STRUCTURED QUESTIONS (70 MARKS)

(INSTRUCTION: Please answer all 7 questions in the answer booklet provided.)

QUESTION 1 [10 Marks]

(a) Categorise the design elements labelled as (i) – (vi) into either computer architecture or computer organization in the Table 1.

(i) ISA format
(ii) Parallel/serial data transfer
(iii) Addressing Mode
(iv) Cache memory type
(v) Data types
(vi) Read memory cycle

[3 *Marks*]

Table 1: Element Design (issue)

Computer Architecture	
Computer Organisation	

(b) List four (4) structural components of a computer and four (4) structural components of a CPU in Table 2. [4 *Marks*]

Table 2: Structural components of Computer and CPU

Structural components of a Computer	(i) (ii) (iii) (iv)
Structural components of a CPU	(i) (ii) (iii) (iv)

(c) List three (3) characteristics of Von Neumann model that was used as basis of developing a computer from first generation until recent technology in Table 3. [3 *Marks*]

Table 3: Characteristics of Von Neumann

(i)
(ii)
(iii)

QUESTION 2 [5 Marks]

(a) Assume numbers are represented in 8-bits representation. Show the addition of the following in 1's complement: [4 *Marks*]

$$\mathbf{-6_{10} + 13_{10}}$$

(b) Represent the following 1's complement of 8-bit representation values in decimal: [1 *Mark*]

10100100

QUESTION 3 [10 *Marks*]

(a) Consider the 1st version of Division Hardware Algorithm as shown in Figure 1 is used to perform a binary multiplication for **`01101`** / **`00101`**.

Steps:
1 – Remainder: (R) = R – D
2 – Test new R
2a: If >= 0 then shift left Q (add 1 at LSB)
2b: If < 0 then R = D + R, shift left Q (add 0 at LSB)
3 – Shift D right
All bits done?
If still < (max bit + 1), repeat
If = (max bit + 1), stop

Figure 1: 1st version of Division Hardware Algorithm

i) Perform and complete binary division after iteration 3 (Consider steps 1 and 2 already completed, step 3 is given) using the provided table in Table 4 on page 4. Create more rows accordingly to the numbers of iterations.

[7 *Marks*]

ii) In the Table 4, circle the division results for quotient and remainder. Then, specify both in decimal.

[3 *Marks*]

Table 4: For Question 3(a) – Division

Iteration	Steps	Quotient (Q)	Divisor (D)	Remainder (R)
0	Initial value	00000	00101 00000	00000 01101
1 & 2	(Consider completed)			
3	1: R = R – D			11111 00101
	2: R = D + R, Shift left Q Add 0 LSB	00000		00000 01101
	3: Shift right D		00000 10100	

QUESTION 4 [10 Marks]

(a) Represent decimal number (–720) in IEEE754 single precision floating-point format by answering the following questions. Show all your working.

i) Convert the number to binary. [1 *Mark*]

ii) Express the answer (i) in normalized form. [2 *Marks*]

iii) Express the bit pattern in this floating-point format structure. Label the sign, biased exponent, and mantissa/significand/fraction bit clearly. [2 Marks]

(b) Show how the following floating-point calculations are performed (where significands are truncated to 6 decimal digits). Show the results in normalized form. [5 Marks]

$$3.314 \times 10^{1} + 822.7 \times 10^{-1}$$

QUESTION 5 [10 Marks]

Consider an x86 assembly program illustrated in Figure 2.

```
   include Irvine32.inc

   Str1 equ <'UTM',0>

   .data
   var1 WORD 2 DUP(0FFh),1000h,2000h
   var2 BYTE 0FFh
   msg  BYTE str1
   var3 WORD 4000h

   .code
   main PROC
   MOV  AX, var1
   MOV  AH, var2
   MOV  BX, var1 + 4
   ADD  BX, var1 + 6
   MOV  var3, BX
   exit
   main ENDP
END main
```

Figure 2: Assembly program

Answer the following questions based on the program.

(a) Determine the final content of the destination register in each of the instructions Table 5 after its execution. [4 *Marks*]

Table 5: Final content of the destination register

Instruction		Register
(i)	`MOV AX, var1`	AX = _____h
(ii)	`MOV AH, var2`	AX = _____h
(iii)	`MOV BX, var1+4`	BX = _____h
(iv)	`ADD BX, var1+6`	BX = _____h

(b) Table 6 illustrates the memory spaces horizontally starting with the memory address offset for variable var1 is `4000h`. Complete the Table 6 by filling up the memory contents of all variables with the correct byte ordering. Note that ASCII code for 'A' is `41h` and so on. Let the space empty for unused memory. [6 *Marks*]

Table 6: Memory spaces

Offset Address	Content (Hex)							
`4000h`								

QUESTION 6 [15 Marks]

Based on the equation **Y**, answer the following questions:

Y = 7 + (1 + 3) * (6 - 4)

(a) Write the Infix and construct the Expression Tree for **Y.** [3 *Marks*]

(b) Write the Postfix and evaluate the value in Reverse Polish Notation (RPN) for **Y.** [2 *Marks*]

(c) Convert the RPN expression from (b) into assembly codes. The instructions and registers that can be used are: PUSH, POP, IMUL (Integer Multiply), ADD, SUB, EAX and EBX. [10 *Marks*]

QUESTION 7 [10 Marks]

Based on Figure 3, given four arrays with multiple initializers in the x86 assembly program. Assuming current pointer to the base address of the program's data segment is 0x00004000h and the address store into EAX register.

```
   include Irvine32.inc

   .data
   count  DWORD 10h
   array1 BYTE  10h, 11h, 12h, 13h
   array2 WORD  123h, 234h, 345h, 456h
   array3 DWORD 123456h, 23456789h

   .code
   main PROC
   add BL,  [00004005h]          ; (i)
   mov ESI, offset array1        ; (ii)
   mov BX, word ptr[ESI +3]      ; (iii)
   mov ECX, 0
   mov DX, array2[ECX]           ; (iv)
   mov ECX, [EAX + 10h]          ; (v)

   call dumpregs
   exit
main ENDP
```

Figure 3: Assembly program

Write the output for each lines labelled as (i) to (v) in Table 7.

Table 7: Output of register for each lines

Labelled	Register	Output
(i)	BL	
(ii)	ESI	
(iii)	BX	
(iv)	DX	
(v)	ECX	

------- End of Question -------

GOOD LUCK !