AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 1: Numbering Systems

1. Convert the following to decimal numbers.

(a) 111001112

(b) 55778

(c) ABCD H

2. Convert the followings to Base-2, Base-8 and Base-16 respectively.

(a) 3064 D

(b) 201610

(c) 2899 base-10

3. Convert the following numbers to the respective base.

(a) 1111 1010 0011 11002 to base-8 and base-16 respectively

(b) 257ACH to base-2 and base-8 respectively

(c) 7548 to base-16

(d) 44716 to base-8

4. Perform the following conversions. You are required to show the working steps clearly. If the operation(s) is illogical, explain the reason.

(a) 165810 to hexadecimal number (2 marks) (b) 7658 to hexadecimal number (2 marks) (c) 673247 to decimal number (2 marks)

5. Perform the following conversions. You are required to show the working steps clearly. If the operation(s) is illogical, explain the reason.

(a) 123010 to base-6 number (2 marks) (b) 152810 to hexadecimal number (2 marks) (c) 23010 to base-7 number (2 marks)

1

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

6. Perform the following operations and show the answers in the respective number base. You are required to show your working steps clearly

(a) ABC16 + FFF16 (3 marks) (b) 1258 – 778 (2 marks) (c) 110112 x 10112

(d) 100111002 + 010010112

7. Perform the following operations and show the answers in the respective number base. You are required to show your working steps clearly

(a) 4618 + 5158

(b) 173H x AAH

(c) 306416 x 2138 (Show your final answer in Hex format.) (3 marks)

2

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 2: Numerical Data Representation

Section A: Signed Number

1. Under what circumstances the Two’s Complement is used?

2. Convert the 8-bit binary number 11010111 into decimal number if the binary number is a(n): (a) Unsigned number (1 mark) (b) Signed number (2 marks)

3. Assuming that an 8-bit system is being applied, perform the binary subtraction operation for the following decimal numbers using Two’s Complement method.

65 - 54

Verify your answer by showing the answer in signed decimal value. (5 marks)

4. Assuming an 8-bit system is used (i.e. the system uses 8 bits to represent an integer). Given the following decimal numbers:

-12 + -8

(a) Solve the above operation using two’s complement method. (5 marks) (b) Verify your answer by showing the answer in signed decimal value. (3 marks) (c) Justify the validity of the answer obtained. (1 mark) (d) Does overflow occur? Justify your answer. (1 mark)

Section B: Floating Point Number

1. Perform the following number conversions. Show your conversion steps clearly. If the operation is illogical, explain the reason.

(a) 30.3010 to Binary (2 marks) (b) 123.1235 to Decimal (2 marks) (c) 100100011111.112 to Octal (2 marks)

2. Perform the following operations. Show your working steps clearly. If the operation is illogical, explain the reason.

(a) Convert 6A.9610 to hexadecimal number

(b) Convert 1807.6510 into a hexadecimal number (2 marks)

3

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

(c) Convert 101011.01112 into a decimal number (2 marks)

3. Given that:

 An Excess-52 notation is applied.

 The implied decimal point is at the beginning of the mantissa.

 A “5” is used to represent a positive number and a “9” is used to represent a negative number. (a) Convert -357.24610 to the SEEMMMMM format. (2 marks) (b) Convert 55220311 to scientific notation.

(c) Convert 95575321 to scientific notation.

4. The following decimal numbers are stored in excess-50 floating point format. A “1”is used to represent a negative sign, and a “5” for positive sign.

(a) Add these two numbers. Present your result in standard decimal sign-and-magnitude notation. 55020311

15375321 (5 marks)

(b) Multiply these two numbers. Present your result in standard decimal sign-and-magnitude notation.

15176323

15485496 (5 marks)

(c) Find the difference of these two numbers. Present your result in standard decimal sign-and magnitude notation.

55152295

15256608 (5 marks)

5. Represent the binary number -10111.012 into IEEE754 single precision format. You are required to show your conversion steps clearly. (5 marks)

6. Given a decimal number “-30.8125”, how this notation can be represented in the IEEE754 single precision notation. You are required to show your working steps. (5 marks)

7. Given an IEEE754 single precision notation below, show how this notation can be represented in a sign-magnitude notation. You are required to show your working steps.

1 1000 0010 0100 1000 0000 0000 0000 000

Assuming that excess-127 is applied. (5 marks)

4

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 3: Addressing Data in Memory and Segment

1. Explain each of the following terms:

(a) Segment

(b) Offset

(c) Register

2. Explain the purpose of the following items:

(a) Code segment (1 mark) (b) Code segment (CS) register (1 mark) (c) Instruction pointer (IP) register (1 mark) (d) Accumulator (AX)

(e) Count register (CX)

(f) Parity flag (PF)

3. (a) List and explain FOUR (4) main segment in a CPU. (4 marks)

(b) Explain a similarity and THREE (3) differences between a register and a Random Access Memory (RAM). (5 marks)

(c) Differentiate between a segment:offset address and an absolute address.

4. Perform the following calculations and show your answers in hexadecimal format. Draw a diagram for each of them to indicate how the respective answer would store in the AX register. You are required to show your working steps clearly.

(a) 738 + 258 (4 marks) (b) 11112 x 1112 x 112 (4 marks)

5. Given the information in the table below.

|  |  |
| --- | --- |
| Code segment (CS) register: 02B3H Data segment (DS) register: 26D2H Stack segment (SS) register: 09AFH | Base Pointer (BP) register: 2062H  Stack Pointer (SP) register: 0094H  Instruction Pointer (IP) register: 0025H |

(a) Calculate the absolute address for the next instruction to be executed by the CPU.

(b) Calculate the corresponding 20-bit absolute memory address using the SS:SP. (4 marks)

5

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

6. Given a 16-bit CS register consists of the hexadecimal value 12AB and the 16-bit IP register consists of the hexadecimal value 0020. Find the absolute address of the instruction and show the answer in 20-bit binary format. (4 marks)

7. Complete the following table.

Definition

Register

Characteristics

Types of register General purpose Address Status Aim

Size

|  |  |  |  |
| --- | --- | --- | --- |
| Examples |  |  |  |

8. Determine which register is/are used for the following purposes:-

(a) When the result of an arithmetic or logical operation generates a result of zero. (b) Used for error checking when there is a possibility that data might be altered or corrupted. (c) When the result of an unsigned arithmetic operation is too large to fit into the destination. (d) Used for arithmetic and data movement.

(e) Used for counting loops.

9. Write down the values of the Carry, Sign, Zero and Overflow flags after each instruction has executed. Show your working steps clearly.

MOV AX, 6120H

ADD AL, AAH

ADD AH, FFH

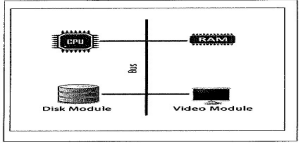
ADD AX, 2

6

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 4: Computer Architecture

1. Given the Figure 1 below.

Figure 1. (a) Define Bus in the computer context. (3 marks)

Bus is a physical connection to transfer data inside computer systems.

(b) Name the bus configuration as shown in the Figure 1 above. (2 marks)

Multipoint bus

(c) List and explain THREE (3) possible types of bus line that can be found in a multipoint bus configuration. (6 marks)

* Data bus: Transports data between the memory CPU
* Address bus: Specifies data receipient, and identifies source and destination of data on data bus.
* Control bus: Provide control for synehronization and control of bus and modules connected.

2. Briefly describe TWO (2) differences of point-to-point bus and multipoint bus. Draw a diagram to illustrate each of them. (6 marks)

Point: Bus carry from specific to specific.

Multipoint :Bus connect servel devices together.

3. Memory registers play an important role in the process of Central Processing Unit (CPU) – Random Access Memory (RAM) communication.”

Explain how memory Address Register (MAR) and Memory Data Register (MDR) involved in the CPU-RAM communication. (5 marks)

When attempting to retrieve data. MAR holds the address of the data to be used by CPU.

MDR will retrieve and store a copy of data pointed by MAR inside the RAM for read access.

MDR is responsible for storing data back into the RAM if write operation is carried out by the CPU.

4. In the Little Man Computer (LMC) model, Instruction Register (IR), Program Counter (PC), Memory Address Register (MAR), Memory Data Register (MDR) and Accumulator (A) were used.

(a) Describe the purpose for IR, PC and A respectively. (6 marks)

IR stores the current instruction being executed by the CPU.

PC stores the address of the next instruction to be executed by the CPU.

A register used for holding data for processing by the CPU and data transfer.

(b) How do IR, PC, A, MAR and MDR relate to each other in the operation of LMC? (3 marks)

They form the machine cycle inside the LMC. The LMC fetch operation involves the MAR and MDR. Whereas the LMC execute operation involves their. PC AND A.

(c) Give a scenario or example to support your elaboration in Q4 (b) above.

(4 marks)

5. One large modern computer has a 36-bit MAR. How much memory can this computer address?

2^36 = 6.87194767 x 10^10 bytes

7

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

6. Assuming that LMC model is applied. Suppose that the following instructions are found at the given location in memory:

Program counter: 20

Value in memory location 20: 550 (LOAD)

Value in memory location 21: 151 (ADD)

Value in memory location 22: 350 (STORE)

:

Value in memory location 50: 422

Value in memory location 51: 008

Show the changes of contents in IR, PC, MAR, MDR and A during the execution of Instruction 20, 21 and 22.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | HR | PC | MAR | MDR | A |
| 20 (LOAD) | 550 | 21 | 50 | 422 | 422 |
| 21 (ADD) | 151 | 22 | 51 | 008 | 430 |
| 22 (STORE) | 350 | 23 | 50 | 130 | 430 |

7. Show the changes of contents in IR, PC, MAR, MDR and A during the execution of Instruction 20 and Instruction 21. (8 marks)

Program counter: 20

Value in memory location 20: 550 (LOAD)

Value in memory location 21: 151 (MUL)

Value in memory location 22: 350 (STORE)

:

Value in memory location 50: 2316

Value in memory location 51: 516

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | HR | PC | MAR | MDR | A |
| 20 (LOAD) | 550 | 21 | 50 | 2316 | 2316 |
| 21 (MUL) | 151 | 22 | 51 | 516 | AF base 16 |
| 22 (STORE) | 350 | 1 | 50 | 2316\*516  AF BSAE 16 | AF base 16 |

8

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 5: Machine Execution

1. Name the FOUR (4) basic components of an assembly language instruction.

**Label, Mnemonic, Operand, Comment**

2. What is the rule to follow when the following type of operand is used?

(a) Memory as operand

* 1. Segment:Offset
  2. MOV AX, NUM
  3. MOV NUM, AX (register to memory)
  4. MOV NUM, DIGIT (CANNOT variable to variable)

1. Register as operand
   1. Registers must be same type. 8-bit cannot mix with 16-bit register.
   2. MOV AX, BX
   3. MOV AH, BH
   4. MOV AX, BH (Cannot)
2. Immediate value as operand
   1. For 2-operand instruction, only as 2nd operand (first must not be immediate value)
   2. MOV AX, 10
   3. MOV NUM (variable), 10
   4. MOV 30, BX (Cannot)

3. Issue DEBUG commands for the following operations:

(a) Reset current IP value to 200. (2 marks)

* 1. -r ip
  2. 200

1. Insert the machine instruction 10 AB 2A 35 and 5D into Code Segment with Offset Address 100. (2 marks)
   1. -e cs:100 10 ab 2a 35 5d
2. Translate the machine instruction from Offset Address 100 to 108 into assembly code. (2 marks)
   1. -u 100 108
3. Execute a group of 5 instruction starts from the offset address 100H. (2 marks)
   1. -p=100, 5
4. Compare the content at code segment in offset address range 0100H to 010AH with the content in the offset address range 0150H to 015AH (2 marks)
   1. -c 100 10A 150

4. Explain the following DEBUG operations.

(a) –D CS:200 (1 mark)

* 1. Dump the first 128-bytes machine code in hex notation from CS offset 200.

(b) –A 200

XXXX:0200 SUB BL,42 (2 marks)

Assemble the code SUB BL,42 into CS offset 200.

(c) –T 3 (1 mark)

Trace execution of 3 instructions from the current offset

1. H 3064 2130

Calculate the hex sum and difference of 3064H and 2130H.

5. Provide the symbolic code for the following operations and trace the execution result for each of them.

(a) Move the hexadecimal value 4269 to AX register



1. Move the 7012H to BX register



1. Add the value of BX to AX register



1. Exchange the content in AX and BX



9

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

6. Describe each of the following DEBUG commands. Complete the table below. Command Purpose Format / E.g. Output

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Purpose | Format / E.g. | Output |
| A | Convert symbolic code to object code | A [address] | Object codes. (Not shown.) |
| C | Compare 2 groups of content in memory segment | C range address | Segment:offset addresses & content of the selected groups |
| D | Display the content of memory segment | D [range] | Segment:offset addresses, object codes & ASCII codes |
| E | Enter the object to memory segment | -E CS:100 8B | Not shown.  Use D command to confirm entered |
| F | Fill in a group of ASCII code repeatedly | F range list | Not shown.  Use D command to confirm filled in. |
| H | Perform hexadecimal computation | H value1 value2 | Result of addition and subtraction |
| P | Proceed to execute a group of instruction | P [=address] [number] | Changes of registers & next instruction |
| Q | Quit the DEBUG program | -Q | Back to root directory |
| R | Retrieve / edit the content of register(s) | R [register] | Content of register(s) |
| T | Trace the execution of instruction 1 by 1 | -T | Changes of registers & next instruction |
| U | Convert object code to symbolic code | U [range] | Segment:offset address, symbolic codes & object codes |

10

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 6: Computer Architecture

1. Write the command line to perform the followings:

(a) To assemble a source program named CSA.ASM into listing and object files.

* 1. masm csa.asm

1. To link an object file named CSA.OBJ into executable and map files.
   1. Link csa.obj
2. To directly execute CSA.EXE from DOS.
   1. Csa.exe
3. To execute CSA.EXE through DEBUG program.
   1. Debug csa.exe
4. Illustrate the process of editing, assembling, linking and executing assembly language programs in an appropriate diagram

Editing (User) [.ASM] 🡪 Assembling (Assembler) 🡪 Linking (Linker) 🡪 Executing (CPU)

3. Write an assembly program using simplified segment definitions to perform the following operations:

(a) Move the immediate value hex 40 to the AL register

MOV AL, 40

1. Shift AL contents one bit left

SHL AL, 1

1. Move immediate value hex 1A to BL

MOV BL, 1A

1. Multiply AL by BL

MUL BL

4. Modify your answer in Q4 above for the following requirements:

(a) Define a 1-byte item named ITEMA containing hex 40 and another named ITEMB containing hex 1A.

ITEMA DB 40

ITEMB DB 1AH

(b) Define a 2-byte item named ITEMC with no constant

ITEMC DW ?

1. Move the contents of ITEMA to AL and shift left one bit

MOV AL, ITEMA

SHL AL, 1

1. Multiply AL by ITEMB

MOV BL, ITEMB

MUL AL

1. Move the product in AX to ITEMC

MOV ITEMC, AX

What is the content in the destination register for each of the above instruction? Show your working.

11

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 7: Assembly Language Fundamentals 1

1. Define the following data items in assembly language with the name DATA1 to DATA4 respectively. (4 marks)

(a) A 1-byte item containing the hex equivalent to decimal 40.

DATA1 DB 40D

(b) A 2-byte item containing an undefined value.

DATA2 DW ?

(c) A string with the characters H,E,L,L,O

DATA3 DB “HELLO$”

1. An item containing the consecutive word values 1, 3, 5, 7 and 9.

DATA4 1,3,5,7,9

2. Determine whether each of the following names is VALID or INVALID identifier that conforms to the rules of Assembly Language. Give the reason if it is INVALID. (10 marks)

(a) 2NAME

INVALID. First character cannot be digits

(b) Total\_1

VALID

(c) #TelNo

VALID

(d) .Counter

INVALID. First character cannot be `.`

1. @Word

INVALID. Used in @data and other special identifiers

3. Below shows a sample of an assembly program. Examine the code. Identify SIX (6) errors found in the program and correct them. You are required to indicate the Line Number in your answer followed by the correct code statements. (6 marks)

Line number Program content

|  |  |
| --- | --- |
| Line 1  Line 2  Line 3  Line 4  Line 5  Line 6  Line 7  Line 8  Line 9  Line 10  Line 11  Line 12  Line 13  Line 14  Line 15  Line 16  Line 17 | . MODEL SMALL  . STACK 64  . DATA  DATA1 DW 1AH  DATA2 DB 380  DATA3 DW ?  .CODE  MAIN PROC FAR  MOV AX, DATA  MOV DS, AX  MOV AX, DATA1  ADD AX, DATA2  MOV DATA3, AX  MOV FX, 4C00H  INT 21H  MAIN END  ENDP MAIN |

Error:

Line 5:DATA2 DW 380

LINE 9:MOV AX,@DATA

LINE 12: ADD AL, DATA2

LINE 14:MOV AX,4C00H

LINE 16: MAIN ENDP

LINE 17:END MAIN

12

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

4. Determine the final content of AL register after running the following assembly instructions. (2 marks)

MOV AX, 0H

MOV AL, 34H MOV BL, 04H MOV CL, AL

DIV CL

MUL BL

SHL AL, 1

DEC AL

AX

AH AL BL CL

5. Consider the code fragment below. Examine the code and show the contents of the AX and BX registers. (10 marks)

.DATA

DATA1 DB 32H DATA2 DB 12H .CODE

MOV AX, 0004H MOV BX, 0000H MOV BL, DATA1 SHR BL, 1

ADD BL, DATA2 MUL AL

AX BX

AH AL BH BL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SUB BX, AX |  |  |  |  |

6. Using only MOV, ADD, SUB, INC, DEC, and NEG, translates the following high-level language assignment statements into assembly language. Assume that A, B, and C are word variables.

1. A = B – A
   1. MOV AX, A
   2. SUB B, AX
2. C = A + B

MOV AX, A

ADD AX, B

MOV C, AX

1. A = - (A + 1)

INC A

DEC A

1. B = 4 \* B + 6

MOV AX, B

ADD AX, AX

ADD AX, AX

ADD AX, 6

MOV B, AX

1. A = B - A – 1

MOV AX, B

SUB AX, A

DEC AX

MOV A, AX

13

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 8: Assembly Language Fundamentals II

1. Using int 21h function 01h and LOOP instruction, write a segment of assembly program to accept 3 characters and store them into a variable named INPUT. (3 marks)

.DATA

INPUT DB 3 DUP (" "), "$"

.CODE

; - Write the assembly codes here –

MOV SI,0

MOV CX, 3

L1:

MOV AH, 01H

INT 21H

MOV INPUT[SI], AL

INC SI

LOOP L1

2. Based on the assembly program shown below, answer the following questions:

Line number Program Contents

|  |  |
| --- | --- |
| Line 1  Line 2  Line 3  Line 4  Line 5  Line 6  Line 7  Line 8  Line 9  Line 10  Line 11  Line 12  Line 13  Line 14  Line 15  Line 16  Line 17  Line 18 | .MODEL SMALL  .STACK 64  .DATA  MESSAGE1 DW "HI ASSEMBLY!", 10, 13, "$"  -----------------------------------------------------------------------  .CODE  MAIN PROC FAR  MOV AX, DATA  MOV DX, AX << ERROR 1, @DATA  MOV CX, 9 << ERROR 2, DS  L1: MOV AH, 09H  LEA DS, MESSAGE1 << ERROR 3, DX  INT 21H  LOOP L1  MOV AX, 4C00H  INT 21H  MAIN END  END MAIN |

(a) The assembly program consists of FIVE (5) errors. Identify and correct the errors. (5 marks)

(b) How many times will the message “HI ASSEMBLY!” be displayed on the screen when the program runs? (1 mark)

**9 times**

3. Given the following variable definition in the data segment.

Data definition Sample output

|  |  |
| --- | --- |
| .DATA  VAR1 DB 2, 0, 1, 1 | 2011 |

Write a program segment to display consecutively each byte of the data defined in VAR1 using indirect-offset address with LOOP operator. (4 marks)

MOV DI, 0

MOV CX, 4

LOOPn:

MOV AH, 02H

MOV DL, VAR1[DI]

ADD DL, '0'

INT 21H

INC DI

LOOP LOOPn

14

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

4. Identify and explain how many times each of the following LOOP operations loop:

(a) MOV CX , 1

L1 :

….

LOOP L1

1 times

(b) L2 : MOV CX , 10

…..

LOOP L2

Endless loop

(c) MOV CX , 10

L3 :

INC CX

LOOP L3

Endless loop

8. Given the following table and initializing instructions, show the effect of the MOV

VALUE\_TBL DB 1, 2, 3, 4, 5, 6, 7, 8

…

LEA BX, VALUE\_TBL

MOV DI, 4

(a) MOV CL, [BX] ; CL = 1

(b) MOV DL, [BX + 3] ; DL = 4

(c) MOV AL, [BX + DI] ; AL = 5

9. Using nested loop structure, write an assembly program that print out the right-angled triangle as below:

Sample output :

|  |
| --- |
| \*\*\*  \*\*  \* |

.MODEL SMALL

.STACK 100

.DATA

tmp DB ?

times DW 3

nl DB 13, 10, "$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV CX, times

; outer loop : 3 times

L1:

MOV tmp, CL

MOV CX, times

; inner loop (3,2,1)

l2:

MOV AH, 2

; print star

MOV DL, "\*"

INT 21h

LOOP l2

MOV AH, 9

LEA DX, nl

INT 21h

DEC times

MOV CX, tmp

LOOP l1

MOV AX,4C00H

INT 21H

MAIN ENDP

END MAIN

10. Modify the assembly program in Q9 to generate the result as below:

Sample output :

|  |
| --- |
| 123  45  6 |

.MODEL SMALL

.STACK 100

.DATA

tmp DB ?

times DW 3

num DB "1"

nl DB 13, 10, "$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV CX, times

; outer loop : 3 times

L1:

MOV tmp, CL

MOV CX, times

; inner loop (3,2,1)

l2:

MOV AH, 2

; print star

MOV DL, num

INC num

INT 21h

LOOP l2

MOV AH, 9

LEA DX, nl

INT 21h

DEC times

MOV CX, tmp

LOOP l1

MOV AX,4C00H

INT 21H

MAIN ENDP

END MAIN

15

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 9: Conditional Processing

1. Assume that AL and BL contain unsigned data and that CL and DL contain signed data. Determine the CMP and conditional jump instructions for the following.

(a) Is CL equal to or smaller than DL?

CMP CL, DL

JLE L1

1. Is AL equal to or smaller than BL?

CMP AL, BL

JBE L2

1. Is AL greater than BL?

CMP AL, BL

JA L3

1. Is CL greater than DL?

CMP CL, BL

JG L4

1. Does DL contain zero?

CMP DL, 0

JZ L5

1. Is there an overflow?

JO L6

2. Assume that DL contains 01111001 and that an item named BOOL\_AMT contains 11100011. Determine the effect on DL for the following unrelated operations:

(a) AND DL , BOOL\_AMT

01100001

1. OR DL , BOOL\_AMT

11111011

1. XOR DL , BOOL\_AMT

1001010

1. AND DL , 00000000B

00000000

1. XOR DL , 11111111B

10000110

1. Write instructions that first clear bit 0 and bit 1 in AL. Then, if the destination operand is equal to zero, the code should jump to label L3. Otherwise, it should jump to label L4.

AND AL, 1100B

JZ L3 (Can straight away jump because check AL)

JNZ L4

4. Translate the following C statements into assembly language.

(a) if (num1 == num2)

{

X = 1; Y = 2;

}

ANSWER:

MOV BL, num1

CMP BL, num2

JE L1

JMP FINISH

L1:

MOV X, 1

MOV Y, 2

FINISH:

(b) if (al > bl && bl > cl)

{

X = 1;

}

CMP AL, BL

JLE FINISH

CMP BL, CL

JLE FINISH

MOV X, 1

FINISH:

(c) if (al > bl || bl > cl)

{

X = 1;

}

CMP AL, BL

JG L1

CMP BL, CL

JG L1

JMP FINISH

L1:

MOV X,1

FINISH:

16

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

5. Write an assembly program to count the vowels in a character array “This is my favorite”. Print out the number of occurrences of each of the vowels a, e, i, o, and u.

Sample output:

a =1 ;e = 1 ; i = 3 ; o = 1 ; u = 0

.MODEL SMALL

.STACK 100

.DATA

CHAR DB "This is my favorite"

txtA DB "a count:"

countA DB '0'

txtE DB "e count:"

countE DB '0'

NL DB 13,10,'$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; COUNTING VOWELS

MOV CX, 19

MOV SI, 0

L1:

CMP CHAR[SI], 'a'

JMP CA

CMP CHAR[SI], 'e'

JE CE

JMP CONTINUE

CA:

INC countA

JMP CONTINUE

CE:

INC countE

JMP CONTINUE

LOOP 1

; DISPLAY VOWELS

MOV AH, 09H

LEA DX, txtA

INT 21H

MOV AH, 02H

MOV DL, countA

INT 21H

MOV AH, 09H

LEA DX, NL

INT 21H

MOV AH, 09H

LEA DX, txtE

INT 21H

MOV AH, 02H

MOV DL, countE

INT 21H

MOV AH, 09H

LEA DX, NL

INT 21H

MOV AX, 4C00H

INT 21H

MAIN ENDP

END MAIN

6. Using simplified segment directive, write an assembly language program that will find the largest value from a list. (15 marks)

The program should:

(a) Prompt the user to enter 5 decimal digits.

(b) Accept the user input. (Assume user will only enter 5 digits)

(c) Find the largest value inside the list.

(d) Display the largest value on the screen.

Sample output:

|  |
| --- |
| Please enter 5 decimal digits >> 35827  The largest value in the list is >> 8 |

|  |
| --- |
| .MODEL SMALL |
|  | .STACK 100 |
|  | .DATA |
|  | INPROMPT DB "Please enter 5 decimal digits >> $" |
|  | OUTPROMPT DB "The largest value in the list is >> $" |
|  | NL DB 13,10,'$' |
|  | DIGITS LABEL BYTE |
|  | MAXN DB 6 |
|  | ACTN DB ? |
|  | ACTSTR DB 20 DUP('$') |
|  | LARGEST DB ? |
|  | .CODE |
|  | MAIN PROC |
|  | MOV AX, @DATA |
|  | MOV DS, AX |
|  |  |
|  | ; ASK FOR 5 DIGITS |
|  | MOV AH, 09H |
|  | LEA DX, INPROMPT |
|  | INT 21H |
|  |  |
|  | MOV AH, 0AH |
|  | LEA DX, DIGITS |
|  | INT 21H |
|  |  |
|  | MOV AH, 09H |
|  | LEA DX, NL |
|  | INT 21H |
|  |  |
|  | ; MOVE LAST NUMBER IN |
|  | LEA DI, ACTSTR |
|  | MOV BL, [DI] |
|  | MOV LARGEST, BL |
|  | INC DI |
|  | ; CHECK-AND-COMPARE |
|  |  |
|  | MOV CX, 4 |
|  | MAX: |
|  | MOV BL, [DI] |
|  | CMP BL, LARGEST |
|  | JLE CONTINUE |
|  | ; IF LARGEST, MOVE IN |
|  | MOV BL, [DI] |
|  | MOV LARGEST, BL |
|  | ; ELSE, JUST CONT |
|  | CONTINUE: |
|  | INC DI |
|  | LOOP MAX |
|  |  |
|  | ; DISPLAY LARGEST VALUE |
|  | MOV AH, 9H |
|  | LEA DX, OUTPROMPT |
|  | INT 21H |
|  |  |
|  | MOV AH, 02H |
|  | MOV DL, LARGEST |
|  | INT 21H |
|  |  |
|  | MOV AX,4C00H |
|  | INT 21H |
|  | MAIN ENDP |
|  | END MAIN |

17

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 10: Keyboard and Screen Processing

1. List out the 3 main usages of the INT 21H operations.

Input, output, terminate program

2. What are the purposes of the following functions?

(a) INT 10H function 02H

Set cursor

(b) INT 10H function 06H

Scroll screen

(c) INT 21H function 01H

Input byte

1. INT 21H function 02H

Output byte

1. INT 21H function 07H

Input byte (no echo)

1. INT 21H function 09H

Output string

1. INT 21H function 0AH

Input string

1. Write the instructions required to set the cursor to row 12, column 24. Then rewrite the instructions set so that it involves DX register.

MOV AH, 02H

MOV BH, 00 ; page number

MOV DH, 11 ; row (start from 0) ;

MOV DX, 0B17H

MOV DL, 23 ; column (start from 0)

INT 10H

1. Explain the effect of the following program segment?

MOV AX , 0600H ; Note: can simplify to AH, 06H (scroll & clear screen)

MOV BH , 71H ; (BG)(FG)H

MOV CX , 0000H ; START, Divide into 2; (ROW)|(COL)H

MOV DX , 184FH ; END, Divide into 2; (ROW)|(COL)H

INT 10H

5. “When using INT 21H function 09H for displaying, a delimiter ($) is defined immediately following the display area.”

(a) Why the delimiter is needed?

To indicate the end of string

(b) What will happen if the delimiter is missing?

The remaining data will be displayed.

6. INT 21H function 0AH for keyboard input requires a parameter list.

(a) What will the first byte store?

The maximum number of bytes

(b) What will the second byte store?

The total number of bytes currently stored

(c) What will the continuous bytes store?

The field that is to contain the typed chars

7. What register works closely with INT 21H to perform various functions?

AH/AX

8. In order to display a character using INT 21H function 02H, the character to be displayed must be loaded to which register?

DL

18

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

9. Write instructions to reset cursor in the new line and at the left most position.

MOV AH, 02H

MOV DL, 13

INT 21H

MOV DL, 10

INT 21H

10. For INT 21H function 09H, when LEA command is applied, where will the assembler load the effective (offset) address to?

DX

11. Identify the hex values of AH, DX and INT if the cursor is set at row 7, column 14 on screen using assembly language. (3 marks)

MOV AH, 02H

MOV BH, 00

MOV DX, 060DH

INT 21H

12. Write an assembly program that will prompt the user for input. The program will displays all the numeric data input by the user, while filters all other characters. (Assume user will not enter $.)

Sample output:

Please enter any data and press enter >> 12A3%C&4

The numerical value is >> 1234

|  |
| --- |
| .MODEL SMALL |
|  | .STACK 100 |
|  | .DATA |
|  | INPROMPT DB "Please enter any data and press enter >> $" |
|  | OUTPROMPT DB "The numerical value is >> $" |
|  | NL DB 13, 10, "$" |
|  | KBINPUT LABEL BYTE |
|  | MAXC DB 50 |
|  | ACTC DB ? |
|  | ACTSTR DB 50 DUP('$') |
|  |  |
|  | STRIPSTR DB 50 DUP('$') |
|  |  |
|  | .CODE |
|  | MAIN PROC |
|  | MOV AX, @DATA |
|  | MOV DS, AX |
|  |  |
|  | ; PRINT LABEL |
|  | MOV AH, 09H |
|  | LEA DX, INPROMPT |
|  | INT 21H |
|  |  |
|  | ; ACCEPT INPUT |
|  | MOV AH, 0AH |
|  | LEA DX, KBINPUT |
|  | INT 21H |
|  |  |
|  | MOV AH, 09H |
|  | LEA DX, NL |
|  | INT 21H |
|  |  |
|  | ; STRIP ALL NON-NUMERIC FIGURE |
|  | MOV CH, 0 |
|  | MOV CL, ACTC |
|  | MOV SI, 0 |
|  | MOV DI, 0 |
|  |  |
|  | OPERATE: |
|  | MOV BL, ACTSTR[SI] |
|  | ; CHECK IF BETWEEN '0' AND '9' |
|  | CMP BL, '0' |
|  | JB CONTINUE |
|  | CMP BL, '9' |
|  | JA CONTINUE |
|  |  |
|  | MOV STRIPSTR[DI], BL |
|  | INC DI |
|  |  |
|  | CONTINUE: |
|  | INC SI |
|  | LOOP OPERATE |
|  |  |
|  | ; PRINT LABEL |
|  | MOV AH, 09H |
|  | LEA DX, OUTPROMPT |
|  | INT 21H |
|  |  |
|  | ; PRINT OUTPUT |
|  | MOV AH, 09H |
|  | LEA DX, STRIPSTR |
|  | INT 21H |
|  |  |
|  | MOV AX, 4C00H |
|  | INT 21H |
|  | MAIN ENDP |
|  | END MAIN |

19

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 11: Input and Output Facilities

1. Define I/O module. Discuss the roles of I/O module in CPU – I/O communication. (2, 8 marks)

2. An I/O technique is known as Interrupt-driven I/O. explain the technique. (5 marks) 3. Explain the FOUR (4) usages of Interrupt. (8 marks)

4. Interrupt is a signal sent to the CPU to indicate an event that requires immediate attention. Servicing the interrupt is a process where the CPU branches to the interrupt handler program from its current program. Explain the flow of servicing an interrupt in detail. (4 marks)

5. Suppose you are sending a block of data from a disk to memory.

(a) Identify the most appropriate I/O handling technique to support the data transfer process as mentioned. Explain your choice. (2 marks)

(b) Based on the scenario given in Q7, explain the process of the I/O handling technique that you mentioned in Q7 (a) with the aid of an appropriate diagram. (6 marks)

6. Distinguish between a bus architecture and a channel architecture in terms of their characteristics and operations. (6 marks)

7. In the event of multiple interrupts, illustrate how these multiple interrupts could be handled by using vectored interrupt and polled interrupt respectively. Support your answer with a diagram for each. (10 marks)

20

AACS3064 COMPUTER SYSTEMS ARCHITECTURE

TUTORIAL 12: Computer Architecture

1. Briefly explain how a system performance is affected.

Multiple CPUs

- Over heads that are required to distribute the instruction among the different CPUs. The conflicts among CPUs in sharing resources

Faster system clock speed

- Speed up the instruction cycle of the system and directly affect the performance of the system. Design of faster CPU circuits and buses will increase the clock speed

Wider instruction and data paths

- A wider interface between CPUs and memory bus allows the CPU to carry more instructions in a single operation and reduce disk accesses. It also allows the CPU to perform pipelining or instruction pools.

Faster disk access

- CPU is able to perform thousand of instruction in the time required for a single disk access. Thus, Small improvements in the disk access speeds will increase the system performance.

Memory access time

- Memory accesses can be reduced by providing more registers (to reduce memory access instructions) or apply extremely fast memory (known as cache memory)

Increased amount of memory and faster memory

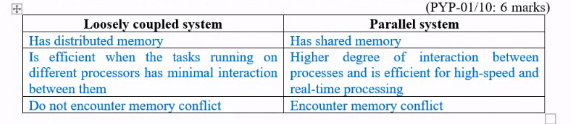
- Increased amount of memory provides larger buffer spaces to hold additional data to enable system to continue processing during disk accesses.

Disk access

- Minimize the number of disk access is the best way in using memory

2. What is the main different between loosely-coupled system and parallel system?

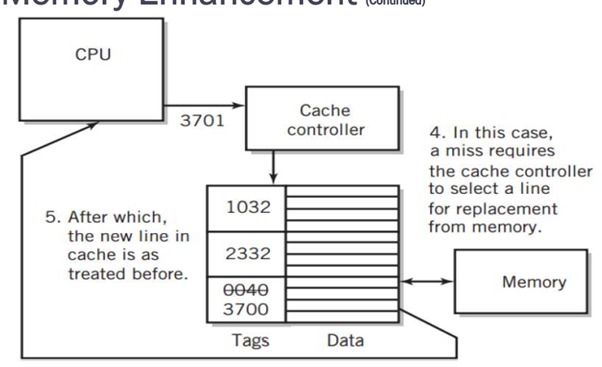
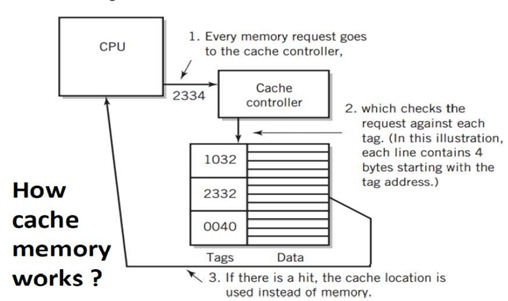
(PYP-01/10: 6 marks)



3. Cache memory is used as a technique to enhance memory performance.

(a) With the aid of a diagram, explain how a cache memory helps to enhance memory performance.

(PYP-08/13: 5 marks)



(b) What will happen when there is a cache miss? (PYP-08/13: 2 marks)

When a cache miss occurs, the system or application searching for the data in the underlying data store, lengthening the time it takes to complete the request. In most cases, the system will write the data to the cache, raising latency once further, but this will be balanced by cache hits on other data.

(c) Analyze the impact of cache miss on the execution of instruction. (PYP-08/13: 4 marks)

(d) Explain what will happen when a cache memory is full. (PYP-08/15: 4 marks)

some of the contents of the cache memory has to be “forcibly removed” to make room for the new information that needs to be written there.

4. List and explain TWO (2) configurations of a multiprocessing system. (PYP-08/13: 6 marks)

Tightly coupled system

- All the computers are tightly coupled configuration and have access to the same program and data.

- Program execution are divided among CPU, each CPU operates independently. No communication channel is required since each CPU accesses to the same memory and I/O devices.

Loosely coupled systems

- Each system is complete in itself, each has its own CPU, memory and I/O facilities.

- Each system is known as a node of the cluster.

5. Explain the following techniques in enhancing memory performance.

(a) Wider path for memory access

Retrieve multiple bytes instead of 1 byte at a time

Several bytes or words can be read / written between CPU and memory with each access, these bytes can be separated as required and processed in the usual way.

(b) Cache memory (PYP-04/13: 4 marks)

Organized into blocks. These blocks are used to hold an exact reproduction of a corresponding amount of storage somewhere in the main memory.

Each block hold a tag which identifies the corresponding location of data in the main memory

6. What performance improvement is offered by the use of memory interleaving?

Memory interleaving may be a technique that CPUs use to extend the memory information measure on the market for Associate in Nursing application. while not interleaving, consecutive memory blocks, usually cache lines, square measure scan from an equivalent memory bank.

7. Differentiate between logical address and physical address.

Logical addresses are relative location of data, instructions and branch target & are separate from physical addresses

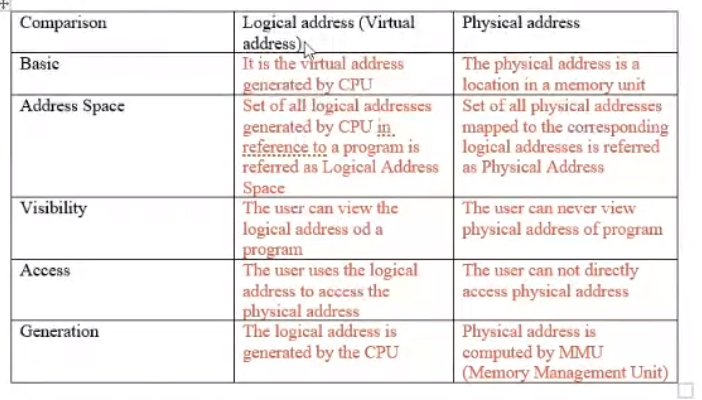
Provide important capabilities

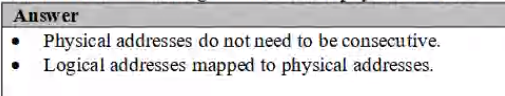
- The ability of relocating programs easily from one part of memory to another. This ability is very important to multi-tasking.

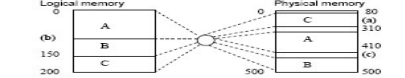
- The ability to split program into smaller pieces that can be loaded into different part of memory.

- Same logical address of 2 different program transformed into different physical location or vice versa.

- Make a program think itself has more memory than actually exist on the computer





8. Complete the diagram below.21

