕	↕	↕	↕	·
DCX	B	Decrement Registers B and C	(B)(C) ← (B)(C) - 1	0	0	0	0	1	0	1	1	1	1	5	·	·	·	·	·
DCX	D	Decrement Registers D and E	(D)(E) ← (D)(E) - 1	0	0	0	1	1	0	1	1	1	1	5	·	·	·	·	·
DCX	H	Decrement Registers H and L	(H)(L) ← (H)(L) - 1	0	0	1	0	1	0	1	1	1	1	5	·	·	·	·	·
DCX	SP	Decrement Stack Pointer	(SP) ← (SP) - 1	0	0	1	1	1	0	1	1	1	1	5	·	·	·	·	·
INR	M	Increment Memory	((H)(L)) ← ((H)(L)) + 1	0	0	1	1	0	1	0	0	1	3	10	↕	↕	↕	↕	·
INR	r	Increment Register	(r) ← (r) + 1	0	0	*D*	*D*	*D*	1	0	0	1	1	5	↕	↕	↕	↕	·
INX	B	Increment Registers B and C	(B)(C) ← (B)(C) + 1	0	0	0	0	0	0	1	1	1	1	5	·	·	·	·	·
INX	D	Increment Registers D and E	(D)(E) ← (D)(E) + 1	0	0	0	1	0	0	1	1	1	1	5	·	·	·	·	·
INX	H	Increment Registers H and L	(H)(L) ← (H)(L) + 1	0	0	1	0	0	0	1	1	1	1	5	·	·	·	·	·
INX	SP	Increment Stack Pointer	(SP) ← (SP) + 1	0	0	1	1	0	0	1	1	1	1	5	·	·	·	·	·
SBB	M	Subtract Memory with Borrow	(A) ← (A) - ((H)(L)) - (CY)	1	0	0	1	1	1	1	0	1	2	7	↕	↕	↕	↕	↕
SBB	r	Subtract Register with Borrow	(A) ← (A) - (r) - (CY)	1	0	0	1	1	*S*	*S*	*S*	1	1	4	↕	↕	↕	↕	↕
SBI		Subtract Immediate with Borrow	(A) ← (A) - (byte 2) - (CY)	1	1	0	1	1	1	1	0	2	2	7	↕	↕	↕	↕	↕
SUB	M	Subtract Memory	(A) ← (A) - ((H)(L))	1	0	0	1	0	1	1	0	1	2	7	↕	↕	↕	↕	↕
SUB	r	Subtract Register	(A) ← (A) - (r)	1	0	0	1	0	*S*	*S*	*S*	1	1	4	↕	↕	↕	↕	↕
SUI		Subtract Immediate	(A) ← (A) - (byte 2)	1	1	0	1	0	1	1	0	2	2	7	↕	↕	↕	↕	↕
LOGICAL GROUP																			
ANA	M	AND Memory	(A) ← (A) ∧ ((H)(L))	1	0	1	0	0	1	1	0	1	2	7	↕	↕	↕[d]	↕	0
ANA	r	AND Register	(A) ← (A) ∧ (r)	1	0	1	0	0	*S*	*S*	*S*	1	1	4	↕	↕	↕[d]	↕	0
ANI		AND Immediate	(A) ← (A) ∧ (byte 2)	1	1	1	0	0	1	1	0	2	2	7	↕	↕	↕[d]	↕	0
CMA		Complement Accumulator	(A) ← ($\overline{A}$)	0	0	1	0	1	1	1	1	1	1	4	·	·	·	·	·
CMC		Complement Carry	(CY) ← ($\overline{CY}$)	0	0	1	1	1	1	1	1	1	1	4	·	·	·	·	↕
CMP	M	Compare Memory	(A) — ((H)(L))	1	0	1	1	1	1	1	0	1	2	7	↕	↕[a]	↕	↕	↕[a]
CMP	r	Compare Register	(A) — (r)	1	0	1	1	1	*S*	*S*	*S*	1	1	4	↕	↕[b]	↕	↕	↕[b]
CPI		Compare Immediate	(A) — (byte 2)	1	1	1	1	1	1	1	0	2	2	7	↕	↕[c]	↕	↕	↕[c]
ORA	M	OR Memory	(A) ← (A) ∨ ((H)(L))	1	0	1	1	0	1	1	0	1	2	7	↕	↕	0	↕	0
ORA	r	OR Register	(A) ← (A) ∨ (r)	1	0	1	1	0	*S*	*S*	*S*	1	1	4	↕	↕	0	↕	0
ORI		OR Immediate	(A) ← (A) ∨ (byte 2)	1	1	1	1	0	1	1	0	2	2	7	↕	↕	0	↕	0
RAL		Rotate Left through Carry	$(A_{n+1}) \leftarrow (A_n)$; (CY) ← (A_7) (A_0) ← (CY)	0	0	0	1	0	1	1	1	1	1	4	·	·	·	·	↕
RAR		Rotate Right through Carry	$(A_n) \leftarrow (A_{n+1})$; (CY) ← (A_0) (A_7) ← (CY)	0	0	0	1	1	1	1	1	1	1	4	·	·	·	·	↕
RLC		Rotate Left	$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$ (CY) ← (A_7)	0	0	0	0	0	1	1	1	1	1	4	·	·	·	·	↕
RRC		Rotate Right	$(A_n) \leftarrow (A_{n-1})$; $(A_7) \leftarrow (A_0)$ (CY) ← (A_0)	0	0	0	0	1	1	1	1	1	1	4	·	·	·	·	↕
STC		Set Carry	(CY) ← 1	0	0	1	1	0	1	1	1	1	1	4	·	·	·	·	1
XRA	M	Exclusive OR Memory	(A) ← (A) ∀ ((H)(L))	1	0	1	0	1	1	1	0	1	2	7	↕	↕	0	↕	0
XRA	r	Exclusive OR Register	(A) ← (A) ∀ (r)	1	0	1	0	1	*S*	*S*	*S*	1	1	4	↕	↕	0	↕	0
XRI		Exclusive OR Immediate	(A) ← (A) ∀ (byte 2)	1	1	1	0	1	1	1	0	2	2	7	↕	↕	0	↕	0

Notes:

a. Z = 1 if (A) = (H)(L); CY = 1 if (A) < (H)(L)

b. Z = 1 if (A) = (r); CY = 1 if (A) < (r)

c. Z = 1 if (A) = (byte 2); CY = 1 if (A) < (byte 2)

d. As if an arithmetic operation were performed.

instruction set (cont'd.)

Mnemonic	Description	Operation	Op Code D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	No. of Bytes	No. of Machine (M) Cycles	No. of States (T)	Condition Flags S	Z	AC	P	CY
BRANCH GROUP																		
CALL	Call Unconditional	((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	0	0	1	1	0	1	3	5	17					
CC	Call on Carry	If CY = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	0	1	1	1	0	0	3	3/5	11/17					
CM	Call on Minus	If S = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	1	1	1	1	0	0	3	3/5	11/17					
CNC	Call on No Carry	If CY = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	0	1	0	1	0	0	3	3/5	11/17					
CNZ	Call on Not Zero	If Z = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	0	0	0	1	0	0	3	3/5	11/17	(Flags Not Affected)				
CP	Call on Positive	If S = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	1	1	0	1	0	0	3	3/5	11/17					
CPE	Call on Parity Even	If P = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	1	0	1	1	0	0	3	3/5	11/17					
CPO	Call on Parity Odd	If P = 0, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	1	0	0	1	0	0	3	3/5	11/17					
CZ	Call on Zero	If Z = 1, ((SP) - 1) ← (PCH) ((SP) - 2) ← (PCL) (SP) ← (SP) - 2 (PC) ← (byte 3)(byte 2)	1	1	0	0	1	1	0	0	3	3/5	11/17					
JC	Jump on Carry	If CY = 1, (PC) ← (byte 3)(byte 2)	1	1	0	1	1	0	1	0	3	3	10					
JM	Jump on Minus	If S = 1, (PC) ← (byte 3)(byte 2)	1	1	1	1	1	0	1	0	3	3	10					
JMP	Jump Unconditional	(PC) ← (byte 3)(byte 2)	1	1	0	0	0	0	1	1	3	3	10					
JNC	Jump on No Carry	If CY = 0, (PC) ← (byte 3)(byte 2)	1	1	0	1	0	0	1	0	3	3	10					
JNZ	Jump on Not Zero	If Z = 0, (PC) ← (byte 3)(byte 2)	1	1	0	0	0	0	1	0	3	3	10					
JP	Jump on Positive	If S = 0, (PC) ← (byte 3)(byte 2)	1	1	1	1	0	0	1	0	3	3	10					
JPE	Jump on Parity Even	If P = 1, (PC) ← (byte 3)(byte 2)	1	1	1	0	1	0	1	0	3	3	10					
JPO	Jump on Parity Odd	If P = 0, (PC) ← (byte 3)(byte 2)	1	1	1	0	0	0	1	0	3	3	10					
JZ	Jump on Zero	If Z = 1, (PC) ← (byte 3)(byte 2)	1	1	0	0	1	0	1	0	3	3	10					
PCHL	H and L to Program Counter	(PCH) ← (H) (PCL) ← (L)	1	1	1	0	1	0	0	1	1	1	5					
RC	Return on Carry	If CY = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	1	0	0	0	1	1/3	5/11					
RET	Return	(PCL) ← ((SP)); (PCH) ← ((SP) + 1); (SP) ← (SP) + 2;	1	1	0	0	1	0	0	1	1	3	10					
RM	Return on Minus	If S = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	1	0	0	0	1	1/3	5/11					
RNC	Return on No Carry	If CY = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	0	0	0	0	1	1/3	5/11					
RNZ	Return on Not Zero	If Z = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	0	0	0	0	1	1/3	5/11					
RP	Return on Positive	If S = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	0	0	0	0	1	1/3	5/11					
RPE	Return on Parity Even	If P = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	1	0	0	0	1	1/3	5/11					
RPO	Return on Parity Odd	If P = 0, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	0	0	0	0	1	1/3	5/11					

instruction set (cont'd.)

Mnemonic	Description	Operation	Op Code								No. of Bytes	No. of Machine (M) Cycles	No. of States (T)	Condition Flags				
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				S	Z	AC	P	CY
BRANCH GROUP (continued)																		
RST	Restart	((SP) − 1) ← (PCH) ((SP) − 2) ← (PCL) (SP) ← (SP) − 2 (PC) ← 8 * (NNN)	1	1	*N*	*N*	*N*	1	1	1	1	3	11					
RZ	Return on Zero	If Z = 1, (PCL) ← ((SP)) (PCH) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	1	0	0	0	1	1/3	5/11					
STACK, I/O, AND MACHINE CONTROL GROUP																		
DI	Disable Interrupts	The Interrupt system is disabled following the execution of the DI instruction.	1	1	1	1	0	0	1	1	1	1	4	·	·	·	·	·
EI	Enable Interrupts	The interrupt system is enabled following the execution of next instruction.	1	1	1	1	1	0	1	1	1	1	4	·	·	·	·	·
HLT	Halt	Processor is stopped; registers and flags are unaffected.	0	1	1	1	0	1	1	0	1	1	7	·	·	·	·	·
IN	Input	(A) ← (data)	1	1	0	1	1	0	1	1	2	3	10	·	·	·	·	·
NOP	No Operation	No operation is performed; registers and flags are unaffected.	0	0	0	0	0	0	0	0	1	1	4	·	·	·	·	·
OUT	Output	(data) ← (A)	1	1	0	1	0	0	1	1	2	3	10	·	·	·	·	·
POP B	Pop Registers B and C off Stack	(C) ← ((SP)) (B) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	0	0	0	0	1	1	3	10	·	·	·	·	·
POP D	Pop Registers D and E off Stack	(E) ← ((SP)) (D) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	0	1	0	0	0	1	1	3	10	·	·	·	·	·
POP H	Pop Registers H and L off Stack	(L) ← ((SP)) (H) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	0	0	0	0	1	1	3	10	·	·	·	·	·
POP PSW	Pop Accumulator and Flags off Stack	(CY) ← $((SP))_0$ (P) ← $((SP))_2$ (AC) ← $((SP))_4$ (Z) ← $((SP))_6$ (S) ← $((SP))_7$ (A) ← ((SP) + 1) (SP) ← (SP) + 2	1	1	1	1	0	0	0	1	1	3	10	‡	‡	‡	‡	‡
PUSH B	Push Registers B and C on Stack	((SP) − 1) ← (B) ((SP) − 2) ← (C) (SP) ← (SP) − 2	1	1	0	0	0	1	0	1	1	3	11	·	·	·	·	·
PUSH D	Push Registers D and E on Stack	((SP) − 1) ← (D) ((SP) − 2) ← (E) (SP) ← (SP) − 2	1	1	0	1	0	1	0	1	1	3	11	·	·	·	·	·
PUSH H	Push Registers H and L on Stack	((SP) − 1) ← (H) ((SP) − 2) ← (L) (SP) ← (SP) − 2	1	1	1	0	0	1	0	1	1	3	11	·	·	·	·	·
PUSH PSW	Push Accumulator and Flags on Stack	((SP) − 1) ← (A) $((SP) - 2)_0$ ← (CY) $((SP) - 2)_1$ ← 1 $((SP) - 2)_2$ ← (P) $((SP) - 2)_3$ ← 0 $((SP) - 2)_4$ ← (AC) $((SP) - 2)_5$ ← 0 $((SP) - 2)_6$ ← (Z) $((SP) - 2)_7$ ← (S) (SP) ← (SP) − 2	1	1	1	1	0	1	0	1	1	3	11	·	·	·	·	·
SPHL	Move H and L to Stack Pointer	(SP) ← (H)(L)	1	1	1	1	1	0	0	1	1	1	5	·	·	·	·	·
XTHL	Exchange Top of Stack with H and L	(L) ↔ ((SP)) (H) ↔ ((SP) + 1)	1	1	1	0	0	0	1	1	1	5	18	·	·	·	·	·

condition flags and standard rules

There are five condition flags associated with the execution of instructions on the INS8080A. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and each flag is represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1, "reset" by forcing the bit to 0. The bit positions of the flags are indicated in the PUSH and POP PSW instructions.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

ZERO (Z): If the result of an instruction has the value 0, this flag is set; otherwise, it is reset.

SIGN (S): If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise, it is reset.

PARITY (P): If the modulo 2 sum of the bits of the result of the operation is 0 (that is, if the result has even parity), this flag is set; otherwise, it is reset (that is, if the result has odd parity).

CARRY (CY): If the instruction resulted in a carry (from addition) or a borrow (from subtraction or a comparison) out of the high-order bit, this flag is set; otherwise, it is reset.

AUXILIARY CARRY (AC): If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise, it is reset. This flag is affected by single-precision additions, subtractions, increments, decrements, comparisons, and logical operations; however, AC is used principally with additions and increments preceding a DAA (Decimal Adjust Accumulator) Instruction.

symbols and abbreviations

The following symbols and abbreviations are used in the subsequent description of the INS8080A instructions:

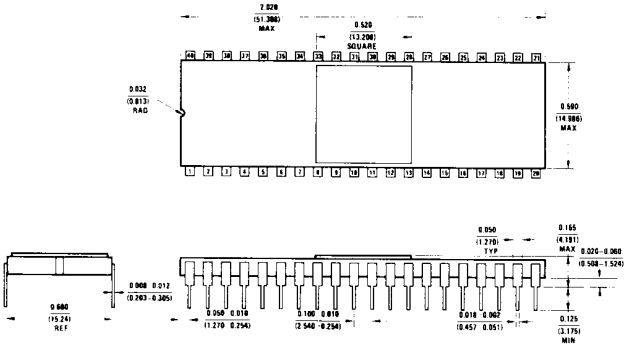
Symbols	Meaning
A	Register A (Accumulator)
B	Register B
C	Register C
D	Register D
H	Register H
L	Register L
DDD, SSS	The bit pattern designating one of the registers A, B, C, D, E, H, L (DDD = destination, SSS = source):

DDD or SSS	Register Name
111	A
000	B
001	C
010	D
011	E
100	H
101	L

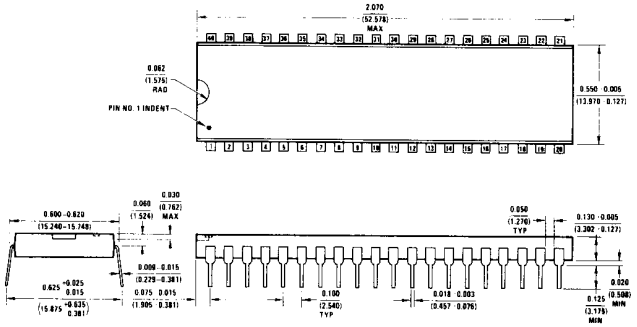
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r, r1, r2	One of the registers A, B, C, D, E, H, L

Symbols	Meaning
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively.)
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively.)
( )	The contents of the memory location or registers enclosed in the parentheses
←	"Is replaced by"
∧	Logical AND
∀	Exclusive OR
∨	Inclusive OR
+	Addition
−	Twos complement subtraction
*	Multiplication
↔	"Exchange"
—	The ones complement (for example, $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively
•	"Not affected"
0	"Reset"
1	"Set"
x	Unknown
↕	Flags affected according to Standard Rules, except as noted.

physical dimensions



40-Lead Ceramic Dual-in-Line Package (D)
Order Number INS8080AD



40-Lead Plastic Dual-in-Line Package (N)
Order Number INS8080N

obsolete - see price list

Manufactured under one or more of the following U.S. patents: 3083262, 3189758, 3231797, 3303356, 3317671, 3323071, 3381071, 3408542, 3421025, 3426423, 3440498, 3518750, 3519897, 3557431, 3560765, 3566218, 3571630, 3575609, 3579059, 3593069, 3597640, 3607469, 3617859, 3631312, 3633052, 3638131, 3648071, 3651565, 3693248.

National Semiconductor Corporation
2900 Semiconductor Drive, Santa Clara, California 95051, (408) 737-5000/TWX (910) 339-9240
National Semiconductor GmbH
808 Fuerstenfeldbruck, Industriestrasse 10, West Germany, Tele. (08141) 1371/Telex 05-27649
National Semiconductor (UK) Ltd.
Larkfield Industrial Estate, Greenock, Scotland, Tele. (0475) 33251/Telex 778-632