

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-3/T-1 B. Sc. Engineering Examinations 2018-2019

Sub : **CSE 315** (Microprocessors, Microcontrollers, and Embedded Systems)

Full Marks : 210

Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION – AThere are **NINE** questions in this section. Answer any **SEVEN**.

1. (a) Explain the following items very briefly with examples.
 i. N-bit processor, ii. Superscalar processor (5×2=10)
 (b) Show the difference of DIP and PGA chips with diagrams only. (5)
2. (a) Suppose you want to access memory location 00111H to 00114H using at 8086 µP.
 Minimum how many clock are required? Draw the signals (timing diagram) of **BHE** and
A0 throughout these clock cycles. (4+6=10)

BHE	A0	Accessed Bank	Data Bits
0	0	Both banks	D0 – D15
0	1	Odd bank	D8 – D15
1	0	Even bank	D0 – D7
1	1	None	None

- (b) Why does 8086 µP have two ground pins? (5)
3. Suppose you want to print a triangular pattern with the following 8086 assembly code.
 Complete the code snippet by writing the necessary instructions after the **PRINT_COL** label (writing only these instructions in your answer script will be sufficient). Note that you cannot make any modification anywhere else in the given code. Check out the expected input-output for understanding the pattern better. (15)

```

;BL contains # of rows
    MOV BH, BL
    MOV AH, 2
PRINT_ROW:
    MOV CL, BL
    MOV CH, 0
PRINT_COL:
    ; write your code
STAR:
    MOV DL, '*'
    JMP PRINT
SKIP:
    MOV DL, ' '
PRINT:
    INT 21H
    LOOP PRINT_COL
    MOV DL, 0DH
    INT 21H
    MOV DL, 0AH
    INT 21H
    DEC BL
    JNZ PRINT_ROW
  
```

Expected Input-Output	
Input	Output
6	***** *...* *..* *.* ** *
5	***** *..* *..* ** *
4	**** *.* ** *
3	*** ** *
2	** *

Figure for Q.3

CSE 315

4. (a) Suppose CS:IP = 0024:02A0 and DS:SI = 0040:00E0 in an 8086 µP. Calculate the physical address for both the segment:offset pairs and comment on the results. (8+2=10)
- (b) Calculate the amount of overlap in the code and data segment for the values of segment registers mentioned in Q.4(a). (5)
5. (a) Suppose CS = 0028H in an 80286 µP. Calculate the starting and ending address of the code segment considering the µP is working in protected mode. You must show the steps in detail. (5+5=10)

Descriptor No.	Global Descriptor Table	Local Descriptor Table
...
8	000012011BBA2BBA	0000A18432001101
7	0000A10BCC13158E	0000B40011220110
6	0000AA7281008564	000072A264793CDF
5	00009A013B2A2B1A	00008323A216320B
4	000000C1A4687A33	000090900912AACD
3	0000273654AA2245	0000606FF221982A
2	0000128112A15689	00002217391BC1D2
1	00000000000040004	0000CDDEFF14457A
0	0000A66219209171	00001522348A1AA3

Bit 3-15: Selector Bit 2: TI Bit 0-1: RPL
Segment Register (Protected Mode)

Byte 6-7: Reserved	Byte 5: Access Rights	Byte 4: Base Address
Byte 2-3: Base Address	Byte 0-1: Limit	

80286 Descriptor

Figure for Q.5(a)

- (b) Calculate the physical addresses for (i) IP = 19F2H & (ii) IP = A016H. Use necessary information of code segment as found by solving Q.5(a). (5)
6. Suppose you have to design a new version of 80386 µP that uses a 3-level paging mechanism. It will support upto 4GB of memory with a fixed page size of 16KB. Moreover, you want the page tables at each level to have twice the number of entries as those of its previous level.

Now answer the following questions. (5×3=15)

- (i) What is the total number of pages?
- (ii) How can you parse the 32-bit linear address received from segmentation unit? Show the splits with a figure. (Note that, the figure to explain the whole paging mechanism is not required here.)
- (iii) How many entries will there be in level 1, 2, 3 page tables?

CSE 315

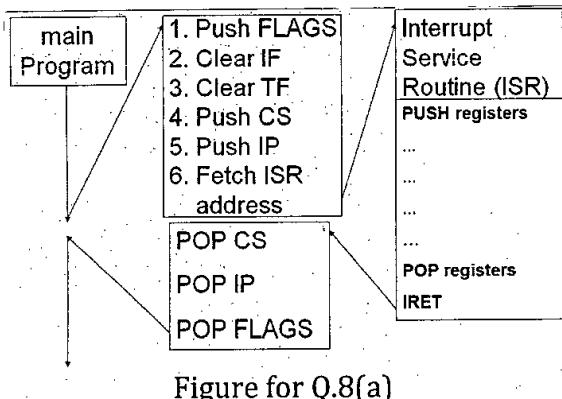
7. (a) Suppose a μ P have to wait for at least 1 second before polling on the I/O devices in a Polled I/O based system. For this purpose you are given the following code snippet. (7+3=10)

```
    MOV CX, n
DELAY:
    MOV m, CX
    MOV CX, n
DELAY_2:
    LOOP DELAY_2
    MOV CX, m
    LOOP DELAY
...
Figure for Q.7(a)
```

If the μ P is running at 10MHz, what should you put as the value of n? What is the actual amount of delay you will achieve with your calculated value of n? You can assume it takes 4 clock cycles to execute a MOV instruction. 17 clock cycles are required to execute a LOOP instruction when it jumps to the target address and 5 clock cycles otherwise.

- (b) Why do the number of available 8086 instructions and opcodes differ by a significant margin? (5)

8. (a) What is wrong with the following interrupt handling procedure? In what situation will it work despite the mistake? (5+5=10)



- (b) Explain double indirect jump with an example. Why is it called so? (5)
9. (a) Briefly explain Hyper Threading (HT) technology with appropriate diagram. (5)
- (b) Why is the address bus bidirectional in Pentium? (5)
- (c) What are the advantages of using physically smaller transistors in Pentium IV? (5)

CSE 315

SECTION – B

There are **FOUR** questions in this section. Answer any **THREE**.

- Assume system clock frequency 1 MHz if not given.
- List of registers and necessary diagrams are at the end of the question.
- If configuration of any required register is missing, just assume a configuration and clearly show the assumed configuration.
- If any control word/bit configuration is missing in the question paper, just assume a pattern of your choice and clearly mention your assumption.

10. (a) Consider the circuit diagram in Figure 10(a). You have to increment an 8 bit counter when switch A is pressed using interrupt. You also have to decrement the counter when switch B is pressed using interrupt. Each interrupt must trigger only a single time as soon as the corresponding switch is pressed. Write two Interrupt Service Routines and the main function in C. (Clearly mention the value assigned to related registers in hexadecimal)

(15)

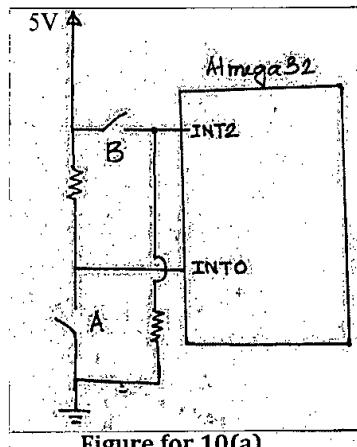


Figure for 10(a)

(b) Write C code for the exact scenario of Q. 10(a) using polling instead of interrupt.

(Clearly mention the value assigned to related registers in hexadecimal)

(10)

(c) Suppose you have to monitor 16 devices using a CPU. Each device has a 4-bit unique ID assigned to it. If multiple devices get triggered at the same time, you must monitor the device having the lowest ID. Which approach should you take to monitor the devices - polling or interrupt? Give three strong reasons supporting your answer.

(10)

11. (a) What steps does the CPU follow to execute an interrupt?

(10)

(b) Briefly write one major problem of using `delay_ms()` of ATmega32.

(5)

(c) Convert the following analog inputs to digital.

(5×2=10)

(i) Reference voltage = 2.56V, $V_{min} = 0V$, 8-bit ADC. Analog input 1.45V

(ii) Reference voltage = 5V, $V_{min} = 0V$, 10-bit ADC, Analog input 4.17V

(d) The digital output of ATmega32 ADC is stored in two 8-bit registers: ADCL and ADCH. Explain why ADCL must be read first before ADCH.

(10)

CSE 315

12. Suppose, Harold wanted to send/receive data using UART with his Atmega32. System clock frequency is 10^6 Hz. In his C code he wrote the following functions to initialize UART and send character data.

```
// code fragment of Question 12
void UART_init(void)
{
    UCSRA = 0b00000010;
    UCSRB = 0b00011100;
    UCSRC = 0b10111110;
    UBRRL = 0x31;
    UBRRH = 0x00;
}
void UART_send(unsigned int data)
{
    while((UCSRA & (1 << UDRE)) == 0x00);
    UDR = data;
}
```

He saw that data were not transmitting correctly. He then found an error in his `UART_send` function and corrected it later.

Answer the following three questions based on this scenario.

- (a) How many bits will be transmitted/received in 1 minute for this configuration? (10)
- (b) Assume that data is now transmitting correctly. Draw the framing diagram for transmitting data = 100010100? (10)
- (c) What was the error in `UART_send` function? Explain briefly. Add at most three lines of code to correct the given function. (15)
13. (a) Suppose you want to use SPI for sending data from Atmega32 to four different devices A, B, C and D. You also must occupy minimum number of Atmega32 pins. Draw a block diagram showing how you can achieve this. (15)
- (b) Suppose you wrote a code to measure elapsed time of a code segment. At the beginning of the code segment, you have reset TCNT1. During the execution of the code segment, TIMER1 overflow interrupt was triggered 50 times. After execution of the code segment TCNT1 value was 1200. What was the elapsed time in seconds? (TIMER1 Prescaler is 8) (10)
- (c) Eliot wrote the following function `EEDROM_write` to write character into a specific address of EEPROM. Unfortunately this function will not work correctly some time. Explain why this will not work correctly. Also insert line(s) to correct the function. (10)

```
// code segment for Question 13(c)
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    EEAR = uiAddress;
    EEDR = ucData;
    EECR |= (1<<EEMWE);
    EECR |= (1<<EEWE);
}
```

Figures and Charts for Section B

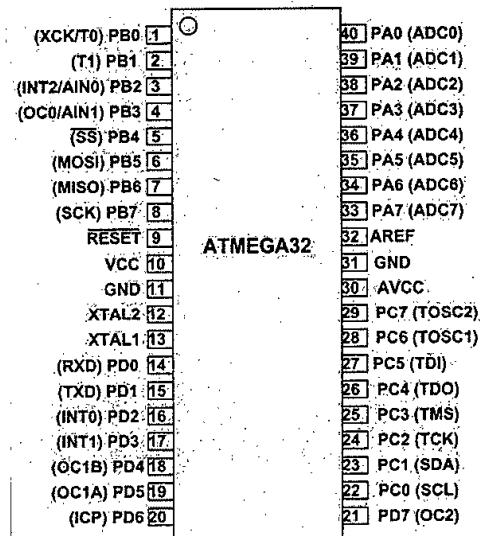


Figure: Atmega32 MCU Pinout

Register Name	Configuration							
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
MCUCR	-	-	-	-	ISC11	ISC10	ISC01	ISC00
Trigger codes: 00 → low level, 01 → any logical change, 10 → falling edge, 11 → rising edge								
MCUCSR	JTD	ISC2	-	-	-	-	-	-
ISC2: 0 → falling edge trigger, 1 → rising edge trigger								
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
URSEL, UMSL, UPM1, UPM0, USBS, UCSZ1, UCSZ0, UCPOL								
UCSRC	UCSZ2, UCSZ1, UCSZ0: 000 → 5-bit, 001 → 6-bit, 010 → 7-bit, 011 → 8-bit, 111 → 9-bit UPM1, UPM0: 00 → No parity, 10 → even parity, 11 → odd parity USBS: 0 → 1 stop bit, 1 → 2 stop bits							
ECCR	-	-	-	-	EERIE	EEMWE	EEWE	EERE

Table: Configurations of Atmega32 registers

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-3/T-1 B. Sc. Engineering Examinations 2018-2019

Sub : CSE 315 (Microprocessors, Microcontrollers, and Embedded Systems)

Full Marks : 210

Time : 3 Hours

The figures in the margin indicate full marks.
USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION - A

There are NINE questions in this section. Answer any SEVEN.

- ✓ 1. (a) Explain the following items very briefly with examples.

i. N-bit processor, ii. Superscalar processor

(5x2=10)
(5)

- (b) Show the difference of DIP and PGA chips with diagrams only.

2. (a) Suppose you want to access memory location 00111H to 00114H using 8086 µP. Minimum how many clock cycles are required? Draw the signals (timing diagram) of BHE and A0 throughout these clock cycles.

(4+6=10)

BHE	A0	Accessed Bank	Data Bits
0	0	Both banks	D0 - D15
0	1	Odd bank	D8 - D15
1	0	Even bank	D0 - D7
1	1	None	None

- (b) Why does 8086 µP have two ground pins?

(5)

- ✓ 3. Suppose you want to print a triangular pattern with the following 8086 assembly code. Complete the code snippet by writing the necessary instructions after the PRINT_COL label (writing only these instructions in your answer script will be sufficient). Note that you cannot make any modification anywhere else in the given code. Check out the expected input-output for understanding the pattern better.

```
;BL contains # of rows
MOV BH, BL
MOV AH, 2
PRINT_ROW:
    MOV CL, BL
    MOV CH, 0
PRINT_COL:
    ; write your code
STAR:
    MOV DL, '*'
    JMP PRINT
SKIP:
    MOV DL, '.'
PRINT:
    INT 21H
    LOOP PRINT_COL
    MOV DL, 0AH
```

Expected Input-Output	
Input	Output
6	***** * 1 2 3 4 5 6 * ... * * .. * * . * ** *
5	***** * .. * * . * ** *
4	**** * .. * **

label (writing only these instructions in your answer script will be sufficient). Note that you cannot make any modification anywhere else in the given code. Check out the expected input-output for understanding the pattern better.

(15)

```

;BL contains # of rows
    MOV BH, BL
    MOV AH, 2
PRINT_ROW:
    MOV CL, BL
    MOV CH, 0
PRINT_COL:
    ; write your code
STAR:
    MOV DL, '*'
    JMP PRINT
SKIP:
    MOV DL, '.'
PRINT:
    INT 21H
    LOOP PRINT_COL
    MOV DL, 0DH
    INT 21H
    MOV DL, 0AH
    INT 21H
    DEC BL
    JNZ PRINT_ROW

```

Expected Input-Output	
Input	Output
6	***** *123456 *...* *.* *.* ** *
5	**** *..* *. ** *
4	*** *.* ** *
3	** ** *
2	** *

Figure for Q.3

Contd P/2

CSE 315

- Q4 (a) Suppose CS:IP = 0024:02A0 and DS:SI = 0040:00E0 in an 8086 µP. Calculate the physical address for both the segment:offset pairs and comment on the results. (8+2=10)
(b) Calculate the amount of overlap in the code and data segment for the values of segment registers mentioned in Q.4(a). (5)

- Q5 (a) Suppose CS = 0028H in an 80286 µP. Calculate the starting and ending address of the code segment considering the µP is working in protected mode. You must show the steps in detail. (5+5=10)

Descriptor No.	Global Descriptor Table	Local Descriptor Table
...
8	000012011BBA2BBA	0000A18432001101
7	0000A10BCC13158E	0000B40011220110
6	0000AA7281008564	000072A264793CDF
5	00009A013B2A2B1A	00008323A216320B
4	000000C1A4687A33	000090900912AACD
3	0000273654AA2245	0000606FF221982A
2	0000128112A15689	00002217391BC1D2
1	0000000000040004	0000CDDEFF14457A
0	0000A66219209171	00001522348A1AA3

Bit 3-15: Selector | Bit 2: TI | Bit 0-1: RPL

Bit 3-15: Selector Bit 2: TI Bit 0-1: RPL		
Segment Register (Protected Mode)		
Byte 6-7: Reserved	Byte 5: Access Rights	Byte 4: Base Address
Byte 2-3: Base Address	Byte 0-1: Limit	
80286 Descriptor		

Figure for Q.5(a)

(b) Calculate the physical addresses for (i) IP = 19F2H & (ii) IP = A016H. Use necessary information of code segment as found by solving Q.5(a). (5)

(c) Suppose you have to design a new version of 80386 μ P that uses a 3-level paging mechanism. It will support upto 4GB of memory with a fixed page size of 16KB. Moreover, you want the page tables at each level to have twice the number of entries as those of its previous level.

Now answer the following questions. $(5 \times 3 = 15)$

(i) What is the total number of pages?

(ii) How can you parse the 32-bit linear address received from segmentation unit? Show the splits with a figure. (Note that, the figure to explain the whole paging mechanism is not required here.)

(iii) How many entries will there be in level 1, 2, 3 page tables?

Contd P/3

- Q (a) Suppose a μ P have to wait for at least 1 second before polling on the I/O devices in a Polled I/O based system. For this purpose you are given the following code snippet. (7+3=10)

```
    . . .
    MOV CX, n
DELAY:
    MOV m, CX
    MOV CX, n
DELAY_2:
    LOOP DELAY_2
    MOV CX, m
    LOOP DELAY
    . . .
```

Figure for Q.7(a)

If the μ P is running at 10MHz, what should you put as the value of n? What is the actual amount of delay you will achieve with your calculated value of n? You can assume it takes 4 clock cycles to execute a MOV instruction. 17 clock cycles are required to execute a LOOP instruction when it jumps to the target address and 5 clock cycles otherwise.

- Q (b) Why do the number of available 8086 instructions and opcodes differ by a significant margin? (5)

CS

Q8 (a) What is wrong with the following interrupt handling procedure? In what situation will it work despite the mistake?

(5+5=10)

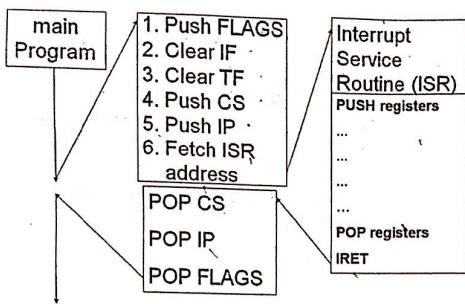


Figure for Q.8(a)

- ✓(b) Explain double indirect jump with an example. Why is it called so? (5)
9. (a) Briefly explain Hyper Threading (HT) technology with appropriate diagram. (5)
(b) Why is the address bus bidirectional in Pentium? (5)
(c) What are the advantages of using physically smaller transistors in Pentium IV? (5)

Contd P/4

CSE 315

SECTION - B

There are **FOUR** questions in this section. Answer any **THREE**.

- Assume system clock frequency 1 MHz if not given.
- List of registers and necessary diagrams are at the end of the question.
- If configuration of any required register is missing, just assume a configuration and clearly show the assumed configuration.
- If any control word/bit configuration is missing in the question paper, just assume a pattern of your choice and clearly mention your assumption.

Q10 (a) Consider the circuit diagram in Figure 10(a). You have to increment an 8 bit counter when switch A is pressed using interrupt. You also have to decrement the counter when switch B is pressed using interrupt. Each interrupt must trigger only a single time as soon as the corresponding switch is pressed. Write two Interrupt Service Routines and the main function in C. (Clearly mention the value assigned to related registers in hexadecimal)

(15)

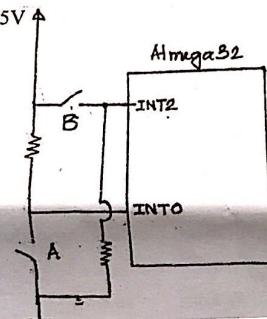


Figure for 10(a)

✓(b) Write C code for the exact scenario of Q. 10(a) using polling instead of interrupt.
(Clearly mention the value assigned to related registers in hexadecimal)

(10)

✓(c) Suppose you have to monitor 16 devices using a CPU. Each device has a 4-bit unique ID assigned to it. If multiple devices get triggered at the same time, you must monitor the device having the lowest ID. Which approach should you take to monitor the devices - polling or interrupt? Give three strong reasons supporting your answer.

(10)

- ✓(1) (a) What steps does the CPU follow to execute an interrupt? (10)
(b) Briefly write one major problem of using delay_ms() of ATmega32. (5)
(c) Convert the following analog inputs to digital. (5x2=10)
✓(d) Reference voltage = 2.56V, $V_{min} = 0V$, 8-bit ADC. Analog input 1.45V
✓(e) Reference voltage = 5V, $V_{min} = 0V$, 10-bit ADC, Analog input 4.17V
✓(f) The digital output of ATmega32 ADC is stored in two 8-bit registers: ADCL and ADCH Explain why ADCL must be read first before ADCH. (10)

Contd P/5

= 5 =

CSE 315

- Q12 Suppose, Harold wanted to send/receive data using UART with his Atmega32. System clock frequency is 10^6 Hz. In his C code he wrote the following functions to initialize UART and send character data.

```
// code fragment of Question 12
void UART_init(void)
{
    UCSRA = 0b00000010;
    UCSRB = 0b00011100;
    UCSRC = 0b10111110;
    UBRRL = 0x31;
    UBRRH = 0x00;
}
void UART_send(unsigned int data)
{
    while((UCSRA & (1 << UDRE)) == 0x00);
    UDR = data;
}
```

He saw that data were not transmitting correctly. He then found an error in his UART _ send function and corrected it later.

Answer the following three questions based on this scenario.

- ✓(a) How many bits will be transmitted/received in 1 minute for this configuration? (10)
- ✓(b) Assume that data is now transmitting correctly. Draw the framing diagram for transmitting data = 100010100? (10)
- ✓(c) What was the error in UART _ send function? Explain briefly. Add at most three lines of code to correct the given function. (15)

transmitting data = 100010100?

(10)

Q6) What was the error in UART _send function? Explain briefly. Add at most three lines of code to correct the given function.

(15)

13. (a) Suppose you want to use SPI for sending data from Atmega32 to four different devices A, B, C and D. You also must occupy minimum number of Atmega32 pins. Draw a block diagram showing how you can achieve this.

(15)

(b) Suppose you wrote a code to measure elapsed time of a code segment. At the beginning of the code segment, you have reset TCNT1. During the execution of the code segment, TIMER1 overflow interrupt was triggered 50 times. After execution of the code segment TCNT1 value was 1200. What was the elapsed time in seconds? (TIMER1 Prescaler is 8)

(c) Eliot wrote the following function EEPROM _ write to write character into a specific address of EEPROM. Unfortunately this function will not work correctly some time. Explain why this will not work correctly. Also insert line(s) to correct the function.

(10)

(10)

```
// code segment for Question 13(c)
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    EEAR = uiAddress;
    EEDR = ucData;
    EECR |= (1<<EEMWE);
    EECR |= (1<<EEWE);
}
```

— 6 —

Figures and Charts for Section B

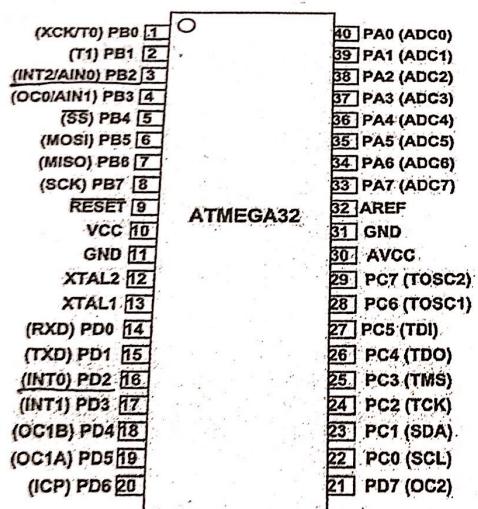


Figure: Atmega32 MCU Pinout

Register Name	Configuration																
GICR	<table border="1"> <tr> <td>INT1</td><td>INT0</td><td>INT2</td><td>-</td><td>-</td><td>-</td><td>-</td><td>IVSEL</td><td>IVCE</td></tr> </table>								INT1	INT0	INT2	-	-	-	-	IVSEL	IVCE
INT1	INT0	INT2	-	-	-	-	IVSEL	IVCE									
MCUCR	<table border="1"> <tr> <td>-</td><td>-</td><td>-</td><td>-</td><td>ISC11</td><td>ISC10</td><td>ISC01</td><td>ISC00</td></tr> </table> <p>Trigger codes: 00 → low level, 01 → any logical change, 10 → falling edge, 11 → rising edge</p>								-	-	-	-	ISC11	ISC10	ISC01	ISC00	
-	-	-	-	ISC11	ISC10	ISC01	ISC00										
MCUCSR	<table border="1"> <tr> <td>JTD</td><td>ISC2</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> </table> <p>ISC2: 0 → falling edge trigger, 1 → rising edge trigger</p>								JTD	ISC2	-	-	-	-	-	-	-
JTD	ISC2	-	-	-	-	-	-	-									
UCSRA	<table border="1"> <tr> <td>RXC</td><td>TXC</td><td>UDRE</td><td>FE</td><td>DOR</td><td>PE</td><td>U2X</td><td>MPCM</td></tr> </table>								RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM										
UCSRB	<table border="1"> <tr> <td>RXCIE</td><td>TXCIE</td><td>UDRIE</td><td>RXEN</td><td>TXEN</td><td>UCSZ2</td><td>RXB8</td><td>TXB8</td></tr> </table>								RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8										
UCSRC	<table border="1"> <tr> <td>URSEL</td><td>UMSEL</td><td>UPM1</td><td>UPM0</td><td>USBS</td><td>UCSZ1</td><td>UCSZ0</td><td>UCPOL</td></tr> </table> <p>UCSZ2, UCSZ1, UCSZ0: 000 → 5-bit, 001 → 6-bit, 010 → 7-bit, 011 → 8-bit, 111 → 9-bit UPM1, UPM0: 00 → No parity, 10 → even parity, 11 → odd parity USBS: 0 → 1 stop bit; 1 → 2 stop bits</p>								URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL										
ECCR	<table border="1"> <tr> <td>-</td><td>-</td><td>-</td><td>-</td><td>EERIE</td><td>EEMWE</td><td>EEWE</td><td>EERE</td></tr> </table>								-	-	-	-	EERIE	EEMWE	EEWE	EERE	
-	-	-	-	EERIE	EEMWE	EEWE	EERE										

Table: Configurations of Atmega32 registers

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-3/T-1 B. Sc. Engineering Examinations 2017-2018

Sub : CSE 315 (Microprocessors and Microcontrollers)

Full Marks : 210

Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION – AThere are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) A system comprising a microprocessor 8086 or 8088, a peripheral device, and a clock generator needs to be designed in such a way that the clock generator feeds both the microprocessor and the peripheral device with 5 MHz clock signals. To ensure it, a hardware designer presents the following design where he applies 15 MHz crystal in between X1 and X2, and another 30 MHz signal to EFI to feed necessary clock signals to both the microprocessor and the peripheral device (not shown in the figure). **(20)**

Now, you need to answer the following (with other necessary diagrams)-

- (i) Does the microprocessor get the intended clock signal? If so, then you need to explain how it is getting the clock signal from corresponding input to the clock generator to the clock input of the microprocessor. If not, then you need to explain why the two clock inputs to the clock generator are not enough or appropriate in supplying the intended clock signal to the microprocessor.
- (ii) Does the peripheral device get the intended clock signal? If so, then you need to explain how it is getting the clock signal from corresponding input to the clock generator to the clock input of the peripheral device. If not, then you need to explain why the two clock inputs to the clock generator are not enough or appropriate in supplying the intended clock signal to the peripheral device.

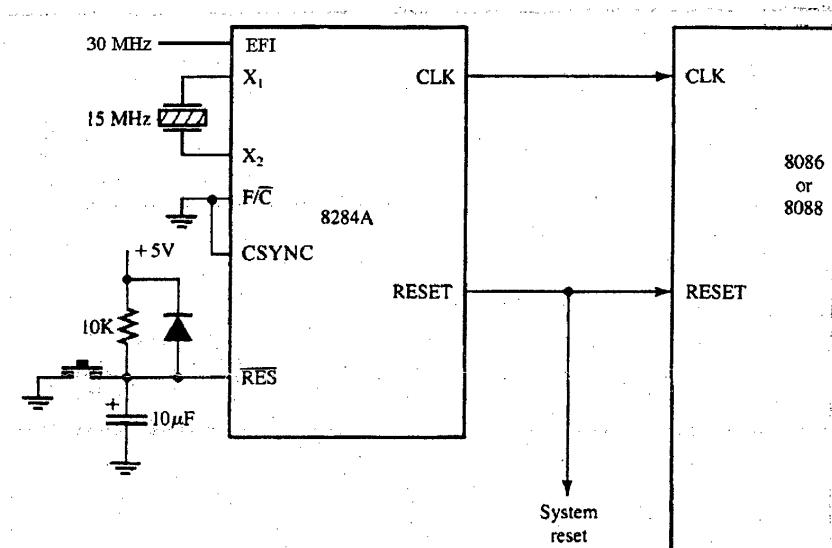


Figure for Question 1(a)

CSE 315

Contd ... Q. No. 1

(b) The following circuit is intended for a buffered system of 8086, where all address, control, and data buses are fully buffered. Do you think that the circuit is enough or appropriate for the purpose of fully buffered system of 8086? If so, then explain how it serves the purpose. If not, then present necessary correction(s) or enhancement(s) needed to be made in the circuit.

(15)

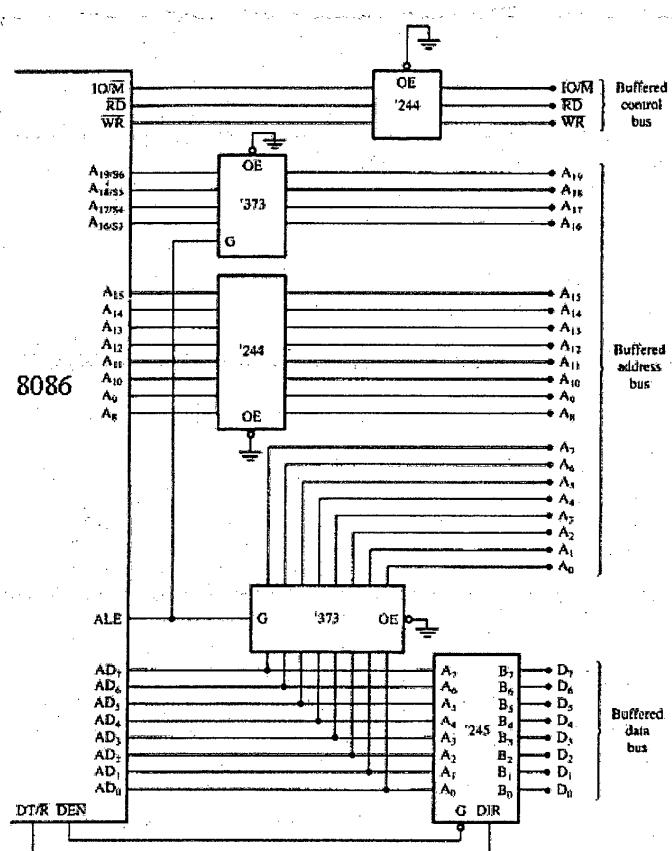


Figure for Question 1(b)

2. (a) "The following circuit interrupts the microcontroller with NMI in case there happens any single-bit, 2-bit or multi-bit error" – validate or invalidate the statement with necessary elaboration and/or figure(s).

(20)

CSE 315

Contd ... Q. No. 2(a)

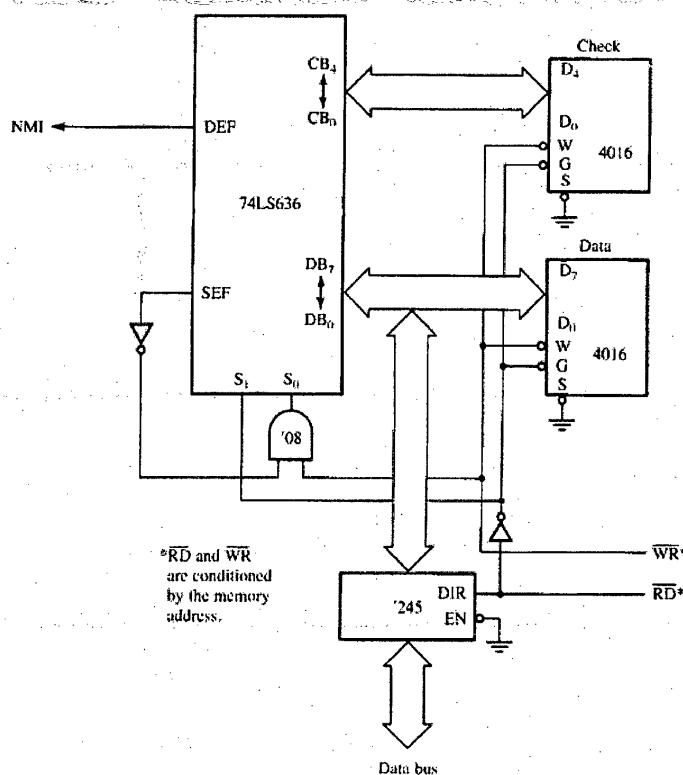


Figure for Question 2(a)

(b) Design separate Bank Write Strobes (for example, WR0, WR1, etc., all in low-enable) for 8086 and 80486. (15)

3. (a) 82C55 can operate in three different modes of operations. Timing diagram of one of the modes is given below: (5+15=20)

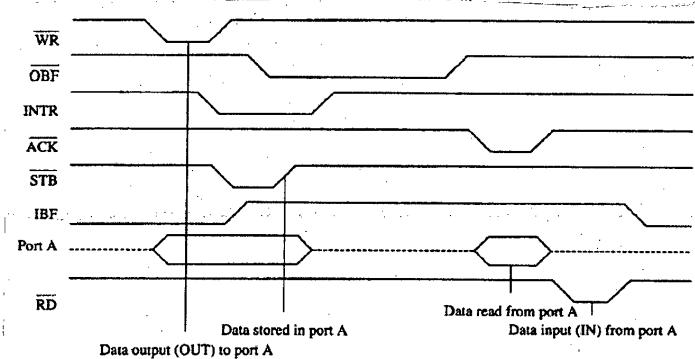


Figure for Question 3(a)

Now, you need to answer the following:

- Which mode of operation is related to the above timing diagram? What can be its possible application?

CSE 315

Contd ... Q. No. 3(a)

(ii) What are the input and output signals (to 82C55) in the figure? How do the input signals control the output signal? You need to show it through pointing an arrow from a change in input signal to corresponding change in output signal for all cases.

(b) The following DRAM circuit needs to be refreshed one row at a time. The whole DRAM needs to be refreshed in 4 ms. In case of connecting it with a microprocessor having a clock rate of 5 MHz with 800 ns for a read or write operation, what will be the percentage (%) of loss in computer time due to the refreshing task? Show necessary calculations as needed.

(15)

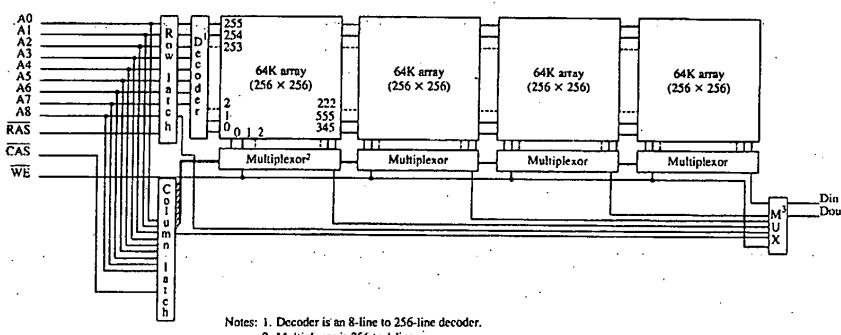


Figure for Question 3(b)

4. (a) How can you do the following to a microprocessor?

(20)

- (i) Apply always a fixed interrupt vector type number (for example FFH),
- (ii) Place a fixed interrupt vector type number (for example 80H) only when it is required and have high-impedance state when it is not required, and
- (iii) Expand the number of applicable interrupt vector type numbers.

(b) How can you generate necessary DMA control signals for 8086 or 8088? Elaborate with necessary figure(s).

(15)

SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE** questions.

Find the Pin Configuration of ATmega32 MCU and its different register configurations at the end of the questions. If configuration for any of the required register is missing, just assume a configuration and clearly show the assumed configuration. If you believe any control word/bit configuration is missing in the question paper, just assume a pattern of your choice and clearly mention your assumption.

5. (a) Write a simple C program for ATmega16/32 to take an 8-bit input from Port A every second and output that to Port B.

(10)

(b) What is wrong with the following push button connection? Describe two different ways to tackle the problem in ATmega16/32.

(5+5+5)

CSE 315

Contd ... Q. No. 5(b)

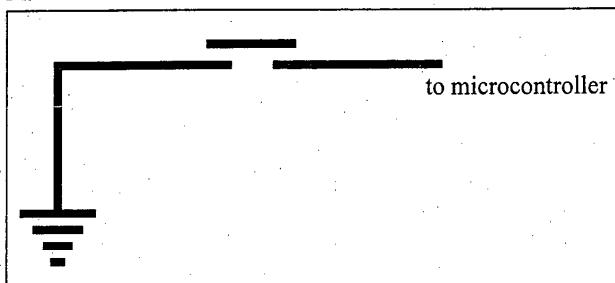


Figure 7: Push button connection for question 5(b)

(c) How can you enable nested interrupts in ATmega16/32? (10)

6. (a) Suppose two **active high** push switches A and B are connected to INT0 and INT2 pin of an ATmega32 MCU, respectively. Also, eight **active high** LEDs (LED0 – LED7) are connected to PA0 – PA7. Write a C code using external interrupt to implement a ring counter, which counts up upon pressing the switch A and counts down upon pressing the switch B. Use 0 for don't care bits. The three external interrupt vector names are INT0_vect, INT1_vect, and INT2_vect. (15)

The codes for external interrupt events of INT0 and INT1 are as following:

Code	Interrupt Triggering Events
00	Low Level
01	Any Logical Change
10	Falling Edge
11	Rising Edge

The codes for external interrupt events of external INT2:

Code	Interrupt Triggering Events
0	Falling Edge
1	Rising Edge

- (b) What is the difference between Phase Correct and Fast PWM modes? Explain with appropriate examples and figures as needed. (10)

- (c) Write a program to write the characters 'a' to 'z' in the first 26 bytes of EEPROM. (10)

7. (a) Consider, you are continuously receiving data from a PC using UART in polling approach. The connection details are: Baud rate: 9600 bps, no parity (code: 0x0), 1 stop bit (code: 0x0), 8 data bits (code: 0x3), and asynchronous communication (code: 0x0). Assume a clock speed of 8MHz. Write a C code that continuously receives a word from the PC, writes that word to PORTA, and sleeps for 1 second and repeats. Clearly specify the initialization of different registers and necessary calculations of their values. (15)

CSE 315

Contd ... Q. No. 7

- (b) Consider you have configured the ATmega32 ADC with a reference voltage of 4V and left justified the result. Show the calculation to find out the step size in volts when: (10)
- (i) You are only reading ADCH
 - (ii) You are reading both ADCL and ADCH
- (c) Describe with example(s) the necessity and application of using “volatile” variables in a C program in the context of ATmega16/32 interrupts. (10)
8. (a) You need to write a C code for an ATmega32/16 based system to control temperature of a medicine storage facility. The facility requires the room temperature to be within 4-10 degree Celsius. If the temperature falls below the range, it should be increased by turning on the heater and when the temperature exceeds the range, the room needs to be cooled down by turning on the air cooler. Both the heater and air cooler stay off while temperature is within the desired range. A temperature sensor is used to measure the temperature of the room.
The output voltage of the temperature sensor is linearly proportional to the temperature and produces an output of 0V to 5.0 V for 0 degree to 20 degree Celsius linearly. Assume that you are using polling mode, reference voltage of 5 V (code: 0x1) and a prescalar of 2 (code:0x1). The sensor is connected to the pin ADC0 (code:0x0). You can turn on the heater and air cooler by writing a logic 1 to PB0 and PB1, respectively. Each time you turn on the heater or the air cooler, wait for 20 seconds before taking a new reading from temperature sensor. (15)
- (b) Determine the values of TOP and OCR1 (The value of OCR1 at which compare match occurs) to create a 1 KHz signal with 40% duty cycle using Fast PWM modes. The system clock frequency is 1 MHz and prescalar is 8. (10)
- (c) Write a C function *unsigned int UART_receive ()* for ATmega16/32 that can receive data (character size 9 bits) sent through UART. If there is a frame or parity error, return -1. (10)

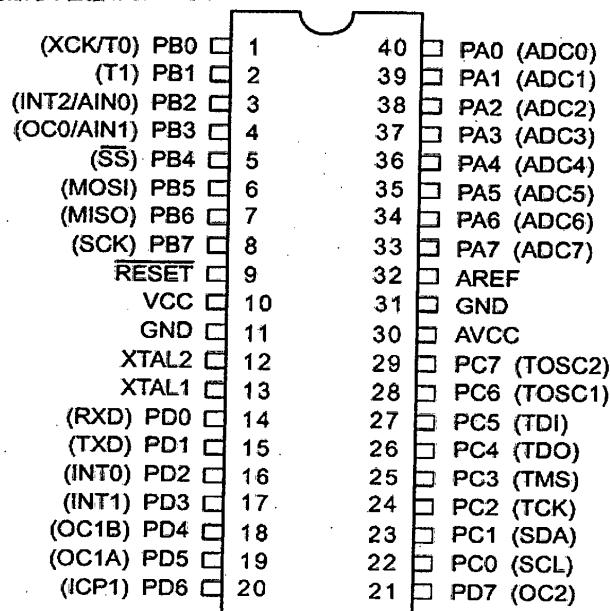


Figure 7: ATmega 32 MCU pinout (for Section B)

Register Name	Configuration							
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
GIFR	INTF1	INTF0	INTF2	-	-	-	-	-
MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR1
SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
ADCCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Table 7: List of registers (for Section B)

SECTION - A

There are **FOUR** questions in this section. Answer any **THREE** questions.

1. (a) What is the mechanism of changing the privilege level in 80386? Explain in details. (17)
 (b) Consider the instruction: "call 0010:12345678". After executing this instruction, the next execution begins at 0020:00004672. All numbers are in hexadecimal. Assuming that no task switching has occurred, can you explain this in detail? (18)

2. (a) What special steps should you take while enabling paging? (12)
 (b) Consider the following linear address: "01405000". Assume that PDBR contains 10000000h. Now for the partial memory map in Figure 2.b, find out the translated physical address. You have to show your calculations for the translation. Also, comment on the content of the page this address belongs to. (14)

Address in Hex	Contents in Hex							

20000018	4	0	0	0	0	X	X	X
20000014	1	0	0	0	0	X	X	X
20000010	3	5	0	0	B	X	X	X
2000000C	3	3	A	0	0	X	X	X
20000008	4	0	0	0	0	X	X	X
20000004	3	0	0	0	0	X	X	X
20000000	4	0	0	1	0	X	X	X
...
10000018	1	0	0	0	0	X	X	X
10000014	2	0	0	0	0	X	X	X
10000010	1	0	0	0	0	X	X	X
1000000C	1	2	0	0	0	X	X	X
10000008	1	0	2	A	0	X	X	X
10000004	0	0	1	A	0	X	X	X
10000000	0	1	0	0	A	X	X	X
...

Figure 2.b: partial memory map for Q2.b

- (c) Give three examples where 80386 references direct physical addresses when paging is enabled. (9)

- (a) Identify whether the following statements are True or False and justify your answer. (20)

CSE 315

Contd... Q. No. 3(a)

Serial	Statement
(i)	There are a total of 4 Privilege levels.
(ii)	You can transfer control only to a code that is equal or lower in privilege than yours
(iii)	Transferring control always triggers selector validation.
(iv)	When you (with PL = 2) have passed through a call gate and the control has just been transferred to a higher PL code (PL = 1), your data segments will initially have DPL < 1.
(v)	Suppose a caller code (PL = 2) used a call gate to run a callee code at PL = 1. When the control will have returned to the caller, the RPL field of the DS register will have 00.
(vi)	RPL of CS is always maintained to be the CPL, hence, RPL will always match the DPL of the current code segment.
(vii)	The only way to change the CPL is to use Call Gate.
(viii)	When you have a call gate, you can transfer the control anywhere in the segment pointed by the destination selector of the call gate.
(ix)	You can never use a FAR JMP to use a call gate.
(x)	If you are at PL2 and there is no call gate with gate DPL = 2, you will never be able to transfer control to a code segment with PL < 2.

(b) Why does TSS have three different sets of stack related registers? Describe a scenario when a stack fault can result in a shut down condition. (15)

4. (a) As a System Software Developer of a world renowned device manufacturing company, you are writing an exception handler for some possible interrupt conditions of a new device. Your handler code contains the following sections:

(9+15)

Section PL: This section contains one or more privileged instruction (s).

Section ST1: This section contains stack operations.

Section H: This section contains the instructions related to handling the interrupt.

Section ST2: This section contains stack operations.

Section IR: This section contains an IRET instruction.

- (i) Can you explain what could be the reason for having Section PL, Section ST1 and Section ST2?

- (ii) Looking at your code, your genius little brother suggests two alternatives as follows: in alternative #1, you do not need Section PL and in alternative #2, you do not need Section S1 and S2, but you need another section with one instruction after Section IR. Explain the two alternatives.

(b) When a page fault occurs, how does the processor know the nature of the fault?

How does it know the offending linear address? (8+3)

CSE 315

SECTION-B

There are **FOUR** questions in this section. Answer any **THREE**.

Students may refer to Table 1 and Figure 1 if necessary.

For the questions related to interfacing LCD display with an ATmega32, assume that there is a header “LCD.h”. To initialize the LCD, you need to call the function “LCD-init()” of that header. To show a string in the LCD, you call the function “LCD-string(str)”. Assume that the ATmega32 is running at a speed of 1 MHz, unless specified otherwise.

5. (a) Describe four modes of parallel data transfer with detailed timing diagrams. Mention the advantages and disadvantages of each one of them. Then, answer with appropriate reasoning: what is the best parallel data transfer mode when you are interfacing your microprocessor with (15)

- (i) Keyboard
- (ii) Dot matrix

- (b) Determine the control word of an 8255A so that (10)

- (i) Port A is configured as input (mode 1), port B is configured as output (mode 0).
- (ii) Port A is configured as output (mode 0), port B is configured as input (mode 0).

Show the detailed steps of obtaining the control word.

(c) Suppose you have 64 bits of information in the character array *char arr[8]*, that defines the contents of an 8x8 LED matrix. Here, arr[0] is the data for the leftmost column, arr[7] is the data for the rightmost column. The LSB of arr[0] is the data for the bottom-left LED of the matrix, whereas MSB of arr[7] is the top-right LED of the matrix. (10)

Now, write necessary code to be written in MDA-8086 processor so that the pattern saved in the character array is shown in the LED matrix of the MDA board. The pattern is shown for two seconds in **green**, and the next two seconds in **red**; and this alternating pattern repeats forever.

6. (a) A PWM wave generation in ATmega32 is using **Phase Correct PWM** with **Non-inverting mode**. Specially, the bits chosen are WGM = 1010 and COM1A = 10. In these settings, TOP = ICR1 and compare match event occurs at OCR1A. (15)

Now, the TOP is not fixed in this scenario. The TOP is updated in runtime as follows:

```
// register set ups
OCR1A = 100;
while (1)
{
    for (i = 0; i < 5; i++)
    {
        ICR1 = 200 + 100 * i;
        _delay_us(400);
    }
}
```

CSE 315

Contd... Q. No. 6(a)

_delay_us() gives delay in microseconds. Here, no prescaler has been used.

Now, draw the generated waveform and determine its frequency and duty cycle.

(b) Write C program to be written in an ATmega32 so that it determines the time period of a uniform square waveform and shows the time period in seconds on LCD. Assume that the LCD is connected to the ATmega according to the requirements of the header "LCD.h".

(10)

Note that ICNC = 1 activates the noise canceller, ICES = 0 selects the falling edge as the input capture event generator, and WGM = 0000 makes the timer operate in normal mode. Clock select "001" makes the timer operate without any prescaling.

(c) Point out the advantages and disadvantages of Harvard architecture over von Newman architecture. Which one is used in ATmega32? Explain how the chosen architecture aids in the overall performance of ATmega32 MCU.

(10)

7. (a) Write code to read all the bytes from EEPROM of an ATmega32, and if the byte stands for a character of capital letter, replace it with the corresponding small letter. For example, say a byte that you have read is 'B'. Then, you need to replace it with 'b'. However, say the byte that you have read is 'd'. Then, keep it unchanged.

(15)

(b) Discuss pipelining in AVR CPUs. What are things considered for obtaining efficiency in this pipelining?

(10)

(c) Point out the advantages and disadvantages of RISC architecture over CISC architecture. Which one is used in ATmega32? Explain how the chosen architecture aids in the overall performance of ATmega32 MCU.

(10)

8. (a) Two sensors S1 and S2 are connected to an ATmega32, namely MCU0. The microcontroller MCU0 is responsible for reading the sensors and collecting sensor data every 2 seconds. However, MCU0 does not have any LCD connected to it. MCU1 and MCU2 (two other ATmega32's) have two LCD displays connected to them, as described in "LCD.h".

(20)

Now, write three programs to be written in MCU0, MCU1 and MCU2 so that MCU0 collects the sensor data and sends the data to MCU1 and MCU2. Readings(s) collected from S1 is sent to MCU1; and reading(s) collected from S2 is sent to MCU2. This data transfer is done in SPI. MCU0 selects which MCU to send the data to, and next sends the data. MCU1 and MCU2, after receiving, show the received data on LCD display. Assume that the LCD displays are connected to MCU1 and MCU2 as described in "LCD.h".

You can use any clock for SPI. You can choose the ADC to be left or right adjusted; either of these two is fine.

(b) Explain with appropriate reasons: which one among serial and parallel data transfer is preferred for long range data communications.

(10)

(c) How does ATmega32 work in sleep modes? Write appropriate code to demonstrate.

(5)

Register Name	Configuration							
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
GIFR	INTF1	INTF0	INTF2	-	-	-	-	-
MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
UCSRC	URSE L	UMSE L	UPM 1	UPM 0	USB S	UCSZ 1	UCSZ 0	UCPO L
SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR1
SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
ADCCSR A	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Table1: Register descriptions

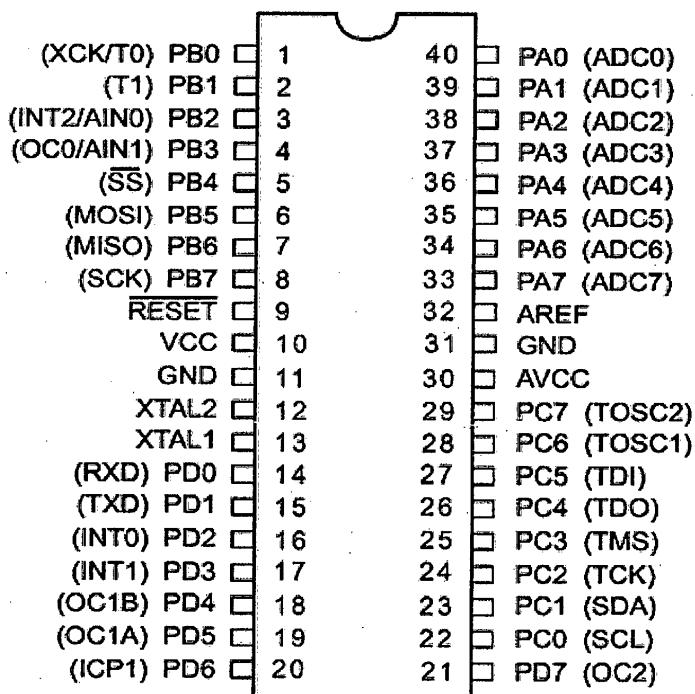


Figure1: ATmega32 pin diagram

SECTION – AThere are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Consider the instruction: "call 0020: 00ABCDEF". Here, a call gate has been used. Deduce the segment and offset of the instruction that will be executed next. Use Table A for GDT. You may be use Table B and Table C to refresh your memory. (15)
- (b) Modify the appropriate entry of the GDT so that the next instruction to be executed is at offset 1234. You can change only one entry in the GDT (Table A); you cannot add any new entry. You need to explain your modification and write down the modified entry. You have to assume that a call gate is being used for the call. Would it be possible to make the offset FE12? Justify your answer. (15)

Table A: GDT

Entry	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	3	1	F	1	1	0	F	F	F	F	0	0	0	0
2	0	0	3	1	F	1	0	1	F	F	F	F	0	0	0	0
3	0	0	3	1	F	1	0	0	F	F	F	F	0	0	0	0
4	0	0	0	0	E	C	0	0	0	0	2	8	4	6	7	2
5	0	2	3	0	F	C	1	F	F	F	F	F	0	0	0	0

Table B: Segment Descriptor

63-56	55	54	53	52	51-48	47	46-45	44	43	42	41	40	39-16	15-0
Base	G	D	L	AV	Lim	P	DPL	S	E	ED/C	R/W	A	Base	Lim

Table C: Call Gate descriptor

63-48	47	46-45	44-40	39-37	36-32	31-16	15-0
Destination Offset 31-16	P	DPL	01100	000	WC	Destination Selector	Destination Offset 15-0

- (c) Your genius little brother looks at the GDT and the above two questions and comments with a crude smile that the call gate was not at all necessary! Do you agree with him? Justify your answer. (5)

CSE 315

2. (a) Describe the logical address to linear address translation in the microprocessor considering both segmentation and paging with appropriate illustrations. (10+15)
- (b) What is identity mapping with respect to paging? What could be the use of such a mapping? (5+5)
3. (a) What is a program-invisible register? Describe the program-invisible registers in advanced microprocessors. (15)
- (b) What is a translation Look-aside Buffer and how is it used? (10)
- (c) Describe the Flat Mode Memory System. (10)
4. (a) Describe the Task State Segment with an appropriate illustration. Why does TSS have three different sets of stack related registers? (10+5)
- (b) Describe the task switch operation with appropriate illustrations. (20)

SECTION-B

There are **FOUR** questions in this section. Answer any **THREE** questions.

5. (a) Suppose, Machine A is trying to send data to Machine B in double handshake mode. However, they are facing a problem. The timing diagram they are following is shown below. (7+8=15)
- (i) What are the problems of their timing diagram?

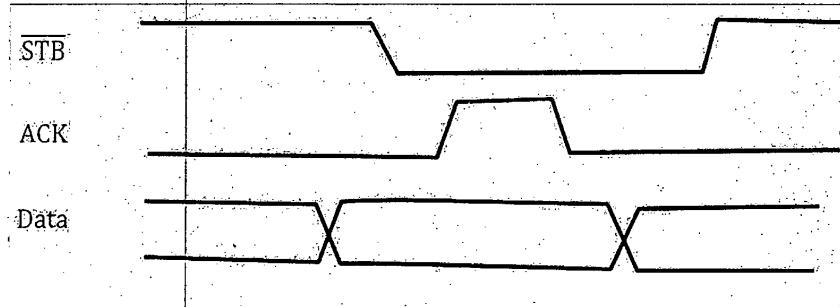


Figure1. (For Question 5(a))

- (ii) Correct the timing diagram. In your corrected timing diagram mark each transition of every signal with a number surrounded by a circle. For each transition you will have to write down in a separate table who initiates the transition (Machine A or B) and what the transitions signifies.
- (b) Implement a C function void UART_send (unsigned char data) which receives a character as an argument and transmits it using UART by **polling on the TXC bit of UCSRA**. (10)
- (c) Suppose you want to use SPI for sending data to three different devices A, B, and C one after another. Briefly explain how you can achieve this with necessary block diagrams. (10)

CSE 315

6. (a) Suppose one active low push switch A and two active high push switches, B and C are connected to INT0, INT1, and INT2 pin of an ATmega32 MCU, respectively. Also, Eight LEDs are connected to PORTB. Write a C code to implement an 8 bit ring counter which counts up when the push switch A is pressed and counts down when B is pressed. Pressing C will reset the counter to 0. The output is shown with the LEDs. Keep in mind that the buttons bounce a lot. Briefly describe how the relevant registers were set and how debouncing was achieved.

The codes for external interrupt events of INT0 and INT1 are as following:

(15)

Code	Interrupt Triggering Events
00	Low Level
01	Any Logical Change
10	Falling Edge
11	Rising Edge

The code for external interrupt events of external INT2:

Code	Interrupt Triggering Events
0	Falling Edge
1	Rising Edge

- (b) Consider you are using ADC with a reference voltage of 5V and you are reading the ADC value only from ADCH. Calculate the maximum precision error you will get when the ADC value is (i) left adjusted and (ii) right adjusted.

(10)

- (c) Consider you are reading a tape recorder with an 8088 microcomputer system with a built-in 8255 PPI (e.g., MTS-88.C). The tape recorder is connected to PORTA and it is read using strobed I/O mode. You also have an output device connected to PORTB which works in simple I/O mode. You will have to design a flowchart to continuously read data from the tape and output it to PORTB. The 8255 is connected with the 8088 microprocessor in the address 010000xxb. Clearly specify the control word.

(10)

7. (a) Consider a buggy C code "Buggy.c" in Fig.3, which attempts to receive a byte every second using UART with the following connection parameters: 1200 bps baud rate, even parity, 1 stop bit, normal speed mode, and 8 data bits. Rewrite the code correcting all the mistakes. Clearly mark the portion of your code added or, modified and specify what was the mistake before.

(15)

Assume the clock speed of the ATmega32 MCU is set at 8MHz.

The code for parity is as follows: 00, 10, and 11 is for no parity, even parity, and odd parity, respectively.

The code for stop bit is as follows: 0 for 1 and 1 for 2 stop bits.

The code for 8 data bits is 011.

Ignore the time needed for polling and also the status of error bits.

CSE 315

Contd... Q. No. 7

- (b) The digital output of ATmega32 is stored in two 8 bit registers: ADCL and ADCH. While reading the 16 bit digital output, does the order of reading from these two registers matter? Explain why or why not. (10)
- (c) Suppose two active high push switches are connected to PA0 and PA4 of an ATmega32 MCU. Also, eight active low LEDs are connected to PORT B. Write a C code to implement an 8 bit ring counter which counts up when the push switch connected to PA0 is pressed and counts down when the other one is pressed. The output is shown with the LEDs. Use polling approach. Keep in mind that the buttons bounce a lot. Also assume the push switches will not be pressed simultaneously. (10)
8. (a) Suppose you are using a particular temperature sensor which produces an output of 0V to 3.3 for 0 degree to 330 degree Celsius linearly. (15)
Write A C code to use ATmega32 ADC in polling mode to read the sensor value and determine the temperature using (i) by reading only ADCH (ii) by reading ADCH and ADCL. Just store the temperature in a variable.
Assume that you are using internal reference voltage of 5V (code: 0x1) and a prescalar of 2 (code: 0x1). The sensor is connected to the pin ADC0 (code: 0x0).
- (b) Draw a timing diagram showing SPI data transfer format for SPI mode 1 (CPOL = 0 and CPHA = 1). Clearly show with respect to clock pulse when the data is sampled and when relevant pins are changed. In SPI mode 1 reading (rising) edge is setup edge and trailing (falling) edge is sample edge. (10)
- (c) Briefly explain how the start bit and data bits are sampled for UART in ATmega32. Your explanation must mention the key difference of sampling process between the normal and double speed modes. (10)

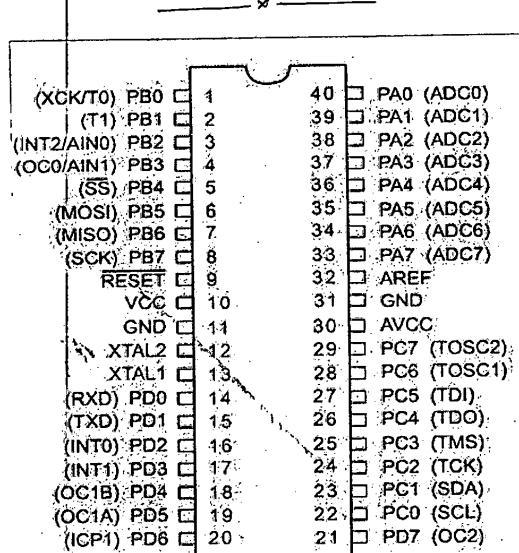


Figure 2. ATmega 32 MCU pinout

P.T.O

```

1 #include <avr/io.h>
2 #include <util/delay.h>
3
4
5     unsigned char UART_receive(void){
6         while ((UCSRA & (1<<RXC)));
7         return UDR;
8     }
9
10
11    int main(void)
12 {
13     DDRB = 0xFF;
14     UCSRA = 0b00000010;
15     UCSRB = 0b00001000;
16     UCSCC = 0b00001100;
17     UBRL = 0x04;
18     UBRRH = 0x16;
19
20     while(1)
21 {
22         unsigned char c = UART_receive();
23         PORTB = c;
24         delay_ms(1000);
25     }
26
27 }

```

Figure 3 (For Question 7(a))

Register Name	Configuration							
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
GIFR	INTF1	INTF0	INTF2	-	-	-	-	-
MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR1
SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
ADCCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Table 1 List of registers

SECTION – AThere are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Explain what determines whether a μ P is a 16-bit or 32-bit device. (8)
 (b) Discuss briefly 80286 protection mechanism. (20)
 (c) How many modes of operation are there in 80286? How can you switch back and forth between them? (7)

2. (a) What are different types of interrupts available in 8086? Briefly describe the interrupts of 8086 pre-defined by INTEL. (5+10)
 (b) List and describe the steps of INTEL microprocessor will take when it response to an external interrupt. (13)
 (c) What addresses in the interrupt-vector table are used for a Type 4 interrupt? (7)

3. (a) Discuss 80286 segment descriptor. (15)
 (b) Discuss virtual mode of operation of 80386. (8)
 (c) Discuss 80386 paging scheme. (12)

4. (a) Discuss how internal architecture of Pentium Pro is different from that of earlier μ Processors. (15)
 (b) Write short notes on (i) Pentium II and (ii) Pentium III. (10+10)

SECTION – BThere are **FOUR** questions in this section. Answer any **THREE**.

Find the pin diagram of ATmega32 microcontroller and its different register configuration at the end of the questions.

5. (a) Suppose in a particular project an ATmega32 microcontroller needs to communicate with four different devices. So you have to connect these four devices with ATmega32. Now answer the following questions: (15)
 - (i) Which serial communication protocol allows you to connect these four devices to the microcontroller using the minimum number of wires?
 - (ii) What are the preconditions for using the serial communication protocol asked in question 5(a)(i)?
 - (iii) Draw a block diagram showing the required connections among the devices if the serial communication protocol asked in question 5(a)(i) is used? You must label those pins of ATmega32 that will be used for the connection.

(b) Suppose, MCAM100 is a programmable pan-tilt video camera. It can be controlled using UART (Universal Asynchronous Serial Receiver & Transmitter). You have to use USART of Atmega32 microcontroller to rotate the camera repeatedly. The required connection parameters are: 8 data bit (code: 011), 1 stop bit (code: 0), no parity bit (code: 00), baud rate 9600 bps and asynchronous mode (code: 0). Sending character 'L' or 'R' will turn the camera left or right respectively. Assume the system clock frequency of ATmega32 is 1 MHz. Now answer the following questions:

(20)

- (i) Calculate the appropriate value for UBRR (USART Baud Rate Register).
- (ii) Write a C function named USART init which initializes ATmega32 USART for the given parameters.
- (iii) Write a C function named USART send which sends a character through ATmega32 USART using polling.
- (iv) Write the main function which continuously does the following things:
 - a. Rotate the camera left for 5 times. Give 500 milliseconds delay between each successive left rotation.
 - b. Rotate the camera right for 5 times. Give 200 milliseconds delay between each successive right rotation.

6. (a) Consider the following C program for Atmega32 microcontroller:

(18)

```
1. #include <avr/io.h>
2. int main(void)
3. {
4.     unsigned char mask = 1<<TOV1;
5.     TCCR1A = 0b00000000; // Timer1 Normal mode
6.     while(1)
7.     {
8.         TCNT1 = 0xFFFF2;
9.         TCCR1B = 0b00000001; // no prescaling,
10.        while(!(TIFR & mask)); // delay
11.        TCCR1B = 0;
12.        TIFR = mask;
13.    }
14. }
```

Answer the following questions considering this program:

- (i) Calculate the amount time delay (in μ sec.) caused by line 10. You have to justify your calculation. Assume that system clock frequency is 8 MHz. Do not consider the clock cycles needed to execute machine instructions.
- (ii) What is the effect of line 11 on Timer1?
- (iii) What is the effect of line 12 on Timer1?

- (b) Consider the following C program which is running on an ATmega32 microcontroller:

(17)

```
1. #include <avr/io.h>
2. #include "avr/interrupt.h"
3. unsigned char d = 31;
4. ISR (TIMER1_COMPA_vect)
5. {
6.     d = ~d;
7.     OCR1A = d;
8. }
9. int main(void)
10. {
11.     DDRD = 1<<5; //Set PD5 as output
12.     OCR1A = d;
13.     TCCR1A = 0b11000001; //inverted mode: Set on match, clear on TOP
14.     TCCR1B = 0b00001001; //Fast PWM, TOP=0xFF, no prescaling
15.     TIMSK = (1<<OCIE1A); //enable output compare A match interrupt
16.     sei();
17.     while (1);
18. }
```

Draw the waveform generated by pin PD5 (OC1A) i.e. pin 19. Represent time in μ sec. using the horizontal axis and represent voltage using the vertical axis. Assume that the time is 0 when Timer1 starts counting. You have to draw the waveform upto 1024 μ sec. System clock frequency = 1 MHz.

Hints: In Fast PWM mode, when we write a value into the OCR1A register, the value will not be loaded into OCR1A until TCNT1 reaches its TOP.

7. (a) Suppose you need to sample an analog signal precisely at sampling frequency 20000 kHz using the on-chip ADC peripheral of ATmega32 microcontroller. Assume system clock frequency = 16 MHz and Timer0 prescaler = 8. Answer the following questions:

(20)

- (i) How can you use Timer0 Compare Match event of ATmega32 to ensure proper ADC conversion timing? Mention the ADC registers that must be properly configured for this purpose?
- (ii) Calculate the appropriate value of OCR0 register.
- (iii) Calculate the optimal value of ADC prescaler.
- (iv) How many ISRs you need to write for this purpose? Draw the flowchart for each ISR.

(b) Consider the following C program for ATmega32 microcontroller:

(15)

```
1. #include <avr/io.h>
2. #include <avr/interrupt.h>
3. #include <avr/sleep.h>
4. #define MAX 60
5. volatile unsigned char sleepNow = 0;
6. unsigned char ovfCount = 0;
7. ISR(TIMER1_OVF_vect)
8. {
9.     if (ovfCount < MAX)
10.    {
11.        ovfCount++;
12.    }
13.    else
14.    {
15.        sleepNow = 1;
16.    }
17. }
18. ISR(INT0_vect)
19. {
20. }

21. int main(void)
22. {
23.     DDRD &= ~ ( 1 << PD2 ); // INT0: input ...
24.     PORTD |= ( 1 << PD2 ); // Enable pullup.
25.     // Level interrupt INT0 (low level)
26.     MCUCR &= ~ ( ( 1 << ISC01 ) | ( 1 << ISC00 ) );
27.     GICR = GICR | (1<<INT0);
28.     TCCR1A = 0b00000000; // normal mode
29.     TCCR1B = 0b00000001; // no prescaler, internal clock
30.     TIMSK = 0b00000100; // Enable Overflow Interrupt
31.     set_sleep_mode(SLEEP_MODE_PWR_DOWN);
32.     sei();
33.     while (1)
34.     {
35.         cli();
36.         if (sleepNow == 1)
37.         {
38.             sleep_enable();
39.             sei();
40.             sleep_cpu();
41.             sleep_disable();
42.             sleepNow = 0;
43.             TCNT1 = 0;
44.         }
45.         sei();
46.     }
47. }
```

In this program, ATmega32 is placed into a sleep mode when a certain condition is met to reduce power consumption. Answer the following questions considering this program:

CSE 315

Contd... Q. No. 7(b)

- (i) When ATmega32 will be placed into sleep mode?
- (ii) How ATmega32 can be waked up from this sleep mode?
- (iii) What is the first thing done by ATmega32 after waking up from sleep?
- (iv) Which clock domains ($\text{clk}_{\text{I/O}}$, clk_{ADC} , clk_{CPU} , $\text{clk}_{\text{FLASH}}$, clk_{ASY}) will be active during this sleep mode?
- (v) What is the function of line 38 and 40? What is the effect of removing line 39 from the program?

8. (a) Consider the following CUDA program:

(17)

```
1. #include <stdio.h>
2. #include "cuda_runtime.h"
3. #include "device_launch_parameters.h"
4. #define ARRAY_SIZE 10
5. float h_in[ARRAY_SIZE];
6. float h_out[ARRAY_SIZE];
7. // the kernel
8. __global__ void vecSquare(float * in , float * out, int n)
9. {
10.     int i = threadIdx.x;
11.     if(i < n)
12.     {
13.         out[i] = in[i] * in[i];
14.     }
15. }
16. int main()
17. {
18.     const int ARRAY_BYTES = ARRAY_SIZE * sizeof(float);
19.     // generate the input array on the host
20.     for(int i=0; i<ARRAY_SIZE; i++)
21.     {
22.         h_in[i] = float(i);
23.     }
24.     // declare GPU memory pointers
25.     float * d_in;
26.     float * d_out;
27.     // allocate GPU memory
28.     cudaMalloc( &d_in, ARRAY_BYTES);
29.     cudaMalloc( &d_out, ARRAY_BYTES);
30.     // transfer the array to the GPU
31.     cudaMemcpy(d_in, h_in, ARRAY_BYTES, cudaMemcpyHostToDevice);
32.     // launch the kernel
33.     vecSquare<<< 1,ARRAY_SIZE >>>(d_in,d_out, ARRAY_SIZE);
34.     // copy back the result array to the CPU
35.     cudaMemcpy(h_out, d_out, ARRAY_BYTES, cudaMemcpyDeviceToHost);
36.     // free GPU memory allocation
37.     cudaFree(d_in);
38.     cudaFree(d_out);
39.     return 0;
40. }
```

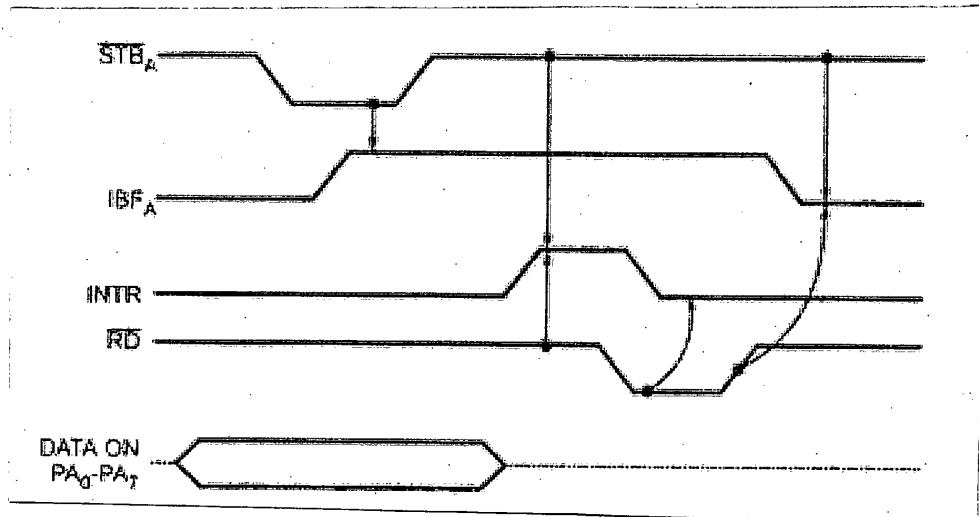
CSE 315

Contd... Q. No. 8(a)

This program performs square of each array element using NVIDIA GPU. Answer the following questions considering this program:

- (i) If you run this program varying the value of ARRAY_SIZE (defined in line 4), you would notice that this program successfully works for ARRAY_SIZE upto 1024. What is the reason for this? Hints: A thread block can contain maximum 1024 threads.
- (ii) Modify this program so that it can work for arbitrary value of ARRAY_SIZE. Rewrite the complete program.
- (b) Suppose, 8086 is reading data bytes from a tape reader through Port A of the 8255 using strobed input mode. The timing diagram for one byte of data transfer is given below:

(18)



Now consider the following list of events:

Event ID	Event description
1.	Data byte from tape recorder is no more valid.
2.	8255 forbids tape recorder to send the next data byte.
3.	8255 signals tape recorder that it is now safe to send the next data byte.
4.	8255 loads the data byte into the input latch of Port A.
5.	8255 informs 8086 about the reception of data using interrupt.
6.	8086 starts reading the data

Redraw the timing diagram and mark the first occurrence of each event with the corresponding event ID surrounded by a circle on the timing diagram.

(XCK/T0)	PB0	1	40	PA0 (ADC0)	
(T1)	PB1	2	39	PA1 (ADC1)	
(INT2/AIN0)	PB2	3	38	PA2 (ADC2)	
(OC0/AIN1)	PB3	4	37	PA3 (ADC3)	
(SS)	PB4	5	36	PA4 (ADC4)	
(MOSI)	PB5	6	35	PA5 (ADC5)	
(MISO)	PB6	7	34	PA6 (ADC6)	
(SCK)	PB7	8	33	PA7 (ADC7)	
	RESET	9	32	AREF	
	VCC	10	31	GND	
	GND	11	30	AVCC	
	XTAL2	12	29	PC7 (TOSC2)	
	XTAL1	13	28	PC6 (TOSC1)	
	(RXD)	PD0	14	27	PC5 (TDI)
	(TXD)	PD1	15	26	PC4 (TDO)
	(INT0)	PD2	16	25	PC3 (TMS)
	(INT1)	PD3	17	24	PC2 (TCK)
	(OC1B)	PD4	18	23	PC1 (SDA)
	(OC1A)	PD5	19	22	PC0 (SCL)
	(ICP1)	PD6	20	21	PD7 (OC2)

Figure 1: ATmega32 pin diagram

Register Name	Configuration							
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
ADCMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
ADCCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
SFIOR	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10

Table 1: List of ATmega32 registers

Figure for Q. No. 5 & 7

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-3/T-2 B. Sc. Engineering Examinations 2013-2014

Sub : **CSE 315** (Microprocessor and Microcontrollers)

Full Marks: 210

Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

SECTION - AThere are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Explain why 8086 is known as a 16-bit processor whereas Pentium is known as a 32-bit processor. (7)
(b) What are the minimum and maximum modes of operation of 8086 processors? How can the modes be selected? (8)
(c) List different types of addressing modes of 8086. Give one example for each. (20)

2. (a) Describe the steps an Intel microprocessor (8086/286/386/Pentium) will take when it responds to an interrupt. (10)
(b) Describe how 8086/286/386 find the address of interrupt service routine (ISR) for an interrupt. (10)
(c) Discuss protected mode memory addressing scheme of 80286 processors. (15)

3. (a) How much physical and virtual memory can an 80386 processor address? Explain how 80386 addresses them. (7)
(b) What differences are there when you compare registers of 80386 with that of 80286 processors? (8)
(c) Discuss a 80386 segment descriptor. (15)
(d) What differences has an 80386 descriptor from that of an 80286 descriptor? (5)

4. (a) Discuss why Pentium processor is known as a superscalar processor. (9)
(b) Discuss how paging memory management of Pentium is different from that of 80386. (10)
(c) Write short notes on any two: (8+8=16)
 (i) Pentium Pro
 (ii) Pentium II
 (iii) Pentium III

CSE 315

SECTION - B

There are **FOUR** questions in this section. Answer any **THREE**.

Find the Pin Configuration of A Tmega32 MCU and its different register configurations at the end of the questions.

5. (a) Draw and briefly explain the timing diagrams for the following methods of parallel data transfer (7+8=15)
- (i) Single Handshake I/O
 - (ii) Double Handshake I/O
- (b) Suppose an 8088 system is connected to a hypothetical input device and a printer via an 8255 PPI. The printer is connected to PORT A and operates in single handshake mode. The hypothetical input device is connected to PORT B and operates in simple I/O mode. Write a pseudo code to continuously read the input device and output it to the printer. Clearly specify the control word. (10)
- (c) Suppose two **active high** push switches are connected to PA0 and PA4 of an ATmega32 MCU. Also, eight **active low** LEDS are connected to PORT B. Write a C code to implement an 8 bit counter which counts up when the push switch connected to PA0 is pressed and counts down when the other one is pressed. The output is shown with the LEDs. Use polling approach. You can assume the push switches will not be pressed simultaneously. (10)
6. (a) What steps does the CPU follow to execute an interrupt? (5)
- (b) Suppose two **active high** push switches A and B are connected to INT0 and INT1 pin of an ATmega 32 MCU, respectively. Also, Eight LEDs, LED0 - LED7 are connected to PB0 - PB7. (10)
- Write a C code using external interrupt to implement the following functionality:
- (i) At any instance, only one LED is turned on and initially it is LED0.
 - (ii) Pressing push switch A will left **rotate** the Led configuration.
That is if currently LED2 is on, pressing A will turn LED3 on.
 - (iii) Pressing push switch B will right **rotate** the LED configuration.
That is if currently LED2 is on, pressing A will turn LED1 on.

The three external interrupt vector names are INT0_vect, INT1_vect, and INT2_vect.

The codes for external interrupt events are as follows:

Code	Interrupt Triggering Events
00	Low Level
01	Any Logical Change
10	Falling Edge
11	Rising Edge

CSE 315

Contd... Q. No. 6

- (c) How many times TIMER1 will overflow to create a delay of 4s, when: **(3+3+4=10)**
- (i) System Clock = 8MHz, no prescalar
 - (ii) System Clock = 4 MHz, Prescalar = 8
- How much time does it take to overflow TIMER1 50 times, with a system clock of 16 MHz and prescalar = 64. Show all the calculations clearly.
- (d) Write down the steps to write a byte in ATmega32 EEPROM. **(10)**
7. (a) Consider, you are continuously receiving data from a PC using UART in polling approach. The connection details are: Baud rate: 4800 bps, no parity, 1 stop bit (code: 0), 8 data bits (code: 011), and asynchronous communication (code: 0). Assume a clock speed of 4 MHz. **(3+3+6+3=15)**
- (i) Calculate the baud rate.
 - (ii) Initialize ATmega32 for the given parameters.
 - (iii) Write a C code that continuously receives a word from the PC, write that word to PORTA, and sleeps for 2 seconds and repeats again.
 - (iv) If double speed transmission mode is used, what changes are needed to be made in steps a to c?
- (b) What is the smallest positive value that can be represented by a 32 bit floating point number in 80×87 Math Coprocessors? Determine both the binary representation and the decimal value. **(10)**
- (c) What are the key characteristics of Harvard Architecture? What are its advantages over Von Neumann Architecture? **(10)**
8. (a)(i) Determine the value of TOP and OCR1 (The value at which compare match occurs) to create a 1 KHz signal with 30% duty cycle using Fast PWM modes. The system clock frequency is 4 MHz and prescalar is 8. **(7)**
- (ii) Find out the highest frequency that can be achieved using Fast PWM mode to generate a signal of 25% duty cycle. Assume, the system clock frequency is 4 MHz and prescalar is 8. What will be the highest frequency when we increase the duty cycle to 50%. **(8)**
- (b) Briefly discuss the in-system flash memory and data SRAM organization of ATmega32 MCU. **(10)**
- (c) The digital output of ATmega32 ADC is stored in two 8 bit registers: ADCL and ADCH. While reading the digital output, which one should be read first and why? **(10)**

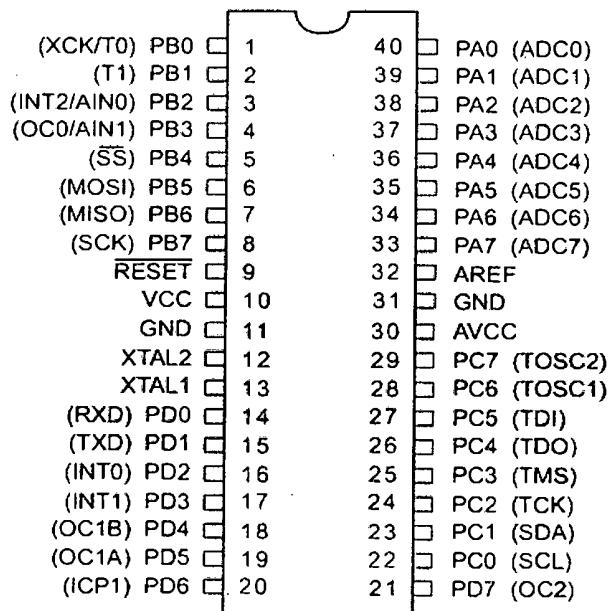


Figure 1. ATmega 32 MCU pinout

Register Name	Configuration								
GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	
MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	
MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF	
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	
TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	
UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	

Table 1 List of registers

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-3/T-2 B. Sc. Engineering Examinations 2012-2013

Sub : **CSE 315** (Microprocessors and Microcontrollers)

Full Marks : 210

Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION - AThere are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) What do you know about **RISC** concept? How RISC-based instruction set is allowed in ATmega16 architecture? **(3+4=7)**
 (b) What is the basic property of Harvard architecture? Even though ATmega16 processor is designed following Harvard Architecture format, ATmega 16 is equipped with 3 main memory sections rather than just instruction and data memories. What is the specific need of this special 3rd type of memory? **(4+4=8)**
 (c) We want to build a simple **Fibonacci counter**. Our counting system is connected with a push button to take pulse input and with 8 LEDs to display current counter value. When the system starts up, our counter will show 0. After successive presses at the push button, the counter will display following sequences,

$$1, 1, 2, 3, 5, 8, 13, \dots$$

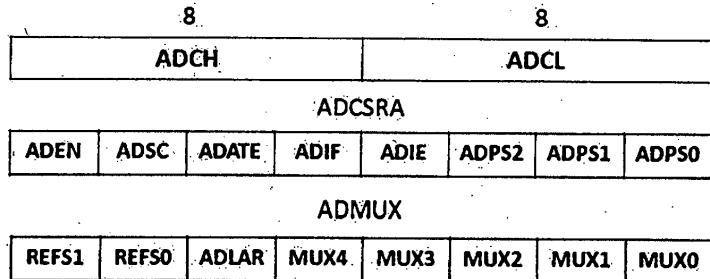
-Since we are using only 8-bits for our counter, when the counter overflows it goes to its initial value, i. e., 0. Using ATmega 16 digital I/O write a **complete C program** to implement this counter system. **(20)**

2. (a) How many timers are there in ATmega32? What are the applications of Pulse Width Modulation? **(1+2=3)**
 (b) How can you use the same hardware to implement both counter and timer? Using this mechanism, describe two ways to generate a time delay. **(5+5=10)**
 (c) Using **Timer0** of a ATmega32, we want to generate an output signal of 32 KHz frequency. Our clock speed is 8 MHz. Find out the value of **TCNT0** if we use overflow method along with no prescaling. **(12)**
 (d) What is the basic difference between polling and interrupt? What are the advantages of interrupts over polling? **(10)**

3. (a) Upon activation of an interrupt, what are the steps that ATmega16 goes through? **(10)**
 (b) What is the difference between the **RET** and **RETI** instructions? Explain why we can use **RETI** instead of **RET**, however, **RET** cannot be used instead of **RETI**. **(5)**
 (c) Write short notes on – Sampling, Quantization, Resolution, and Conversion Time. **(3+3+3+3=12)**
 (d) For a 3-bit ADC, find out the binary output value of V_{out} using **successive approximation** technique. Give that, $V_{in} = 3.22V$, $V_{gnd} = 0V$ and $V_{ref} = 5V$. **(8)**

CSE 315

4. (a) Major relevant registers for ATmega32 ADC subsystem are- (15)



-write a **complete C program** that takes analog input in **ADC0** pin, uses external V_{ref} in **AREF** pin and shows the digital output using **PORTA** and **PORTB**. You have to use the **interrupt** approach for the ADC conversion.

(b) What is the difference between serial and parallel data communication? Also note the advantages and disadvantages of them. (4+6=10)

(c) What is the main problem that we face while interfacing RS232 standard devices with ATmega16 microcontrollers? How can we solve this problem? (3+2=5)

(d) If we want to transfer data at a baud rate of **4800**, what values should be loaded in **UBRRH** and **UBRRL** register? (5)

SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

5. (a) Draw the functional unit block diagram for the internal architecture of 80386 and briefly describe them. (20)
- (b) What are the basic differences between real mode and virtual 86 modes of 80386. (5)
- (c) Explain how the linear address 00200000H will be converted to physical address by the paging unit of 80386. (10)
6. (a) Discuss memory management scheme of 80286. (20)
- (b) Describe a segment descriptor of 80286. (15)
7. (a) How 8255 will be configure if FFH is written to the control register of 8255. (8)
- (b) Show the connections between 8255 and 8086 such that 8255 will have an address of AAH. Show the connection of 8255's following signals only: cs , address bus and data bus. (10)
- (c) What address in the interrupt-vector table are used for a type 2 interrupt? (5)
- (d) List and describe the steps an 8086 will take when it responds to an interrupt. (12)

CSE 315

8. (a) Explain what the following signals do in 8086 : (i) \overline{BHE} , (ii) \overline{TEST} , ALE and (iv)
MN/ \overline{MX} . $(4 \times 2.5 = 10)$
- (b) List different types of addressing modes of 8086. Give one example for each type. (20)
- (c) If the stack segment register of an 8086 contains 1234H and stack pointer register
contains 4321H, what is the physical address of the top of stack? (5)
-

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

SECTION – AThere are **FOUR** questions in this section. Answer any **THREE**.

1. (a) Explain what determines whether a microprocessor is a 16-bit or 32-bit device. (5)
 (b) Draw the internal architecture of 8086 and describe all the registers of 8086. (5+15=20)
 (c) What are the minimum and maximum modes of operation of a 8086 microprocessor? How can they be selected? (5)
 (d) Describe the advantage of queue in 8086 architecture. (5)

2. (a) How much physical and virtual memory can 80286 address? Explain how 80286 addresses them. (5)
 (b) Discuss the protected mode memory addressing scheme of 80286. (15)
 (c) Discuss 80286 protection mechanism. (15)

3. (a) Discuss 80386 descriptors. What are the differences they have from 80286 descriptors? (15)
 (b) Discuss 80386 virtual mode of operation. (8)
 (c) Discuss 80386 paging unit and paging scheme. (12)

4. (a) Narrate how 8086 finds the interrupt service routine (ISR) address for an interrupt. (10)
 (b) What are the different types of interrupts available in 8086? Briefly describe the interrupts of 8086 pre-defined by INTEL. (13)
 (c) Compare the features of 80386 with those of 80286 from the point of view of internal registers. (12)

SECTION – BThere are **FOUR** questions in this section. Answer any **THREE**.

5. (a) Suppose you are to design an automated fire control system for a highly sophisticated room. To detect the fire, a pair of sensors (a smoke(S) sensor and a temperature (T) sensor) are used. The sensors work in this way: the S sensor provides 0 volt if there is not smoke in the room when it detects smoke, it jumps to 5 volt. The T sensor provides analog voltage in the range of 0 to 5 volt in proportional to the temperature between 0°C to 100°C. To be sure that the smoke is indeed caused by the fire you have to check the temperature of the room. So, the system continuously listens

CSE 315

Contd... Q. No. 5(a)

to the S sensor and collects the temperature from the T sensor only after detection of smoke in the room. If the temperature of the room is found to be greater than 60°C after detection of smoke, the system will initiate an alarm to buzz. There will be a switch to stop the alarm. If the alarm is not switched off within one minute of the initiation of buzz, the system will send a start signal to the automated fire fighting system, which is connected to the fire control system.

(8)

Now, draw the block diagram with appropriate connections of the system along with pin names. You cannot use polling approach to collect data from any of the sensors.

(b) Write necessary steps to program for an external interrupt using ATmega 16.

Configure necessary registers to handle INT0.

(10)

(c) Write short notes on:

(10)

(i) Interrupt Vector

(ii) Interrupt Service Routine

(d) Briefly explain ADC features of ATmega 16.

(7)

6. (a) Draw the internal block diagram of 8087 math-coprocessor.

(12)

(b) Draw the timing diagram for the nonpipelined read cycle of the Pentium processor.

(8)

(c) Explain the configuration of 8255 if hex 75 is sent to 8255 as a control word.

(5)

(d) Show the circuit connections between 8255 and 8086 microprocessor, where port addresses of 8255 are: 0580H (PORT-A), 0582H (PORT-B), 0588H (PORT-C), and 058AH (Control Register). Here 8086's address bus is 20 bit and data bus is 16 bit. You need to show the connection of 8255's following pins in your diagram: CS, RD, WR, RESET, A1, A0, and Data bus.

(10)

7. (a) What is Paging? What is the difference between paging and Segmentation?

(3+4=7)

(b) Explain different types of segmentation of Pentium processor with appropriate figure.

(12)

(c) Write short notes on:

(12)

(i) Inquire Cycle

(ii) Virtual 8086 Mode

(iii) NMI

(d) Write four applications where micro-controllers are used.

(4)

8. (a) Explain Real mode and Protected mode architectures of a Pentium processor.

(8)

(b) Explain the transition of a logical address into a physical address in protected mode of Pentium while paging is enabled.

(20)

(c) Configure the value of necessary registers for USART communication using ATmega 16 for baud rate of 4800 bps. Assume system clock is configured as 1 MHz.

(7)

SECTION - A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Suppose you are to design an automated fire control system for a highly sophisticated room. To detect the fire, a pair of sensors (a smoke (S) and a temperature (T) sensor) is used. The sensors work in this way: the S sensor provides 0 volt if there is no smoke in the room. When it detects smoke, it jumps to 5 volt. The T sensor provides analog voltage in the range of 0 to 5 volt proportional to the temperature between 0 C to 100 C. To be sure that the smoke is indeed caused by the fire you have to check the temperature of the room. So, the system continuously listens to the S sensor and collects the temperature from T sensor only after detection of smoke in the room. If the temperature of the room is found greater than 60 C after detection of smoke, the system will initiate an alarm to buzz. There will be a switch to stop the alarm. If the alarm is not switched off within one minute of the initiation of buzz, the system will send a start signal to the automated fire fighting system which is connected to the fire control system. (8+8=16)

Now,

- (i) Draw the block diagram with appropriate connections of the system along with pin names. You cannot use polling approach to collect data from any of the sensors.
- (ii) Show the working procedure of your designed system using a flow chart.
- (b) Explain with figure the Program memory map and Data memory map of ATmega micro-controller. (7)
- (c) Explain Harvard architecture and von-Neumann architecture. What architecture is followed by the ATmega micro-controller? (7)
- (d) Write five applications where micro-controllers are used. (5)

2. (a) Explain how an ATmega CPU can read or write the 16-bit Timer/Counter (TCNT1) register in a single clock cycle using its 8-bit data bus. Write the necessary assembly code fragment for the following two atomic operations: (5+4+4=13)

- (i) Write the value 0x3F55 to TCNT1
- (ii) Read the value of TCNT1 to the registers r15 & r16, where r16 contains the high byte.

- (b) Draw the block diagram of the input capture unit of the Timer1 of an ATmega micro-controller and explain its operation assuming input is coming from an external source. (12)

- (c) Explain the working principle of the three major applications of Timer1 of an ATmega micro-controller. Specify the interrupts that are needed to use for each of those applications. (10)

CSE 315

3. (a) Suppose you want to produce a custom waveform having duty cycle = 0.5 and period = 1000 μ s. Show with figure how you can generate the custom waveform using each of the CTC, Fast PWM, and Phase Correct PWM operational modes of Timer1 of an ATmega micro-controller. Specify the necessary values of the related registers for each mode. Assume that the micro-controller is operating at 1MHz clock rate. (4+4+4=12)
- (b) Draw the block diagram of the memory and central processing unit of an ATmega Micro-controller. How does it ensure instructions to be executed in every clock cycle? (16)
- (c) What precautions must be taken when using Data Register empty Interrupt (UDRI) and Receive Complete Interrupt (RXCI) in interrupt-driven data transmission for serial communication using an ATmega micro-controller? (7)
- (d) Why ADCL must be read before ADCH when ADC of an ATmega 16 generates 10-bit result. (6)
4. (a) Suppose you are designing a micro-controller based automated system that takes input from an analog device and sends it to a computer via USART after the necessary conversion is done. The analog device generates analog voltage precisely in every 200 μ s and the analog voltage persists for a very brief period of time. Assume that the micro-controller is operating at 1MHz clock rate. You cannot use polling approach in any step of your automated system. Now answer the following: (3+3+3+3+6=18)
- (i) Which mode of the ADC is appropriate in this case and why?
 - (ii) How can you ensure that ADC samples analog voltage precisely in every 200 μ s?
 - (iii) What will be triggering event to start the A-to-D conversions?
 - (iv) How many ISRs you need to write and what are the purposes of those ISRs?
 - (v) Write the necessary steps to configure the micro-controller for the automated system. You need to mention the name of the necessary registers and action on the register.
- (b) Describe with figure the Start Bit Sampling process used by the clock recovery logic of an ATmega micro-controller to synchronize its internal clock to the incoming serial frames for both normal mode and double speed mode. (11)
- (c) What are the values of the necessary registers to configure the baud rate of 9600 bps in Asynchronous Double Speed Mode for serial communication using an ATmega 16? Assume system clock is configured as 2MHz. (6)

CSE 315

SECTION - B

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) Discuss with appropriate figure how memory addressing is done. Assume that the current segment selector is pointing to GDT. **(20)**
(b) Suppose DS has the value 000B (in Hex). The following table describes the GDT: **(15)**

Entry	Descriptor Value (in Hex)
0	—
1	0031E110FFFF0000
2	0031E101FFFF0000
3	0031E100FFFF0000

The base address stored in GDTR is 0000FFFF.

Now answer the following questions:

- (i) What should be the value of the limit field of GDTR?
(ii) Suppose you are using DS as the segment selector (with the value given above) to access a memory location for reading and writing data. Will the instruction execute successfully? Justify your answer.
(iii) Suppose you want to read some data from the GDT. Using the above configuration, would it be possible? Explain.
6. (a) Describe with appropriate figure how calling is done through a call gate. **(20)**
(b) Describe the rule that determines who can use a call gate. **(5)**
(c) What problems may arise with regards to the stack when a call gate is used and how this problem is tackled? **(10)**
7. (a) Describe the linear-to-physical address translation procedure with appropriate figure. **(18)**
(b) How can you effectively "turn off" segmentation? **(5)**
(c) Describe how you can produce contiguous linear addresses through page translation. **(12)**
8. (a) Describe the task switch operation with appropriate figure. **(25)**
(b) Consider the effect of a task switch on the TSS. Present in a tabular format the status of NT bit and Busy bit in the Old TSS and New TSS respectively for JMP, Call Interrupt/Exception and IRET instructions. **(10)**