

DATATEKNIK ApS

Microprocessorsystem

ID-7000

System description

September 1975

Table of Contents.

1.	<u>Introduction</u>	1
2.	<u>The bus</u>	1
3.	<u>Modules</u>	5
3.1.	ID-7000 CPU Module	5
3.2.	ID-7003 Interrupt Priority Module	5
3.3.	ID-7009 Multiple DMA Module	6
3.4.	ID-7005 Manual Control Module	7
3.5.	ID-7010 Bus Extender Module	7
3.6.	ID-7011 PROM-Writer Kit	7
3.7.	ID-7002 Static RAM Module	8
3.8.	ID-7007 Dynamic RAM Module	8
3.9.	ID-7008 Refresh Module	8
3.10.	ID-7001 PROM Module	8
3.11.	ID-7004 Async. Communication Module	9
3.12.	ID-7012 Sync. Transmitter Module	9
3.13.	ID-7013 Sync. Receiver Module	9
3.14.	ID-7014 Flexible Disc Module	10
3.15.	ID-7006 Papertape Reader Module	10
3.16.	ID-7015 Interval Timer Module	10
3.17.	ID-7016 A/D-Conversion Module	11
3.18.	ID-7017 D/A-Conversion Module	11
3.19.	ID-7018 Standard I/O Module	11
4.	<u>Basic Evaluation Software</u>	12
4.1.	ID-7000 Debugger/Monitor	12
4.2.	ID-7000 Assembler	14
4.3.	ID-7000 Text Editor	14
	Appendix 1: 8080A Central Processor, Data Sheet	

1. Introduction.

ID-7000- MICROPROCESSOR SYSTEM is a modular microprocessor system based on the INTEL 8080 microprocessor. The CPU-module, memory modules and I/O-modules communicate via a common bus-system. The different modules can be placed anywhere on the bus. Addresses for memory- and I/O modules are determined by switch registers situated on the modules. The modules are built up on printed-wire-boards 144mm x 138mm. The connection to the bus is made through a loop-pole edge-card connector. Connection to the environment is by means of a 64-pole edge-card connector placed at the top of the module. The bus consists of a motherboard placed at the bottom of the cabinet. A maximum of 19 modules can be plugged into the bus. The cabinet contains power supplies to the system.

The system is intended to be used in user defined applications in cases where the use of microcomputers is reasonable, and where the number of units prohibits a more integrated design. Even in such cases, the modular microprocessor system can be useful in prototyping the equipment.

The ID-7000 microprocessor system has been developed in co-operation with the Institute of Computer Science, at the Technical University of Denmark. It is a further development on a previous 8080 microprocessor system made by the Institute.*)

2. The bus.

The bus carries the following signals and supplies:

ADR(15:0): 16 bit three-state bus, holding addresses for memory- and I/O-modules.

ADR?
D(7:0): 24 } 8 bit three-state bus, used for data transmission between CPU and memory-/input-output modules and between memory- and input-output modules.

\overline{IR} (7:0): 8 (active-low) signals for interrupt requests to the priority interrupt-module.

These signals are open-collector to enable several I/O-modules to share the same interrupt request line.

\overline{RR} : Low-active read-request signal, indicating a CPU - (or I/O-unit with DMA-status) request for fetching data or instruction from memory addressed by ADR(15:0).

\overline{WR} : Low-active write-request signal, indicating a CPU - (or I/O-unit with DMA-status) request for transmitting data to memory addressed by ADR(15:0).

\overline{INR} : Low-active input-request signal, indicating a CPU request for fetching data from I/O-unit addressed by ADR(7:0)

*) This system is described in the publication: "ID-8000, Et mikro-processorsystem" part 1 and 2. Edited by Ole Lading.

OUTR: Low-active output-request signal, indicating a CPU request for transmitting data to I/O-unit addressed by ADR(7:0).

AFBR: Low-active interrupt-request signal, indicating a CPU request for fetching a RST instruction from the Interrupt Priority module (if available).

By means of these 5 control signals, the CPU indicates which type of machine-cycle is going on. The control signals are low-active open-collector signals. This permits modules other than the CPU modules (DMA I/O-Modules), to generate the signals. The signals are timed in such a way that they can be used directly for gating and clocking purposes by memory-, interrupt- and I/O-modules.

HOLDR: Low-active open-collector signal from a DMA-requesting I/O-module. HOLDR is answered from the CPU-module by HOLDA, when the CPU is in hold-state. In installations with multiple DMA capability, the HOLDR-signal is generated by the Multiple DMA module.

HOLDA: Low-active, open-collector signal from the CPU-module, indicating the CPU hold state. The signal can be used by a DMA-I/O-unit, or, in installation with multiple DMA-capability, by the Multiple DMA module.

ABUSDISABLE: This low-active input signal to the CPU-module can be used by other modules (DMA I/O-modules etc.) to release the CPU address buffers.

DBUSDISABLE: This low-active input signal to the CPU-module can be used by other modules (DMA I/O-module etc.) to release the CPU data buffers.

The above mentioned 4 signals are only used in installations including DMA I/O-modules. In installations serving more than one DMA I/O-module, a Multiple DMA module is needed, and the following two buses are used:

DMAREQ(7:0): 8 (active-low) signals for DMA requests to the Multiple DMA module.

DMAACK(7:0): 8 (active-low) signals from the Multiple DMA module. These signals allow the requesting modules to control the buses and execute a memory read or write cycle.

VENT: Low-active, open-collector control signal to the CPU module. The signal can be used to force the CPU into wait state. It may be generated by memory and I/O modules to extend the current machine cycle.

RESET: This low-active bus line is used to reset I/O-modules and CPU-modules. After reset the CPU will start program execution in memory location 0. The RESET signal is generated by the power supply during power-up conditions or by activating a manual push-button.

+5V: +5 volt power supply for TTL- and MOS-logic.

+12V: +12 volt power supply for MOS-logic.

+5V: + 5 volt power supply for MOS-logic.

GND: , Common ground for the system.

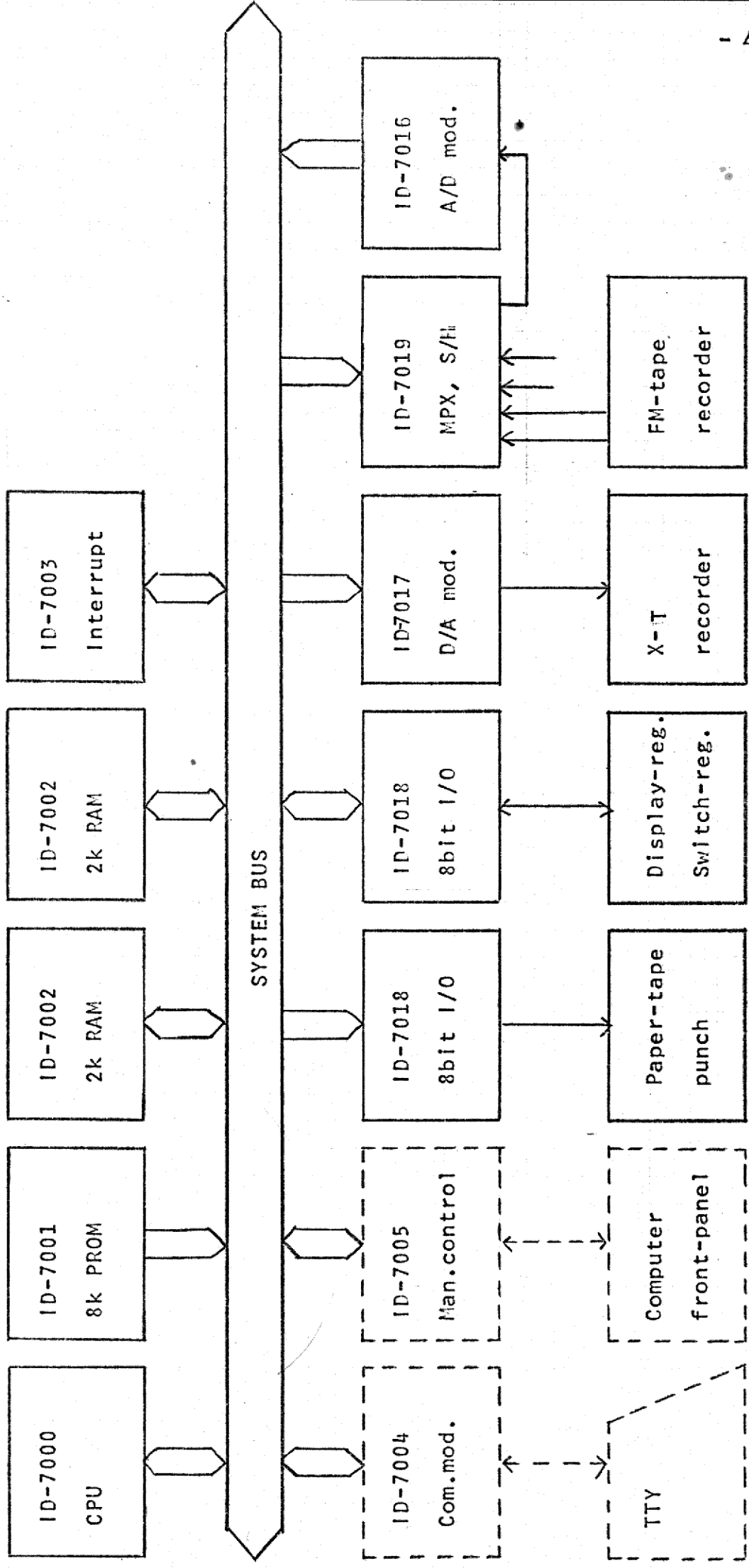
=====

In addition to the above mentioned general buses and control signals, the bus system contains some special signals used for test purposes and for communication between the CPU-module and some of the special purpose system modules. (Interrupt Priority module, Manual Control module, Multiple DMA module etc.). These signals are explained in the detailed descriptions of the different modules. In table I is shown the allocation of pin-numbers in the motherboard connector to the different signals.

Motherboard connections
ID-7000 Microprocessor system.

Side A (back)	Pin no.	Side B (component side)
+5V	1	GND
+5V	2	GND
ADR(1)	3	<u>MEMDISABLE</u>
(1)	4	HLTA
(2)	5	STACK
(3)	6	INTE
(4)	7	
(5)	8	
(6)	9	
(7)	10	
(8)	11	<u>IR(o)</u>
(9)	12	(1)
(10)	13	(2)
(11)	14	(3)
(12)	15	(4)
(13)	16	(5)
(14)	17	(6)
(15)	18	(7)
	19	
<u>WR</u>	20	
<u>OUTR</u>	21	
<u>INR</u>	22	
<u>RR</u>	23	
<u>AFBR</u>	24	
	25	
<u>ABUSDISABLE</u>	26	<u>DBUSDISABLE</u>
<u>WG</u>	27	<u>HOLDR</u>
<u>VENT</u>	28	<u>HOLDA</u>
<u>INT</u>	29	<u>DMAACK(o)</u>
	30	(1)
	31	(2)
D(o)	32	(3)
(1)	33	(4)
(2)	34	(5)
(3)	35	(6)
(4)	36	(7)
(5)	37	<u>DMAREQ(o)</u>
(6)	38	(1)
(7)	39	(2)
	40	(3)
	41	(4)
	42	(5)
	43	(6)
	44	(7)
<u>RESET</u>	45	<u>POWERLOSS</u>
<u>SYNC</u>	46	M1
<u>Ø1</u>	47	<u>Ø2</u>
<u>16M</u>	48	<u>16M</u>
GND	49	+12V
GND	50	+5V

TABLE I



Typical application of the ID-7000 Microprocessor System in medical electronics:
Equipment for automatic analysis of electromyographic recordings.

3. Modules.

The basic part of the system is the ID-7000 CPU module. The functions of the CPU module can be expanded by use of some special purposes modules (ID-7003 Interrupt Priority module and ID-7009 Multiple DMA module). Another special purpose module is the ID-7005 Manual control module, used for system evaluation and test in connection with the ID test panel. The ID-7010 Bus extender module is used for extension of the bus, to an external galvanic isolated low-speed bus. For PROM-programming, the ID-7011 PROM-writer module can be used. The more general modules include RAM- and PROM-memories, A/D- and D/A-modules, modules for asynchronous and synchronous data communication, modules for connection of paper-tape reader, floppy disc drives etc. A wire-wrap board is available for custom-designed modules and special modules can be developed upon request. This section contains a short description of the different modules. Further description and programming information is given in the manuals on the different modules.

3.1 ID-7000 CPU module.

This module contains the INTEL 8080 microprocessor with associated logic and buffers. This includes:

- Clock generator for clock phases for 8080 and associated logic.
- Three-state buffers for addresses (ADR(15:0)).
- Three-state two-way buffers for data (D(7:0)).
- Latch for 8080-status information.
- Logic for generating the request signals. \overline{RR} , \overline{WR} , \overline{INR} , \overline{OUTR} and \overline{AFBR} .
- Buffers for control signals to and from the 8080 microprocessor.

The CPU module is self-contained, i.e. it is able to run a program together with a single memory-module.

In complex applications, the performance can be expanded by use of special modules, to include multiple interrupt requests and multiple DMA requests.

3.2 ID-7003 Interrupt priority module.

This module contains logic for making priority of 8 incoming interrupt request lines IR(7:0) from I/O modules. In the module is included an 8-bit interrupt mask register IM(7:0) for selective activating/deactivating of interrupt levels.

The interrupt mask register is loaded and stored by means of normal I/O instructions. When at least one interrupt line corresponding to an activated interrupt level is on, the Interrupt priority module transmits an interrupt to the CPU module. When this is answered with the \overline{AFBR} signal, indicating an interrupt cycle going on, the 7003-module transmits a RST instruction corresponding to the interrupt request with highest priority.

This makes vectored interrupt possible at 8 priority levels. The module includes facilities for programmed power-up/power-down handling (power loss interrupt).

3.3 ID-7009 Multiple DMA module.

This module expands the number of possible DMA I/O modules from one in the basic system to eight with the ID-7009 module present in the installation. When at least one incoming request line (DMAREQ(7:0)) is activated, the multiple DMA module activates the hold request line (HOLDR) to the CPU. When the CPU enters the hold state, indicated by an active HOLDA-line, the inputs to an internal priority encoder are frozen by means of a latch. The DMA-acknowledge lines (DMAACK(7:0)) now indicate which requesting unit is to be served first. When all DMA I/O modules (requesting service before HOLDA was activated) have been served, the Multiple DMA module deactivates the HOLDR-line to the CPU. When the CPU deactivates the HOLDA-line, the latch is opened for the new request situation.

3.4 ID-7005 Manual control module.

This module is used during evaluation of actual systems and during test and trouble shooting. It can be used in connection with a simple control box containing 4 switches or in connection with a control panel giving indication of bus states and control signals. The functions of the 4 switches are

- START/STOP: This switch controls the state of the CPU-module. In STOP the CPU is forced into wait-mode, until START is activated. When stopped, the indicators display the next machine cycle to be executed.
- SINGLE CYCLE: When activated, the CPU executes one cycle. The switch is only active, when the START/STOP switch is in STOP position.
- SINGLE INSTRUCTION: When activated, the CPU executes one instruction. The CPU is stopped in the first cycle of the next instruction. The switch is only active, when the START/STOP switch is in STOP position.
- DEBUG-CALL: This switch can be used to call a special debugging program, used during system evaluation. After activating the DEBUG-CALL switch, the status of the user program can be examined by the debug-program.

3.5 ID-7010 Bus extender module.

This module converts the internal high-speed bus of the system to an external low-speed bus for connection of process-control equipment. The low-speed bus is electrically isolated from the high-speed bus and facilitates use of high level logic (HiN11, CMOS etc.) connected to the bus, for use in noisy environments. By means of a switch register situated on the module, the user can select the limit between the addresses for high-speed I/O modules and the addresses for I/O modules connected to the low-speed bus. When input and output instructions with addresses corresponding to modules on the low-speed bus are executed the instruction execution is prolonged by means of the "Vent" - signal until the addressed I/O module has sampled or transmitted data from/to the bus.

A typical use of the low-speed bus system, is an application, where a large number of points is scanned and/or set by the microprocessor.

3.6 ID-7011 PROM-writer kit.

This kit is used for programming the INTEL 2708/2704 erasable read-only memory IC's used in the system. The kit contains three parts:

- a. ID-7011-1 PROM-writer module
- b. ID-7011-2 PROM-writer control box
- c. ID-7011-3 PROM-writer program (1 2704 PROM).

An ID-7000 microprocessor system including a CPU module, 1 RAM module, 1 PROM module and 1 TTY module (with ASR-33 Teletype writer) is necessary for using the 7011 PROM-writer kit. An ID-7006 Papertape Reader module with a GNT 26 Papertape reader may be used instead of the TTY module and the ASR-33 Teletype.

3.7 ID-7002 Static RAM module.

This module is a 2K x 8 bit writeable memory containing 16 static 1K MOS memory circuits (2102-1 or equivalent). A 5 bit switch register on the module determines the address of the first location on the memory module. A write protect switch on the module, accessible from the upper edge of the board, may be used during program testing, when RAM memory is used instead of PROM-memory. The memory module is fast enough to match the speed of the CPU, without using the wait function. (A circuit for activating the wait function is present on the module, for use if the lowest speed memory circuits (2102 etc.) are used).

3.8 ID-7007 Dynamic RAM module.

This module is a 8K x 8 bit writable memory containing 16 dynamic 4K MOS-memory circuits. The module is intended to be used instead of the ID-7002 2K x 8 bit static memory in installations with large RAM memories. Another application is in installations where power back-up for the memory is required, because of the low stand-by power of the dynamic RAM circuits.

Installations using the ID-7007 dynamic RAM memory need one ID-7008 Refresh module for a maximum of 8 ID-7007 modules. (in some applications, the refreshing can be done by software, and the refresh module may be omitted). The memory module is fast enough to match the speed of the CPU, without using the wait function. Even the refresh operation is done without affecting the instruction execution of the CPU.

3.9 ID-7008 Refresh module.

This module is used for refreshing the information in the ID-7007 Dynamic RAM memory. The RAM memory is divided into 64 blocks. To retain the information in the memory, each block must be accessed every 2 msec. A counter on the refresh module performs this task and supplies addresses to the memory module by means of 6 special wires in the bus. The refresh is timed to avoid loss of processor speed. One ID-7008 Refresh module can be used to refresh a maximum of 8 ID-7007 Dynamic RAM memories (64K byte).

3.10 ID-7001 PROM module.

This module contains sockets and associated logic for a maximum of 8 INTEL 2708 or 2704 erasable PROM circuits. (The 2708 contains 1024 x 8 bits and the 2704 512 x 8 bits). A switch register situated on the module determines the address of the first memory location on the module. By means of straps the module can be programmed in the following ways according to the memory circuits used:

- a. 512 - 2048 bytes memory using 1-4 2704 IC's
- b. 512 - 4096 bytes memory using 1-8 2704 IC's
- c. 1024 - 4096 bytes memory using 1-4 2708 IC's
- d. 1024 - 8192 bytes memory using 1-8 2708 IC's

The memory module is fast enough to match the speed of the CPU, without using the wait function.

3.11 ID-7004 Asynchronous Data Communication module (TTY-module).

This module is used for connecting asynchronous data communication equipment such as teletypewriter, CRT-terminals or modems to the microprocessor system.

The interface to the data communication equipment is in accordance with the CCITT V24 (or EIA RS-232) standard.

By means of straps on the module a large range of baud-rates can be selected from 75-9600 bits/sec.

By means of a control word from the computer, the module can be programmed to use different character formats. The module communicates with the CPU by interrupt or by sensed status word.

The module uses two consecutive I/O addresses. The address of the module is determined by a 7-bit switch register.

3.12 ID-7012 Synchronous Data Communication Transmitter module.

This module is used for connecting synchronous data communication equipment to the microprocessor system. The module can be used alone for simplex connections, or in connection with the ID-7013 Synchronous Data Communication Receiver module for full-duplex or half duplex connections. The interface to the data communication equipment is in accordance with the CCITT V24 (or EIA RS-232) standard.

By means of straps on the module, the baud-rate can be set to 1200, 2400, 4800 or 9600 bits/sec. The module communicates with the CPU by interrupt or by sensed status word. Logic to detect data overrun has been incorporated.

The module uses two consecutive I/O-addresses. The address of the module is determined by a 7 bit switch register.

3.13 ID-7013 Synchronous Data Communication Receiver module.

This module is used for connecting synchronous data communication equipment to the microprocessor system. The module can be used alone for simplex connections, or in connection with the ID-7012 Synchronous Data Communication Transmitter module for full-duplex or half duplex connections. The interface to the data communication equipment is in accordance with the CCITT V24 (or EIA RS-232) standard. The module contains logic to synchronize on the ASCII SYNC-character.

By means of straps on the module, the baud-rate can be set to 1200, 2400, 4800 or 9600 bits/sec. The module communicates with the CPU by interrupt or by sensed status word. Logic to detect data overrun has been incorporated.

The module uses two consecutive I/O addresses. The address of the module is determined by a 7 bit switch register.

3.14 ID-7014 Flexible Disc module.

This module is a control unit for interfacing a maximum of four CDS 140 flexible disc drives*) to the microprocessor system, for storing a maximum of 1Mbyte with a data rate of 31.25 Kbytes/sec.

The module contains logic for head-positioning and sector reading and writing using the IBM format (with soft sector marks). Also included is logic for CRCC-generation and checking. With appropriate software drivers, the disc format is fully compatible with the IBM-3740 system. The module uses four consecutive I/O addresses in the system, two for control/status and two for data transfer. The address of the module is set by means of a 6 bit switch register situated on the module.

The module consists of three printed circuit boards connected together by flat cable on the top connectors of the boards. The module can be plugged into the bus in any three consecutive bus locations. The three boards contain the following logic:

- ID 7014-1: Address logic, control registers, bus drivers for control and status, disc-drive buffers, interrupt logic.
- ID 7014-2: Input and output buffers, shift register, register control logic, overrun detection, bus drivers for data communication.
- ID 7014-3: Read circuit, write circuit, CRCC circuit.

3.15 ID-7006 Papertape Reader module.

This module contains logic for interfacing a GNT 26,**) 500 ch/sec optical reader to the microprocessor system. The reader can be controlled in either a continuous mode or in a start/stop mode. In both modes, communication with the microprocessor system can be performed by means of interrupts or by using sensed status words.

The module uses two consecutive I/O addresses. The module address is set by a 7 bit switch register.

3.16 ID-7015 Interval Timer module.

This module can be used in process control installations and other applications where a crystal controlled time base is needed. The module can be programmed by the CPU, to generate interrupts with a certain repetition rate. A control word transmitted to the module selects an internal clock rate on the module to be 10 KC/S, 1KC/S, 100 C/S or 10 C/S. This time base is used for clocking a programmable counter. The modulus of the counter is determined by another control word transmitted from the CPU. It is possible to preset the time between interrupts to values from 200 usec to 25.6 sec. with a resolution corresponding to the internal clock rate of the module (100 usec, 1 msec, 10 msec or 100 msec).

The module uses two consecutive I/O addresses. The module address is set by a 7 bit switch register.

*) CDS 140 is a product of Century Data Systems Inc.

***) GNT 26 is manufactured by the Danish Company GNT Automatic A/S.

3.17 ID-7016 A/D-Conversion module.

This module contains associated logic and a socket for a ZELTEX A/D-converter.*) A wide range of speed- and precision requirements can be served by use of different ZELTEX converters:

ZD 460	50 usec	8 bit
ZD 461	100 usec	10 bit
ZD 462	200 usec	12 bit
ZD 470	15 usec	8 bit
ZD 471	30 usec	10 bit

By means of switches on the module, it is possible to select one of three different analog ranges (-+ 5V, -+ 10V or 0-10V).

The module uses two consecutive I/O addresses. The address of the module is determined by a 7 bit switch register. A conversion is started by transmitting a control word to the module. When the conversion is completed the module generates an interrupt.

3.18 ID-7017 D/A-Conversion module.

This module contains associated logic and a socket for ZELTEX D/A-converters:

ZD 430	20-25 usec	8 bit
ZD 431	20-25 usec	10 bit
ZD 432	20-30 usec	12 bit
ZD 440	2-5 usec	8 bit
ZD 441	5-10 usec	10 bit

By means of switches on the module, it is possible to select one of three different analog ranges (-+ 5V, -+ 10V and 0-10V).

The module uses two consecutive I/O addresses. The address of the module is determined by a 7 bit switch register.

3.19 ID-7018 Standard I/O module.

This module contains an 8 bit output register, an 8 bit input register and associated logic, including two flip-flops for hand-shaking communication between the microprocessor system and other equipment, typically I/O units or minicomputers. The inputs and outputs to and from the module are available together with the control signals from the top connector of the module. Communication between the module and the CPU is performed by means of interrupts or through sensed status words.

The module uses two consecutive I/O addresses. The address of the module is determined by a 7 bit switch register.

*) ZELTEX converters is a product from ZELTEX Inc.

4. Basic evaluation software.

The CPU in the ID-7000 microprocessor system is an INTEL 8080 microprocessor. This means, that all software from the INTEL Corporation (cross-assembler, PL/M-compiler and simulator) which runs on a variety of time-sharing systems and university computing centers, can be used in generating programs for the ID-7000 microprocessor system.

The following system software is available from or under development by Datateknik ApS:

1. ID-7000 DEBUGGER/MONITOR. (Available now)
2. ID-7000 ASSEMBLER. (Ultimo 1975)
3. ID-7000 TEXT EDITOR. (Ultimo 1975).

Except for the ID-7000 DEBUGGER/MONITOR, the programs are delivered either in PROM's for use in the ID-7001 PROM modules or in papertapes for loading into ID-7002 RAM-modules. The DEBUGGER/MONITOR can only be delivered in PROM's.

In the following section is given a short description of these software modules. Detailed descriptions are given in the manuals for the programs.

4.1 ID-7000 DEBUGGER/MONITOR.

This program is an essential tool in evaluation of user software and hardware. It also functions as an operating system or a monitor for the system. It has a large number of features including facilities for:

- a. Input (from the keyboard) of instructions in mnemonic with hexadecimal or decimal addresses.
- b. Input (from the keyboard) of hexadecimal and decimal numbers or ASCII strings.
- c. Listing of memory in mnemonic instruction form with hexadecimal addresses.
- d. Listing of memory in hexadecimal form.
- e. Examining or loading the 8080 stack-pointer.
- f. Binary dump of memory into papertape (by use of TTY-Punch or high-speed punch).
- g. Binary dump of memory into PROM-IC's (by use of the ID-7011 PROM-programmer).
- h. Binary input from papertape to memory (by use of TTY-reader or high-speed reader).
- i. Insertion of breakpoints (with loop counting facilities) into user programs.

- j. Starting user programs from specified addresses with specified PSW (register content).
- k. Returning to user programs (after a manual DEBUG call or a break point) with stacked PSW.

The DEBUGGER/MONITOR occupies the upper 4K of the memory (from X.F000-X.FFFF). The program is entered by use of the DEBUG call push-button or from user programs by means of breakpoints. In both cases the contents of the registers and the program counter are stored in the stack. It is possible in this way to examine the status of the user program, when it was interrupted, and to restore the status of the user program when started again.

The DEBUGGER/MONITOR is programmed not to use the stack or other parts of RAM memory, except when the DEBUG call or the break point facilities are used.

4.2 ID-7000 Assembler.

This program is a two-pass assembler for translating programs written in assembler language into 8080 machine language. The source language is compatible with the language used for the INTEL assemblers. Input is taken from the ASR-33, teletype reader or from a high-speed papertape reader. The object output is generated on the ASR-33; teletype punch or on a high-speed punch device. Listing of the program together with error messages and the symbol table are performed by the Teletype (or equivalent equipment).

The ID-7000 Assembler is most convenient used as a PROM-program on a single ID-7001 PROM-module. Memory requirements for storing the symbol table and the stack depends on the number of symbols used in the program to be assembled. When a 2K RAM module is used (1 ID-7002 module) a program with 200 symbols can be assembled.

The ID-7000 Assembler is also available on papertape for loading into RAM-memory.

4.3 ID-7000 Text editor.

The ID-7000 Text editor is a program for generating and editing texts on the microprocessor system. The program is intended for generating source texts for the ID-7000 Assembler but may be used for other purposes too.

Input to the text editor is taken from the Teletype keyboard and the Teletype reader (or a high speed reader). Listing of the text is performed by the Teletype-printer and output of the text is done by the Teletype punch (or a high speed punch-device).

The ID-7000 Text editor is most convenient used as a PROM-program on a single ID-7001 PROM-module.

Furthermore an ID-7002 RAM-module is necessary for text buffer in the editing process.

The ID-7000 Text editor is also available on papertape for loading into RAM memory.



Silicon Gate MOS 8080A

SINGLE CHIP 8-BIT N-CANNEL MICROPROCESSOR

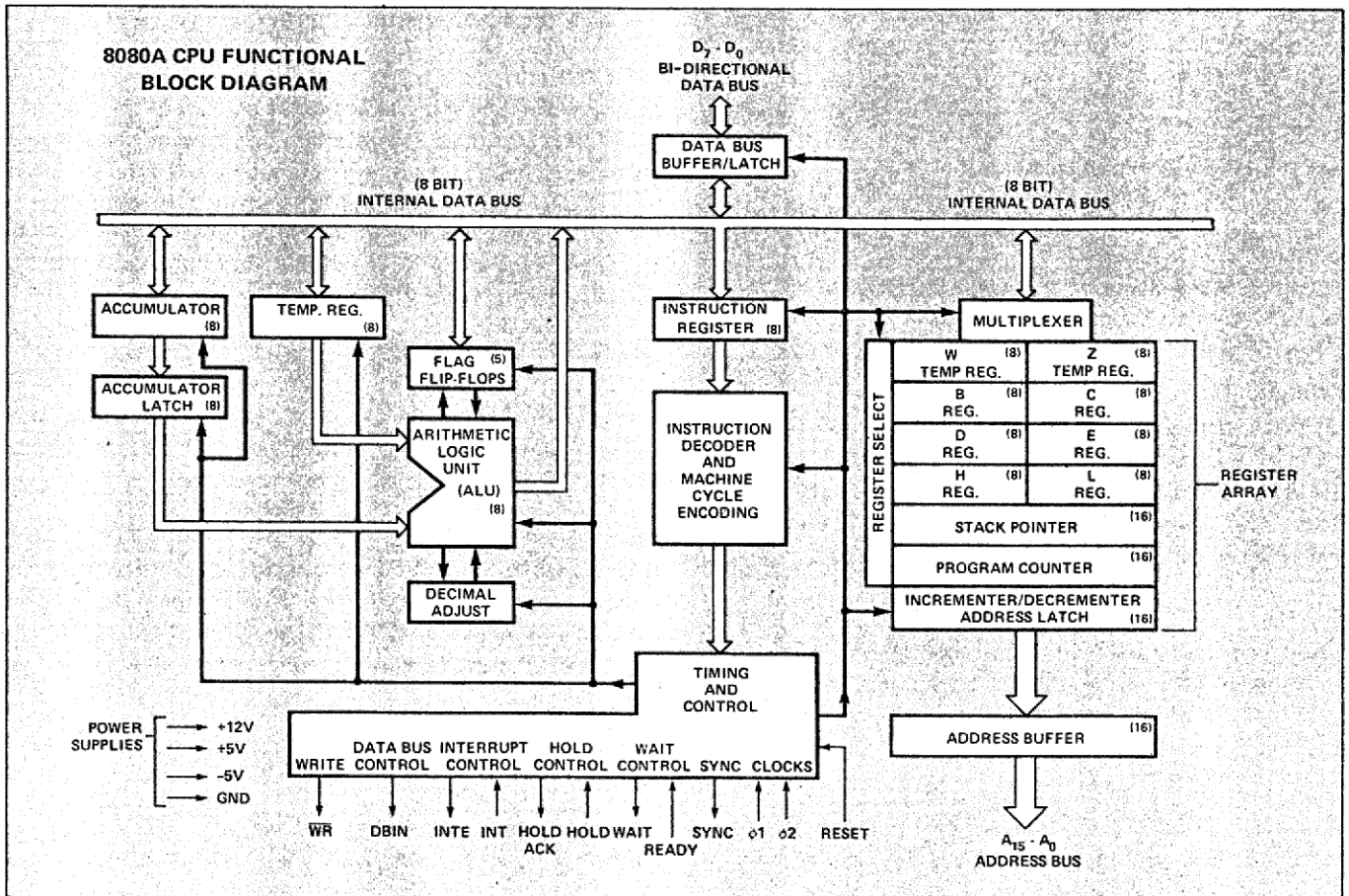
The 8080A is functionally and electrically compatible with the Intel® 8080.

- TTL Drive Capability
- 2 μ s 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 buses 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 buses resides with the HOLD signal. It provides the ability to suspend processor operation and force the address and data buses into a high impedance state. This permits OR-tying these buses 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.

A₁₅-A₀ (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. A₀ is the least significant address bit.

D₇-D₀ (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. D₀ 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.

WR (output)

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

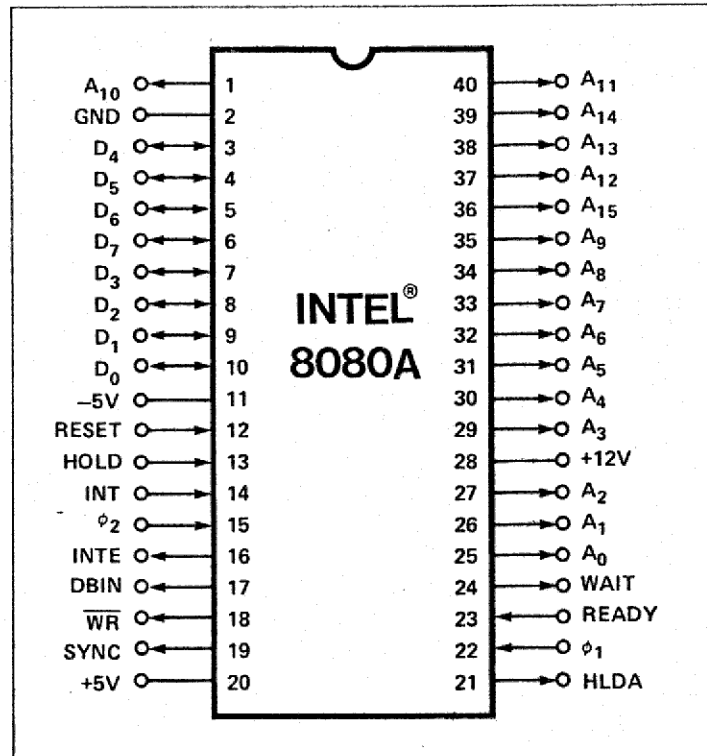
HOLD (input)

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 T_W state and the READY signal is active.
- As a result of entering the HOLD state the CPU ADDRESS BUS (A₁₅-A₀) and DATA BUS (D₇-D₀) 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



Pin Configuration

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 ϕ_1 and high impedance occurs after the rising edge of ϕ_2 .

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).
- ϕ_1, ϕ_2 2 externally supplied clock phases. (non TTL compatible)

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 sub-routines 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

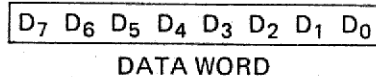
increment and decrement memory, the six general registers and the accumulator is provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer. Further capability is provided by the ability to rotate the accumulator left or right through or around 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.

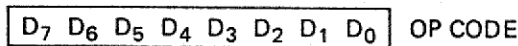
Data and Instruction Formats

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.



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.

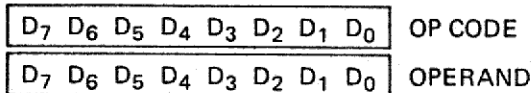
One Byte Instructions



TYPICAL INSTRUCTIONS

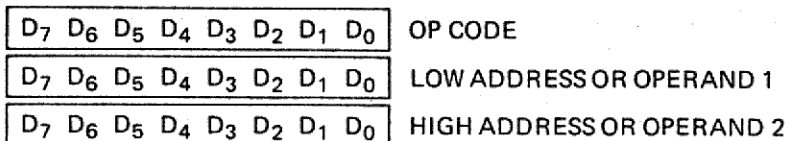
Register to register, memory reference, arithmetic or logical, rotate, return, push, pop, enable or disable
Interrupt instructions

Two Byte Instructions



- Immediate mode or I/O instructions

Three Byte Instructions



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⁾								Clock ⁽²⁾ Cycles	Mnemonic	Description	Instruction Code ⁽¹⁾								Clock ⁽²⁾ Cycles
		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀				D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
MOV _{r1,r2}	Move register to register	0	1	D	D	D	S	S	S	5	RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
MOV _{M,r}	Move register to memory	0	1	1	1	0	S	S	S	7	RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
MOV _{r,M}	Move memory to register	0	1	D	D	D	1	1	0	7	RP	Return on positive	1	1	1	1	0	0	0	0	5/11
HLT	Halt	0	1	1	1	0	1	1	0	7	RM	Return on minus	1	1	1	1	1	0	0	0	5/11
MVI _r	Move immediate register	0	0	D	D	D	1	1	0	7	RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
MVI _M	Move immediate memory	0	0	1	1	0	1	1	0	10	RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
INR _r	Increment register	0	0	D	D	D	1	0	0	5	RST	Restart	1	1	A	A	A	1	1	1	11
DCR _r	Decrement register	0	0	D	D	D	1	0	1	5	IN	Input	1	1	0	1	1	0	1	1	10
INR _M	Increment memory	0	0	1	1	0	1	0	0	10	OUT	Output	1	1	0	1	0	0	1	1	10
DCR _M	Decrement memory	0	0	1	1	0	1	0	1	10	LXI _B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
ADD _r	Add register to A	1	0	0	0	0	S	S	S	4	LXI _D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
ADC _r	Add register to A with carry	1	0	0	0	1	S	S	S	4	LXI _H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
SUB _r	Subtract register from A	1	0	0	1	0	S	S	S	4	LXI _{SP}	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
SBB _r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4	PUSH _B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
ANA _r	And register with A	1	0	1	0	0	S	S	S	4	PUSH _D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
XRA _r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4	PUSH _H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
ORA _r	Or register with A	1	0	1	1	0	S	S	S	4	PUSH _{PSW}	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
CMP _r	Compare register with A	1	0	1	1	1	S	S	S	4	POP _B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
ADD _M	Add memory to A	1	0	0	0	0	1	1	0	7	POP _D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
ADC _M	Add memory to A with carry	1	0	0	0	1	1	1	0	7	POP _H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
SUB _M	Subtract memory from A	1	0	0	1	0	1	1	0	7	POP _{PSW}	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
SBB _M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7	STA	Store A direct	0	0	1	1	0	0	1	0	13
ANA _M	And memory with A	1	0	1	0	0	1	1	0	7	LDA	Load A direct	0	0	1	1	1	0	1	0	13
XRA _M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7	XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
ORA _M	Or memory with A	1	0	1	1	0	1	1	0	7	XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
CMP _M	Compare memory with A	1	0	1	1	1	1	1	0	7	SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7	PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7	DAD _B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7	DAD _D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7	DAD _H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
ANI	And immediate with A	1	1	1	0	0	1	1	0	7	DAD _{SP}	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7	STAX _B	Store A indirect	0	0	0	0	0	0	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7	STAX _D	Store A indirect	0	0	0	1	0	0	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7	LDAX _B	Load A indirect	0	0	0	0	1	0	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4	LDAX _D	Load A indirect	0	0	0	1	1	0	1	0	7
RRC	Rotate A right	0	0	0	0	1	1	1	1	4	INX _B	Increment B & C registers	0	0	0	0	0	0	1	1	5
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4	INX _D	Increment D & E registers	0	0	0	1	0	0	1	1	5
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4	INX _H	Increment H & L registers	0	0	1	0	0	0	1	1	5
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10	INX _{SP}	Increment stack pointer	0	0	1	1	0	0	1	1	5
JC	Jump on carry	1	1	0	1	1	0	1	0	10	DCX _B	Decrement B & C	0	0	0	0	1	0	1	1	5
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10	DCX _D	Decrement D & E	0	0	0	1	1	0	1	1	5
JZ	Jump on zero	1	1	0	0	1	0	1	0	10	DCX _H	Decrement H & L	0	0	1	0	1	0	1	1	5
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10	DCX _{SP}	Decrement stack pointer	0	0	1	1	1	0	1	1	5
JP	Jump on positive	1	1	1	1	0	0	1	0	10	CMA	Complement A	0	0	1	0	1	1	1	1	4
JM	Jump on minus	1	1	1	1	1	0	1	0	10	STC	Set carry	0	0	1	1	0	1	1	1	4
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10	CMC	Complement carry	0	0	1	1	1	1	1	1	4
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10	DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CALL	Call unconditional	1	1	0	0	1	1	0	1	17	SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
CC	Call on carry	1	1	0	1	1	1	0	0	11/17	LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17	EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17	DI	Disable interrupt	1	1	1	1	0	0	1	1	4
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17	NOP	No-operation	0	0	0	0	0	0	0	0	4
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											

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.