Military College of Signals-NUST
Mid Term Examinations – Solution BESE 15 B
Computer Organization & Architecture

Instr: Asst Prof Athar Mohsin Time: 1 ½ hrs
Marks: 30

Q#1. Briefly answer following questions **(12 Marks)**

a. What is EU & BIU? Explain their purpose.

Ans: The 8088/ 8086 CPUs are organized as two processing units, called the

a. Bus Interface Unit (BIU) and the Execution Unit (EU). Each unit has a dedicated function and both operate at the same time

BIU

a. The BIU provides H/W functions, including generation of the memory and I/O addresses for the transfer of data between the outside world -outside the CPU and the EU.
b. BIU contains
    a. A prefetch queue - FIFO
    b. A bus controller
    c. Segment register
    d. Instruction pointer (IP)
c. Main purpose is
    a. To fill the prefetch queue filled with instructions
    b. To generate and accept the system control signals
    c. To provide system with a memory address or I/O port number
    d. To act as a window between EU and memory for data

The EU

a. Fetch program instruction codes and data from the BIU, executes these instructions, and store the results in the general registers.
b. The EU is responsible for decoding and executing instructions

The purpose

a. To carryout instructions that are fetched from the prefetch queue

It contains

a. An Arithmetic Logic Unit (ALU)
b. Status and control flags
c. An instruction register – to receive instructions from prefetch queue
d. A register array – to hold information temporarily
e. Also contain pointers and index register – to address data located in the memory

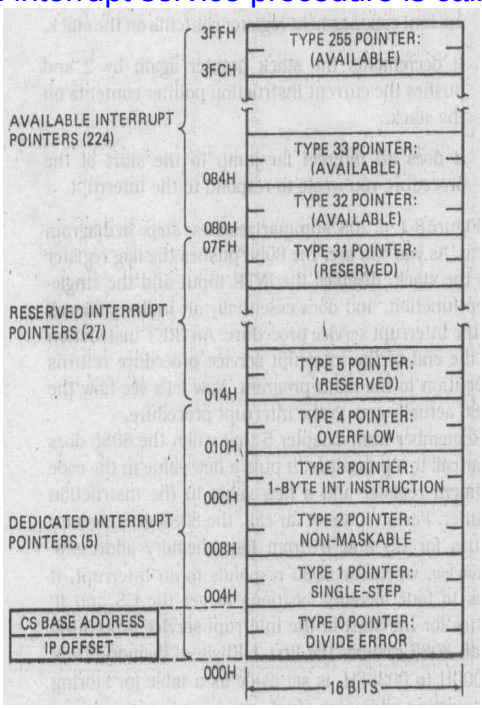
b. What is prefetch queue? What is the advantage of having a prefetch queue in 8086 microprocessor

Ans: While EU is decoding or executing an instruction (bus is not required)

a. BIU fetches up to six instructions for the next instruction
b. BIU stores these perfected instructions in Queue (FIFO)
c. When EU is ready for its next instruction it simply read the instruction from the queue
d. The advantage of having a prefetch queue is to Speedup the process as Prefetching allows EU to obtain the next instruction directly from BIU instead of the memory- allow higher speed
e. When EU is not asking to read/ write data from memory- BIU looks ahead in the program to prefetch next instruction
    i. Prefetch instructions are held in FIFO queue
    ii. Code is held until EU accepts it

c. What is the purpose of interrupt vector table (IVT), and where it is located.

Ans: In 8086 the first 1 Kbyte of memory (00000H to 003FFH) is set aside as a table for storing the starting addresses of interrupt service procedures. Table can hold starting addresses of 256 interrupt procedures. Starting address of an interrupt service procedure is called “interrupt vector”



d. What is the difference between SRAM & DRAM in terms of application and characteristics like speed, size & cost?

SRAM- Static RAm	DRAM- Dynamic RAM

Storage cells are made of flip-flops and therefore do not require refreshing in order to keep their data • The problem with the use of flip-flops for storage cells is that each cell requires at least 6 transistors to build, and the cell holds only 1 bit of data • SRAMs are widely used for cache memory. • retain data as long as the power is on, which do not need to be refreshed, and whose contents can be accessed at any time	The use of a capacitor as a means to store data cuts down the number of transistors needed to build the cell; • it requires constant refreshing due to leakage • This is in contrast to SRAM (static RAM), whose individual cells are made of flip-flops • Since each bit in SRAM uses a single flip-flop, SRAM has much larger memory cells and consequently lower density • The use of capacitors as storage cells in DRAM results in much smaller memory cell size • it must be refreshed periodically, due to the fact that the capacitor cell loses its charge;

e. Explain what is von Neumann architecture, and why most digital computers, suffers from the von Neumann bottleneck.

Ans: Under the von Neumann architecture, a program and its data are both stored in memory. It is therefore possible for a program, thinking a memory location holds a piece of data when it actually holds a program instruction, to accidentally (or on purpose) modify itself.

The von Neumann architecture stores both program code and data in memory. There is a single path between main memory and the CPU, forcing instruction fetch cycles and execution cycles to share this pathway.

f. In performing following operation on the data items which interrupt will occur (show your result in binary notation) and how 8086 will respond to this interrupt.

(1) Data 1 = + $(64)_{10}$
(2) Data 2 = + $(64)_{10}$

Ans: the result of this operation will be : OVERFLOW" as addition of two signed number will result in changing the sign bit, moreover the result of six bit data will requires seven bit storage hence overflow, the microprocessor will enter in type 4 or overflow interrupt.

When an interrupt occurs the CPU completes the current instruction and then:-–

(1) Disables the maskable interrupt (This prevents the interrupt from itself being interrupted. Programmer may over-ride this behavior within the ISR
(2) Saves the IP program counter, CS code segment register, and Flags register on the stack.
(3) Jumps to an address found in memory locations 4*N, where N is the number of the interrupt.

(4) Executes an ISR found at that address
(5) At the end of the ISR executes an iret instruction which pulls IP, CS and the Flags register off the stack, restoring the CPU to the status it had before the interrupt occurred.

Q#2. **(6 Marks)**

a. How many bits would be needed to address if the memory is byte addressable and word addressable to address following?
(1) A 2M x 32 Memory
(2) A 4M x 16 main memory

Ans:

2M x 32 Memory

(1) There are 2M × 4 bytes which equals $2 \times 2^{20} \times 2^{2} = 2^{23}$ total bytes, so 23 bits are needed for an address
(2) There are 2M words which equals $2 \times 2^{20} = 2^{21}$, so 21 bits are required for an address

4M x 16 memory

(1) There are 4M × 2 bytes which equals $2^{2} \times 2^{20} \times 2 = 2^{23}$ total bytes, so 23 bits are needed for an address
(2) There are 4M words which equals $2^{2} \times 2^{20} = 2^{22}$, so 22 bits are required for an address

b. How many 256 × 8 RAM chips are needed to provide a memory capacity of 4096 bytes?
(1). How many bits will each memory address contain?
(2). How many address lines must go to each chip?
(3). How many lines must be decoded for the chip select inputs? Specify the size of the decoder.

To get 4096 bytes of memory, one would need 16, 256x8 RAM chips (256x16=4096).

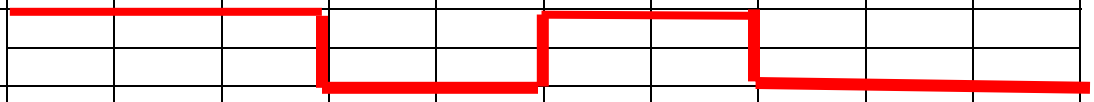
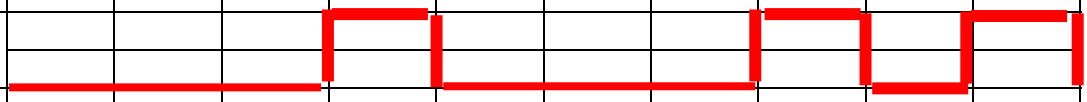
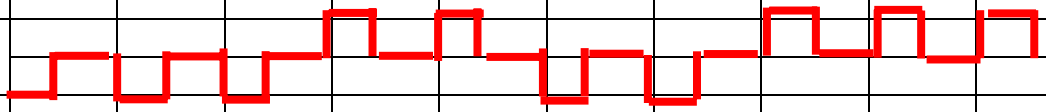
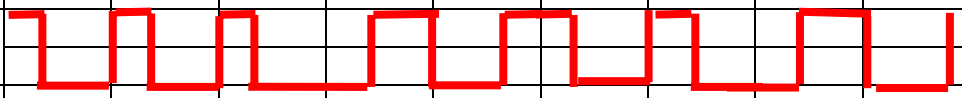
a. each memory address contain 8 bit
b. 8 address line needed
c. 4 lines are requires for decoder, which implies a 4-to-16 decoder.

Q#3. Represent the data stream (three 0s followed by two 1s followed by two 0s and three 1s [**0001100111**]) for NRZ-L, NRZ-I, RZ and Manchester encoding schemes. For claiming the marks, Show your work neatly and with clear interval levels. **(6 Marks)**

Q#4. Suppose we are working with an error-correcting code that will allow all single-bit errors to be corrected for memory words of length 7**(1011001)**. How many check bits are needed, and what will be the length of the code words according to the Hamming Algorithm. If we receive the code word including check bits and the data bits (excluding check bits) are: (**1011011).** Present the

actual hamming code word (Assuming even parity), Check correct received data bits, according to error-correcting code. **(6 Marks)**

Ans:

	0	0	0	1	1	0	0	1	1	1
NRZ-L										
NRZ-I										
RZ										
Manch hester										

Ans:

Data sent: 1011001, the hamming code construction is as fol:

- 7 data bits
- 4 Check bits will be required
  - $2^k - 1 >= m+k$
    → $2^4 - 1 >= 7 + 4$
    → 15 >= 11 so 4 redundant bits and 11 bit code

D7	D6	D5	R4	D4	D3	D2	R3	D1	R2	R1
1	0	1	-	1	0	0	-	1	-	-
1		1		1		0		1		R1=0
1	0			1	0			1	R2=1	
				1	0	0	R3=1			
1	0	1	R4=0							
1	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**

So the code word sent → **10101001110**
10101011110

The received word → 1011011

D7	D6	D5	R4	D4	D3	D2	R3	D1	R2	R1
1	0	1	0	1	0	1	1	1	1	No R1=~~0~~, 1
1	0			1	0			1	Yes R2=~~1~~ 0	
				1	0	1	No R3=~~1~~ 1			
1	0	1	yes R4=0							
1	**0**	**1**	**0**	**1**	**0**	1	**1**	**1**	**0**	**1**

Error was detected on Bit no 5 that can be removed