**CMPE325 Quiz 2 Sample Questions**

**Q1)** Consider the following code segment:

**addi $a0, $zero, 100**

**Loop: sub $a0, $a0, 1**

**bne $a0, $zero, Loop**

**lw $a1, -4($sp)**

For a single-cycle processor, answer the following:

* How many cycles are needed to execute this code segment? …………………………
* Assuming the following functional units time delays: Instruction and Data Memory (200 ps), ALU and adderss (200 ps), Register file access (reads or writes) (100 ps).
* Calculate the clock frequency. ………………………………
* Calculate the total CPU time ………………………………

**Q2)** It is required to execute the following instruction in the single-cycle datapath:

**addi $29, $29, 16**

**A/** The single-cycle datapath shown below shows the execution of this instruction. Several of the datapath values are filled, and you are requested to provide values for the other remaining signals in the diagram, which are marked with a ? symbol. Write your answers (in decimal) directly on the diagram. Assume register $29 initially contains the number 129, and if a value cannot be determined, mark it as ‘X.’



**B/** Fill the table below the control signal for evaluating bne instruction.

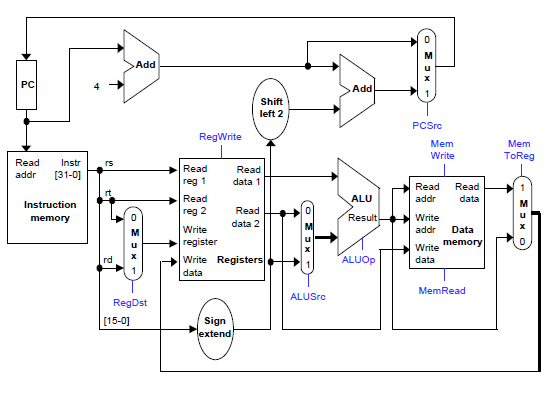
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RegDst** | **RegWrite** | **ALUSrc** | **ALUOp** | **MemWrite** | **MemRead** | **MemToReg** | **PCSrc** |
|  |  |  |  |  |  |  |  |  |

**Q3)** Consider the following Single-Cycle Processor data-path.

**A/** High-light the data-path for executing the instruction

**bne rs, rt, immediate**

**B/** Draw implementation of the logical function of PCSrc for this instruction.



**ZF**

**C/** Fill the table below the control signal for evaluating bne instruction.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RegDst** | **RegWrite** | **ALUSrc** | **ALUOp** | **MemWrite** | **MemRead** | **MemToReg** | **PCSrc** |
|  |  |  |  |  |  |  |  |  |

**Q4)** Consider the single-cycle data-path shown below. Assume that we wish to add the following new instruction **jm** (jump memory) to this data-path.

**jm offset($rs)**

The **jm** instruction loads a word from effective address (**$rs + offset**), this is similar to **lw** except the loaded word is put in the **PC** instead of register **$rt**.

1. Add any necessary data-path(s) and justify the need for the modification(s).
2. High-light the data-path for executing this instruction.

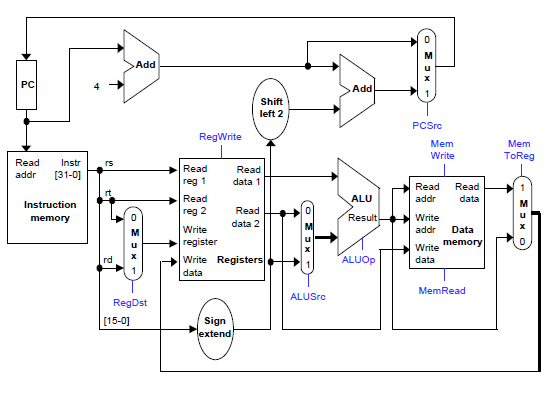
c) In the given table below, provide the corresponding control signals to support the **jm** instruction.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RegDst** | **RegWrite** | **ALUSrc** | **ALUOp** | **MemWrite** | **MemRead** | **MemToReg** | **PCSrc** |
|  |  |  |  |  |  |  |  |

**Q5)** In a single-cycle processor, assume that it is required to simplify the MIPS instruction set architecture by removing the original lw and sw instructions and replacing them with ones that do not contain a constant offset. The new loads and stores will have the following general forms.

lw rt, rs # rt = Mem[rs]

sw rt, rs # Mem[rs] = rt

1. Show what changes must be made to the single-cycle datapath shown below to implement the new lw and sw instructions without using the ALU. 
2. Fill in the table below the correct control signals for the new lw and sw instructions. Use ‘X’ to indicate any don’t-care conditions (if necessary).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RegDst | RegWrite | ALUSrc | ALUOp | MemWrite | MemRead | MemToReg | PCSrc |
| sw |  |  |  |  |  |  |  |  |
| lw |  |  |  |  |  |  |  |  |

1. Assume that memories and the ALU have 2ns delays, and the registers have a 1ns delay. Find the minimum clock cycle times for both the original single-cycle datapath and the new modified one (without using the ALU for sw/lw instructions). Show your work.

Original: ……………………………

Modified: ……………………………

1. Based on the new modified processor, how to implement the instruction

lw $t0, 4($sp) ……………………………

……………………………

In this case, what would be the required CPU time for executing lw $t0, 4($sp)?

CPU time: ……………………………

Comment on the result: ……………………………

**Q6)**  Assume that we want to add the following instruction to the MIPS single cycle datapath:

**lwd rd, rs, rt # rd = Mem[rt + rs]**

1. Highlight the complete datapath shown below for executing this instruction



1. Fill in the table below the correct control signals for executing **lwd** instruction. Use ‘X’ to indicate any don’t-care conditions (if necessary).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RegDst** | **RegWrite** | **ALUSrc** | **ALUOp** | **MemWrite** | **MemRead** | **MemToReg** | **PCSrc** |
|  |  |  |  |  |  |  |  |  |

**Q7)** Assume that it is required to add the following instruction

**Loop r1, r2, offset**

to the single-cycle data-path shown below. Let this instruction be of I-type with the format of

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | op | rs (r2) | rt (r1) | offset |
| Bits | 31-26 | 25-21 | 20-16 | 15-0 |

This is a branch instruction that increments register r1 by 1, and compares it to r2. If these two values are not equal then the PC is set to PC + offset<<2. The instruction has the same effect as sequentially executing the following two instructions on the MIPS architecture:

**addi r1, r1, 1**

**bne r1, r2, offset**

**A/ [15pts]** Draw the additional parts of the datapath (if any) to execute the loop instruction. **Note:** In the single-cycle processor design, it is not allowed to use any component more than once during the instruction execution. Instead, it is allowed to duplicate the components.



**B/** Provide the control signals to support this instruction in the following table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RegDst** | **RegWrite** | **ALUSrc** | **ALUOp** | **MemWrite** | **MemRead** | **MemToReg** | **Loop** |
|  |  |  |  |  |  |  | 1 |