

Experiment – 10

Interfacing with 8253/8254 and 8259

Introducing 8253/8254:

8253 and 8254 are software programmable timer/counter designed for use with microcomputer systems for the generation of accurate time delays under software control instead of setting up timing loops in software. They are nearly identical in function. The major differences are: a) the maximum clock frequency for 8253 is 2.6MHz and for 8254 it is 8MHz. b) 8254 only has a read-back feature, by which you can latch the count value at any instant.

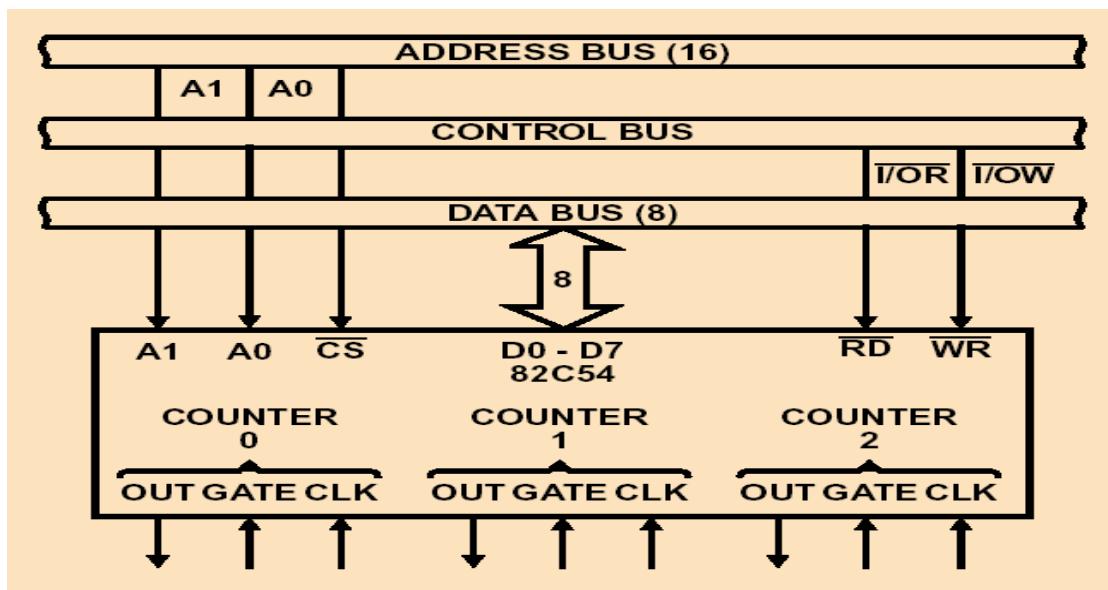


Fig. 8253/8254 system interface

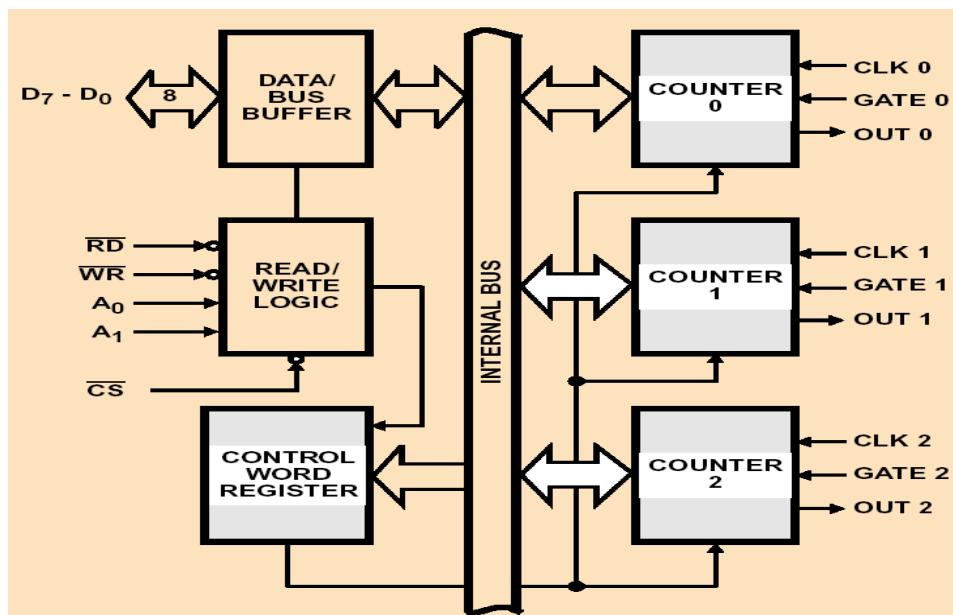


Fig. The internal block diagram

Programming the 8254

Counters are programmed by writing a Control Word and then an initial count. A1 and A0 are Select inputs for one of the three counters or Control Word Register for read/write operations. Normally connected to the system address bus.

A1	A0	SELECTS
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

Control Word Format:

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC - Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW - Read/Write

RW1	RW0	
0	0	Counter Latch Command (See Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M - Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD - Binary Coded Decimal

0	Binary Counter 16-bit
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Read Operations:

The two software handled methods are:

Counter Latch command:

Like a Control Word, this command is written to the Control Word Register, which is selected when A1, A0 = 11. Also, like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

Read Back Command:

The read-back command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

D7	D6	D5	D4	D3	D2	D1	D0
1	1	COUNT	STATUS	CNT 2	CNT 1	CNT 0	0

D5: 0 = Latch count of selected Counter (s)

D4: 0 = Latch status of selected Counter(s)

D3: 1 = Select Counter 2

D2: 1 = Select Counter 1

D1: 1 = Select Counter 0

D0: Reserved for future expansion; Must be 0

Fig. READ-BACK COMMAND FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D7: 1 = Out pin is 1

0 = Out pin is 0

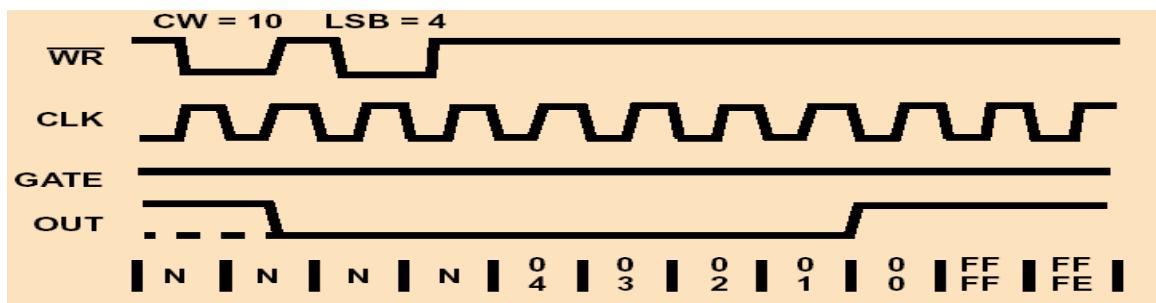
D6: 1 = Null count

0 = Count available for reading

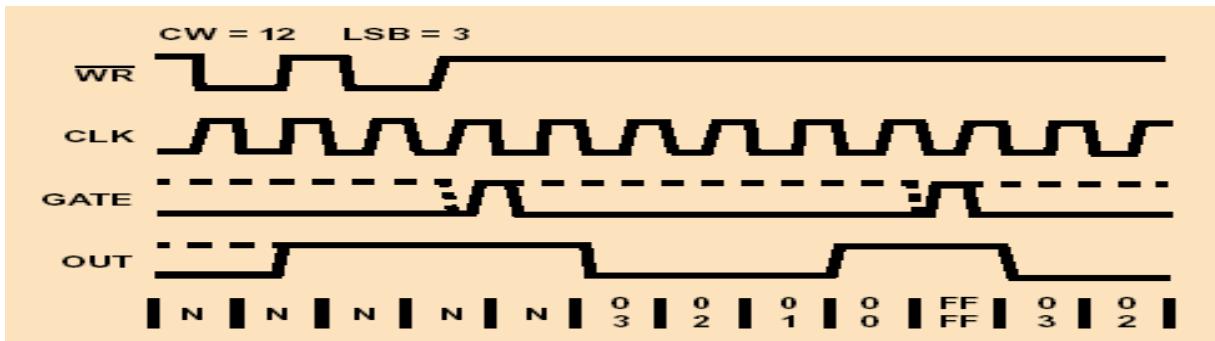
D5 - D0 = Counter programmed mode (See Control Word Formats)

Fig. The counter status format

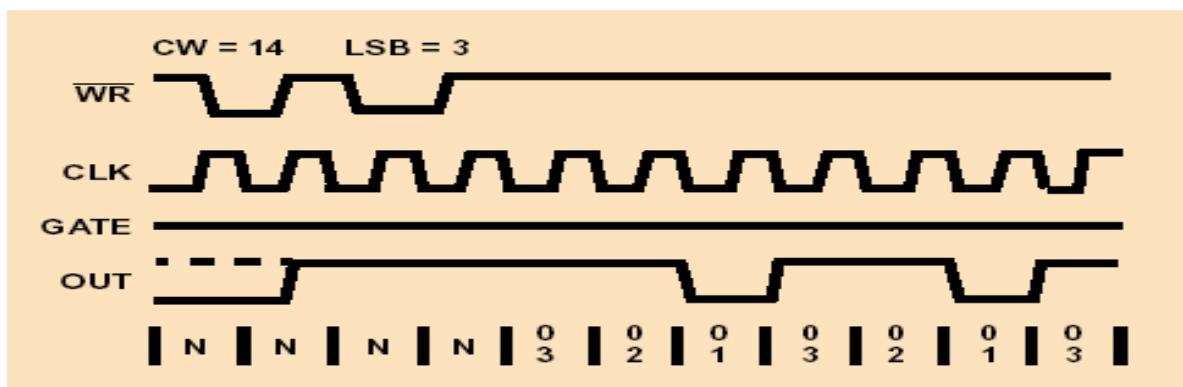
Modes of operation:



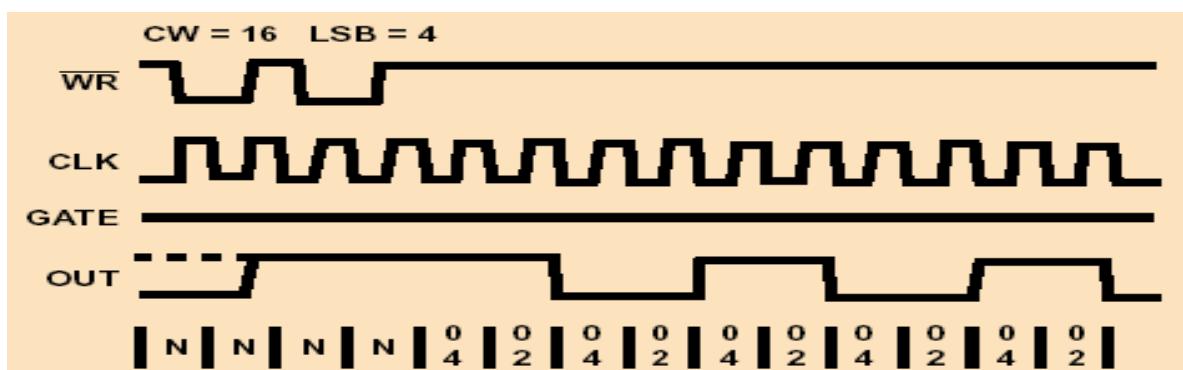
Mode – 0 : Interrupt on Terminal Count



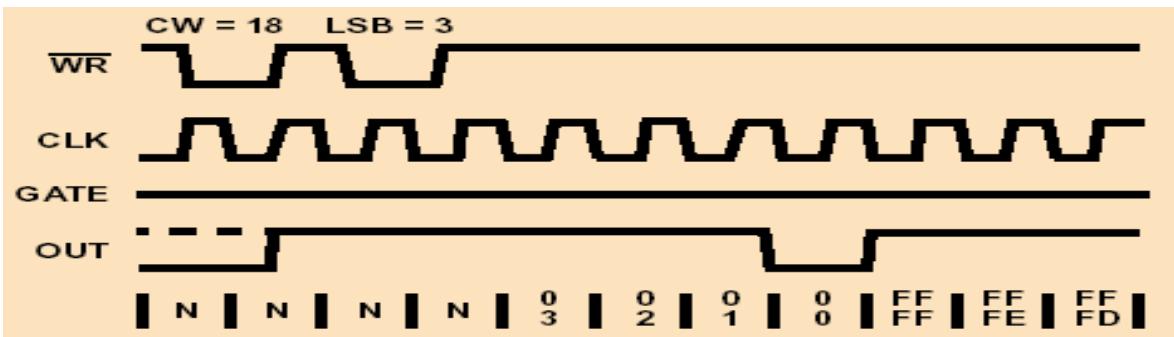
Mode – 1 : Hardware Retriggerable One-Shot



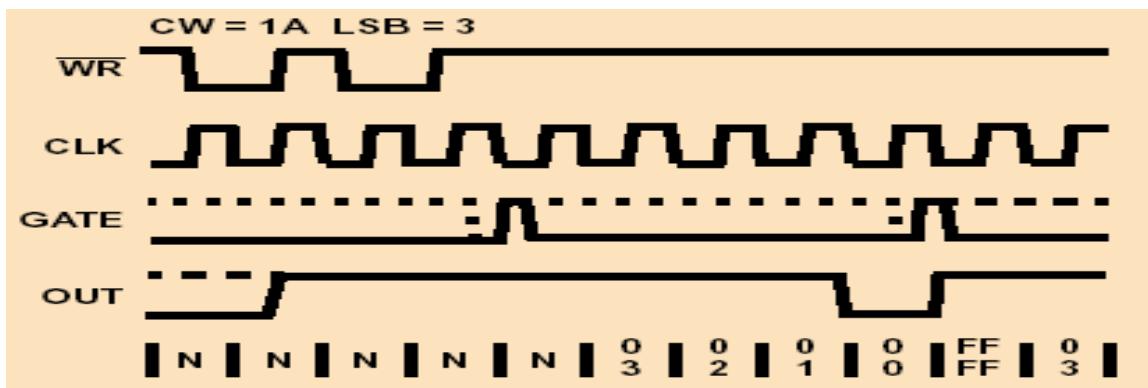
Mode – 2 : Timed Interrupt Generator



Mode – 3 : Square Wave Mode



Mode – 4 : Software Triggered Mode



Mode – 5 : Hardware Triggered Strobe (Retriggerable)

Introducing 8259:

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or ``routine'' that is associated with its specific functional or operational requirements; this is referred to as a ``service routine''. The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can 'point' the Program Counter to the service routine associated with the requesting device. This 'pointer' is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete

interrupt structure can be defined as required, based on the total system environment.

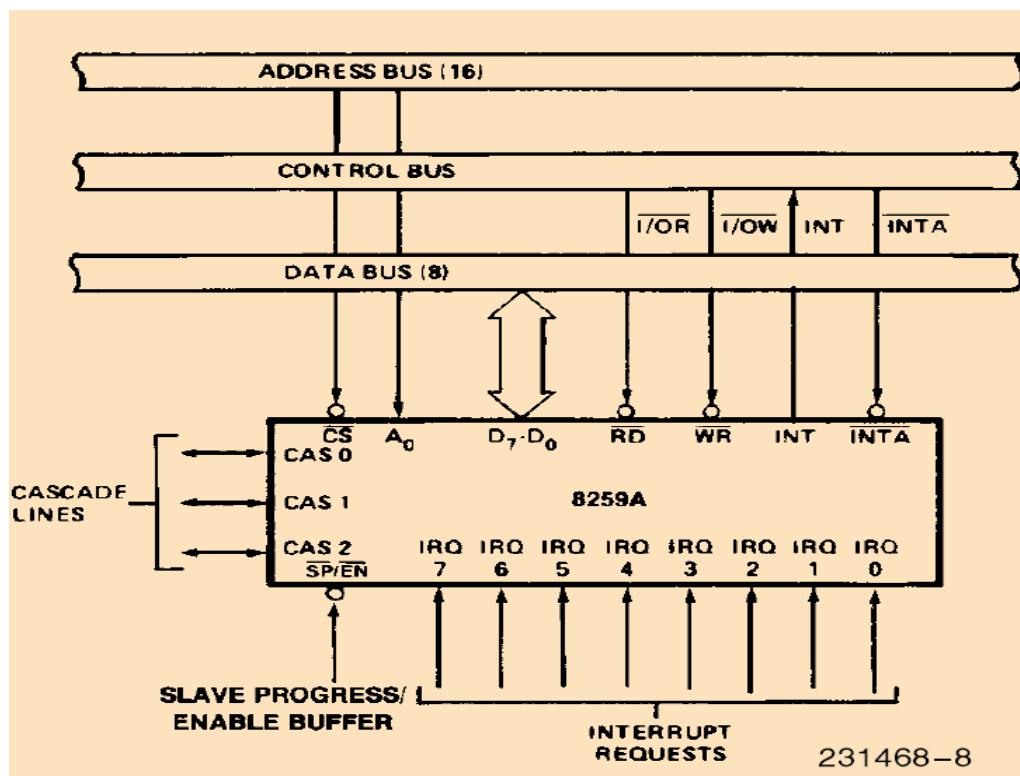


Fig. Interface to system bus.

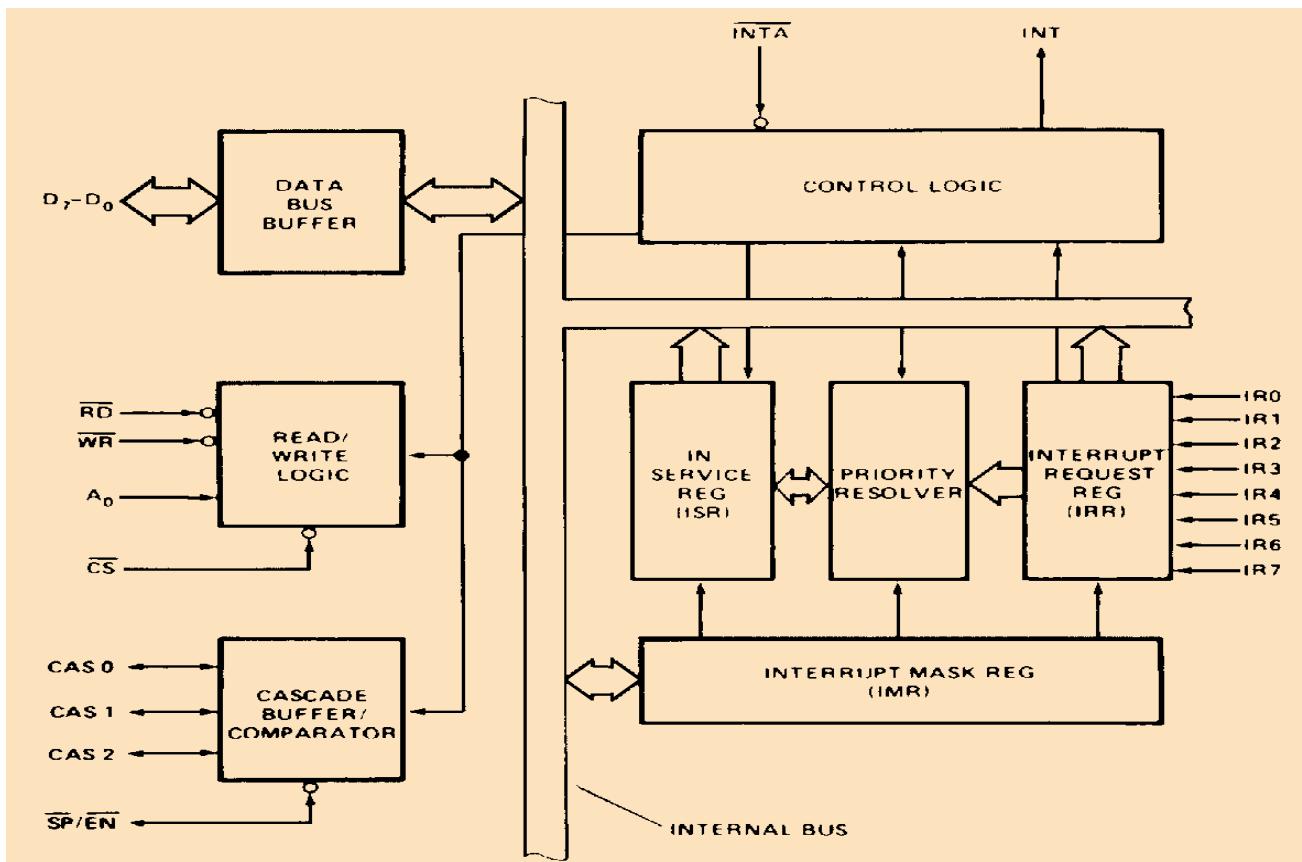


Fig. Internal Block Diagram

Programming the 8259:

The 8259A accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs): Before normal operation can begin, each 8259A in the system must be brought to a starting point—by a sequence of 2 to 4 bytes timed by WR pulses.

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence. According to the flow-chart an ICW1 and an ICW2 must be sent to any 8259 in the system. If the system has any slave 8259s (cascade mode) then an ICW3 must be sent to the master, and a different ICW3 to the slave. If the system is an 8086, or if you want to specify certain special conditions then you have to send an ICW4 to both master and slave.

ICW2 is used to tell the 8259 the type number to send in response to an interrupt signal on the IR0 input. In response to an interrupt on some other IR input, the 8259 will automatically add the number of the IR input to this base number and send the result to the 8086 as the type number for that input.

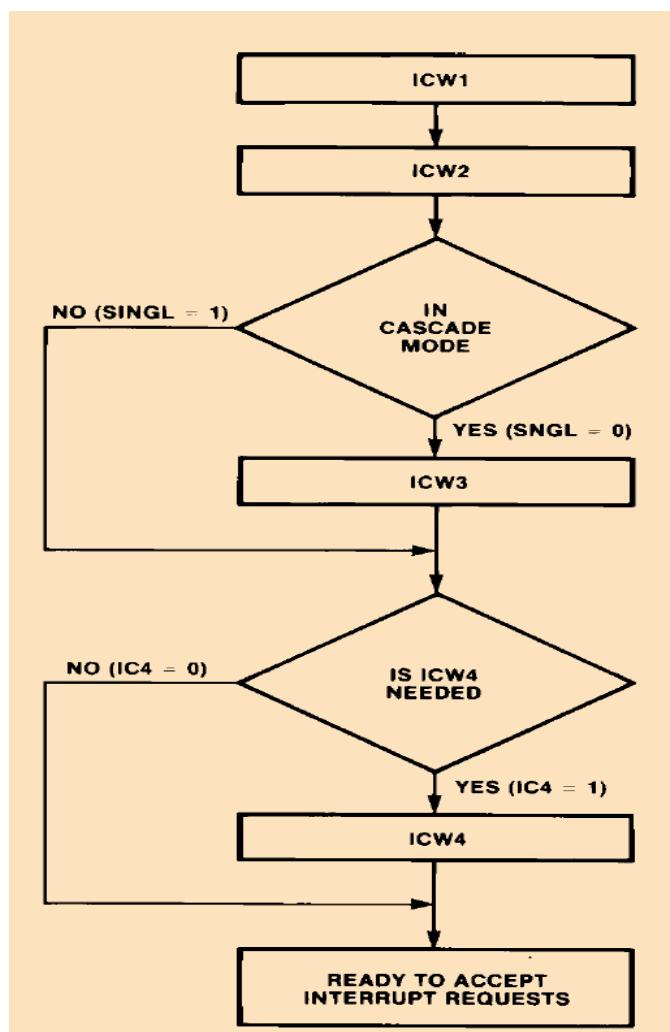


Fig. Initialization Sequence.

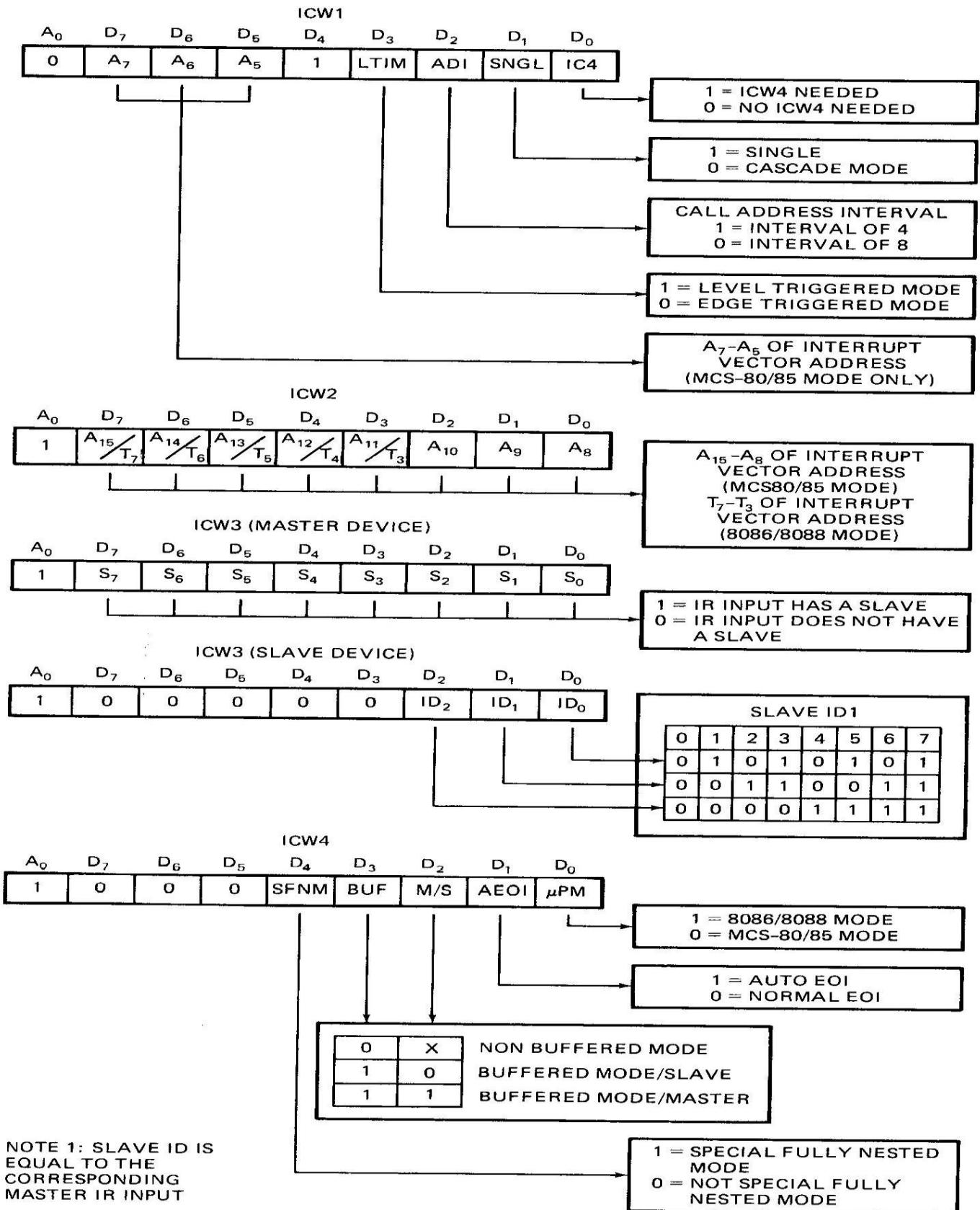


Fig. ICW Formats

2. Operation Command Words (OCWs): These are the command words which command the 8259A to operate in various interrupt modes.

An **OCW1** must be sent to an 8259 to unmask any IR inputs that you want it to respond to.

OCW2 is mainly used to reset a bit in the in-service register. This is usually done at the end of the interrupt procedure. The effect of resetting the ISR bit for an interrupt level is that once the bit is reset, the 8259 can then respond to interrupt signals of lower priority. We usually use the nonspecific end-of-interrupt. If you want to reset a specific ISR bit, you can send the 8259 an OCW2 with 011 in bits D7, D6, D5 and the number of the ISR bit you want to reset in the lowest 3 bits of the word. You can also use OCW2 to tell the 8259 to rotate the priorities of the IR inputs so that after an IR input is serviced, it drops to the lowest priority. We are not discussing about the OCW3 this time. You can see the datasheets from internet for more details.

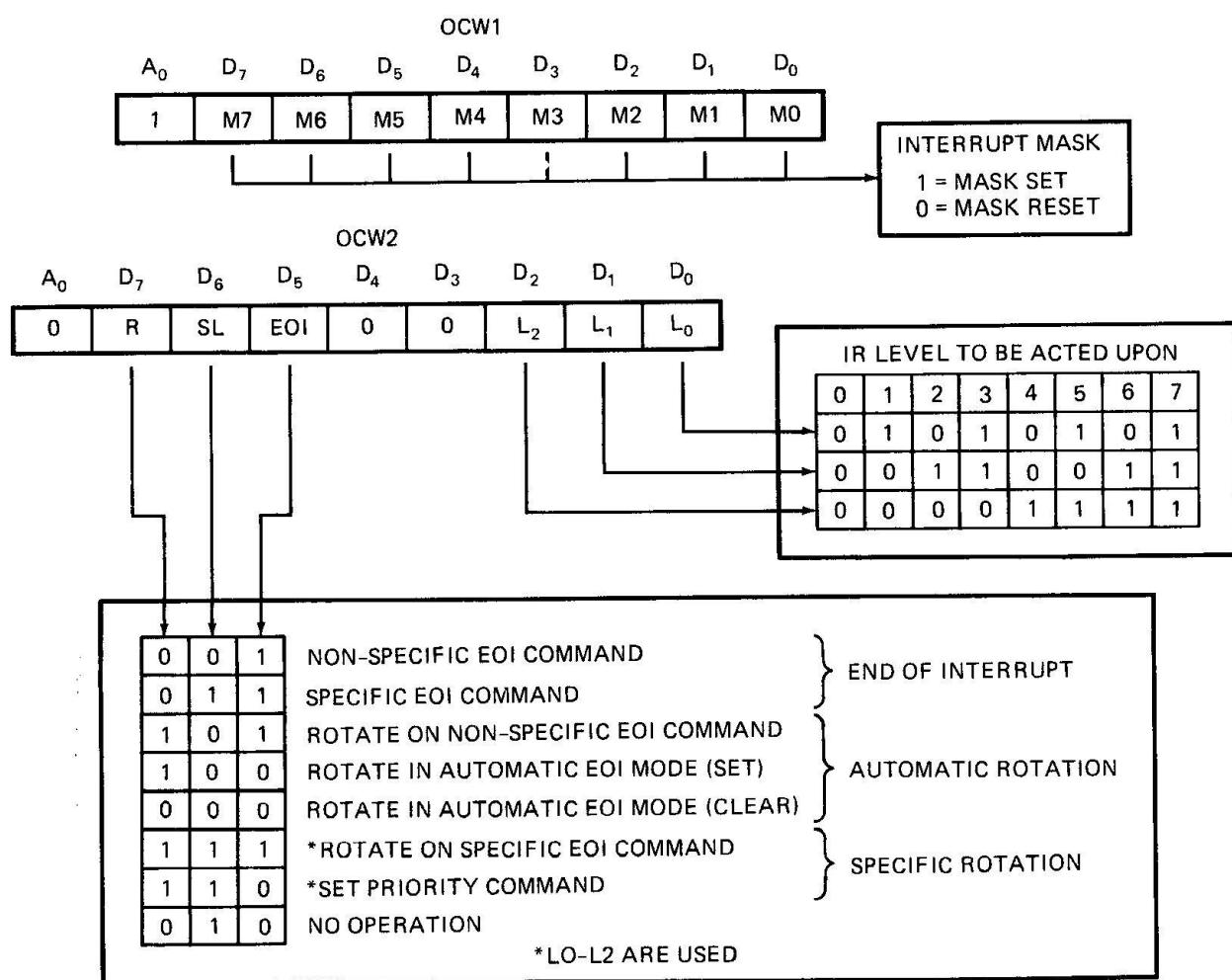


Fig. OCW Formats

Program Examples: Only for MTS – 86C Boards

1. 8253/8254

```
; place the second from the top of dip-switch at the top left corner of the board at right position : OUT1
COUNT1    EQU    OFFDAH ; address to select 8253 counter 1
CSR        EQU    OFFDEH ; address to select 8253 control word
CNT3      EQU    3FD6H  ; address to select 8255 control word register (PPI - 3)
APORT3    EQU    3FD0H  ; address to select 8255 Port A
BPORT3    EQU    3FD2H  ; address to select 8255 Port B
CPOR3     EQU    3FD4H  ; address to select 8255 Port C
FND       EQU    3FF0H  ; address to select Seven Segment Display

        ORG    0

CODE SEGMENT
ASSUME CS:CODE,DS:CODE

START:    MOV    SP,4000H    ; Setup Stack Pointer
          MOV    AX,CS      ; CS = DS
          MOV    DS,AX

          MOV    AX,0
          MOV    ES,AX      ; Setup ES to '0'
          MOV    BX,2*4      ; Setup Vector Address of NMI to BX
          MOV    ES:WORD PTR[BX],OFFSET NMI   ; To write IP in Vector Table
          MOV    ES:2[BX],CS  ; To write CS in the Vector Table

          MOV    DX,CNT3    ; Address of control word register of 8255 to initialize
          MOV    AL,90H      ; To set up ports as : A PORT = input, B and C PORT = Output
          OUT    DX,AL      ; DX goes to the Address Bus and AL to the Data Bus

          MOV    DX,CPOR3   ; Outputting to C PORT all High
          MOV    AL,0FFH
          OUT    DX,AL

I8253:    MOV    DX,CSR    ; Initializing 8253
          MOV    AL,01110110B ; selecting Counter 1, LSB then MSB, Mode-3-Square Wave Mode, Binary Count
          OUT    DX,AL      ; DX goes to the Address Bus and AL to the Data Bus

          MOV    DX,COUNT1  ; Address of 8253 Counter 1

PLAY:     PUSH   DX
          MOV    DX,APORT3 ; Address of 8255 A PORT : Which is connected to the 8 push buttons to take input
          IN     AL,DX      ; The data in the PORT A is taken to AL
          NOT   AL
          MOV    DX,BPORT3 ; Address of 8255 B PORT : Which is connected to the 8 LEDs to output any value.
          OUT    DX,AL      ; DX goes to the Address Bus and AL to the Data Bus
          POP    DX

          CMP    AL,BL      ; Same as previous inputted button ?
          JZ     PLAY
          MOV    BL,AL      ; To store the 1 Byte information of which button pressed to BL
          CMP    AL,0
          JZ     PLAY
          ; Whether any button pressed ?
          ; If not, Jump to PLAY to see whether any button is pressed

          PUSH   AX
          MOV    AX,10
          OUT    DX,AL      ; store the information of which button pressed to the STACK
          ; Data of Soundless ( the rest ) transfer to 8253
          ; sending LSB

          MOV    AL,AH
          OUT    DX,AL      ; sending MSB
          MOV    CX,1500    ; Time Delay
          LOOP   $
          POP    AX
          ; Return input value of buttons

          PUSH   DX
          MOV    DX,FND    ; temporary storage to stack
          ; The seven segment display address

          TEST   AL,80H    ; '7' key/button ?
          JNZ   DO
          TEST   AL,40H    ; '6' key/button ?
          JNZ   RE
          TEST   AL,20H    ; '5' key/button ?
          JNZ   MI
          TEST   AL,10H    ; '4' key/button ?
```

	JNZ	FA	; in case of '4' key then jump to FA
	TEST	AL,8	; '3' key/button ?
	JNZ	SOL	; in case of '3' key then jump to SOL
	TEST	AL,4	; '2' key/button ?
	JNZ	RA	; in case of '2' key then jump to RA
	TEST	AL,2	; '1' key/button ?
	JNZ	SY	; in case of '1' key then jump to SY
	TEST	AL,1	; '0' key/button ?
	JNZ	DO1	; in case of '0' key then jump to DO1
DO:	MOV	AL,11011000B	; seven segment value for : 7
	OUT	DX,AL	
	MOV	AX,4697	; Divide Rate of : DO
	JMP	SET8253	
RE:	MOV	AL,10000010B	; seven segment value for : 6
	OUT	DX,AL	
	MOV	AX,4184	; Divide Rate of : RE
	JMP	SET8253	
MI:	MOV	AL,10010010B	; seven segment value for : 5
	OUT	DX,AL	
	MOV	AX,3728	; Divide Rate of : MI
	JMP	SET8253	
FA:	MOV	AL,10011001B	; seven segment value for : 4
	OUT	DX,AL	
	MOV	AX,3519	; Divide Rate of : FA
	JMP	SET8253	
SOL:	MOV	AL,10110000B	; seven segment value for : 3
	OUT	DX,AL	
	MOV	AX,3135	; Divide Rate of : SOL
	JMP	SET8253	
RA:	MOV	AL,10100100B	; seven segment value for : 2
	OUT	DX,AL	
	MOV	AX,2793	; Divide Rate of : RA
	JMP	SET8253	
SY:	MOV	AL,11111001B	; seven segment value for : 1
	OUT	DX,AL	
	MOV	AX,2491	; Divide Rate of : SI
	JMP	SET8253	
DO1:	MOV	AL,11000000B	; seven segment value for : 0
	OUT	DX,AL	
	MOV	AX,2352	; Divide Rate of : High DO
SET8253:	POP	DX	; Return DX from Stack – the counter 1 address
	OUT	DX,AL	; LSB first
	MOV	AL,AH	
	OUT	DX,AL	; MSB next
	MOV	CX,1500	; Delay
	LOOP	\$	
	JMP	PLAY	; Repeat
NMI:	PUSH	DX	; Service routine for NMI
	PUSH	AX	
	MOV	DX,COUNT1	; select address for Counter1
	MOV	AX,10	; Soundless transfer
	OUT	DX,AL	; LSB
	MOV	AL,AH	
	OUT	DX,AL	; MSB
	POP	AX	
	POP	DX	
	IRET		; Return from interrupt
CODE	ENDS		
	END	START	

2. 8259

```
CNT      EQU  3FD6H      ; address to select 8255 control word register (PPI - 3)
BPORT    EQU  3FD2H      ; address to select 8255 Port B
CSR      EQU  0FFC8H      ; address to select 8259 Command Word : A0 = 0
PORT     EQU  OFFCAH      ; address to select 8259 Command Word: A0 = 1

CODE SEGMENT
ASSUME CS:CODE,DS:CODE

START:   CLI              ; Clear Interrupt flag
         MOV  SP,4000H      ; Setup Stack Pointer
         MOV  AX,CS
         MOV  DS,AX          ; CS = DS

MAIN:    MOV  AX,0
         MOV  ES,AX          ; Setup ES = 0
         MOV  BX,40H*4        ; Setup Vector Address
         MOV  ES:WORD PTR[BX],OFFSET INTRO      ; Set IP in the Vector Table
         MOV  ES:2[BX],CS      ; Set CS in the Vector Table

         MOV  DX,CSR          ;A0=0
         MOV  AL,00010011B    ;ICW1 : ICW4 needed, Single 8259, Edge Triggered, When in 8086 D2 is don't care
         OUT  DX,AL

         MOV  DX,PORT          ;A0=1
         MOV  AL,40H            ;ICW2 : Interrupt type for IR0 : 64 Decimal
         OUT  DX,AL

         MOV  AL,00000101B    ;ICW4 : In 8086 Mode
         OUT  DX,AL

         MOV  AL,11111110B    ; OCW1 : Interrupt at IR0 is only unmasked or enabled
         OUT  DX,AL

         MOV  DX,CNT          ; Initializing 8255
         MOV  AL,80H            ; Output to all PORT
         OUT  DX,AL

         MOV  AL,01H            ; Initializing LED flickering Pattern
         MOV  DX,BPORT          ; Address for 8255 Port : B which is connected to LEDs.
         OUT  DX,AL
         STI                  ; Enable all type of Interrupt

         JMP  $

INTRO:   ROL  AL,1          ; LED Display manipulation
         OUT  DX,AL
         MOV  CX,5FFFH          ; Delay
         LOOP $                  ; End of loop
         PUSH AX
         PUSH DX
         MOV  DX,CSR
         MOV  AL,20H            ; OCW2 : EOI (End Of Interrupt) command : nonspecific EOI : to clear the ISR bit
                                ; that is servicing.
         OUT  DX,AL
         POP  DX
         POP  AX
         IRET                 ; Return to main routine.

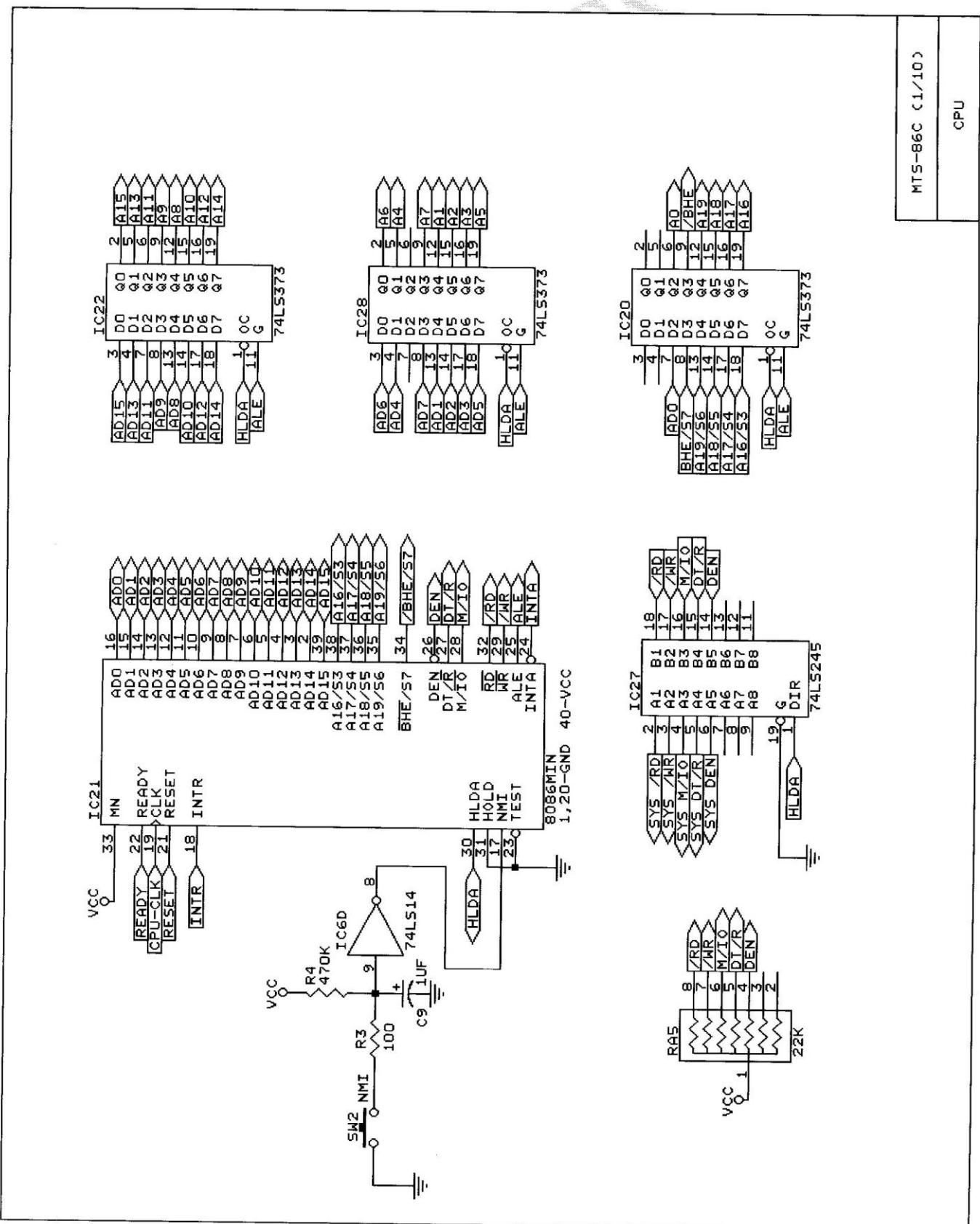
CODE ENDS
END      START
```

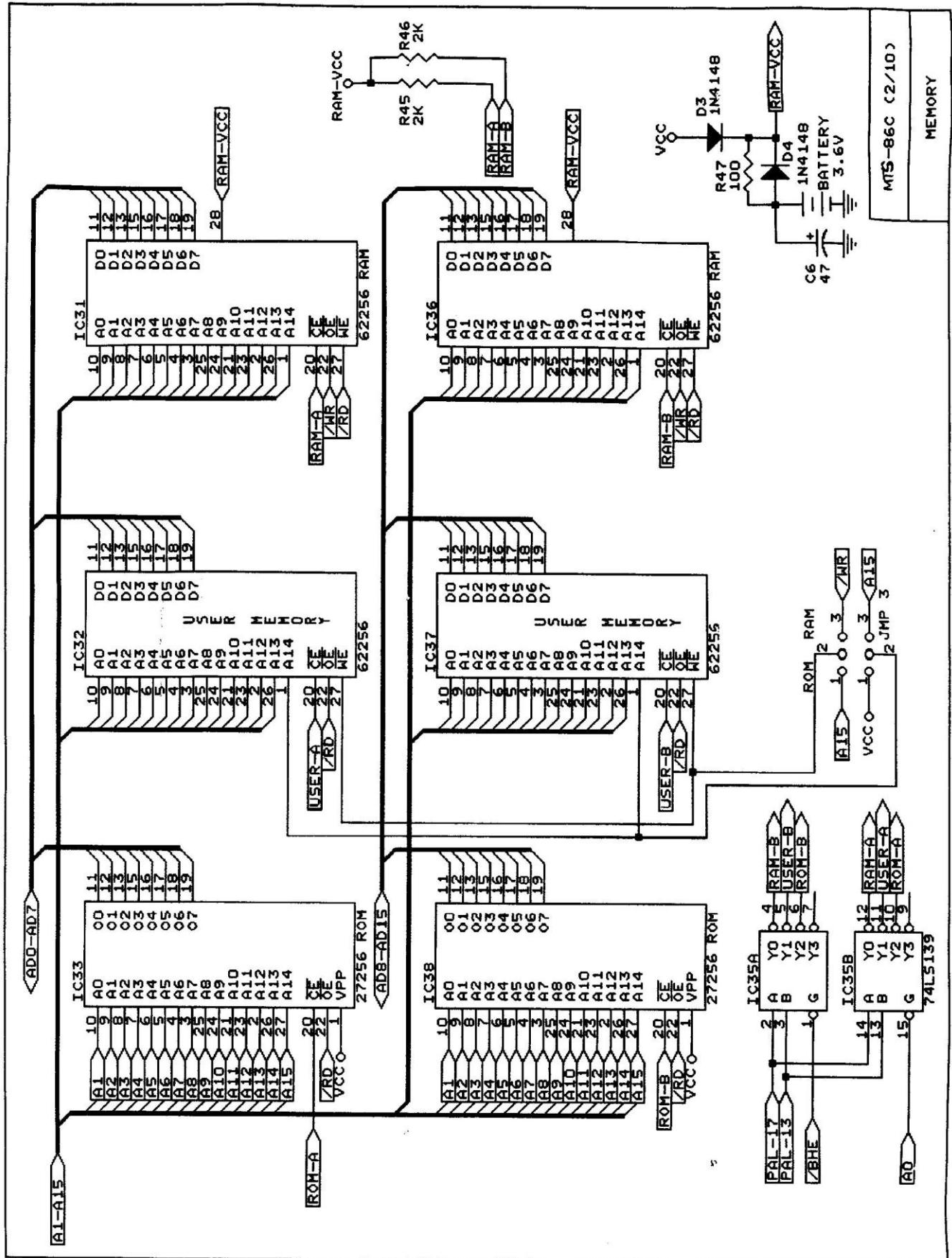
Exercise: Make a stop watch that will start at IR0 interrupt and Stop also at IR0 interrupt. Show the value of hour: min : sec in the LCD Display.

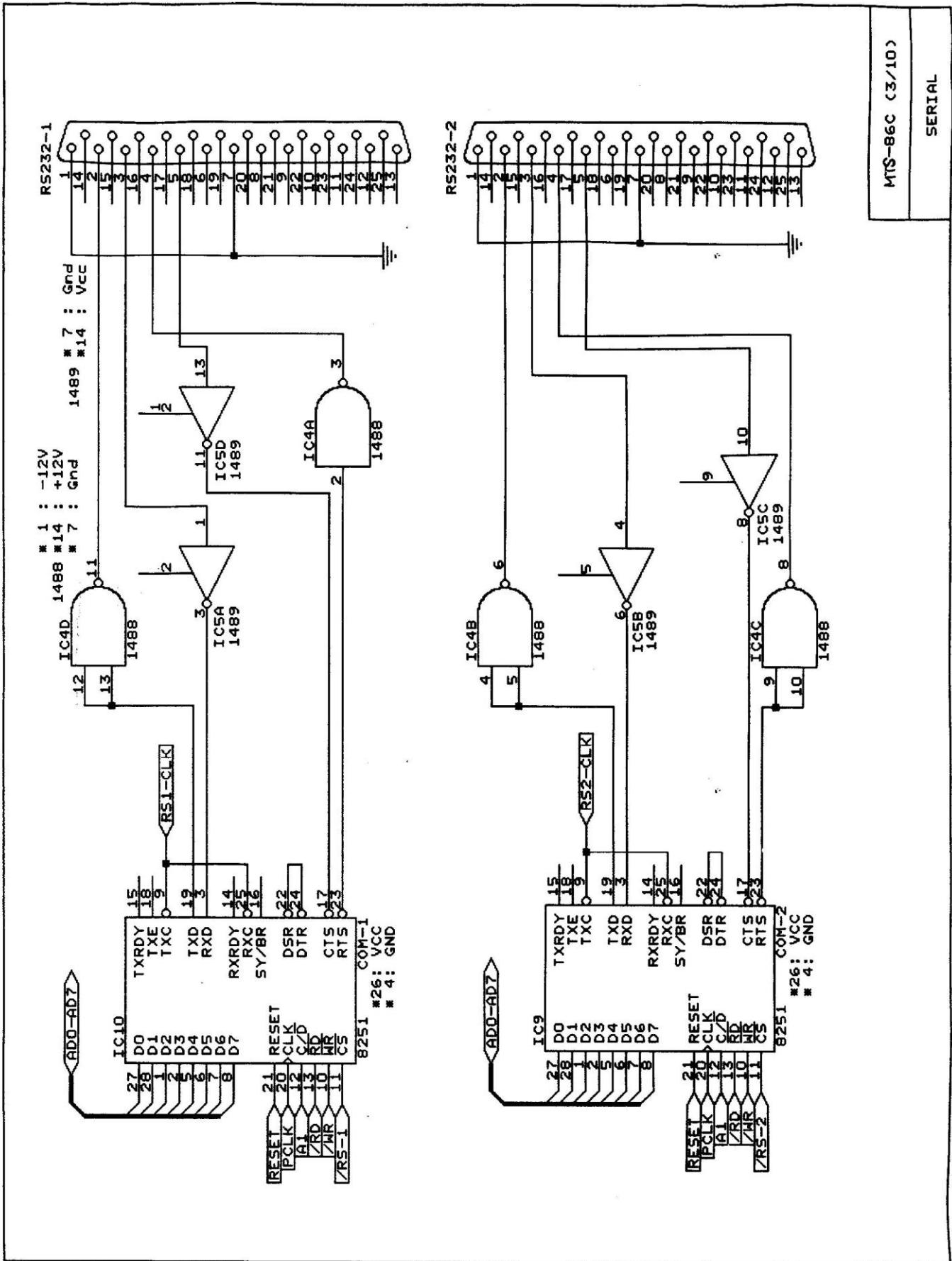
Prepared by : Rajib Mikail , Lecturer , EEE , BUET

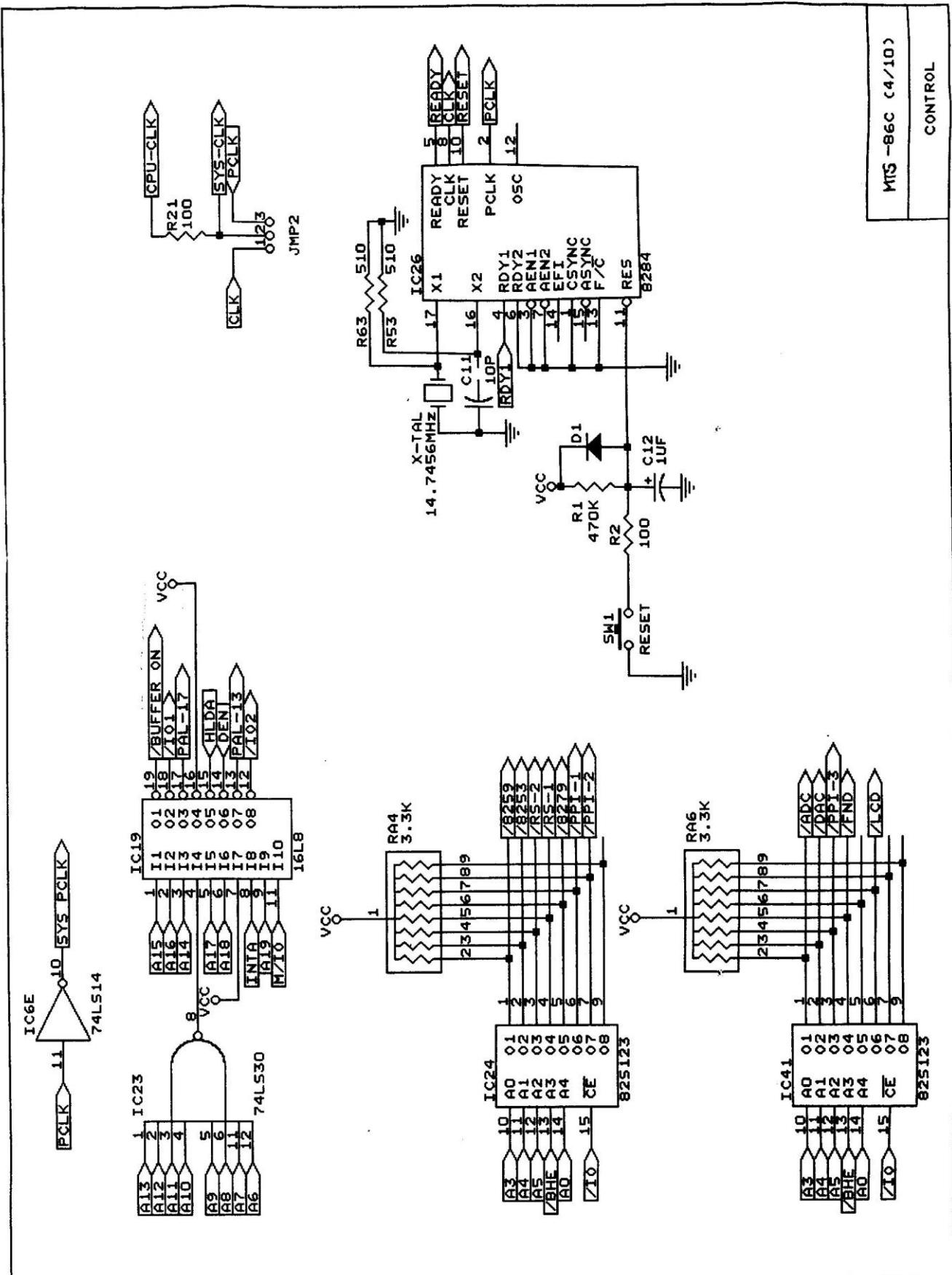
Appendix:

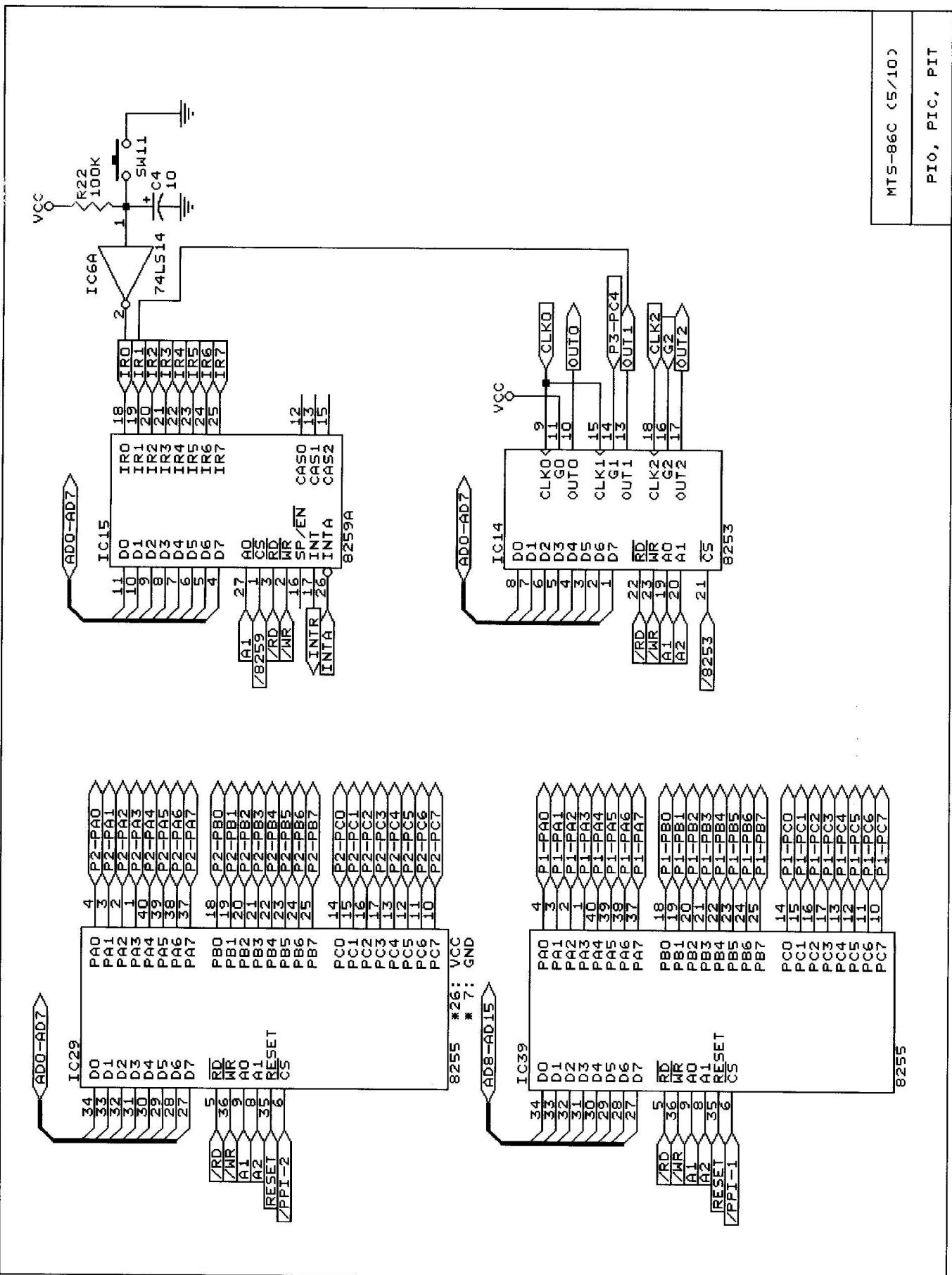
11-5. Circuit Diagram of MTS-86C











MTS-86C (5/10)

PIO, PIC, PIT

