DANSK DATA ELEKTRONIK ApS

Microprocessorsystem
ID-7ooo

System description
Second edition
March 1976

Dansk Data Elektronik ApS, Generatorvej 6A, 273o Herlev, Denmark
Phone: o2 84 5o 11

Table of Contents.

1. Introduction 1

2. The bus 1

3. Modules 5

3.1. ID-7ooo CPU Module 5
3.2. ID-7oo3 Interrupt Priority Module 5
3.3. ID-7oo9 Multiple DMA Module 6
3.4. ID-7oo5 Manual Control Module 7
3.5. ID-7olo Bus Extender Module 7
3.6. ID-7oll PROM-Writer Module 7
3.7. ID-7oo2 Static RAM Module 8
3.8. ID-7oo7 Dynamic RAM Module 8
3.9. ID-7oo8 Refresh Module 8
3.1o. ID-7ool PROM Module 8
3.11. ID-7oo4 Async. Communication Module 9
3.12. ID-7ol2 Sync. Receiver/Transmitter Module 9
3.13. ID-7ol4 Flexible Disc Module 9
3.14. ID-7oo6 Papertape Reader/Punch Module 9
3.15. ID-7ol5 Interval Timer Module 1o
3.16. ID-7ol6 A/D-Conversion Module 1o
3.17. ID-7ol7 D/A-Conversion Module 1o
3.18. ID-7ol8 Standard I/O Module 1o
3.19. ID-7ol9 Analog Multiplexor Module 11
3.2o. ID-7o2o Wire-Wrap Board 11
3.21. ID-7o21 Read Scanner Module 11

4. Basic Evaluation Software 11

4.1. ID-7ooo Debugger/Monitor 11
4.2. ID-7ooo Assembler 12
4.3. ID-7ooo Text Editor 13

Appendix 1: 8o8oA Central Processor, Data Sheet

1. Introduction.

ID-7ooo- MICROPROCESSOR SYSTEM is a modular microprocessor system based on the INTEL 8o8o 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 loo-pole edge-card connector. Connection to the environment is by means of a 72-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-7ooo microprocessor system has been developed in co-operation with the Department of Computer Science, at the Technical University of Denmark. It is a further development on a previous 8o8o 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.

D(7:0): 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-8ooo, Et mikroprocessorsystem" part 1 and 2. Edited by Ole Lading.

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

$\overline{\text{AFBR}}$: 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.

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

HOLDA: High-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.

$\overline{\text{ABUSDISABLE}}$: 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.

$\overline{\text{DBUSDISABLE}}$: 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:

$\overline{\text{DMAREQ(7:0)}}$: 8 (active-low) signals for DMA requests to the Multiple DMA module.

$\overline{\text{DMAACK(7:0)}}$: 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.

$\overline{\text{VENT}}$: 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.

$\overline{\text{RESET}}$:	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 $\overline{\text{RESET}}$ 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.
ANALOG GND: +15V ANALOG SUPPLY: ÷15V ANALOG SUPPLY:	In configurations using A/D-, D/A- or ANALOG MPX-I/O units, these optional power supplies must be present.
$\overline{\text{REF}}$:	Low-active signal, indicating a refresh cycle. (Used in dynamic RAM modules).
RAD(5:0):	6 address signals used instead of ADR(5:0) in dynamic RAM modules.
DW:	Write signal to dynamic RAM modules.
DS:	Strobe signal for reading dynamic RAM modules.
CE:	Clock pulse to dynamic RAM modules.

==========

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-7ooo Microprocessor system.

Side A (back)	Pin no.	Side B (component side)
+5V	1	GND
÷5V	2	GND
ADR(0)	3	$\overline{\text{MEMDISABLE}}$
(1)	4	HLTA
(2)	5	$\overline{\text{STACK}}$
(3)	6	$\overline{\text{INTE}}$
(4)	7	ANALOG GND
(5)	8	+15V ANALOG SUPPLY
(6)	9	÷15V ANALOG SUPPLY
(7)	1o	
(8)	11	$\overline{\text{IR}}$(o)
(9)	12	(1)
(1o)	13	(2)
(11)	14	(3)
(12)	15	(4)
(13)	16	(5)
(14)	17	(6)
(15)	18	(7)
DW	19	RAD(0)
$\overline{\text{WR}}$	2o	RAD(1)
$\overline{\text{OUTR}}$	21	RAD(2)
$\overline{\text{INR}}$	22	RAD(3)
$\overline{\text{RR}}$	23	RAD(4)
$\overline{\text{AFBR}}$	24	RAD(5)
$\overline{\text{REF}}$	25	CE
$\overline{\text{ABUSDISABLE}}$	26	$\overline{\text{DBUSDISABLE}}$
$\overline{\text{WG}}$	27	$\overline{\text{HOLDR}}$
$\overline{\text{VENT}}$	28	HOLDA
$\overline{\text{INT}}$	29	$\overline{\text{DMAACK}}$(0)
DS	3o	(1)
	31	(2)
D(o)	32	(3)
(1)	33	(4)
(2)	34	(5)
(3)	35	(6)
(4)	36	(7)
(5)	37	$\overline{\text{DMAREQ}}$(o)
(6)	38	(1)
(7)	39	(2)
	4o	(3)
	41	(4)
	42	(5)
	43	(6)
	44	(7)
$\overline{\text{RESET}}$	45	$\overline{\text{POWERLOSS}}$
SYNC	46	M1
$\overline{\text{Ø1}}$	47	$\overline{\text{Ø2}}$
$\overline{\text{16M}}$	48	$\overline{\text{16M}}$
GND	49	+12V
GND	5o	+5V

TABLE I

3. Modules.

The basic part of the system is the ID-7ooo CPU module. The functions of the CPU module can be expanded by use of some special purposes modules (ID-7oo3 Interrupt Priority module and ID-7oo9 Multiple DMA module). Another special purpose module is the ID-7oo5 Manual control module, used for system evaluation and test in connection with the ID test panel. The ID-7olo Bus extender module is used for extension of the bus, to an external galvanic isolated low-speed bus. For PROM-programming, the ID-7oll 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-7ooo CPU module.

This module contains the INTEL 8o8o microprocessor with associated logic and buffers. This includes:

- Clock generator for clock phases for 8o8o and associated logic.
- Three-state buffers for addresses (ADR(15:0)).
- Three-state two-way buffers for data (D(7:0)).
- Latch for 8o8o-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 8o8o 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-7oo3 Interrupt priority module.

This module contains logic for making priority of 8 incoming interrupt request lines $\overline{IR(7:0)}$ 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 7oo3-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-7oo9 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-7oo9 module present in the installation. When at least one incoming request line ($\overline{\text{DMAREQ}}$(7:0)) is activated, the multiple DMA module activates the hold request line ($\overline{\text{HOLDR}}$) to the CPU. When the CPU enters the hold state, indicated by an active $\overline{\text{HOLDA}}$-line, the inputs to an internal priority encoder are frozen by means of a latch. The DMA-acknowledge lines ($\overline{\text{DMAACK}}$(7:0) now indicate which requesting unit is to be served first. When all DMA I/O modules (requesting service before $\overline{\text{HOLDA}}$ was activated) have been served, the Multiple DMA module deactivates the $\overline{\text{HOLDR}}$-line to the CPU. When the CPU deactivates the $\overline{\text{HOLDA}}$-line, the latch is opened for the new request situation.

3.4 ID-7oo5 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-7olo 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 (HiNil, 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 "$\overline{\text{Vent}}$" - 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-7oll PROM-writer module.

This module is used for programming the INTEL 27o8/27o4 erasable read-only memory IC'S used in the system. The module is used in conjunction with a PROM-writer program contained in the ID 7ooo DEBUGGER/MONITOR program (see section 4.1). The PROM to be programmed is placed in an IC-socket on the module. Optionally a PROM-writer box, in connection with the module via the top connector, can be used.

An ID-7ooo microprocessor system including a CPU module, 1 RAM module, 1 PROM module (containing the DEBUG/MONITOR program) and 1 TTY module (with ASR-33 Teletype writer) is necessary for using the ID-7oll PROM-writer module.

The ASR-33 Teletype writer can be replaced by a CRT-terminal combined with an ID-7oo6 Papertape Reader module and a high speed reader.

3.7 ID-7oo2 Static RAM module.

This module is a 2K x 8 bit writeable memory containing 16 static 1K MOS memory circuits (21o2-1 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 (21o2 etc.) are used).

3.8 ID-7oo7 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-7oo2 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-7oo7 dynamic RAM memory need one ID-7oo8 Refresh module for a maximum of 8 ID-7oo7 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-7oo8 Refresh module.

This module is used for refreshing the information in the ID-7oo7 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-7oo8 Refresh module can be used to refresh a maximum of 8 ID-7oo7 Dynamic RAM memories (64K byte).

3.1o ID-7oo1 PROM module.

This module contains sockets and associated logic for a maximum of 8 INTEL 27o8 or 27o4 erasable PROM circuits. (The 27o8 contains 1o24 x 8 bits and the 27o4 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 - 2o48 bytes memory using 1-4 27o4 IC's
b. 512 - 4o96 bytes memory using 1-8 27o4 IC's
c. 1o24 - 4o96 bytes memory using 1-4 27o8 IC's
d. 1o24 - 8192 bytes memory using 1-8 27o8 IC's

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

3.11 ID-7oo4 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, or TTY 2oMA current loop standard.

By means of straps on the module a large range of baud-rates can be selected from 75-96oo 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-7o12 Synchronous Data Communication Receiver/Transmitter module.

This module is used for connecting synchronous data communication equipment to the microprocessor system. The interface to the data communication equipment is in accordance with the CCITT V24 (or EIA RS-232) standard.

Baud rates in the range from DC-56K Baud is available. The module communicates with the CPU by interrupt or by sensed status word. Logic for SYNC detection and automatic SYNC transmission is included.

The module uses 4 consecutive I/O-addresses. The address of the module is determined by a 6 bit switch register.

3.13 ID-7o14 Flexible Disc module.

This module is a control unit for interfacing a maximum of four CDS 14o[1] 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). With appropriate software drivers, the disc format is fully compatible with the IBM-374o 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 2 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 2 consecutive bus locations.

3.14 ID-7oo6 Papertape Reader/Punch module.

This module contains logic for interfacing a GNT 26[2], 5oo ch/sec optical reader and/or a FACIT 4o7o[3], 7o ch/sec punch to the microprocessor system. The communication with the microprocessor system can be performed by means of interrupts or by using sensed status words.

[1] CDS 14o is a product of Century Data Systems Inc.,
[2] GNT 26 is manufactured by the Danish Company Gnt Automatic A/S.
[3] FACIT 4o7o is manufactured by the Swedish Company FACIT AB.

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

3.15 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 100 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.

3.16 ID-7016 A/D-Conversion module.

This module contains an 8/10 bit analog-digital converter. Included on the module is an 8 input differential analog multiplexor, a high precision differential instrumentation amplifier and sample/hold circuit. By means of 8 potentiometers on the module, the amplification in each channel can be selected individually from 1 to 20 times. The basic range can be selected to either 0-10V or ÷10V-+10V.

The module uses two consecutive I/O adresses. 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 or by an external signal. When the conversion is completed, the module generates an interrupt and a ready status bit is set.

The module uses the AD 7570 A/D-converter from Analog Devices.

3.17 ID-7017 D/A-Conversion module.

This module contains a maximum of four 8/10 bit digital-analog converters. The analog outputs from the module is in the range 0-+10V or 0-÷10V.

The module uses 8 consecutive I/O addresses for loading digital data to the converters. The address of the module is determined by a 5 bit switch register.

The module contains 4 AD 7522 DAC'S and 4 AD518 high speed amplifiers and associated logic.

3.18 ID-7018 Standard I/O module.

This module contains two INTEL 8255 programmable peripheral interface IC'S, giving 48 programmable input/output pins in the top connector of the board. Also included is logic for generating interrupts to the ID-7003 Interrupt Priority Module.

The module uses 8 consecutive I/O addresses. The address of the module is determined by a 5 bit switch register.

3.19 ID-7o19 Analog Multiplexor module.

This module can be used when a big number of analog inputs must be connected to the ID-7o16 A/D-Conversion module. The module can be used to multiplex 64 single ended or 48 differential analog channels to one single ended or differential channel.

The module uses one I/O address. The address of the module is determined by an 8 bit switch register.

3.2o Wire-Wrap board.

This board is used for custom designed interfaces. It can be equipped with a maximum of 35 16 pin wire-wrap sockets. All kinds of wire-wrap sockets with a 3/1o" or 6/1o" spacing may be used.

3.21 ID-7o21 Reed Scanner module.

This moduel is used for scanning a maximum of 256 contacts organized in a 16 by 16 matrix. The connection to the contact matrix is done via the top connector of the module.

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

4. Basic evaluation software.

The CPU in the ID-7ooo microprocessor system is an INTEL 8o8o 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-7ooo microprocessor system.

The following system software is available:

1. ID-7ooo DEBUGGER/MONITOR.
2. ID-7ooo ASSEMBLER.
3. ID-7ooo TEXT EDITOR.

Except for the ID-7ooo DEBUGGER/MONITOR, the programs are delivered either in PROM's for use in the ID-7oo1 PROM modules or in papertapes for loading into ID-7oo2 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-7ooo 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 8o8o 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-7oll 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 5K of the memory (from X.EC00 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-7ooo Assembler.

This program is a two-pass assembler for translating programs written in assembler language into 8o8o 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-7ooo Assembler is most convenient used as a PROM-program on a single ID-7ool 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-7oo2 module) a program with 2oo symbols can be assembled.

The ID-7ooo Assembler is also available on papertape for loading into RAM-memory.

4.3 ID-7ooo Text editor.

The ID-7ooo Text editor is a program for generating and editing texts on the microprocessor system. The program is intented for generating source texts for the ID-7ooo 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-7ooo Text editor is most convenient used as a PROM-program on a single ID-7oo1 PROM-module.

Furthermore an ID-7oo2 RAM-module is necessary for text buffer in the editing process.

The ID-7ooo Text editor is also available on papertape for loading into RAM memory.

Silicon Gate MOS 8080A

SINGLE CHIP 8-BIT N-CHANNEL MICROPROCESSOR

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

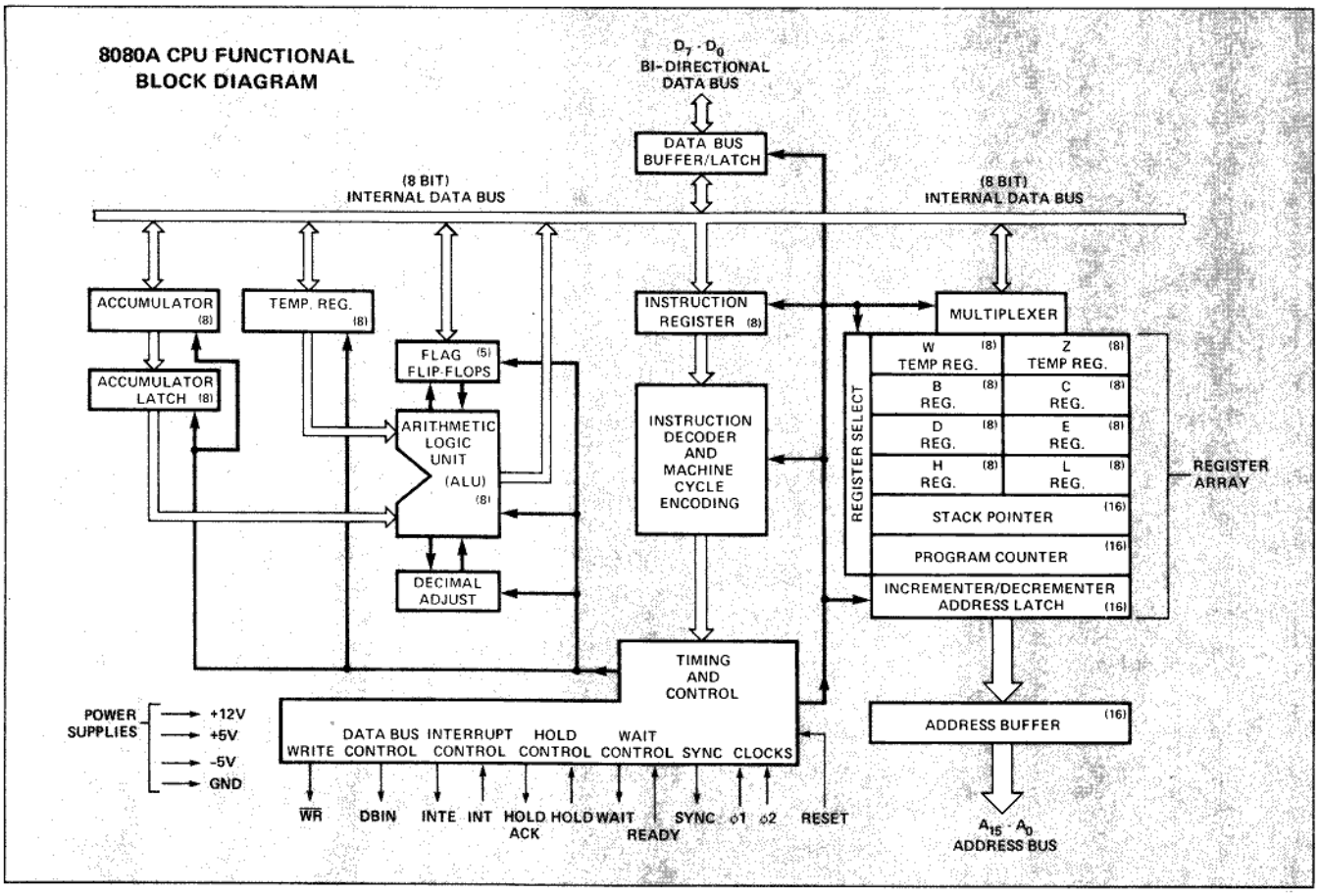
- **TTL Drive Capability**
- **2 μs Instruction Cycle**
- **Powerful Problem Solving Instruction Set**
- **Six General Purpose Registers and an Accumulator**
- **Sixteen Bit Program Counter for Directly Addressing up to 64K Bytes of Memory**
- **Sixteen Bit Stack Pointer and Stack Manipulation Instructions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

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

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

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

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



8080A FUNCTIONAL PIN DEFINITION

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

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

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

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

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

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

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

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

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

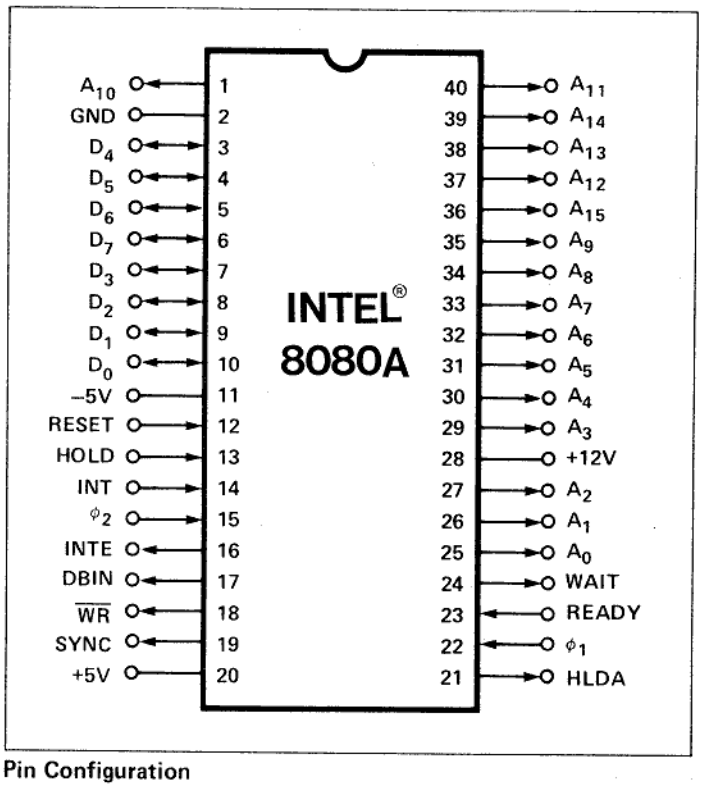
- the CPU is in the HALT state.
- the CPU is in the T2 or TW state and the READY signal is active.

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

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

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

In either case, the HLDA signal appears after the rising edge of ϕ_1 and high impedance occurs after the rising edge of ϕ_2.



Pin Configuration

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

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

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

V_{SS}	Ground Reference.
V_{DD}	+12 ± 5% Volts.
V_{CC}	+5 ± 5% Volts.
V_{BB}	-5 ±5% Volts (substrate bias).
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 subroutines is provided both conditionally and unconditionally. The RESTART (or single byte call instruction) is useful for interrupt vector operation.

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

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

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

Data and Instruction Formats

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

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

DATA WORD

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

One Byte Instructions

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	OP CODE

TYPICAL INSTRUCTIONS

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

Two Byte Instructions

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	OP CODE
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	OPERAND

Immediate mode or I/O instructions

Three Byte Instructions

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

Jump, call or direct load and store instructions

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

INSTRUCTION SET

Summary of Processor Instructions

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

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