**Sample Question Format**

**KIIT Deemed to be University**
**Online Mid Semester Examination(Spring Semester-2021)**

**Subject Name & Code:** COA, CS 2006 **Applicable to Courses:**

**Full Marks=20** **Time:1 Hour**

**SECTION-A(Answer All Questions. All questions carry 2 Marks)**

**Time:20 Minutes** **(5×2=10 Marks)**

<table>
<tr><th>Question No</th><th>Question Type( MCQ/ SAT)</th><th>Question</th><th>Answer Key(if MCQ)</th><th>CO Mapping</th></tr>
<tr><td>Q.No:1(a)</td><td></td><td>An LOAD instruction is kept in memory at an address 400 and the memory address 401 occupies the address field of the instruction which is shown below. The Opcode is used to add the content of accumulator with an operand. The content of accumulator is 100 and the content of register R600 is 500. What will be the effective address of the operand and the content of accumulator after the execution of the instruction, if the addressing mode is<br><br>
<table>
<tr><th>Addresses</th><th>Address field of Instructions Opcode</th></tr>
<tr><td>401</td><td>600</td></tr>
<tr><td></td><td></td></tr>
<tr><td>500</td><td>800</td></tr>
<tr><td></td><td></td></tr>
<tr><td>600</td><td>700</td></tr>
<tr><td></td><td></td></tr>
<tr><td>700</td><td>900</td></tr>
</table>
(i) Indirect<br>(ii) Register indirect<br><br>a) 700, 1000,500,900<br>b) 600,800,600,500<br>c) 700,1000,600,500<br>d) 600,800,500,900</td><td>a</td><td>CO1</td></tr>
<tr><td></td><td></td><td>How many memory references are required for the execution of the following code, where LOCX LOCY denote memory locations.<br><br>MOV (LOCX), R1<br>ADD R1, (LOCY)<br>INCR LOCY<br><br>a) 7</td><td>d</td><td>CO1</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | | b) 9<br>c) 10<br>d) 11 | | |
| | | Consider the code given below. What will be the offset for the instruction BGTZ to branch to the location labeled LOOP? All the instructions are 4 bytes in length.<br><br>1000 MOV #10,R1<br>MOV #LOCA,R2<br>CLEAR R3<br>LOOP ADD (R2)+, R3<br>DEC R1<br>BGTZ ?<br><br>a) -8<br>b) -12<br>c) 8<br>d)12 | b | CO1 |
| | | Consider the below given memory map.<br><br>Address / Content<br>20 / 40<br>30 / 50<br>40 / 60<br>50 / 70<br>60 / 80<br><br>Which of the following instruction will load 60 in the accumulator<br><br>a) LOAD #40<br>b) LOAD 60<br>c) LOAD (20)<br>d)LOAD (40) | C | CO1 |
| | **Q.No:1(b)** | What is the assembly language program to derive the expression<br>X = T+O/(W*E)-R<br>in a stack based computer with zero address instructions.<br>A. PUSH T; PUSH O; PUSH W; PUSH E; MULT; DIV; ADD; PUSH R; SUB; POP X<br>B. PUSH T; PUSH O; PUSH W; DIV; PUSH E; MULT; ADD; PUSH R; SUB; POP X<br>C. PUSH T; PUSH O; SUB; PUSH W; PUSH E; PUSH R;; SUB; MULT; ADD; POP X<br>PUSH T; PUSH O; PUSH W; ADD; DIV; PUSH E; PUSH R; MULT; SUB; POP X | **A** | CO1 |
| | | What is the assembly language program to | **B** | **CO1** |

| | | | | |
|---|---|---|---|---|
| | | derive the expression<br>X = T+(O/W)*E-R<br>in a stack based computer with zero address instructions.<br>A. PUSH T; PUSH O; PUSH W; PUSH E; MULT; DIV; ADD; PUSH R; SUB; POP X<br>B. PUSH T; PUSH O; PUSH W; DIV; PUSH E; MULT; ADD; PUSH R; SUB; POP X<br>C. PUSH T; PUSH O; SUB; PUSH W; PUSH E; PUSH R;; SUB; MULT; ADD; POP X<br>**D.** PUSH T; PUSH O; PUSH W; ADD; DIV; PUSH E; PUSH R; MULT; SUB; POP X | | |
| | | What is the assembly language program to derive the expression<br>X = T-O+W*(E-R)<br>in a stack based computer with zero address instructions.<br>A. PUSH T; PUSH O; PUSH W; PUSH E; MULT; DIV; ADD; PUSH R; SUB; POP X<br>B. PUSH T; PUSH O; PUSH W; DIV; PUSH E; MULT; ADD; PUSH R; SUB; POP X<br>C. PUSH T; PUSH O; SUB; PUSH W; PUSH E; PUSH R;; SUB; MULT; ADD; POP X<br>**D.** PUSH T; PUSH O; PUSH W; ADD; DIV; PUSH E; PUSH R; MULT; SUB; POP X | **C** | **CO1** |
| | | What is the assembly language program to derive the expression<br>X = T/(O+W)-E*R<br>in a stack based computer with zero address instructions.<br>A. PUSH T; PUSH O; PUSH W; PUSH E; MULT; DIV; ADD; PUSH R; SUB; POP X<br>B. PUSH T; PUSH O; PUSH W; DIV; PUSH E; MULT; ADD; PUSH R; SUB; POP X<br>C. PUSH T; PUSH O; SUB; PUSH W; PUSH E; PUSH R;; SUB; MULT; ADD; POP X<br>**D.** PUSH T; PUSH O; PUSH W; ADD; DIV; PUSH E; PUSH R; MULT; SUB; POP X | **D** | **CO1** |
| **Q.No:1(c)** | | When using Branching, the usual sequencing of the PC is altered. A new instruction is loaded which is called as _____<br><br>A. CALL Instruction<br>B. Return Instruction<br>C. Branch target | c | CO1 |

<table>
<tr><td></td><td></td><td>D. Jump Target</td><td></td><td></td></tr>
<tr><td></td><td></td><td>__ the most suitable data structure used to store the return addresses in the case of nested subroutines.<br>a) circular queue<br>b) Stack<br>c) Linear Queue<br>d) doubly Link list</td><td>b</td><td>CO1</td></tr>
<tr><td></td><td></td><td>MFC is activated by which functional unit of computer.<br>a. Central Processing Unit<br>b. Memory<br>c. Input Unit<br>d. Output Unit</td><td>b</td><td>CO2</td></tr>
<tr><td></td><td></td><td>What is subroutine nesting?<br>a) Having multiple subroutines in a program<br>b) Using a linking nest statement to put many subroutines under the same name<br>c) Having one routine call the other<br>d) None of the mentioned</td><td>c</td><td>CO1</td></tr>
<tr><td><u>Q.No:1(d)</u></td><td></td><td>If PC=2004 then PC will updated to _____ ? (initial value of R1=10, R2=-100)<br><table><tr><td>Memory Location</td><td>Instruction</td></tr><tr><td>2000</td><td>ADD R1, R2</td></tr><tr><td>2004</td><td>Branch>0 1000</td></tr></table><br>A. 2004<br>B. 2008<br>C. 3004<br>D. 3008</td><td>B</td><td>CO2</td></tr>
<tr><td></td><td></td><td>If PC=2048, in 2048 memory location “JUMP 1000” presents then PC will updated to _____ ?<br>A. 2048<br>B. 2052<br>C. 3048<br>D. 3052</td><td>D</td><td>CO2</td></tr>
<tr><td></td><td></td><td>If PC=2004 then PC will updated to _____ ? (initial value of R1=10, R2=100)<br><table><tr><td>Memory Location</td><td>Instruction</td></tr><tr><td>2000</td><td>ADD R1, R2</td></tr><tr><td>2004</td><td>BNZ 1000</td></tr></table><br>A. 2004<br>B. 2008<br>C. 3004<br>D. 3008</td><td>D</td><td>CO2</td></tr>
<tr><td></td><td></td><td>If PC=2004 then PC will updated to _____ ?</td><td>B</td><td>CO2</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | | (initial value of R1=10, R2=100)<br>Memory Location Instruction<br>2000 ADD R1, R2<br>2004 BZ 1000<br><br>A. 2004<br>B. 2008<br>C. 3004<br>D. 3008 | | |
| **Q.No:1(e)** | | Three-bus organization of the datapath inside of a processor, CONSTANT 4 at ALU input is useful ----------.<br><br>A. to increment other addresses like memory addresses in LOADMULTIPLE & STOREMULTIPLE type instructions.<br>B. to add 4 to the contents of PC for updating its location to point to the next instruction in the given sequence in memory.<br>C. for Branching<br>D. None of the Above | A | CO2 |
| | | The CONSTANT 4 at input A of the ALU is useful ----------, in a Three-bus organization of the datapath inside of a processor.<br><br>A. for Branching<br>B. to add 4 to the contents of PC for updating its location to point to the next instruction in the given sequence in memory.<br>C. to increment other addresses like memory addresses in LOADMULTIPLE & STOREMULTIPLE type instructions.<br>D. None of the Above | C | CO2 |
| | | In Three-bus organization of the datapath inside of a processor, Which one of the following is the correct Sequence of Control Steps for Fetching an instruction from memory?<br><br>A.<br>1. PCout, R=B, MAR in, Read<br>2. IncPC, WMFC<br>3. MDRoutB, R=B, IRin<br><br>B.<br>1. PCout, R=B, MAR in,<br>2. IncPC, Read, WMFC<br>3. MDRoutB, R=B, IRin<br><br>C.<br>1. PCout, R=B, MAR in, IncPC<br>2. Read, WMFC<br>3. MDRoutB, R=B, IRin<br><br>D.<br>None of the above | A | CO2 |

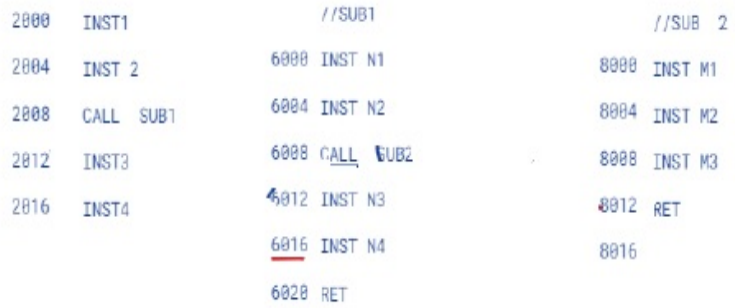
| | | | | |
|---|---|---|---|---|
| | | Which of the following is the correct Sequence of Control Steps required to Fetch an instruction from memory in Three-bus organization of the datapath inside of a processor?<br><br>A.<br>3. PCout, MAR in, Read<br>4. IncPC, WMFC<br>3. MDRoutB, R=B, IRin<br><br>B.<br>1. PCout, R=B, MAR in, IncPC<br>2. Read, WMFC<br>3. MDRoutB, R=B, IRin<br><br>C.<br>1. PCout, R=B, MAR in, Read<br>2. IncPC, WMFC<br>3. MDRoutB, R=B, IRin<br><br>D.<br>None of the above | C | CO2 |

**SECTION-B(Answer Any One Question. Each Question carries 10 Marks)**

**Time: 30 Minutes** **(1×10=10 Marks)**

| Question No | Question | CO Mapping |
|---|---|---|
| 2.a. The contents of memory locations 5000, 5100 and 6000 are 1000, 40 and 90 respectively before the following program is executed. [ Here, the 2nd operand is the destination]<br><br>MOV #5000, R1<br>MOV 100(R1), R2<br>ADD R2, 6000<br>ADD (R1)+, R2<br>MOV R2, (5000)<br><br>What will be the contents memory locations 1000, 5000, 6000 and register R1 and R2 after the program is executed? Find out the number of memory references required for each of the instructions in the above program.<br><br>Solution<br><br>Mem[1000]= 1040<br>Mem[5100]= 40<br>Mem[6000]=130<br>[R1]=5001<br>[R2]=1040<br><br>**Instruction** **No of Memory references**<br>MOV #5000, R1 1 | | CO1 |

| | | |
|---|---|---|
| MOV 100(R1), R2 2<br>ADD R2, 6000 3<br>ADD (R1)+, R2 2<br>MOV R2, (5000) 3<br><br>2.b. Write the assembly language code the following pseudo code using the addressing modes known to you/ applicable for the given situation. Here p is a pointer to an integer.<br><br>p=1000;<br><br>*p=10;<br><br>p++;<br><br>d=*p + 20;<br><br>Answer:<br><br>MOVE #1000, R0<br>MOVE R0,P<br>MOVE #10, R1<br>MOVE R1, (P)<br>INC P<br>MOV #20, R2<br>ADD (P), R2<br>MOVE R2, D | | |
| 3.a) Register R5 is used in a program to point to the top of a stack. Assume that the stack address space ranges from 2000 to 1500 and each stack word consumes 4 bytes and machine is byte addressable. Write a sequence of instructions using the Index, Autoincrement, and Autodecrement addressing modes to perform each of the following tasks:<br><br>i) Pop the top two items off the stack, add them, and then push the result onto the stack.<br>ii) Copy the fifth item from the top into register R3.<br>iii) Remove the top ten items from the stack.<br><br>Answer:<br>Here, register R5 is used as the stack pointer (SP).<br>a) Pop the top two items off the stack, add them, and then push the result onto the stack.<br>Move (R5)+, R0<br>Add (R5)+, R0<br>Move R0, -(R5)<br>b) Copy the fifth item from the top into register R3.<br>Move 16(R5), R3<br>c) Remove the top ten items from the stack. | | CO1 |

| | | |
|---|---|---|
| Add #40, R5<br><br>3.b)<br>  <br><br>Initially the stack pointer SP contains 4000 and keeping a value NULL in the stack.<br>What are the content of PC, SP, and the top of the stack?<br>i) After the subroutine call instruction is executed in the main program?<br>ii) After the subroutine call instruction is executed in the subroutine SUB1?<br>iii) After the return from SUB2 subroutine?<br><br>ANS :<br>(I)PC = 6000 ,SP =3996 STACK[SP] = 2012<br>(ii) PC =8000 SP =3992 Stack[SP] = 6012 | | |
| 4. a. What is Von Neumman Concept? Discuss the Basic Operational Concept of a Computer and explain how it executes a instruction by taking an example of any assembly language instruction.<br><br>5. b. Discuss the factors that affect the performance of the computer. If a 8GHz computer takes 7 clock cycles for ALU instructions, 11 clock cycles for branch instructions and 6 clock cycles for data transfer instructions. Then Find the total time taken by the computer to execute the program that consists of 10 ALU instructions, 5 branch instructions and 5 data transfer instructions.<br><br>The factors that affect the performance of the computer are as follows:<br><br>i) **Clock cycle time/clock period (R)** – It is just the length of a cycle.<br><br>ii) **CPI (S) -** It is is the average number of clock cycles per instruction, for a particular machine and program.<br><br>iii) **Number of instructions in a program (N)** – It is the **dynamic instruction count** that is how many instructions are actually executed when the program runs, not the static instruction count that is how many lines of code are in a program. | | CO1 |

| | | |
|---|---|---|
| Number instructions=10+5+5=20=N<br><br>Total cycles=7*10+11*5+6*5=70+55+30=155<br><br>S=CPI=155/20<br><br>R=8Ghz<br><br>T=NS/R=(20*(155/20))/8x$10^{9}$=19.375 x$10^{-9}$ Sec=19.375 nano-sec | | |
| 5.a. Draw the schematic diagram of the architecture of a single bus CPU. Write the sequence of control steps for the following branch instructions for single bus CPU organization.<br><br>Branch=0 loop<br><br>**Control steps:**<br>**1. PC out, MAR in,Read, Select 4, Add, Z in**<br>**2. Z out, PC in, Y in, WMFC**<br>**3. MDR out, IR in**<br>**4. Offset-field-IR out, Select Y, Add, Z in, IF Z=0, then End**<br>**5. Z out, PC in, End.**<br><br>5.b. Write the sequence of control steps for the following instructions for single bus CPU organization. Assume second operand is the destination operand.<br>MUL #23, (R2)<br><br>**1. PC out, MAR in,Read, Select 4, Add, Z in**<br>**2. Z out, PC in, Y in, WMFC**<br>**3. MDR out, IR in**<br>**4. R2 out, MAR in, Read**<br>**5. Offset-field-IR out, Y in, WMFC**<br>**6. MDR out, Select Y, Add, Z in**<br>**7. R2 out, MAR in**<br>**8. Z out, MDR in, Write**<br>**9. WMFC, End** | | CO2 |
| 6.a. Explain the 3-bus architecture inside CPU with neat diagram. Write<br>the control signals for the following instruction.<br>MUL (R1) ,$R_5$<br><br>.Explanation of 3-bus architecture [2.5]<br><br>MUL (R1) ,R2 [2.5]<br><br>**STEPS:**<br>1. PCout,R=B,$MAR_{in}$,Read,Increment PC | | CO2 |

2. WMFC
3. $MDR_{outB}$,R=B,$IR_{in}$
4. $R_{1outB}$,R=B,$MAR_{in}$,Read
5. $R_{2outA}$,WMFC
6. $MDR_{outB}$,SelectA,MUL,$R_{2in}$,end

6.b. Discuss the advantages of 3-bus architecture inside CPU over single bus organization inside CPU and write the control signal for the following instruction execution in 3-bus architecture inside CPU.

MOVE (R1)+ ,$R_5$

Advantage of 3-bus architecture over single bus architecture: [2]

- Less control signals are required.
- At a time more than one out operation can possible on the data path .
- CPU can increase its speed.
- System performance can be improved.

MOVE (R1)+ ,$R_5$ [3]

**STEPS:**

1.PCout,R=B,$MAR_{in}$,Read,Increment PC
2.WMFC, $MDR_{inE}$
3.$MDR_{outB}$,R=B,$IR_{in}$
4.$R_{1outB}$,R=B,$MAR_{in}$,Read, $MDR_{inE}$
5.Select4,ADD,$R1_{in}$, WMFC
6.$MDR_{outB}$,R=B,$R_{5in}$, END

**Controller of Examinations**