***King Fahd University of Petroleum and Minerals***

***College of Computer Science and Engineering***

***Computer Engineering Department***

COE 301 COMPUTER ORGANIZATION

**ICS 233: COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE**

**Term 161 (Fall 2016-2017)**

**Major Exam 2**

**Saturday Dec. 10, 2016**

**Time: 150 minutes, Total Pages: 13**

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_\_**

**Notes:**

* Do not open the exam book until instructed
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **20** |  |
| **Q2** | **10** |  |
| **Q3** | **17** |  |
| **Q4** | **23** |  |
| **Total** | **70** |  |

Dr. Aiman El-Maleh

Dr. Marwan Abu Amara

#  **[20 Points]**

# **(Q1)** Write MIPS programs with minimal used instructions. Use MIPS programming convention in saving and restoring registers in procedures.

## **[4 points]** Write a procedure **GetAscii** that receives a single hexadecimal digit in register $a0 and returns the ASCII code of that digit in register $v0.For example, if $a0=0x9 the procedure will return 0x39 in $v0 and if $a0=0xA, the procedure will return 0x41 in $a0. Assume the use of capital letters for the digits A to F.

## **[11 points]** Write a procedure **DispHex** that receives a number in register $a0 and displays the hexadecimal representation of that number. Only significant hexadecimal digits need to be displayed. For example, if $a0=0x1E, the procedure will display 1E. Your DisHex procedure should utilize the GetAscii procedure.

## **[5 points]** Write a MIPS program that asks the user to enter a decimal number and displays its hexadecimal content using the **DispHex** procedure. Two sample runs of the program are given below:

 Enter a decimal number: 260

 Your number in hexadecimal is: 0x104

 Enter a decimal number: 0

 Your number in hexadecimal is: 0x0

 **[10 points]**

**(Q2)**

## **[4 Points]** Given that **Multiplicand=0111** and **Multiplier=1011** are signed 2’s complement numbers, show the **signed** multiplication of **Multiplicand** by **Multiplier**. The result of the multiplication should be an 8 bit **signed** number in HI and LO registers. Show the steps of your work.

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration** | **Multiplicand** | **Sign** | **Product =HI,LO** |
| 0 | Initialize |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |
| 4 |  |  |  |  |
|  |  |  |  |

## **[6 Points]** Given that **Dividend=0111** and **Divisor=1011** are signed 2’s complement numbers, show the **signed** division of **Dividend** by **Divisor**. The result of division should be stored in the Remainder and Quotient registers. Show the steps of your work, and show the final result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration** | **Remainder (HI)** | **Quotient (LO)** | **Divisor** | **Difference** |
| 0 | Initialize |  |  |  |  |
| 1 |  |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |  |
|  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |
| Final Result |  |  |  |  |

 **[17 points]**

**(Q3)**

1. **[2 Points]** Find the **decimal value** of the following single precision float:

 [0, 1000 1000, 0000 0100 1100 0000 0000 000]

1. **[2 Points]** Find the **decimal value** of the following single precision float:

 [1, 0000 0000, 0110 0000 0000 0000 0000 000]

1. **[3 Points]** Find the normalized single precision representation of –59.625.
2. **[4 Points]** Round the given single precision float with the given GRS bits using the following rounding modes showing the resulting normalized number:

 **GRS**

 **+1.111 1111 1111 1111 1111 1111 100 x 2-127**

Zero: [  ]

+infinity: [  ]

-infinity: [  ]

Nearest Even: [  ]

1. **[6 Points]** Find the normalized **difference** between **A** and **B** (i.e., A-B) by using rounding to **+infinity**. Perform the operation using **guard**, **round** and **sticky** bits.

**A = +1.000 0101 1100 1010 1000 0001 × 24**

**B = +1.011 1001 0101 0000 0010 1000 × 2-1**

**[23 Points]**

# **(Q4)** Consider the single-cycle datapath and control given below along with ALU design for the MIPS processor implementing a subset of the instruction set:



##

## Show the control signals generated for the execution of the following instructions by filling the table given below: **(5 points)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Op | RegDst | RegWrite | ExtOp | ALUSrc | ALUOp | Beq | Bne | J | MemRead | MemWrite | MemtoReg |
| R-type |  |  |  |  |  |  |  |  |  |  |  |
| slti |  |  |  |  |  |  |  |  |  |  |  |
| sw |  |  |  |  |  |  |  |  |  |  |  |
| beq |  |  |  |  |  |  |  |  |  |  |  |
| j  |  |  |  |  |  |  |  |  |  |  |  |

## Excluding the ALUOp, Beq, Bne and J signals, show the design of the control unit for the control signals given in the table above based on the given instructions. Assume that the opcode of these instructions is a 6-bit opcode such that the opcode for R-type instructions is 0, the opcode for slti is 1, the opcode for sw is 2, and so on for the rest of the instructions. **(5 points)**

## Show the design of the Next PC block. **(4 points)**

## We wish to add the following instructions to the MIPS single-cycle datapath. Add any necessary datapath modifications and control signals needed for the implementation of these instructions. Show only the **modified** and **added** components to the datapath.

* 1. sra **(3 points)**

|  |  |  |
| --- | --- | --- |
| Instruction | Meaning | Format |
|  sra rd, rt, imm5 |  rd= rt>>imm16 | Op6 = 0 | 0 | rt5 | rd5 | Imm5 | f5=3 |

* 1. jr **(3 points)**

|  |  |  |
| --- | --- | --- |
| Instruction | Meaning | Format |
|  jr rs |  PC=rs | op6 = 0 | rs5 | 0 | 0 | 0 | 8 |

## Assume that the propagation delays for the major components used in the datapath are as follows:

* + - Instruction and data memories: 120 ps
		- ALU and adders: 30 ps
		- Register file access (read or write): 14 ps
		- Main control: 8 ps
		- ALU control: 7 ps

Ignore the delays in the multiplexers, PC access, extension logic, and wires. What is the cycle time for the single-cycle datapath given above? **(3 points)**

**Syscall Services:**



**MIPS Instructions:**

















