

DESIGN OF HARDWARE INTERFACE FOR
SBC 8085 INTEL-SINGLE BOARD COMPUTER

SAMIR BAHY EL DIN

A Major Technical Report

in

The Faculty

of

Engineering

Presented in partial fulfillment of the Requirements

for the degree of Master of Engineering at

Concordia University

Montreal, Quebec, Canada

November 1979

© SAMIR BAHY EL DIN 1979

ABSTRACT

SAMIR BAHY EL DIN
DESIGN OF HARDWARE INTERFACE FOR
SBC 8085 INTEL-SINGLE BOARD COMPUTER

An Overall general description of up 8085 and IC 8279 are covered in this report.

The design of INTERFACE for keyboard type input device and the output device using INTEL 8279 is the center of attention of this technical report.

The design deals with the assembly programming of 8279 which is needed in that design.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Dr. S. Gracovetsky, for his helpful suggestions, counsel and encouragement.

Special thanks should go to Mrs. Monica Etwaroo for typing this Major Technical Report.

TABLE OF CONTENTS

	PAGE
ABSTRACT	
ACKNOWLEDGEMENT	
INTRODUCTION	
TECHNICAL REPORT	1
1. REQUIREMENT	1
2. EQUIPMENT AVAILABLE FOR THE PROJECT	1
3. SYSTEM EXPANSION	2
4. DESCRIPTION FOR 80/05 SBC	2
5. DESCRIPTION OF INTEL 8085 UP	3
5.1.1 ARITHMETIC SECTION	3
5.1.2 INSTRUCTION DECODER	4
5.1.3 ARITHMETIC LOGIC UNIT	4
5.1.4 ACCUMULATOR	5
5.1.5 PROGRAM MEMORY	5
5.1.6 BUS	5
5.1.7 TEST AND INT INPUTS	7
5.1.8 PROGRAM COUNTER AND STACK	8
5.1.9 PROGRAM STATUS WORD	11
5.1.10 CONDITIONAL BRANCH LOGIC	12
5.1.12 INTERRUPT	12

	PAGE
5.1.12 INTERRUPT TIMING	13
5.1.13 TIMER/COUNTER	14
5.1.14 CLOCK AND TIMING CIRCUITS	17
5.1.14.1 CLOCK GENERATION	17
5.1.14.2 CYCLE COUNTER	17
5.1.15 INITIALIZATION OR RESET	18
5.1.16 PIN DESCRIPTION FOR 8085 UP	19
5.1.16.1 AD ₀₋₇ (INPUT/OUTPUT 3-STATE)	19
5.1.16.2 A _{8-A15} (OUTPUT 3-STATE)	19
5.1.16.3 S ₀ , S ₁ (OUTPUT)	21
5.1.16.4 RD (OUTPUT 3-STATE)	21
5.1.16.5 WR (OUTPUT 3-STATE)	21
5.1.16.6 ALE (OUTPUT)	21
5.1.16.7 READY (INPUT)	22
5.1.16.8 HOLD (INPUT)	22
5.1.16.9 HOLD A (OUTPUT)	22
5.1.16.10 INTR (INPUT)	23
5.1.16.11 TRAP (INPUT)	23
5.1.16.12 INTA (OUTPUT)	23
5.1.16.13 RESET OUT (OUTPUT)	24
5.1.16.14 RESET IN (INPUT)	24
5.1.16.15 CLK (OUTPUT)	25
5.1.16.16 IO/M (OUTPUT)	25
5.1.16.17 X ₁ , X ₂ (INPUT)	25
5.1.16.18 <u>V_{CC}</u>	25

	PAGE
5.1.16.19 SID (INPUT).....	25
5.1.16.20 SOD (PUTPUT)	25
5.1.16.21 V _{SS}	26
6. DESCRIPTION OF INTEL 8111-RAM	27
7. DESCRIPTION OF INTEL 8155 RAM/IO/TIMER	28
8. DESCRIPTION OF 8279 (PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE)...	30
8.1 FUNCTIONAL DESCRIPTION.....	32
8.1.1 INPUT MODES	32
8.1.2 OUTPUT MODES	32
8.2 HARDWARE DESCRIPTION	34
8.3 PRINCIPLES OF OPERATION	37
8.3.1 I/O CONTROL AND DATA BUFFERS	38
8.3.2 CONTROL AND TIMING REGISTERS AND TIMING CONTROL ..	38
8.3.3 SCAN COUNTER	39
8.3.4 RETURN BUFFERS AND KEYBOARD DEBOUNCE AND CONTROL..	40
8.3.5 FIFO/SENSGR RAM AND STATUS	40
8.3.6 DISPLAY ADDRESS REGISTERS AND DISPLAY RAM	41
8.4 KEYBOARD/DISPLAY MODE SET	42
8.4.1 8279 COMMANDS	42
8.4.2 KEYBOARD/DISPLAY MODE SET	42
8.4.3 PROGRAM CLOCK	43
8.4.4 READ FIFO/SENSOR RAM	43
8.4.5 READ DISPLAY RAM	44
8.4.6 WRITE DISPLAY RAM	45
8.4.7. DISPLAY WRITE INHIBIT/BLANKING	45

	PAGE
8.4.8 CLEAR	46
8.4.9 END INTERRUPT/ERROR MODE SET	47
8.4.10 STATUS WORD	48
8.4.11 DATA READ	48
8.4.12 DATA WRITE	48
8.5 INTERFACE CONSIDERATIONS	49
8.5.1 SCANNED KEYBOARD MODE, 2-KEY LOCKOUT	49
8.5.2 SCANNED KEYBOARD MODE, N-KEY ROLLOVER	50
8.5.3 SCANNED KEYBOARD - SPECIAL ERROR MODES	50
8.5.4 SENSGR MATRIX MODE	51
8.5.5 DATA FORMAT	51
8.5.6 DISPLAY	53
9. DESCRIPTION OF THE HARDWARE DESIGN	54
9.1 COMMUNICATION BETWEEN CPU AND THE OTHER LSI	55
9.2 CIRCUIT ANALYSIS	57
9.3 INITIALIZATION	58
9.4 CLOCK CIRCUITS	59
9.5 INSTRUCTION TIMING	62
9.6 OPCODE FETCH TIMING	67
9.7 MEMORY READ TIMING	69
9.8 I/O READ TIMING	70
9.9 MEMORY WRITE TIMING	71
9.10 I/O WRITE TIMING	74
9.11 MULTIPLEXED ADDRESS/DATA BUS	74
9.12 MULTIBUS INTERFACE	75

	PAGE
10. THE INTERFACE DESCRIPTION	77
11. PROGRAMMING OF 8279	80
11.1 I/O REGISTER ADDRESSING	80
11.2 PROGRAMMING THE COMMAND REGISTER	80
11.2.1 INSTRUCTION REQUIRED FOR COMMAND REGISTER	81
12. DESCRIPTION OF THE MONITOR ROUTINE USED IN THE SYSTEM	83
13. DISCUSSION	86
14. REFERENCES	87

INTRODUCTION

THE INPUT/OUTPUT INTERFACE FOR SBC 80/05 WILL GIVE THE SBC THE CAPABILITY TO COMMUNICATE WITH OUTSIDE WORLD. THE FOLLOWING THREE PARAGRAPH EX-PLAINS THE RELATION BETWEEN INPUT DEVICE, OUTPUT DEVICE AND 8279 CHIP.

i) THE MATRIX KEYBOARD WILL BE USED AS INPUT DEVICE WHICH WILL MAKE THE USER (FOR THIS SMALL SYSTEM) CAPABLE FOR PUTTING ANY KIND OF INFORMATION INSIDE THE RAM AND BY USING THE MONITOR OR (THE SOFTWARE DESIGNED FOR THIS SYSTEM). THE USER WILL BE ABLE TO PROCESS HIS INFORMATION AND GETTING THE FINAL RESULT IN THE OUTPUT DEVICE.

ii) THE TIL - 313 LIGHT EMITTING DIODES WILL BE USED AS OUTPUT DEVICE, AND WILL DISPLAY THE FINAL RESULT FOR THE PROCESSED DATA.

iii) THIS TWO DEVICES COMMUNICATE WITH 8085 UP THROUGH INTEL 8279 CHIP (PROGRAMMABLE KEYBOARD / DISPLAY INTERFACE). THE PURPOSE OF THE PROJECT IS TO GET FAMILIAR WITH THE HARDWARE AND SOFTWARE OF THE 8085 UP AND 8279 PROGRAMMABLE KEYBOARD AND DISPLAY INTERFACE LSI.

HARDWARE INTERFACE DESIGN AND SOFTWARE

TECHNICAL REPORT

1. REQUIREMENT

It is required to design the HARDWARE INTERFACE for BC 80/85 single board computer which is a member of INTEL' complete line of SBC 80 computer.

The HARDWARE INTERFACE will give the capability for the Single Board Computer to use a keyboard as input device or TTY and light emitting diodes led's as output device.

2. EQUIPMENTS AVAILABLE FOR THE PROJECT

- a) SBC 80/05 Single Board Computer which contains:
 - 1) INTEL 8085 CPU
 - 2) INTEL 8111-RAM provide 256 bytes of static read and write memory
 - 3) INTEL 8155 combination RAM/IO/TIMER
 - 4) Two sockets are providing for installing up to 4k bytes of ROM (which stores the required operating system)

SBC 80/05 provides jumber-selectable interrupts to the four interrupt request inputs of the 8085 CPU.

- b) INTEL 3279-PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE
- c) 2 TTL DECODER 74LS156
- d) MATRIX KEYBOARD
- e) 4 TIL-313
- f) 16 TRANSISTOR (2N2907)

3. SYSTEM EXPANSION

Processing power, memory and I/O capacity may be increased in SBC 80/05 based system with single -5v power by adding standard INTEL expansion boards.

Memory may be expanded to 65,536 bytes by adding user-specified combinations of RAM boards, PROM boards or Combination boards. Input/Output capacity may also be increased by adding SBC 80 Analog I/O Boards.

4. DESCRIPTION FOR 80/05 SBC

The SBC 80/05 Single Board Computer is controlled by an Intel 8085 cpu, which includes six 8 bit general purpose registers, may be addressed individually or in pairs, which allows both single-precision and double-precision operations. The minimum on board execution time is 2.03 usec.

The 8085 cpu has a 16 bit program counter which allows direct addressing of up to 65,536 bytes of memory and external stack located within any portion of Read/write memory, may be used

as a last-in/first-out storage area for the contents of the Program Counter. Flags, accumulator and all six general-purpose registers and 16 bit stack pointer registers which control the addressing of the external stack and, this will allow subroutine nesting. This is bounded only by the system memory size.

5. DESCRIPTION OF INTEL 8085 UP

The following sections break the 8085 into functional blocks and describe each in detail.

5.1.1. Arithmetic Section

The arithmetic section of the processor contains the basic data manipulation functions of the 8085 and can be divided into the following blocks:

- a) Arithmetic Logic Unit (ALU)
- b) Accumulator
- c) Carry Flag
- d) Instruction Decoder

In a typical operation data sorted in the accumulator is combined in the ALU with data from another source on the internal bus (such as a register or I/O port) and the result is sorted in the accumulator or another register. The following is a more detailed description of the function of each block.

5.1.2. Instruction Decoder

The operation code (op code) portion of each program instruction is stored in the Instruction Decoder and converted to output which control the function of each of the blocks of the Arithmetic Section. These lines control the source of data and the destination register as well as the function performed in the ALU.

5.1.3. Arithmetic Logic Unit

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under control of the Instruction Decoder. The ALU can perform the following functions:

- Add With or Without Carry
- And, OR, Exclusive OR
- Increment/Decrement
- Bit Complement
- Rotate Left, Right
- Swap Nibbles
- BCD Decimal Adjust

If the operation performed by the ALU results in a value represented by more than 8 bits (overflow of most significant bit) a Carry Flag is set in the Program Status Word.

5.1.4. Accumulator

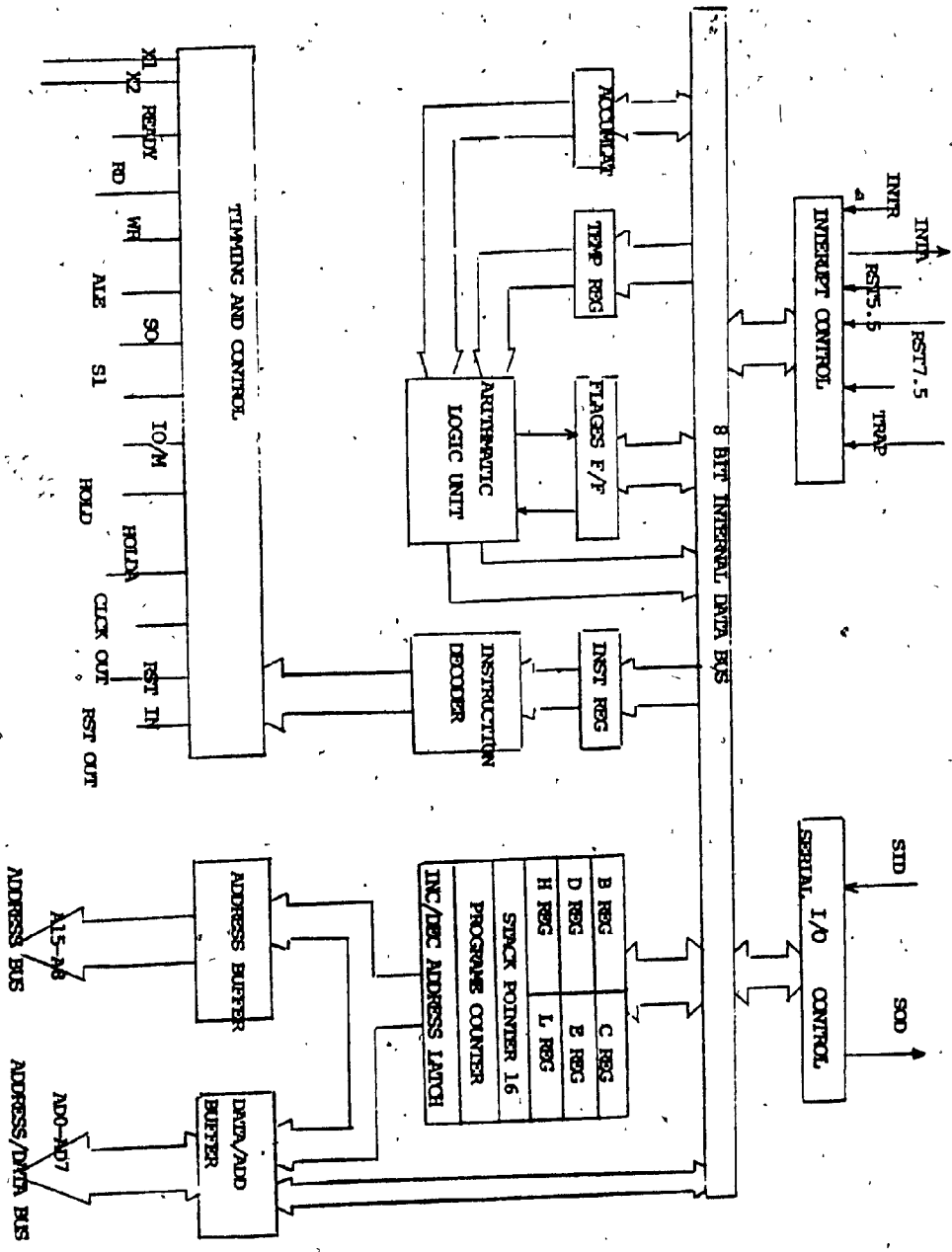
The accumulator is the single most important data register in the processor being one of the sources of input to the ALU and often the destination of the result of operations performed in the ALU. Data to and from I/O ports and memory also normally passes through the accumulator.

5.1.5. Program Memory

Resident program memory consists of 1024 words eight bits wide which are addressed by the program counter. In the CPU, the memory is ROM which is mask programmable at the factory. Program code is completely interchangeable among the three versions.

5.1.6. Bus

Bus is also 8-bit port which is true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, Bus can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed however.



As a static port, data is written and latched using the OUTL instruction and inputted using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding RD and WR output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the WR output line and output data is valid at the trailing edge of Wr. A read of the port generates a pulse on the RD output line and input data must be valid at the trailing edge of RD. When not being written or read, the BUS lines are in a high impedance state.

5.1.7. Test and INT Inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and INT. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and INT pins have other possible functions as well.

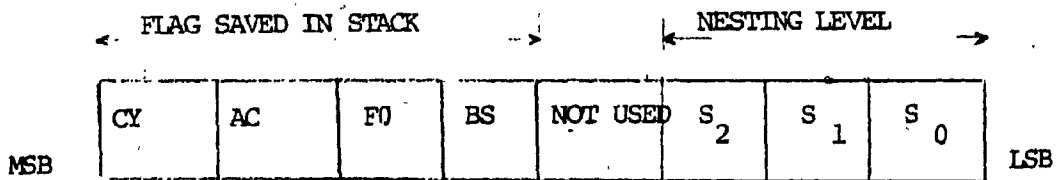
5.1.8. Program Counter and Stack

The Program Counter is an independent counter while the Program Counter Stack is implemented using pairs of registers in the Data Memory Array. Only 10 bits of the Program Counter are used to address the 1024 words of on-board program memory while the most significant two bits are used for external Program Memory fetches. The Program Counter is initialized to zero by activating the Reset line.

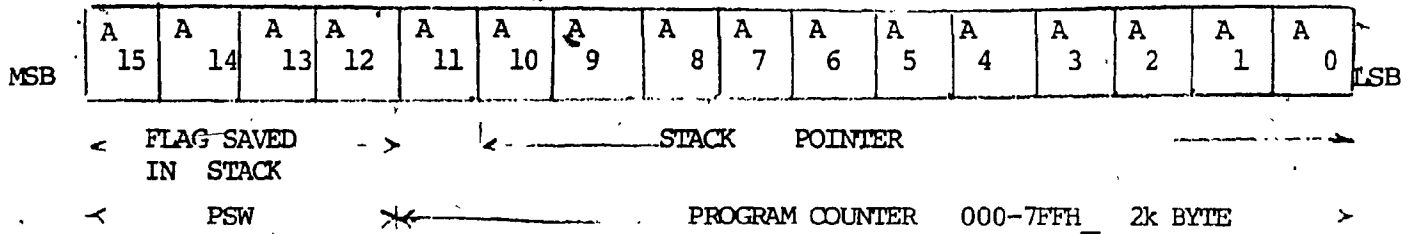
An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit Stack Pointer which is a part of the Program Status Word (PSW). Data RAM locations 8 thru 23 are available as shown in the following figures.

The Stack Pointer when initialized to 000 points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111 (look to the Program Counter Map).

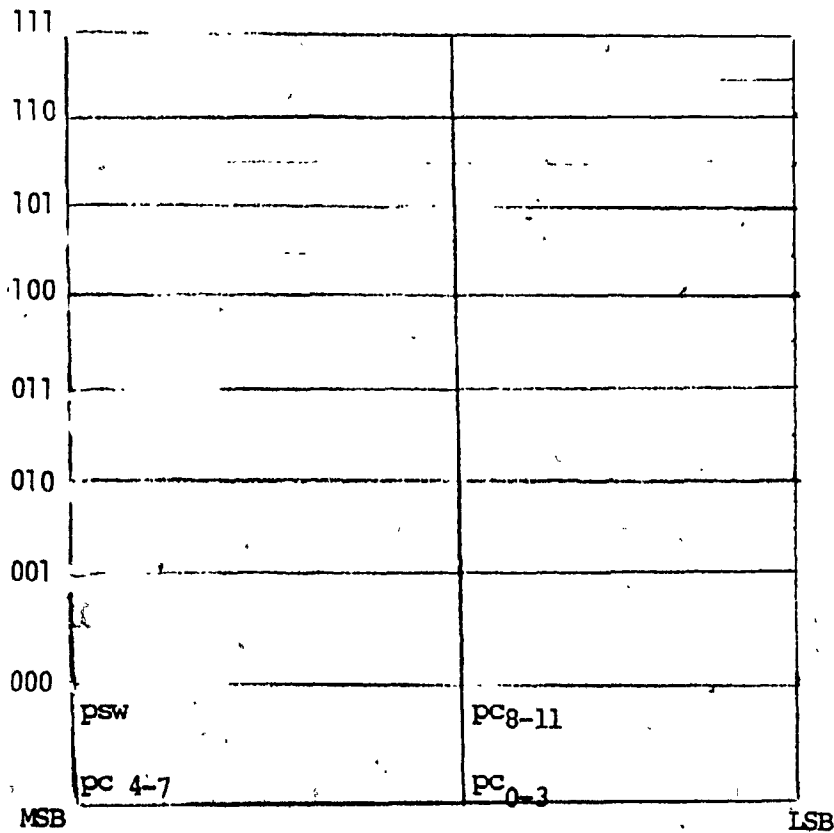
The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the Stack Pointer to be decremented and the contents of the resulting register pair to be transferred to the Program Counter.



PROGRAM COUNTER MAP



NESTING LEVEL OF POINTER



5.1.9 Program Status Word

An 8-bit status word which can be loaded to and from the accumulator exists called the Program Status Word (PSW). The accompanying figure shows the information available in the word. The Program Status Word is actually a collection of flip-flops throughout the machine which can be read or written as a whole. The ability to write to PSW allows for easy restoration of machine status after a power down sequence.

The upper four bits of PSW are stored in the Program Counter Stack with every jump to subroutine or interrupt vector and are optionally restored upon return with the RETR instruction. The RET return instruction does not update PSW.

The PSW bit definitions are as follows:

- Bits 0 - 2: Stack Pointer bits (S₀, S₁, S₂)
- Bit 3: Not used ("1" level when read)
- Bit 4: Working Register Bank Switch Bit (BS)
 - 0 Bank 0
 - 1 Bank 1
- Bit 5: Flag 0 bit (F0) user controlled flag which can be complemented or cleared, and tested with the conditional jump instruction JF0.

Bit 6: Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DAA.

Bit 7: Carry (CY) carry flag which indicates that the the previous operation has resulted in overflow of the accumulator.

5.1.10 Conditional Branch Logic

The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. By using the conditional jump instruction the following conditions can effect a change in the sequence of the program execution.

5.1.11 Interrupt

An interrupt sequence is initiated by applying a low "0" level input to INT pin. Interrupt is level triggered and active low to allow "WIRE ORing" of several interrupt sources at the input pin. The Interrupt line is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at location 3 in program memory as soon as all cycles of the current instruction are complete. As in any CALL to subroutine, the Program Counter and Program Status word are saved in the stack.

Program Memory location 3 usually contains an unconditional jump to an interrupt service subroutine elsewhere in program memory. The end of an interrupt service subroutine is signalled by the execution of a Return and Restore Status instruction RETR. The interrupt system in single level mode in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR re-enables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. This sequence holds true also for an internal interrupt generated by timer overflow. If an internal timer/counter generated interrupt and an external interrupt are detected at the same time, the external source will be recognized. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the Counter (one less than terminal count), and enabling the event counter mode. A "1" to "0" transition on the T1 input will then cause an interrupt vector to location 7.

5.1.12 Interrupt Timing

The interrupt input may be enabled or disabled under Program Control using the EN I and DIS I instructions. Interrupts are disabled by Reset and remain so until enabled by the users program.

An interrupt request must be removed before the RETR instruction is executed upon return from the service routine otherwise the processor will re-enter the service routine immediately. Many peripheral devices prevent this situation by resetting their interrupt request line whenever the processor accesses (Reads or Writes) the peripherals data buffer register. If the interrupting device does not require access by the processor, one output line of the CPU may be designated as an "interrupt acknowledge" which is activated by the service subroutine to reset the interrupt request. The INT pin may also be tested using the conditional jump instruction JN1. This instruction may be used to detect the presence of a pending interrupt before interrupts are enabled. If interrupt is left disabled, INT may be used as another test input like T0 and T1.

5.1.13 Timer/Counter

The CPU contains a counter to aid the user in counting external events and generating accurate time delays without placing a burden on the processor for these functions. In both modes the counter operation is the same, the only difference being the source of the input to the counter. It can work as a counter, an event counter and as a timer.

The 8-bit up binary counter is pre-settable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice versa. The counter content is not affected by Reset and is initialized solely by the MOV T,A instruction. The counter is stopped by a START T instruction or as an event counter by a START CNT instruction. Once started the counter will increment to its maximum count (FF) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or Reset.

The increment from maximum count to zero (overflow) results in the setting of an overflow flag flip-flop and in the generation of an interrupt request. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by executing a JTF or by Reset. The interrupt request is stored in a latch and then ORed with the external interrupt input INT. The timer interrupt may be enabled or disabled independently of external interrupt by the EN TGNTI and DIS TGNTI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer or counter service routine may be stored. If timer and external interrupts occur simultaneously, the external source will be recognized and the Call will be to location 3.

Since the timer interrupt is latched it will remain pending until the external device is serviced and immediately be recognized upon return for the service routine. The pending timer interrupt is reset by the Call to location 7 or may be removed by executing a DISTGNTI instruction. Execution of a START CNT instruction connects the T1 input pin to the counter input and enables the counter. Subsequent high to low transitions on T1 will cause the counter to increment. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 7.5 usec when using a 6MHz crystal)--there is no minimum frequency. T1 input must remain high for at least 100ns after each transition.

As a timer, execution of a START T instruction connects an internal clock to the counter input and enables the counter. The internal clock is derived by passing the basis 400 KHz machine cycle clock ALE through a - 32 prescaler. The prescaler is reset during the START T instruction. The resulting 12.5 KHz clock increments the counter every 80 usec (assuming 6 MHz XTAL). Various delays between 80 usec and 20 msec (256 counts) can be obtained by presetting the counter and detection overflow. Times longer than 20 msec may be achieved by accumulating multiple overflows in a register under software control.

For time resolution less than 80 usec an external clock can be applied to the T1 input and counter operated in the event counter mode. A1e divided by 3 or more can serve as this external clock. Very small delays or "fine tuning" of larger delays can be easily accomplished by software delay loops.

5.1.14 Clock and Timing Circuits

Timing generation for the CPU is completely self-contained with the exception of a frequency reference which can be XTAL, series RC, or external clock source. The Clock and Timing circuitry can be divided into the following functional blocks:

5.1.14.1 Clock Generation

The output of the oscillator is divided by 3 in the State Counter to create a clock which defines the state times of the machine (CLK). CLK can be made available on the external pin T0 by executing an ENTO CLK instruction. The output of CLK on T0 is disabled by Reset of the processor.

5.1.14.2 Cycle Counter

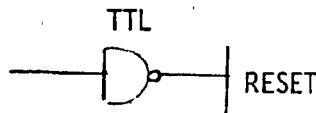
CLK is then divided by 5 in the Cycle Counter to provide a clock which defines a machine cycle consisting of 5 machine states.

This Clock is called Address Latch Enable (ALE).

5.1.15 Initialization or Reset

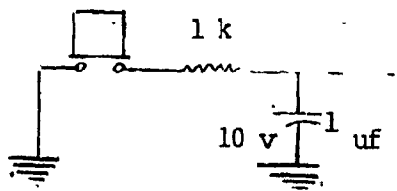
The reset input provides a means for initialization for the processor. This Schmitt-Trigger input has an internal pullup resistor which in combination with an external 1 μ f capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset. If the reset pulse is generated externally the reset pin must be held at ground (.5V) for at least 50 milliseconds after the power supply is within tolerance.

EXTERNAL RESET



OPEN COLLECTOR
OR ACTIVE PULLUP

POWER ON RESET



Reset performs the following functions:

1. Sets program counter to zero.
2. Sets stack pointer to zero.
3. Selects register bank 0.
4. Selects memory bank 0.
5. Sets BUS to high impedance state.
6. Sets Ports 1 and 2 to input mode.
7. Disables interrupts (timer and external)
8. Stops timer.
9. Clears timer flag.
10. Clears F0 and F1.
11. Disables clock output from T0.

5.1.16 Pin Description For 8085 UP

The following describes the function of each pin:

5.1.16.1 AD₀₋₇ (Input/Output 3-State)

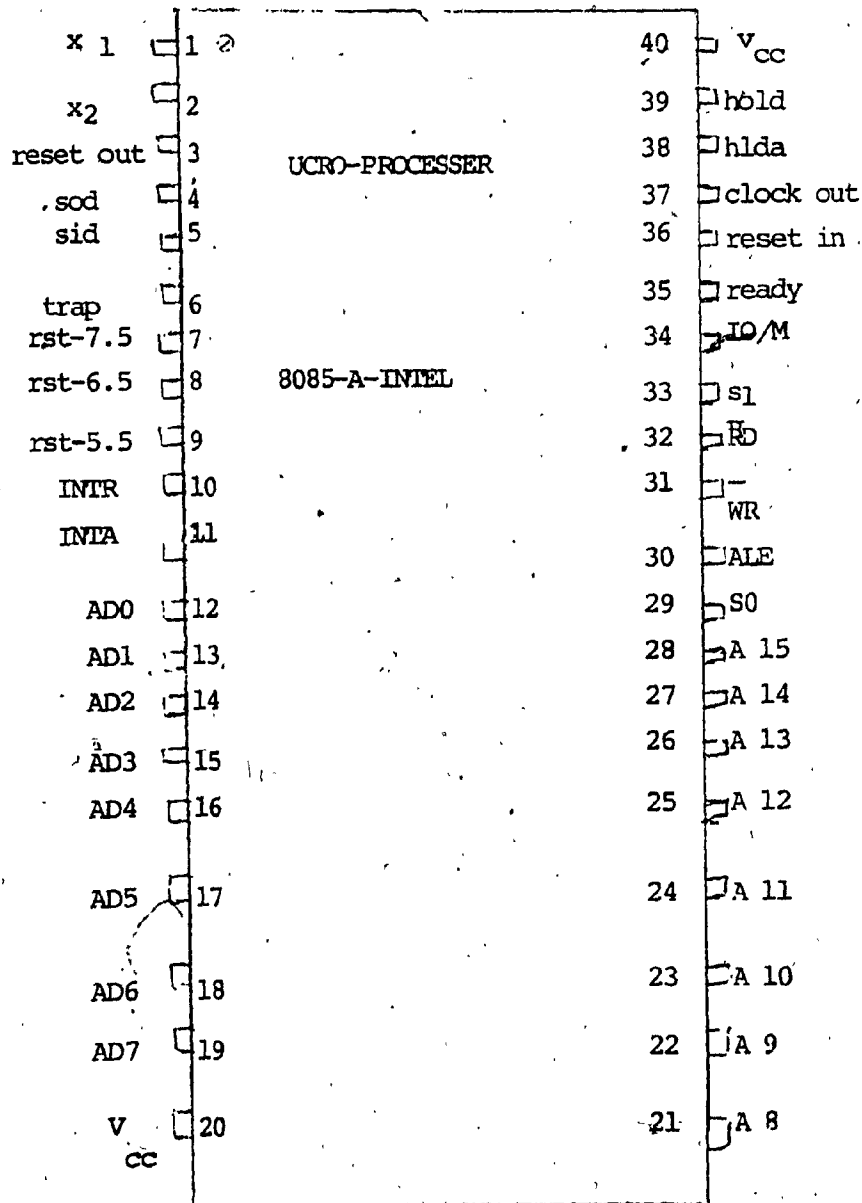
Multiplexed Address/Data Bus; Lower 8-bits of the memory address (or I/O address) appear on the bus during the first clock cycle of a machine state. It then becomes the data bus during the second and third clock cycles.

5.1.16.2 A_{8-A₁₅} (Output 3-State)

Address Bus; The most significant 8-bits of the memory address or the 8-bits of the I/O address,

3-stated during Hold and Halt modes.

8085 PIN CONFIGURATION



5.1.16.3. S₀, S₁ (Output)

Data Bus Status. Encoded status of the bus cycle:

<u>S₁</u>	<u>S₀</u>	
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

S₁ can be used as an advanced R/W Status

5.1.16.4. RD (Output 3-state)

READ; indicates the selected memory of I/O device is to be read and that the Data Bus is available for the data transfer. 3-stated during Hold and Halt.

5.1.16.5. WR (Output 3-state)

WRITE; indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of WR. 3-stated during Hold and Halt modes.

5.1.16.6. ALE (Output)

Address Latch Enable: it occurs during the first clock cycle of a machine state and enables the address to get latched into the on-chip latch of peripherals.

The falling edge of ALE is set to guarantee set-up and hold times for the address information. ALE can also be used to strobe the status information. ALE is never 3-stated.

5.1.16.7. READY (Input)

If Ready is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If Ready is low, the CPU will wait for Ready to go high before completing the read or write cycle.

5.1.16.8. HOLD (Input)

HOLD; indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request, will relinquish the use of buses as soon as the completion of the current machine cycle. Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, RE, WR, and IO/M lines are 3-stated.

5.1.16.9. HLDA (Output)

HOLD ACKNOWLEDGE; indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle.

HLDA goes low after the Hold request is removed.

The CPU takes the buses one half clock after

HLDA goes low.

5.1.16.10 INTR (Input)

INTERRUPT REQUEST; is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an INTA will be issued. During this cycle a RESTART of CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

5.1.16.11 TRAP (Input)

Trap interrupt is nonmaskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or interrupt Enable. It has the highest priority of any interrupt.

5.1.16.12 INTA (Output)

INTERRUPT ACKNOWLEDGE; is used instead of (and has the same timing as) RD during the Instruction cycle after an INTR is accepted.

It can be used to activate the 8259 Interrupt chip or some other interrupt port.

RST 5.5

RST 6.5 (inputs)

RST 7.5

RESTART INTERRUPTS; These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted.

RST 7.5 Highest Priority

RST 6.5

RST 5.5 Lowest Priority

The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR.

5.1.16.13 RESET OUT (Output)

Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.

5.1.16.14 RESET IN (Input)

Reset sets the Program Counter to zero and resets the interrupt Enable and HLDA flip-flops. None of the other flags or registers (except the instruction register) are affected. The CPU is held in the reset condition as long as Reset is applied.

5.1.16.15 CLK (Output)

Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X₁ X₂ input period.

5.1.16.16 IO/M (Output)

IO/M indicates whether the Read/Write is to memory or I/O. Tristated during Hold and Halt modes.

5.1.16.17 X₁, X₂ (Input)

Crystal or R/C network connections to set the internal clock generator. X₁ can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.

5.1.16.18 VCC

5 volt supply

5.1.16.19 SID (Input)

Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

5.1.16.20 SOD (Output)

Serial output data line. The output SOD is set

or reset as specified by the SIM instruction.

5.1.16.21 VSS

Ground Reference.

6. DESCRIPTION OF INTEL 8111-RAM

Two Intel 8111-A4 RAM chips provide 256 bytes of static read/write memory; an Intel 8155 combination RAM/IO/Timer provides an additional 256 bytes of static read/write memory. Two sockets are provided for installing up to 4k bytes of nonvolatile read-only memory (ROM), which may be added in 2k byte increments using Intel 2716 Ultraviolet Erasable and Reprogrammable ROM's (EPROM's) or 8316E Masked ROM's. Optionally, if only 2k bytes are required, ROM may be added in 1k byte increments using Intel 8708 EPROM's or 8308 Masked ROM's.

Twenty-two programmable parallel I/O lines are implemented using the I/O ports of the Intel 8155 RAM/IO/Timer. The system software is used to configure the I/O lines in any combination of unidirectional input or output ports. The I/O interface may be customized to meet specific peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 22 programmable I/O lines and single grounding lines are brought to a 50-pin edge connector (J1).

7. DESCRIPTION OF INTEL 8155 RAM/IO/TIMER

The Intel 8155 RAM/IO/Timer includes a programmable 14-bit interval timer, which is configured by software to meet the system requirements. Whenever a given time delay is needed, software commands to the timer select the desired operating mode. The current contents (present count) of the timer counter and the timer mode bits may be read at any time during system operation. There are four timer operating modes:

- a. Timer Out goes low during the second half of count... Therefore, the count loaded in the Count Length Register should be twice the timeout desired.
- b. Timer Out remains high until the first half of the count has been completed, and goes low for the second half of the count. The count length is automatically reloaded when the terminal count is reached.
- c. A single low pulse is generated upon reaching the terminal count; this function is useful for generating real-time clocks.
- d. A Divide-by-N Counter generates a repetitive Timer Out low pulse; a new pulse train is initiated every time the terminal count is reached.

Serial I/O capability is provided through the Serial Input Data (SID) and Serial Output Data (SOD) functions of the CPU.

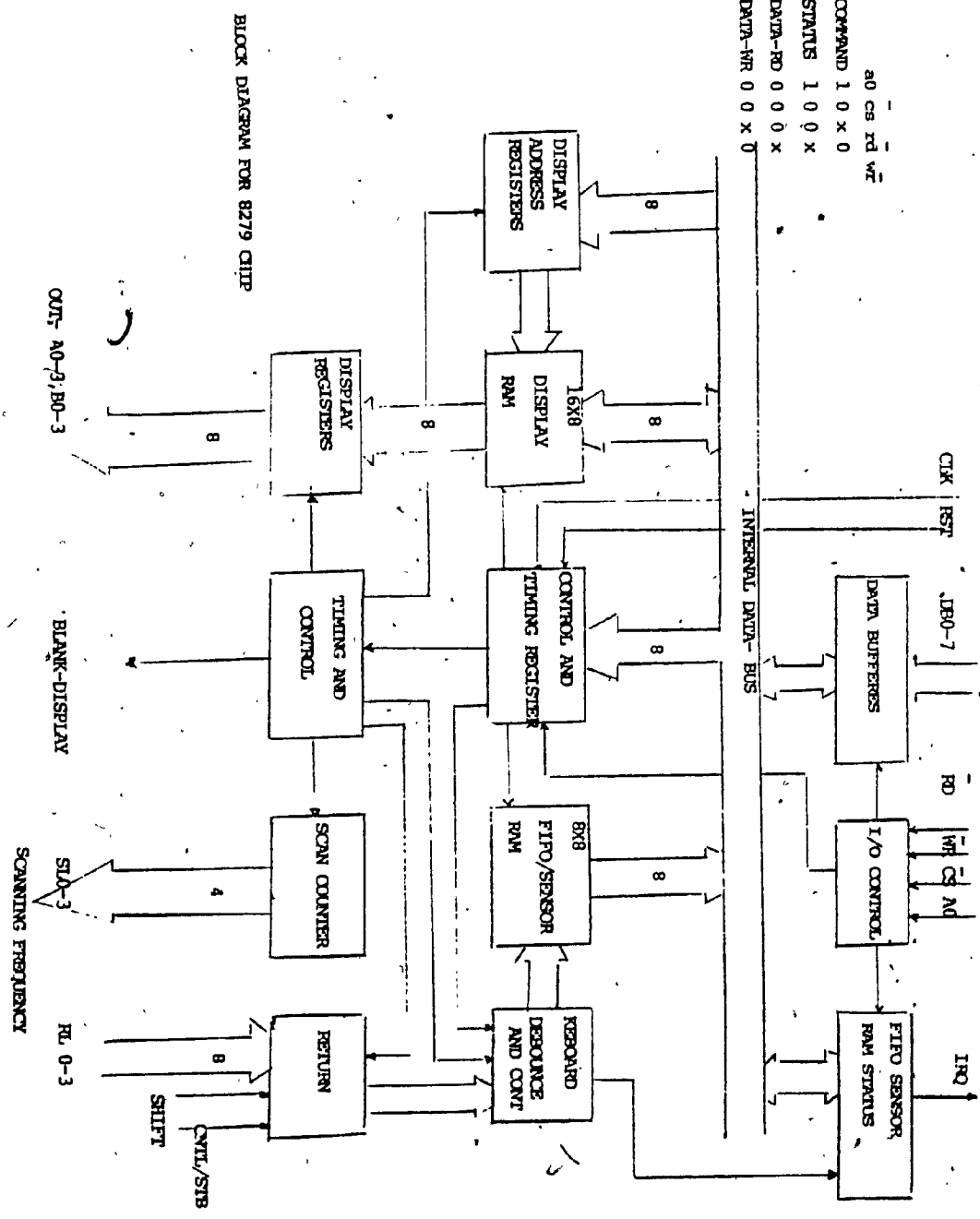
These functions are controlled exclusively by software through the execution of RIM and SIM instructions. The baud rate for the serial I/O interface is determined by the system time available for the execution of serial I/O support software. Hence, the maximum baud rate supported by the SBC 80/05 is solely dependent on the overall system real-time software requirements. Serial I/O signals are TTL compatible and sockets are provided on the board for optional installation of RS232C line drivers and receivers.

8. DESCRIPTION OF 8279 (PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE)

The Intel 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

20 CS rd wr
 COMMAND 1 0 x 0
 STATUS 1 0 0 x
 DATA-RD 0 0 0 x
 DATA-WR 0 0 x 0



8.1 FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

8.1.1 INPUT MODES

.Scanned Keyboard - with encoded (8 X 8 X 4 key keyboard) or decoded (4 X 8 X 4 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.

.Scanned Sensor Matrix - with encoded (8 X 3 matrix switches) or decoded (4 X 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.

.Strobed Input - Data on return lines during control line strobe is transferred to FIFO.

8.1.2 OUTPUT MODES

.8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit.

.Right entry or left entry display formats.

.Other features of the 8279 include:

.Mode programming from the CPU.

.Programmable clock to match the 8279 scan times to the CPU cycle time.

.Interrupt output to signal CPU when there is keyboard or sensor data available.

.An 8 byte FIFO to store keyboard information.

.16 byte internal Display RAM for display refresh.

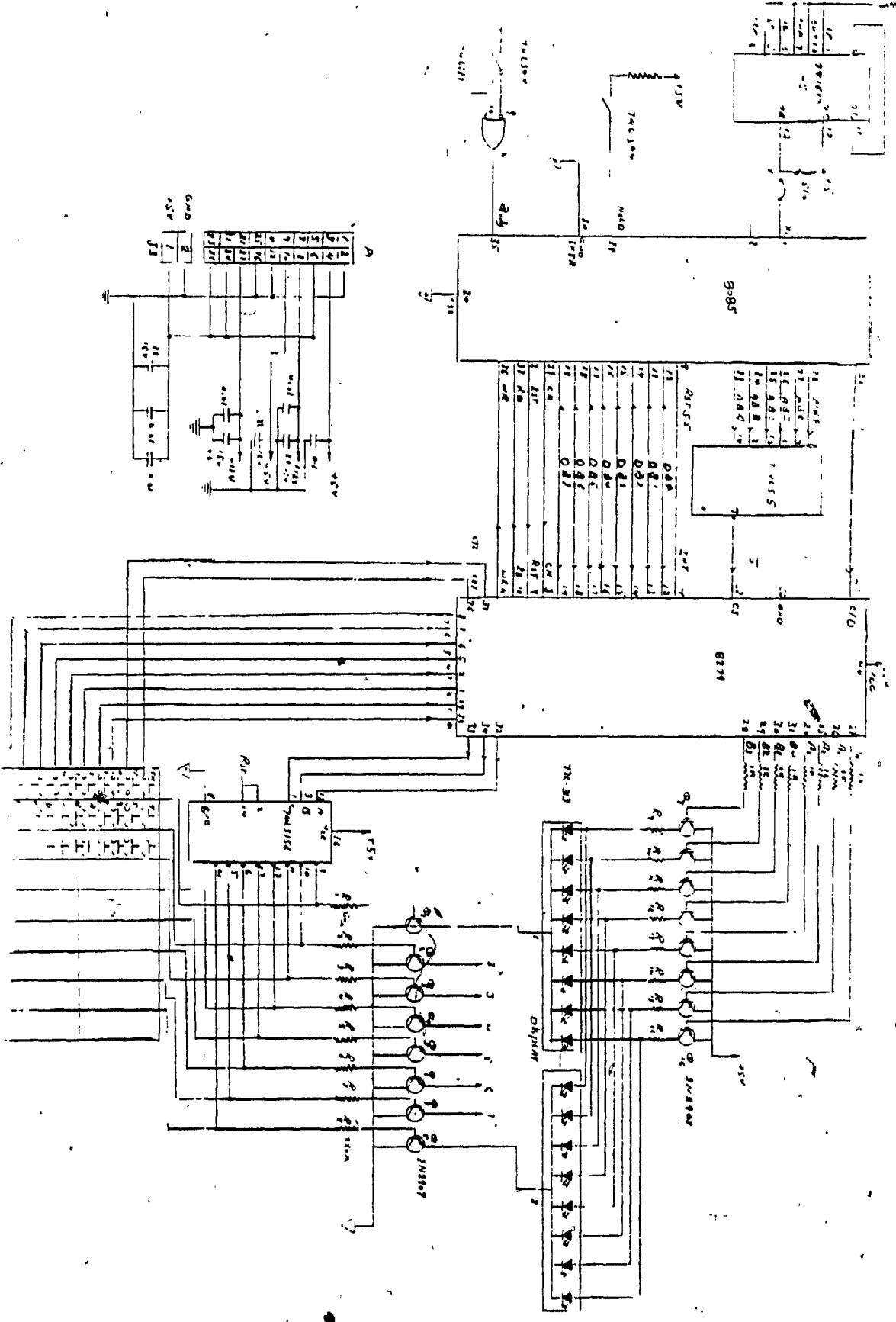
This RAM can also be read by the CPU.

8.2 HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

No. Of

<u>Pins</u>	<u>Designation</u>	<u>Function</u>
8	DB0-DB7	Bi-directional data bus. All data and commands between the CPU and the 8279 are transmitted on these lines.
1	CLK	Clock from system used to generate internal timing.
1	RESET	A high signal on this pin resets the 8279.
1	CS	Chip Select. A low on this pin enables the interface functions to receive or transmit.
1	A0	Buffer Address. A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
2	RD, WR	Input/Output read and write. These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
1	IRQ	Interrupt Request. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM.



No. Of

PinsDesignationFunction

The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.

2

Vss, Vcc

Ground and power supply pins.

4

SL0-SL3

Scan Lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).

8

RL0-RL7

Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pull-ups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.

1

SHIFT

The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.

No. Of		
<u>Pins</u>	<u>Designation</u>	<u>Function</u>
1	CNTL/STB	For Keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
4	OUT A0-OUT A3	These two ports are the outputs for the 16 X 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL0-SL3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8 bit port.
4	OUT B0-OUT B3	
1	BD	Blank Display. This output is used to blank the display during digit switching or by a display blanking command.

8.3 PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device.

8.3.1. I/O CONTROL AND DATA BUFFERS

The I/O control section uses the CS, A₀, RD and WR lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by CS. The character of the information, given or desired by the CPU, is identified by A₀. A logic one means the information is a command or status. A logic zero means the information is data. RD and WR determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected (CS 1), the devices are in a high impedance state. The drivers input during WR.CS and output during RD.CS.

8.3.2. CONTROL AND TIMING REGISTERS AND TIMING CONTROL

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A₀=1 and then sending a WR. The command is latched on the rising edge of WR. The command is then decoded and the appropriate function is set.

The timing control contains the basic timing counter chain. The first counter is a N prescaler that can be programmed to match the CPU cycle time to the internal timing. The prescaler is software programmed to a value between 2 and 31. A value which yields an internal frequency of 100 kHz gives a 5.1 ms keyboard scan time and 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

8.3.3. SCAN COUNTER

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note that when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

8.3.4 RETURN BUFFERS AND KEYBOARD DEBOUNCE AND CONTROL

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrox modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

8.3.5 FIFO/SENSOR RAM AND STATUS

This block is a dual function 8 X 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error.

The status can be read by an RE with CS low and A0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In scanned Sensor Matrix mode, the memory is Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

8.3.6 DISPLAY ADDRESS REGISTERS AND DISPLAY RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See interface Considerations for details.

8.4 SOFTWARE OPERATION

8.4.1 8279 COMMANDS

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with CS low and A0 high and are loaded to the 8279 on the rising edge of WR.

8.4.2 KEYBOARD/DISPLAY MODE SET

	MSB							LSB
CODE:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0 0 8 8-bit character display - Left entry

0 1 16 8-bit character display - Left entry

(default after reset)

1 0 8 8-bit character display - Right entry

1 1 16 8-bit character display - Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard - 2 Key Lockout
0	0	1	Decoded Scan Keyboard - 2-Key Lockout
0	1	0	Encoded Scan Keyboard - N-Key Rollover
0	1	1	Decoded Scan Keyboard - N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

8.4.3 PROGRAM CLOCK

CODE: 0 0 1 P P P P P

Where P P P P P is the Prescaler value 2 to 31. The programmable prescaler divides the external clock by P P P P P to get the basic internal frequency. Choosing a divisor that yields 100 KHz will give the specified scan and debounce times. Default after a reset pulse (but not a program clear) is 31.

8.4.4 READ FIFO/SENSOR RAM

CODE: 0 1 0 AI x A A A - X Don't Care

Where AI is the Auto-Increment flag for the Sensor RAM and AAA is the row that is going to be read by the CPU. AI and AAA are used only if the mode is set to Sensor Matrix. This command is used to specify that the source

of data reads (CS . Rd . Ao) by the CPU is the FIFO/Sensor RAM. No additional commands are necessary as long as data is desired from the FIFO/Sensor RAM. Another command is necessary if reading is desired from a different row than has been selected. If AI is a one, the row select counter will be incremented after each read so the next read will be from the next Sensor Ram row.

In the Auto Increment mode for reading data from the FIFO/Sensor RAM, each read advances the address by one so that the next read is from the next character. This Auto Incrementing has no effect on the display.

8.4.5 READ DISPLAY RAM

CODE: 0 1 1 AI A A A A

Where AI is the Auto-Increment flag for the Display RAM and AAAA is the character that the CPU is going to read next. Since the CPU uses the same counter for reading and writing, this command also sets the next write location and Auto-Increment mode. This command is used to specify the display RAM as the data source for CPU data reads. If AI is set, the character address will be incremented after each read (or write) so that the next read (or write) will be from (to) the next

CA has the combined effect of CD and CF. CA uses the CD clearing code to determine how to clear the Display RAM. CA also resets the internal timing chain to resynchronize it.

8.4.6 WRITE DISPLAY RAM

CODE: 1 0 0 AI A A A A

Where AI is the Auto-Increment flag for the Display RAM and AAAA is the character that the CPU is going to write next. The addressing and Auto-Increment are identical to Read Display RAM. The difference is that Write Display RAM does not affect the source of CPU reads. The CPU will read from whichever RAM (Display or FIFO/Sensor) was last specified. This command will, however, change the location the next Display RAM read will be from if that source was specified.

8.4.7 DISPLAY WRITE INHIBIT/BLANKING

CODE: 1 0 1 X IW IW BL BL
 A B A B

Where IW is Inhibit Writing (nibble A or B) and BL is Blanking (nibble A or B). If the display is being used as a dual 4-bit display, then it is necessary to mask

character.

8.4.8 CLEAR

CODE: 1 1 0 C_D C_D C_F C_A

Where C_D is Clear Display, C_F is Clear FIFO Status (including interrupt), and C_A is Clear All. C_D is used to clear all positions of the Display RAM to a programmable code. All ones, all zeros and hexadecimal 20 are possible. The 2 least significant bits of C_D are also used to specify the blanking code (see below).

C _D	C _D	C _D	
0	X		All Zeros (X Don't Care)
1	0		AB Hex 20 (0010 0000)
1	1		All Ones

Enable clear display when 1 (or by C_A 1)

Clearing the display takes approximately 160 us. During this time the CPU cannot write to the Display RAM. The MSB of the FIFO status word will be set during this time. C_F set the FIFO status to empty and resets the interrupt output line. After execution of a clear command with C_F set, the Sensor Matrix mode RAM pointer will be set to row 0.

one of the 4-bit halves so that entries to the Display from the CPU do not affect the other half. The IW flags allow the programmer to do this. It is also useful to be able to blank either half when that half is not to be displayed. The BL flags blank the display. The next command sets the output code to be used as a "blank". Default after reset is all zeros. Note that to blank a display formatted as a single 8-bit output, it is necessary to set both BL flags to entirely blank the display. A "1" sets the flag. Reissuing the command with "0" resets the flag.

8.4.9 END INTERRUPT/ERROR MODE SET

CODE: 1 1 1 E X X X X - X Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM.

(The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode - if the E bit is programmed to "1", the chip will operate in the special Error mode. (For further details, see Interface Considerations Section).

8.4.10 STATUS WORD

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and CS and RD are low.

See Interface Considerations for more detail on status word.

8.4.11 DATA READ

Data is read when A_0 , CS and Rd are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of ~~RD~~ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set.

FIFO reads always increment (if no error occurs) independent of AI.

8.4.12 DATA WRITE

Data that is written with A_0 , CS and WR low is always written to the Display RAM. The address is specified by the latest Read Display ~~or~~ Write Display command. Auto-Incrementing on the rising edge of WR occurs if AI set by the latest display command.

8.5 INTERFACE CONSIDERATIONS

8.5.1 SCANNED KEYBOARD MODE, 2-KEY LOCKOUT

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

8.5.2 SCANNED KEYBOARD MODE, N-KEY ROLLOVER

With N-Key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

8.5.3 SCANNED KEYBOARD - SPECIAL ERROR MODES

For N-key rollover mode the user can program a special error mode. This is done by the "End interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a ~~simultaneous multiple~~ depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. The error flag is reset by sending the normal CLEAR command with Cf = 1.

8.5.4 SENSOR MATRIX MODE

In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them. The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt Command if the Auto-Increment flag is set to one.

8.5.5 DATA FORMAT

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted).

CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

MSB				LSB
CNTL	SHIFT	SCAN		RETURN

SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch position maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed inputs ports could be tied to the return lines and scanned by the 8279.

MSB								LSB
RL7	RL6	RL5	RL4	RL3	RL2	RL1		RL0

In Strobed Input mode, the data is also entered to the FIFO from the return lines.

The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.

MSB								LSB
RL7	RL6	RL5	RL4	RL3	RL2	RL1	RL0	

8.5.6 DISPLAY

8.5.6.1 Left Entry

Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.

LEFT ENTRY MODE (AUTO INCREMENT)

Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

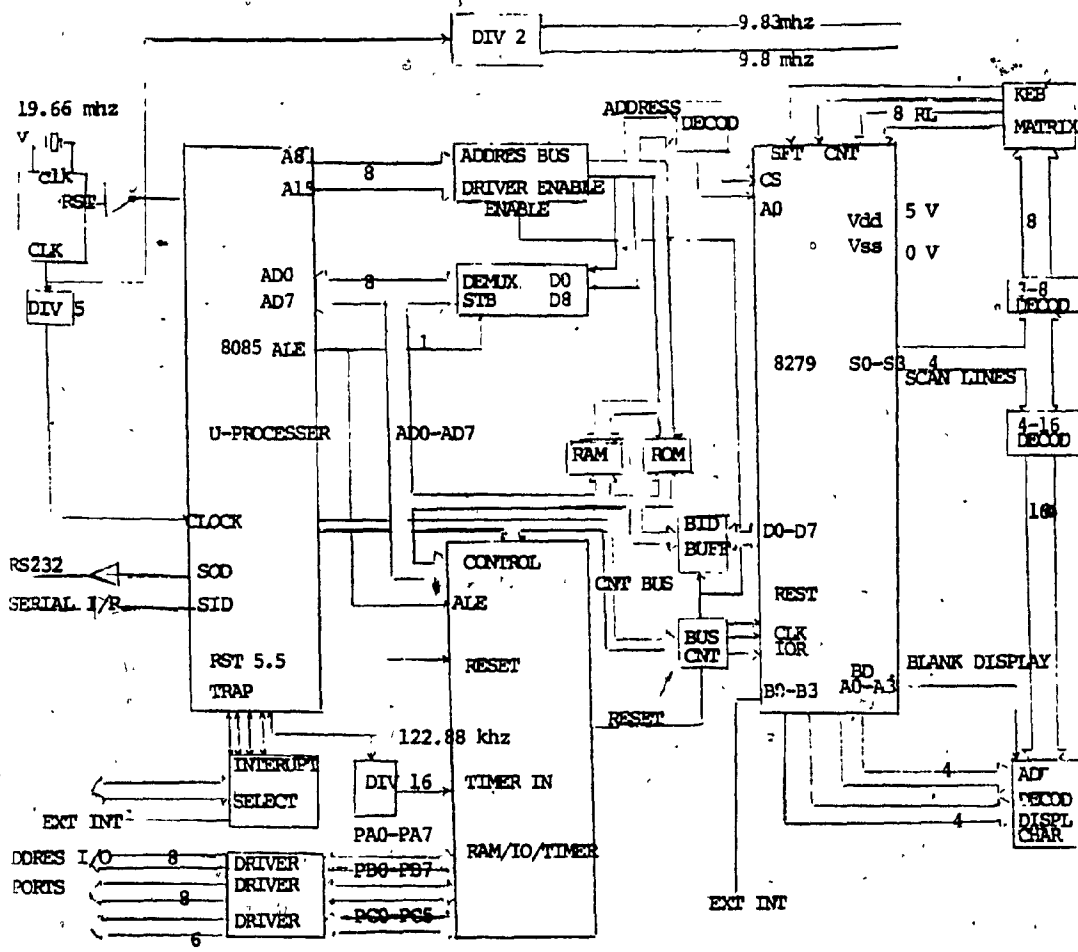
9. DESCRIPTION OF THE HARDWARE DESIGN

The SBC 80/05 is composed of the following functional blocks:

- a) Clock Generator (including power-up reset).
- b) CPU (including interrupt control).
- c) Bus Interface (address bus drivers, data bus drivers, and bus controllers).
- d) Random Access Memory (RAM).
- e) Read - Only Memory (ROM/EPROM).
- f) Programmable Keyboard / Display Interface
- g) Serial I/O Interface
- h) Parallel I/O Interface (including programmable timer).

9.1. COMMUNICATION BETWEEN CPU AND THE OTHER LSI

CPU, which is the heart of the system, performs all the system processing functions and generates the address and control signals required to access memory and I/O ports. The CPU multiplexes the 8 - bit data bus and the lower eight bits of the address bus. During the first part of the machine cycle, the lower eight bits on the address/data bus are latched into Demultiplexer U18 and RAM/I/O/Timer U15. During the remainder of the machine cycle the bus is used for memory and I/O data transfers. The CPU responds to interrupt requests originating from jumper-selectable sources. As shown in block diagram, these interrupt requests may be generated by the on-board timer and parallel I/O ports, by one or more devices via the Multibus, or by an external source.



SBC 80/05 BLOCK DIAGRAM WITH 8279 INCLUDED

The 8155 timer is a programmable 14-bit binary down-counter that counts the input pulses and outputs either a square wave or a pulse when the "terminal-count" is reached. The count length and timer output mode are loaded under program control.

The four selectable timer modes are as follows:

- a) Timer Out goes low during the second half of count. Therefore, the count loaded in the Count Length Register should be twice the timeout desired.
- b) Timer Out remains high until the first half of the count has been completed and goes low for the second half of the count. The count length is automatically reloaded when the terminal count is reached.
- c) A single low pulse is generated upon reaching the terminal count; this function is useful for generating real-time clocks.
- d) A Divide-by-N Counter generates a repetitive Timer Out low pulse; a new pulse train is initiated every time the terminal count is reached.

9.2 CIRCUIT ANALYSIS

The schematic diagram for the SBC 80/05 is given. Both active-high and active-low signals are used. A signal mnemonic that ends with a virgule (e.g., DAT7/) denotes that the signal is active low (0.4V).

Conversely, a signal mnemonic without a virgule (e.g., ALE) denotes that the signal is active high (2.0V).

9.3 INITIALIZATION

When power is applied in a start-up sequence, the contents of the CPU program counter, instruction register and interrupt enable flip-flop are subject to random factors and cannot be predicted. For this reason, a power-up sequence is used to set the CPU (as well as Bus Controller U19 and the I/O ports of U15) to a known internal state.

When power is initially applied to the SBC 80/05, capacitor C1 (1ZC7) begins to charge through resistor R4. The charge developed across C1 is sensed by a Schmitt trigger, which is internal to Clock Generator A3. The Schmitt trigger converts the slow transition appearing at pin 2 into a clear, fast-rising synchronized RESET output signal at pin 1. The RESET signal is inverted by open-collector gate U24-3 to produce Initialize signal INIT/, which is distributed as shown on block diagram. The INIT/ signal clears the CPU program counter, instruction register, and interrupt enable flip-flop; initializes the three I/O ports of U15 to the input mode; and sets Bus controller U19 to a known internal state.

9.4 CLOCK CIRCUITS

The time base for the SBC 80/05 is provided by Clock Generator U3 (1ZC7) and crystal Y1. The 19.6608-MHz output of U3 is divided by U12 (2ZC6) to produce a 9.8304-MHz signal, which is driven through gate U30 to produce Multibus clocks BCLK/ and CCLK/. Jumpers W7 and W8 are provided so that, when removed, some other master module can be used to generate one or both of these clocks if desired.

The 19.6608-MHz output of U3 is divided by U2 (1ZD6) to produce a 3.93216-MHz clock input to CPU U1, which internally divides this into a 1.96608-MHz clock output. This output is further divided by U14 (2ZB6) to produce the 122.88-kHz input to U15 (3ZD4).

The Bus Interface allows the SBC 80/05 to use a system bus that is common to other master devices (e.g., CPU's and DMA controllers), thus allowing system memory and I/O devices to be shared on a priority basis. The primary element of the Bus Interface is Bus Controller U19 which operates synchronously with the bus clock (BCLK/) and consists of the following functional sections:

- a) Bus arbitration logic to resolve bus contention in multiple master systems.
- b) Timing logic, initiated by the bus arbitration logic, to ensure adequate setup and hold times for the address and data placed on the Multibus;

also generates read/write control signals.

- c) Output drive logic for driving the bus memory and I/O command (control) lines.

When the SBC 80/05 gains control of the Multibus to perform a write operation, the Bus Controller gates the device address and data onto the Multibus and issued a Write command. In performing a read operation, the Bus Controller gates the device address onto the Multibus and issues a Read command. Operations between the CPU and the on-board memory and I/O ports do not require the Multibus. Notice in block diagram that the data bus drivers are bidirectional and the address bus drivers are unidirectional. This allows the SBC 80/05 full control of the Multibus but prevents other modules from accessing the SBC 80/05 memory and I/O ports.

The SBC 80/05 provides eight-bit words of static Random Access Memory (RAM). Two Intel 8111-U4 devices provide 256 words in locations 3E00-3EFF. The Intel 8155 RAM/I/O/Timer provides 256 eight-bit words of static RAM in locations 3F00-3FFF. This 512 word RAM storage area requires neither refreshing nor clock inputs, thereby providing the CPU immediate access to the addressed location.

Two IC sockets are provided to allow to install either 2K or 4K bytes of Read Only Memory (ROM). We install two Intel 2716 (2K x 8) or 8708 (1K x 8) ultraviolet erasable and reprogrammable ROM's (EPROM's) for program development or install two Intel 8316E (2K x 8) or 8408 (1K x 8) masked ROM's containing a dedicated program. Depending on the type of ROM or EPROM installed, the address locations are 0000-07FF (two 1K x 8-bit chips) or 0000-0FFF (two 2K x 8-bit chips).

The Serial I/O Interface is accomplished via the Serial Input Data (SID) and Serial Output Data (SOD) pins on the CPU. Data on the SID line is loaded into the CPU by a RIM instruction; data on the SOD line is set or cleared by a SIM instruction. Data buffers are provided for TTL level interface. Sockets are provided for the installation of level converters for FS232C interface.

The Parallel I/O Interface consists of three general purpose ports provided by the Intel 8155 RAM/I/O/Timer. Each of the three ports can be programmed to be either an input port or an output port. One of the three ports (Port 03) can be programmed to be status pins, thus allowing the other two ports (Ports 01 and 02) to operate in a handshake mode. The I/O portion of the 8155 contains four internal registers - one register for command and status and one data register for each of the three ports. Sockets are provided for the installation of input terminators or output drivers as required by the user's configuration.

9.5 INSTRUCTION TIMING

The execution of any program consists of read and write operations, where each operation transfers one byte of data between the CPU and a particular memory or I/O address. Although the CPU can vary the address, data, type, and sequence of operations, it is capable of performing only a basic read or write operation. With the exception of a few control lines, such as Address Latch Enable (ALE); these read and write operations are the only communication necessary between the processor and the other components to execute any instruction.

An instruction cycle is the time required to fetch and execute an instruction. During the fetch phase, the selected instruction (consisting of up to three bytes) is read from memory and stored in the operating registers of the CPU. During the execution phase, the instruction is decoded by the CPU and translated into specific processing activities.

Each instruction cycle consists of up to five machine cycles. A machine cycle is required each time the CPU accesses memory or an I/O port. The fetch phase requires one machine cycle for each byte to be fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instructions from memory; other instructions, however, require an additional machine cycle(s) to write or read data to or from memory or I/O devices.

Every instruction cycle has at least one reference to memory during which time the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of that instruction requires no reference to memory. The first machine cycle in every instruction cycle is therefore a fetch, and beyond that there are no specific rules. For instance, the In (input) and OUT (output) instructions each require three machine cycles: fetch (to obtain the instruction), memory read (to obtain the I/O address of the peripheral), and an input or output machine cycle (to complete the transfer).

Each machine cycle consists of a minimum of three and a maximum of six states designated T1 through T6. A state is the smallest unit of processing activity and is defined as the interval between two successive falling edges of the CPU clock. Each state (or CPU clock cycle) has a duration of 508 nanoseconds (derived by dividing the crystal frequency by 10).

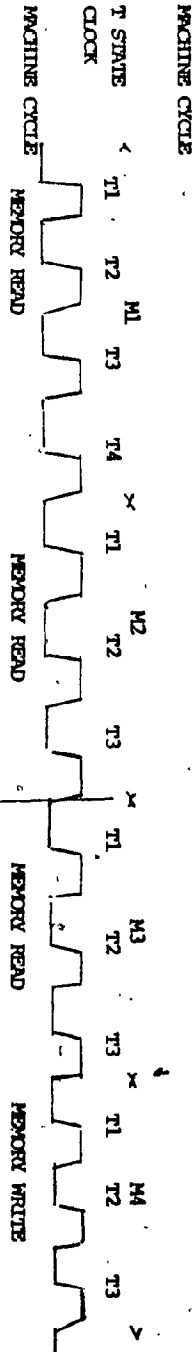
Every machine cycle normally consists of three T-states with the exception of an opcode fetch, which consists of either four or six T-states. The actual number of states required to execute any instruction depends on the instruction being executed, the particular machine cycle within the instruction cycle, and the number of wait states inserted into the machine cycle. The wait state is initiated when the READY input to the CPU is pulsed low.

There is no wait state imposed when the CPU is addressing on-board I/O or memory. As discussed later in this section, the wait state occurs only while waiting for XACK/ to be pulled low in response to an off-board I/O or memory read or write operation. Thus, the wait state depends on how quickly the Multibus can be accessed and speed of the addressed memory or I/O device.

The following figure is presented to show the relationship between an instruction cycle, machine cycle, and T-state. This example shows the execution of a Store Accumulator Direct (STA) instruction involving on-board memory. Notice that for this instruction the opcode fetch (machine cycle M1) requires four T-states and the remaining three cycles each require three T-states.

The opcode fetch is the only machine cycle that requires more than three T-states. This is because the CPU must interpret the requirements of the opcode fetched during T1 through T3 before it can decide what must be done in the remaining T-state(s).

INSTRUCTION CYCLE



ADDRESS BUS THE ADDRESS (PC) POINTS TO THE ADDRESS (PC 1) POINTS THE ADDRESS (PC 2) POINTS THE ADDRESS IS THE DIRECT OP-CODE OF THE INSTRUCTION THE 2ND BYTE OF INSTRUCTION 3RD BYTE OF INSTRUCTION ADDRESS

DATA BUS INSTRUCTION OP CODE LOW ADDRESS BYTE OF THE HIGH ADDRESS BYTE OF THE CONTENT OF THE ACCUMULATOR (STA)

(STA)

At the beginning of T2, the CPU pulls the RD/line low to enable the addressed memory device. The device will then drive the ADO-AD7 lines. After a period of time, as determined by the access time of the addressed memory device, valid data (the DCX instruction, in this example) will be present on the ADO-AD7 lines. During T3 the CPU loads the data on DCO-DC7 into its instruction register and drives RD/ high, disabling the addressed memory device. During T4 the CPU decodes the opcode and decides whether or not to enter T5 on the next clock cycle or start a new machine cycle and enter T1. In the case of the DCX instruction, the CPU will enter T5 and then T6 before beginning a new machine cycle.

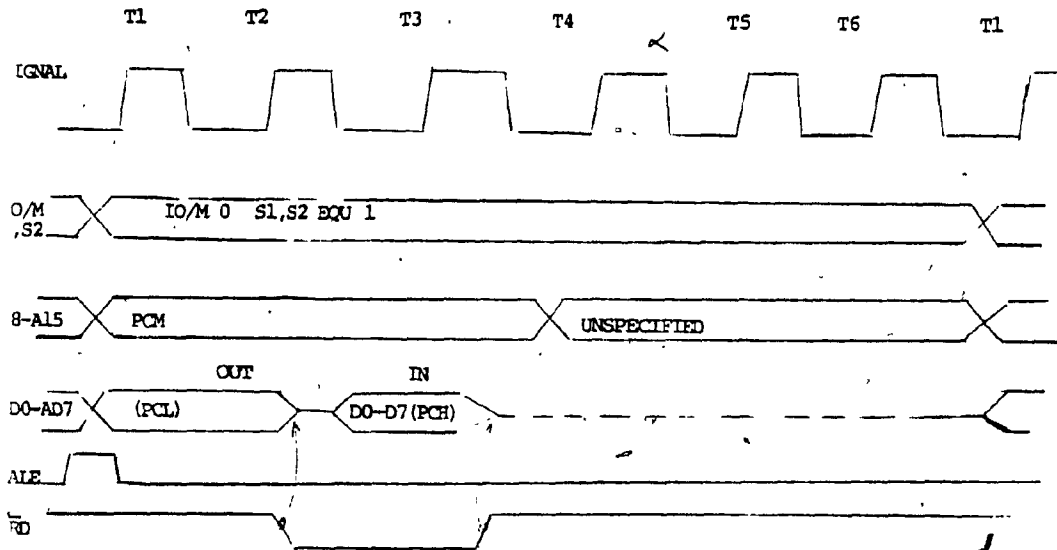
The following figure is identical to the last one with one exception, which is the use of the READY input to the CPU. As shown on this figure, the CPU examines the state of the READY input during T2. If the READY input is high, the CPU will proceed to T3 as shown in last figure. If the READY input is low, however, the CPU will enter the Twait state and stay there until READY goes high. When READY goes high, the CPU will exit the Twait state and enter T3. The external effect of using the READY input is to preserve the exact state of the CPU signals at the end of T3 for an integral number of clock periods before finishing the machine cycle. This "stretching" of the system timing, in effect, increases the allowable access time for memory or I/O devices. By inserting Twait states, the CPU can accommodate slower memory or slower I/O devices.

A common use of the READY input is to single-step the CPU using a manual switch. It should be noted, however, that access to the on-board memory and I/O ports does not impose a Twait state.

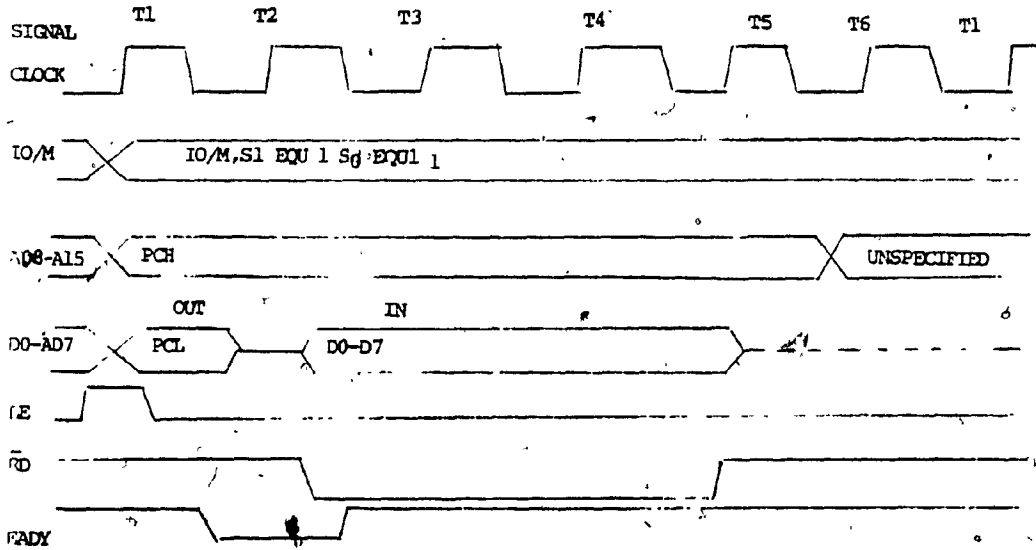
9.6 OPCODE FETCH TIMING

The following figure shows the timing relationship of a typical opcode fetch machine cycle. At the beginning of T1 of every machine cycle, the CPU performs the following:

- a) Pulls IO/M low to signify that the machine cycle is a memory reference operation. (The CPU also drives status lines S0 and S1; however, these lines are not used by the SBC 80/05.)
- b) Places high-order bits (PCH) of program counter onto address lines U8-U15. These address bits will remain true until at least T4.
- c) Places low-order bits (PCL) of program counter onto address-data lines AD0-AD7. These address bits will remain true for only one clock cycle, after which AD0-AD7 go to their high-impedance state as indicated by the dashed line in the following figure.
- d) Activates the Address Latch Enable (ALE) signal.



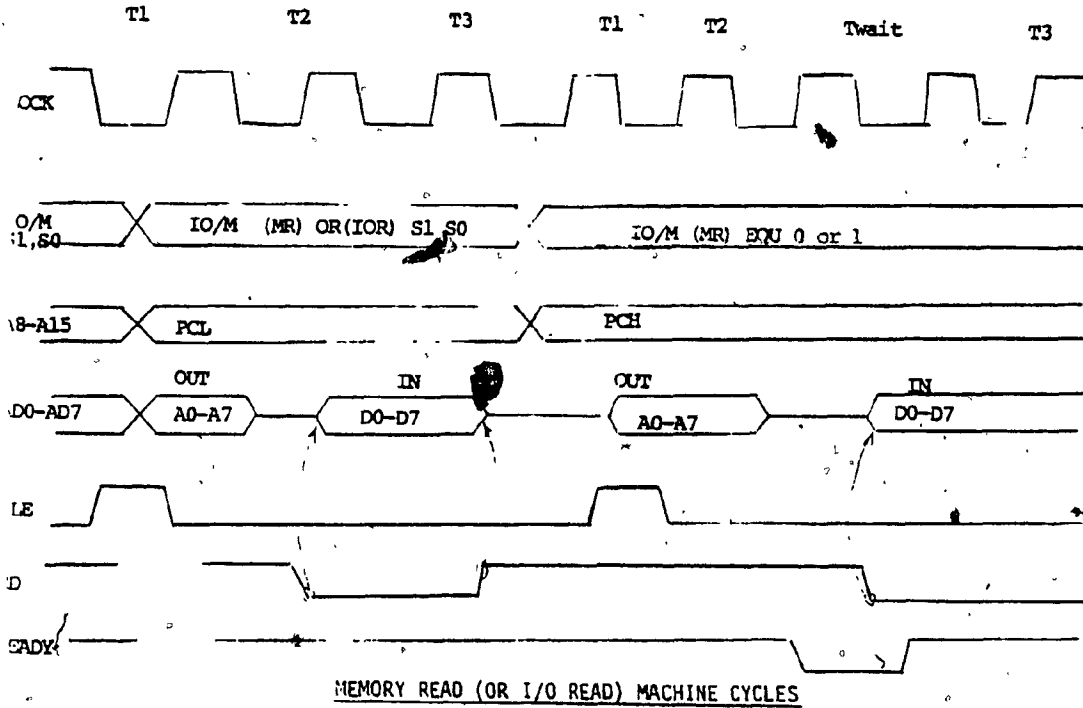
OPCODE FETCH MACHINE CYCLE



OPCODE FETCH MACHINE CYCLE (WITH WAIT)

9.7 MEMORY READ TIMING

The following figure shows the timing of two successive memory read machine cycles, the first without a Twait state and the second with one Twait state. Disregarding the states of the S0 and S1 lines, the timing during T1 through T3 is identical with the opcode fetch machine cycle without wait. The major difference between the opcode fetch and memory read cycles is that an opcode fetch machine cycle requires four or six T-states whereas the memory read machine cycle requires only three T-states. One minor difference between the two cycles is that the memory address used for the opcode fetch cycles always the contents of the program counter (PC), which points to the current instruction; the address used for a memory read cycle can be one of several origins. Also, the data read from memory is placed into the appropriate register instead of the instruction register. Note that a Twait can be imposed by slower memory devices as previously described.



9.8 I/O READ TIMING

The figure at the beginning of the page also illustrates the timing of two successive I/O read machine cycles, the first without a Twait state and the second with one Twait state. With the exception of the IO/M status signal, the timing of a memory read cycle and an I/O read cycle is identical. For an I/O read, IO/M is driven high to identify that the current machine cycle is referencing an I/O port. One other minor exception is that the address used for an I/O read cycle is derived from the second byte of an IN instruction; this address is duplicated onto both the U8-U15 and A0-A7 lines.

The data read from the I/O port is always placed in the accumulator specified by the IN instruction. Note that a Twait may be imposed by slower I/O devices as described for slower memory devices.

9.9 MEMORY WRITE TIMING

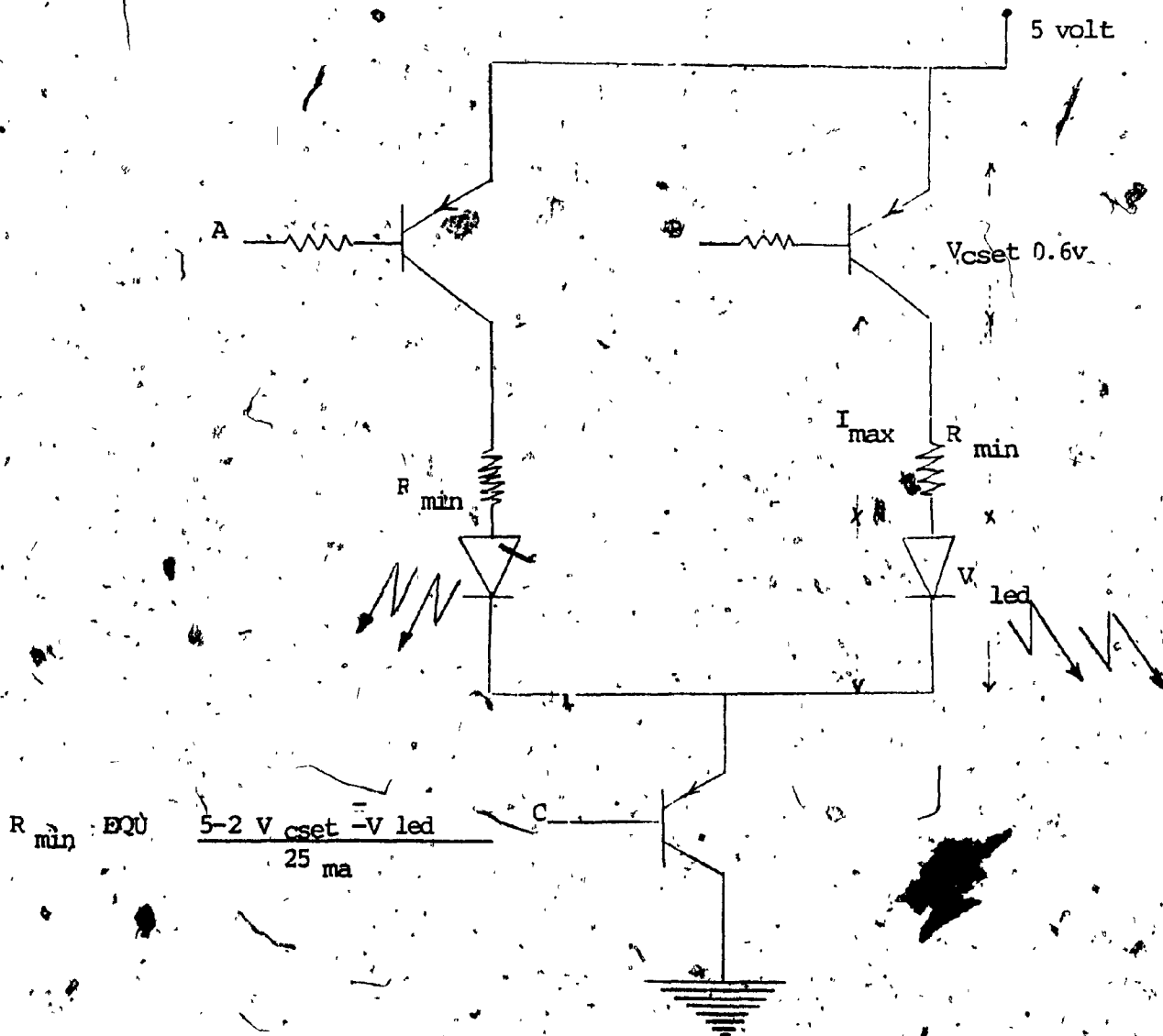
The figure in page 73 shows the timing of two successive memory write machine cycles, the first without a Twait state. Again, disregarding the states of the S0 and S1 lines, the timing during T1 is identical to the timing of an opcode fetch, memory read, and I/O read cycles. The difference occurs, however, at the end of T1. For instance, in a memory read cycle the ADO-AD7 lines are disabled (high impedance) at the beginning of T2 in anticipation of the returned data. In a memory write cycle, the ADO-AD7 lines are not disabled and the data to be written into memory is placed on these lines at the beginning of T2. The Write (WR/) line is driven low at this time to enable the addressed memory device. During T2 the READY input is checked to determine if a Twait state is required. If the READY input is low, Twait states are inserted until READY goes high. During T3, the WR/ line is driven high to disable the addressed memory device and terminate the memory write operation. Note that the contents on the address and data lines do not change until the next T1 state.

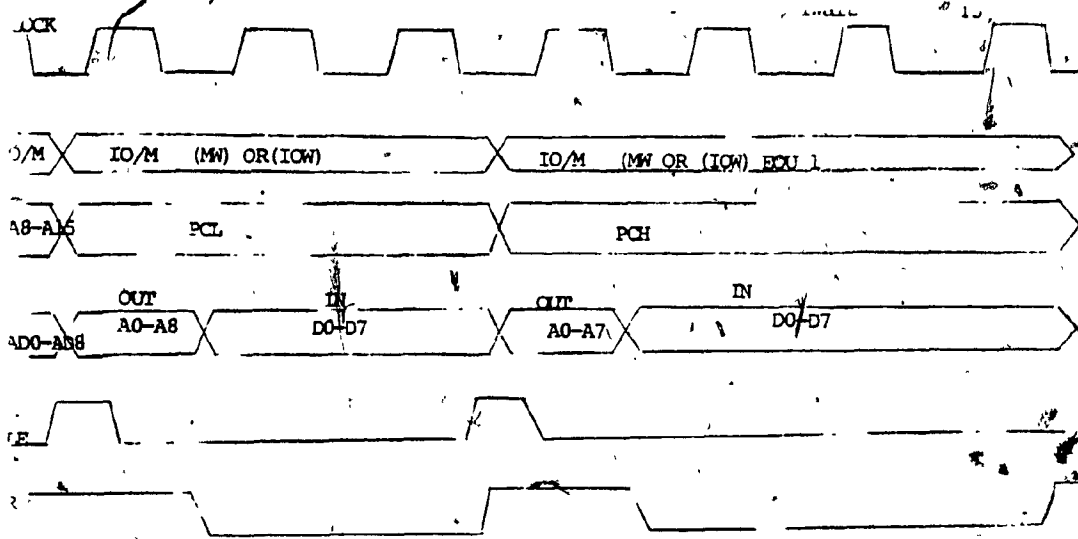
Bus Controller U19 arbitrates all requests for control of the Multibus. When the SBC 80/05 acquires control of the Multibus, the Bus Controller generates the appropriate memory or I/O command and enables the address onto the Multibus by enabling the Address Bus Drivers. The Bus Controller also enables the Data Bus Drivers, which, depending on whether the operation is a read or write, transfers data from or to the Multibus. The RC network (R8 and C13) connected to the DLYADJ input of the Bus Controller provides a 70-nanosecond delay to ensure an adequate setup and hold relationship between the address/data lines and the appropriate control signals.

The falling edge of the BCLK/ signal provides a timing reference for the bus arbitration logic. Bus arbitration begins when the CPU needs access to an external memory or I/O port. When this requirement occurs, the Command (CMD) and Off-Board Request (OFF BD REQ) are both high at the Transfer Start Request (XSTR) input to the Bus Controller. The Bus Controller drives Bus Request (BREQ/) low and forces Bus Priority Out (BPRO/) high. The BREQ/ output from the master modules is used by the Multibus when the bus priority is resolved by a parallel priority scheme.

The SBC 80/05 gains control of the Multibus when the BPRN/ input to the Bus Controller is driven low which, on the next falling edge of BCLK/, drives its BUST/ and ADEN/ outputs low.

The BUSY/ output indicates to all master devices that the bus is in use and prohibits any other master from acquiring control of the bus; the ADEN/ output enables the Address Bus Drivers and Data Bus Drivers. The ADEN/ output also activates the Bus Control (BUS CTL/) signal, which is applied to the input of gate U23-8 (1ZB6). As discussed later, the BUS CTL/ signal is used in conjunction with Transfer Acknowledge (XACK/) to activate the READY input to the CPU.





MEMORY WRITE (OR I/O WRITE) MACHINE CYCLES

9.10 I/O WRITE TIMING

The last figure also illustrates the timing of two successive I/O write machine cycles, the first without a Twait state and the second with one Twait state. With the exception of the I/OM status signal, the timing of a memory write cycle and an I/O write cycle are identical.

9.11 MULTIPLEXED ADDRESS/DATA BUS

The lower eight bits (A0-A7) of the memory address or I/O address (depending on whether a memory reference machine cycle or an I/O reference machine cycle is in progress) are output by the CPU during the first clock cycle (T1). The A0-A7 lines become the data bus during the second and third cycles (T2 and T3).

The trailing edge of the Address Latch Enable (ALE) signal issued by the CPU during T1 strobes these eight address bits into Demultiplexer U18 (1ZC3) and into RAM/IO/Timer U15 (3ZB6). The low-order address bits (AB0-AB7) from U18 are placed on the SBC 80/05 address bus together with the high-order address bits (AB0+AB7). This 16-bit address bus (AB0-ABF) is distributed to Address Bus Drivers U31-U33 (2ZA2), ROM/EPROM U16-U17 (3ZA5), and RAM U39-U40 (3ZB6).

9.12 MULTIBUS INTERFACE

The Multibus interface consists of unidirectional Address Bus Drivers U31-U33 (2ZA2), bidirectional Data Bus Drivers U34-U35 (2ZB4), and Bus Controller U19 (2ZD4).

The Bus Controller now examines the IO/M, RD/, and WT/ inputs and then outputs the appropriate command signal as follows:

IO/M	RD/	WT/	BUS COMMAND
0	0	1	Memory Read Command MRDC/
0	1	0	Memory Write Command MWTC/
1	0	1	I/O Read Command IORC/
1	1	0	I/O Write Command IOWC/

If the bus command is either an IORC/ or a MRDC/, the Bus Controller drives its Read Data (RDD) signal high to the Direction Input Enable (DIEN) input of bidirectional Data Bus Drivers U34-U35. When DIEN is driven high, data is transferred from the Multibus to the SBC 80/05. If the bus command is either an IOWC/ or a MWTC/, RDD is driven low and data is transferred from the SBC 80/05 to the Multibus!

The SBC 80/05 can lose control of the Multibus if its BPRN/ input goes high or when the CMD is completed. This causes the Bus Controller Transfer Complete (XCP) input to go low. In no case, however, will the SBC 80/05 lose control of the bus if the transfer is not complete or if the override function is invoked.

10. THE INTERFACE DESCRIPTION

The programmable keyboard/Display Interface has two sections: Keyboard and Display. The Keyboard section can Interface to regular typewriter style, Keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or bank of indicator lights TIL-3I3.

The 8085 processor will send the address of 8279 "FC" on the Data Bus, and at the same time the UP will send a WR signal to it, So the address FC will enable the chip CS and writing it to 8279 will create WR signal. Those two signal WR.CS will put the Driver of the 8279 in the input MOD.

While reading from 8279 will have another two Combined signal RD.CS which will put the 8279 in the Output Mode.

So to write to Control or Timing Register you have to send a Command by making Buffer address A0 a logic high.

So the following combination of signal will store any command in the Control Register (WR. CS. A0).

Then this Command will be decoded and certain function is set.

To synchronize the Scanning of switch (keyboard) and display the o/p of the Scan Counter must be externally decoded to provide the Scan lines for the keyboard and display in the decoded MOD of the Scan Counter decode the least significant 2 bits and provides a decoded one to four Scan.

Note that when the keyboard is in decoded Scan, si is the display. This means that only the 1st 4 characters in the Display RAM are displayed.

The display part of 8279 which is Pins 24,25,26,27,28, 29,30,31 are connected to the Transistor base each 4 pins represent one character or one digit. These two ports on the Outputs for the 16 X 4 display refresh Register, the data from these Outputs is synchronized to the Scan lines for Multiplexed digit displays.

The two 4 ports may be blanked independently and the two 4 bit port may be considered as one 8 bit port. So when any switch in the Keyboard is depressed (for 10 MS) the debounce circuit detects a closed switch and then the address of this switch in the Matrix plus the status of shift and Control are transferred to the FIFO. In the Scanned sensor Matrix modes, the contents of this return lines is directly transferred to the corresponding row of sensor RAM each key scan time.

This return line input are connected to the Scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls it low. They also serve as an 8 bit input in the strobed input mode.

Since in the mode, the contents of the return line are transferred to the FIFO on the rising edge of CONTROL AND STROB CNTL/STB line pulse.

Display the Data is controlled by CPU command since the CPU can set the MOD. So as I mentioned before, to set the 8279 MOD sent the following format at the combined signal (CS, WR, A0).

The following command is decoded by 8279:

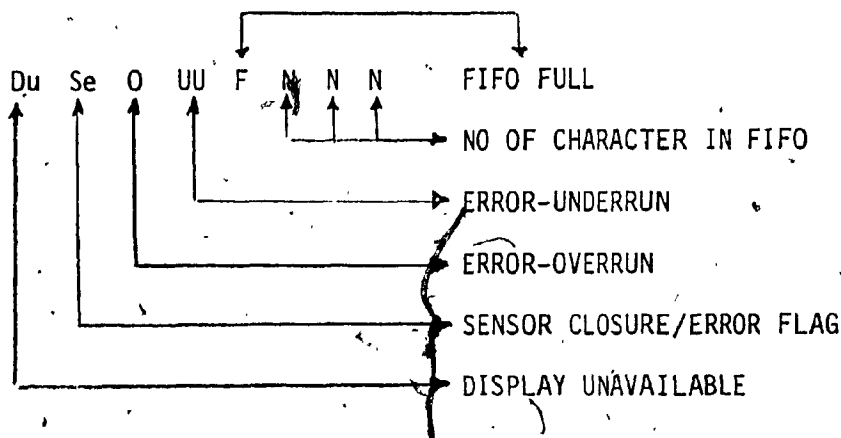
- a) Keyboard/Display MOD
- b) Controlling the Scanning Time by setting the prescaler value 2 to 31
- c) Reading the Status of FIFO/Sensor RAM. In this case they will read this status by their combined signal (CS, RD, A0)
- d) Read Display RAM. Command to enable the CPU to read one of 16 characters.
- e) Write Display RAM. Command to enable the CPU to write the character through 8279.
- f) Display write inhibit/ Blanking.
- g) Clear. This is to clear the Display, clear FIFO Status and clear all the position.

FIFO STATUS

FIFO Status is used in the keyboard and strobed input modes indicate whether an error has occurred. There are two types of errors possible: overrun and underrun.

Overrun occurs when the entry of another character into a full FIFO is attempted.

Underrun occurs when the CPU tries to read on empty FIFO. Also the status word has a bit to indicate that the Display RAM was unavailable because a clear command had not completed.



11. PROGRAMMING OF 8279

11.1 I/O REGISTER ADDRESSING

The I/O section consists of Command/Status (C/S) register and Data Register.

11.2 PROGRAMMING THE COMMAND REGISTER

The Command Register of 8279 consist of the following 8 bit:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	X	X	X	X	X	X
FUNCTION OF COMMAND			PARAMETER OF THE COMMAND				

Bit 7, Bit 6, Bit 5 represent the function of the command. Maximum command available is 8 command.

Bit 4, Bit 3, Bit 2, Bit 1, Bit 0 is 5 bit and is used for the parameter of the command.

The Command Register can be altered at any time by performing an I/O write to location Command Register.

11.2.1 INSTRUCTION REQUIRED FOR COMMAND REGISTER

Those two Instructions are going to set the keyboard and Display MOD.

```
MVI    A,15H    ; set keyboard to 8-8bit character display - Right entry and the display to decoded Scan Sensor Matrix.
```

```
OUT    0FDH    ; Perform I/O write operation to 8279 chip.
```

Also to set the internal Scanning frequency, you need to set the Timer by setting the prescaler by the following instructions:

```
MVI    A,3FH    ; Where the prescaler is set to 31
```

```
OUT    0FDH
```

When you Output FD, the least significant bit is logic High. So to make 8279 interpreted this information as Command.

Also to read FIFO/Sensor RAM and STATUS: This follows Instruction is performing this function:

```
IN    0FDH    ; read the STATUS of FIFO/Sensor RAM
MOV   B,A     ; store the STATUS in Register B.
```

By testing the STATUS byte of the FIFO/Sensor RAM you can know how many character in FIFO RAM and you can check if the Display is available or not.

So by using 8279 chip it saves a lot of hardware design needed for those devices and also save some processing time of the CPU.

12. DESCRIPTION OF THE MONITOR ROUTINE USED IN THE SYSTEM

The Program is organized as Follows:

- a) Cold start routine
- b) Warm start - Register save routine
- c) Interrupt vectors
- d) Keyboard monitor
- e) TTY monitor-
- f) Layout of RAM usage.

The keyboard monitor begins with the command RECOGNIZER followed by the command ROUTINE SECTION, utility routine section and monitor table. The command and utility routines are in alphabetical order within their respective sections.

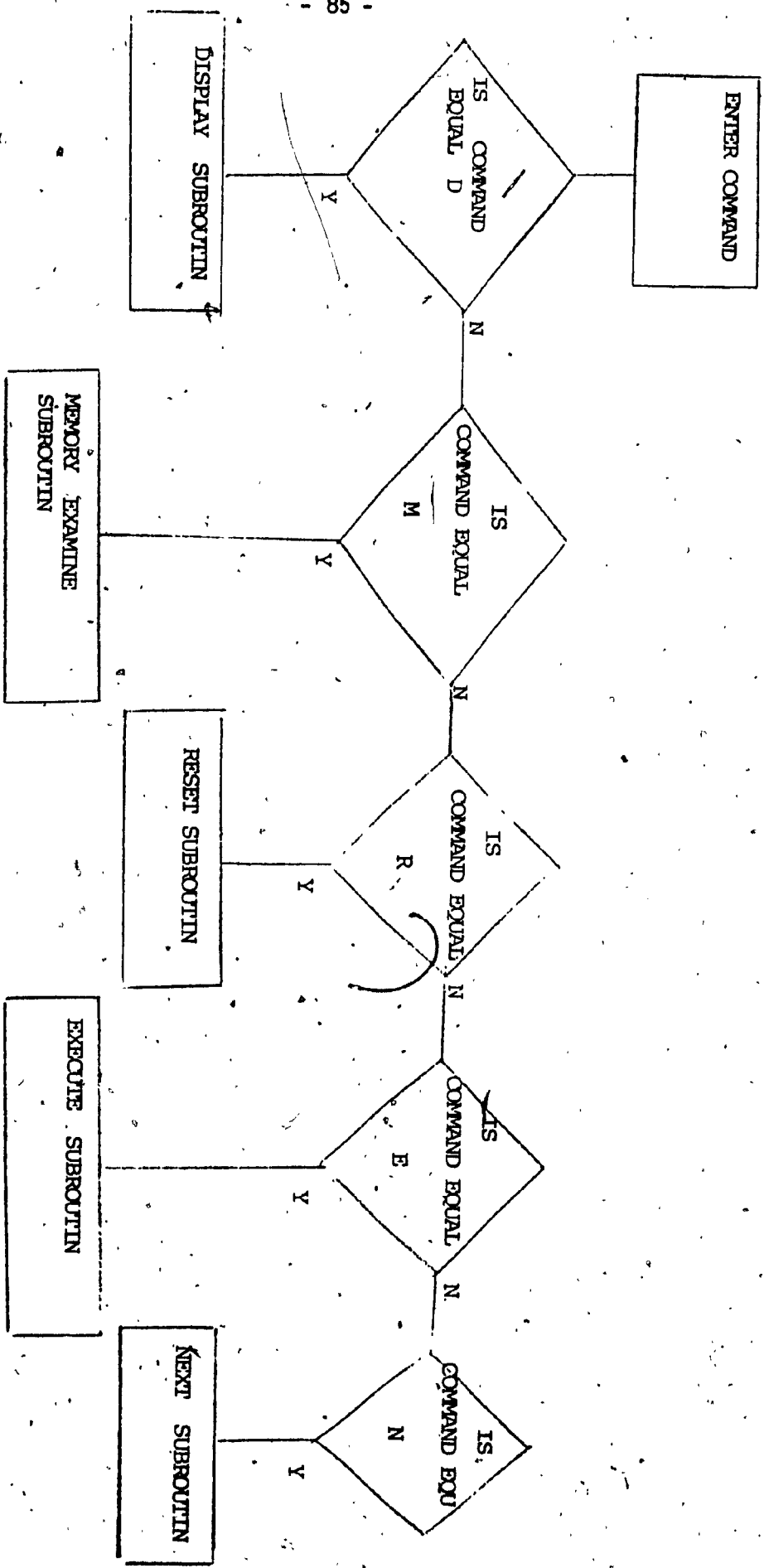
The following is the command assigned by the monitor:

- a) EXAM
- b) GO - CMD
- c) S STOP
- d) SUBST
- e) CLEAR
- f) CLDIS

The following flowchart is describing the command DECODER utility.

In this utility the CPU is looking to the input from the keyboard and check for the right command.

After reading the command from the keyboard the CPU will jump to corresponding subroutine and execute it and give a final result on the Display Unit.



FLOW CHART FOR COMMAND DECODER

DISCUSSION

Since the Data input and data output (Display) are integral part of many microprocessor design the system designers need an Interface that can control those functions without placing a large load on the C.P.U.

This function had been achieved by using 8279.

The input section can be represented by a regular typewriter style.

The output can be represented by display unit TIL 313 LED.

Using 8279 reduce the cost of the hardware for small processing system. But it will increase the cost of the software.

The 8279 is connected direct to the u-processor Data-Bus.

All the operation MODES of 8279 are completely under the control of the programmer using assembly language programming.

REFERENCES

1. INTEL COMPONENT DATA CATALOG 1979
2. SBC/80/05 INTEL MANUAL