__/__/____

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

CACHAR, ASSAM

DEC, 2020

B. TECH. III RD SEMESTER

Subject: EE223

Microprocessor and Microcontroller

Submitted by:

Name : Subhajit Ghimire

Branch : CSE - B

ScholarId : 1912160

Under the Guidance of:

Dr. Risha Mal

Assistant Professor

Department of Electrical Engineering

NIT Silchar

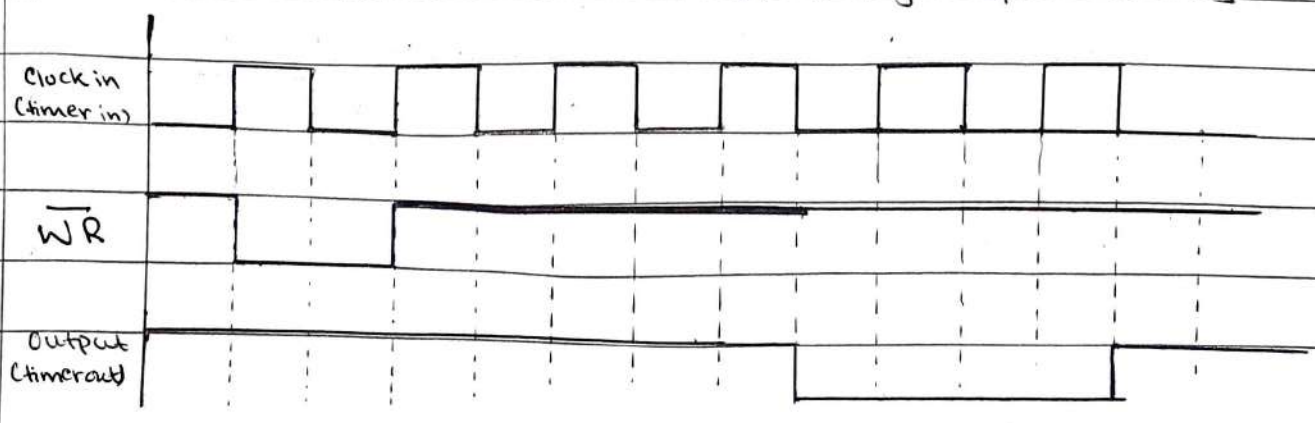
Question Set - 13 __/__/____

Q.1. <a> How can you design a square wave generator using 8155. Explain with appropriate diagram different modes of operation, control logics and instructions for the timer.

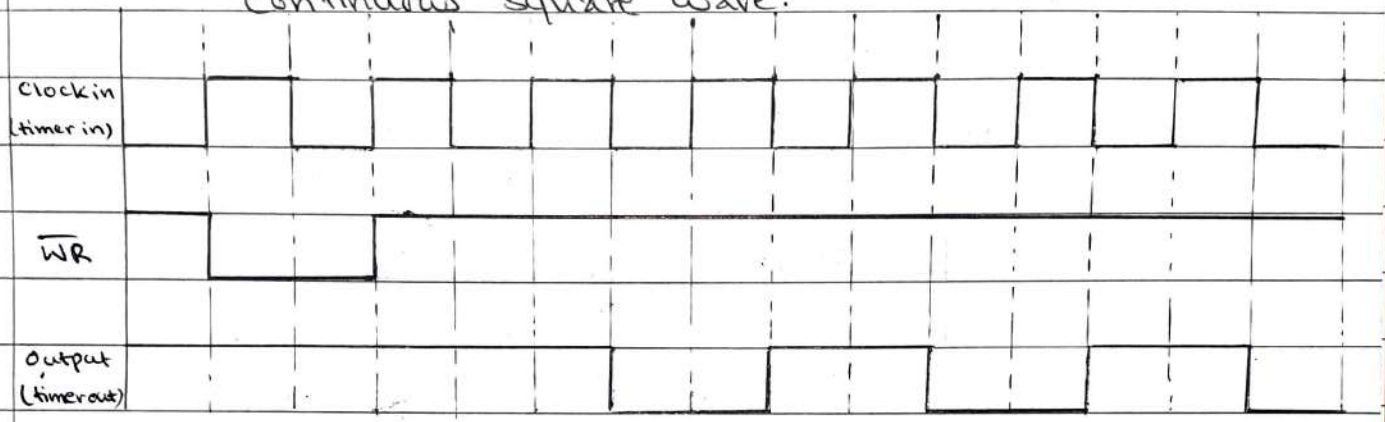
Ans. The 8155 and 8156 are RAM and I/O chips to be used in the 8085A and 8088 microprocessor systems. The RAM portion is designed with 2048 static cells organised as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085A CPU. The 8155-2 and 8156-2 have maximum access times of 330 ns for use with the 8085-A-2 and the full speed 5 MHz 8088 CPU.

Modes of Operation:

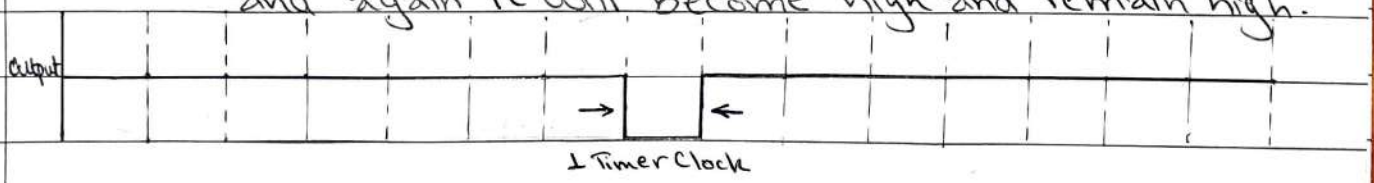
1. Mode 0: In this mode, timer gives only one cycle of square wave, the output remains high for 1/2 count. If count is odd, it remains high for (n+1)/2 and low for (n-1)/2, where n is the count value. Wave width depends on two factor: one is input clock pulse frequency, and the other is count loaded in counter. [Single Square Wave]



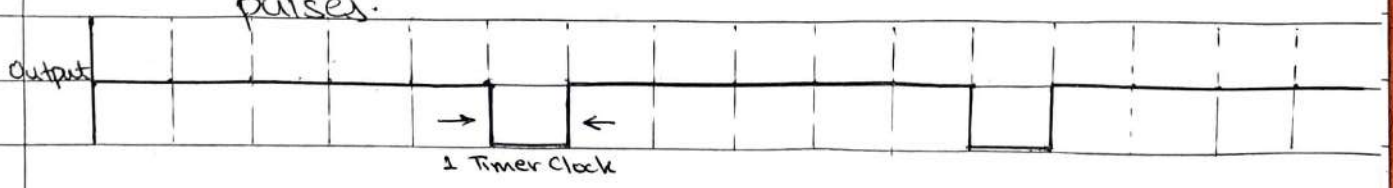
2. Mode 1 : This mode is similar to single square wave in operation but when the counter becomes zero, the count value is automatically reloaded. Thus, it provides continuous square wave.



3. Mode 2 : This mode gives a single clock pulse as a output of the end of the count. The output is normally high, but it becomes low for 1 clock pulse and again it will become high and remain high.



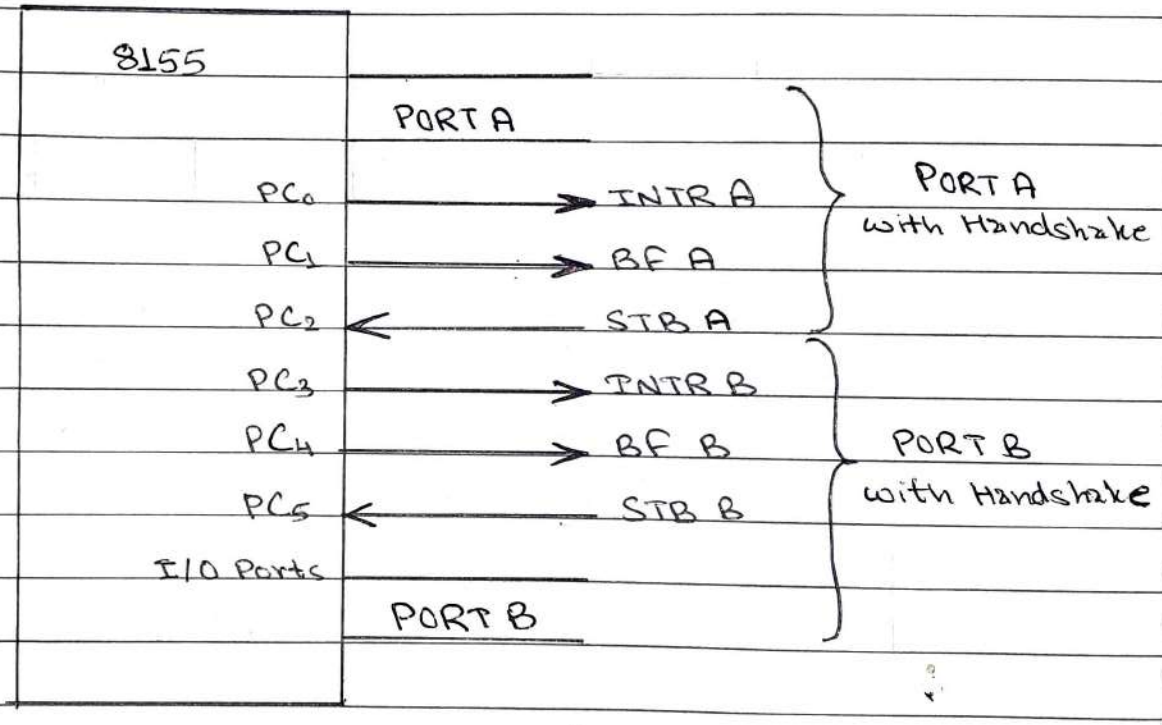
4. Mode 3: This mode is similar to mode 2 but when the counter becomes zero, the count value is automatically reloaded. Thus it provides continuous pulses.



__/__/____

Control Logics:

In handshake mode, when both the ports A and B are configured, port A uses the lower three signals of port C (i.e., PC_0, PC_1, PC_2) and port B uses the upper three signals (i.e., PC_3, PC_4, PC_5).



The functions of the above mentioned signals are:

1. STB (Strobe Input): This is connected from a peripheral to the 8155. The low on this signal informs the 8155 that data are strobed into the input port.

2. BF (Buffer full): This is an active high signal, which basically indicates the presence of a data byte in the port.

3. INTR (Interrupt Request): The rising edge of the STB signal generates this signal. This happens whenever the interrupt flipflop (INTE) is enabled. This can be used to interrupt the MPU.

4. INTE (Interrupt Enable): This is an internal flipflop which is used to enable or disable the interrupt capability of the 8155. The interrupts of port A and port B are controlled by bits D4 and D5, respectively, in the control register.

Instructions for the square wave generation set counter:

```
MVI  A, 0AH        } Configuring 8155
OUT  90H           }

Start: MVI A, 20H  }
       OUT 93H     } Start Conversion
       MVI A, 00H  }
       OUT 93H     }

Status: IN 90H     }
        ANI 02H    } Read Status register
        JZ Status  } Wait till BFA = 1

        IN 91H     ; Read temperature value
        OUT 92H    ; Display on LED
Again:  IN 90H     ; Start Read Status Register
        ANI 02H
        JNZ Again  ; Wait till BFA = 0
        JMP Start  ; Start Conversion Again.
```

Q1. <b> Calculate the count for 8155 timer to obtain square wave of 500 μs time period with clock frequency of 3.072 MHz.

Ans. Given,

Pulse period = 500 μs
= 500×10^{-6} s

Frequency = 3.072 MHz
= 3.072×10^{6} Hz

So, Clock period = $\frac{1}{3.072 \times 10^{6}}$ s

Count = $\frac{\text{Pulse period}}{\text{Clock period}}$

= $\frac{500 \times 10^{-6}}{\frac{1}{3.072 \times 10^{6}}}$

= $500 \times 10^{-6} \times 3.072 \times 10^{6}$

= 1536

∴ Count = $(1536)_{10}$ = $(0600)_{16}$

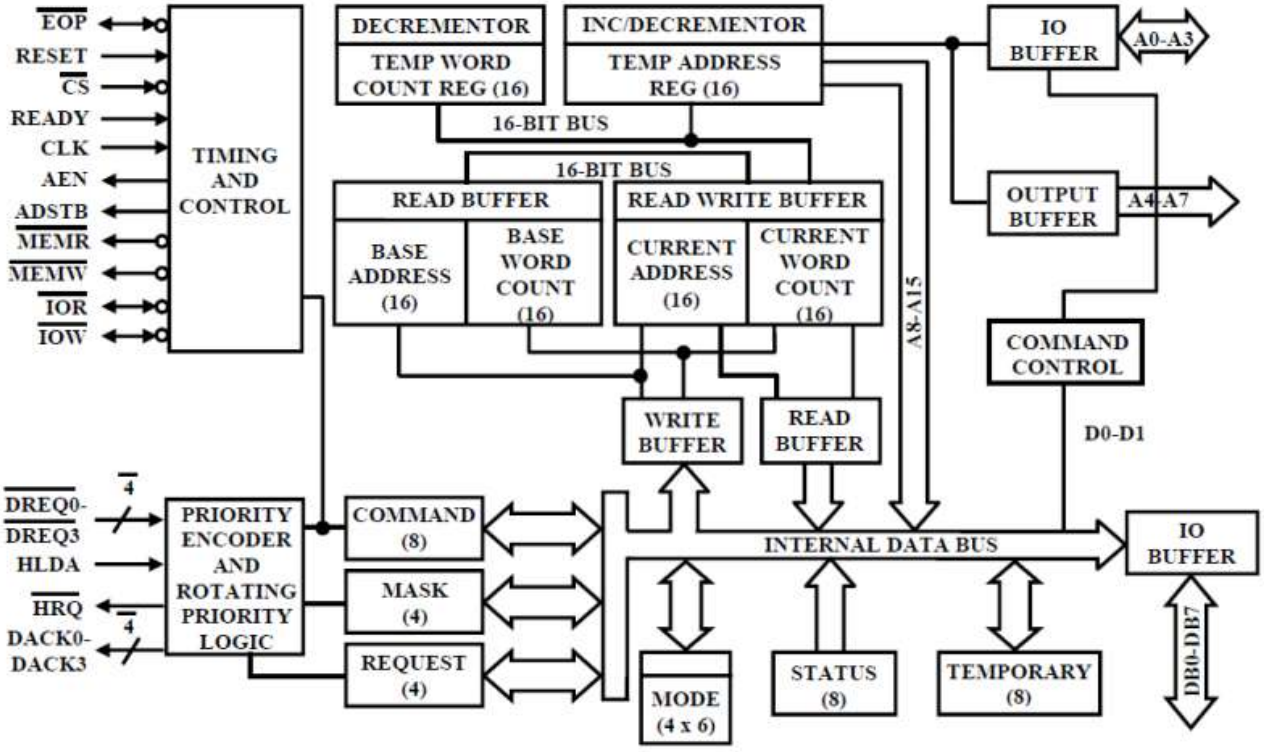
∴ The count is 600H.

__/__/____

Q.2. (a) What is 8237? Draw the block diagram representation of it and explain how it works.

Ans The 8237 is a four channel device that can be expanded to include any number of DMA channel inputs. The 8237 is capable of DMA transfers at rates up to 1.6 megabyte per second. Each channel is capable of addressing a full 64k-byte section of memory and can transfer up to 64K bytes with a single programming.

Block Diagram:



__/__/____

Working of 8237:

The 8237 direct memory access (DMA) controller is designed to improve the data transfer rate in systems which must transfer data from an I/O device to memory, or move a block of memory to an I/O device. It will also perform memory-to-memory block moves, or fill a block of memory with data from a single location. Operation modes are provided to handle single byte transfers as well as discontinuous data streams, which allows the 82C37A to control data movement with software transparency.

The 8237 operates in four different modes, depending upon the number of bytes transferred per cycle and number of ICs used.

i) Single : One DMA cycle, one CPU cycle interleaved until address counter reaches zero.

ii) Block : Transfer progresses until the word count reaches zero or the EOP signal goes active.

iii) Demand : Transfers continue until TC or EOP goes active or DRQ goes inactive. The CPU is permitted to use the bus when no transfer is requested.

iv) Cascade: Used to cascade additional DMA controllers. DREQ and DACK is matched with HRQ and HLDA from the next chip to establish a priority chain. Actual bus signals is executed by cascaded chip.

The timing and control block derives internal timing from clock timing from clock input, and generates external control signals. The priority encoder block resolves priority contention between DMA channels requesting service simultaneously.

Q.2. (b) Write the initialisation instruction for DMA controller with the following specifications.

- Disable the DMA controller and initialise.
- Initialise Channel #3 to transfer 1K bytes from system memory to floppy disk assigned to channel 3.
- Starting address for block data is 4075H.
- Set up the demand mode whereby DMA can complete the data transfer without interruption.

Ans. The initialisation instructions to set up the DMA controller are as follows:

__/__/____

```
MVI A, 00000100B  ; Command: 0 0 0 0 0 1 0 0
                                    |
                                Disable DMA
OUT 08H           ; Send to Command Register.

MVI A, 00000111B  ; Mode: 00 0 0 01 1 1
                          |  | |  |  |
                          |  | |  |  Ch-3
                          |  | |  Write
                          |  | Disable Auto Load
                          |  Increment Adder
                          Demand mode
OUT 0BH           ; Send to Mod Register.

MVI A, 75H        ; Low-order byte of starting address

OUT 06H           ; Output to Ch-3 Memory Address Register.

MVI A, 40H        ; High-order byte of starting address.

OUT 06H           ; Output to Ch-3 Memory Address Register.

MVI A, FFH        ; Low-byte order of count 03FFH

OUT 07H           ; Output to Ch-3 Count Register.

MVI A, 03H        ; High-order byte of count 03ffH

OUT 07H           ; Output to Ch-3 Count Register.

MVI A, 10000000B  ; Command: 10 0 0 0 00 0
                             |  | | | |  |
                             |  | | | |  Disable MEM to MEM
                             |  | | | DMA Enable
                             |  | | Normal Time
                             |  | Fixed Priority
                             |  Late Write
                             DACK DREQ High
OUT 07H           ; Send to Command flag.
```

__/__/___

Q.8. Write a program to generate a rectangular wave with a 200 μs on-period and a 400 μs off-period.

Ans.

```
Main:   MVI A, 90H      ; Stores 90H to accumulator
        SIM             ; Sending 1 to SOD
        CALL delay1     ; Calling delay of 200ms
        MVI A, 40H      ; Storing 40H to accumulator
        SIM             ; Sending 0 to SOD
        CALL delay1     ; Calling delay of 200μs
        CALL delay1     ; Calling delay of 200μs, making 400μs
        JMP Main        ; Unconditional jump to main

delay1: MVI C, 0EH      ; Storing 0EH to C register
Loop1:  DCR C           ; Decreasing C content by 1
        JNZ Loop1       ; If C≠0, Jumps to Loop1
        RET             ; Return
```