**Name: KEY** **Id#**

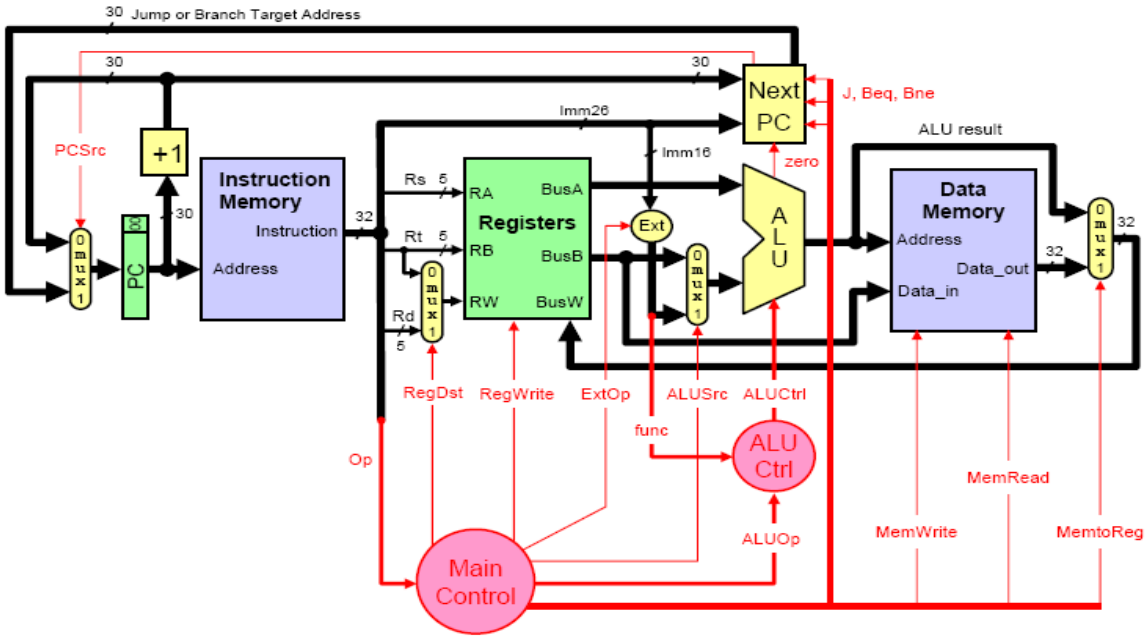
# ICS 233, Term 141

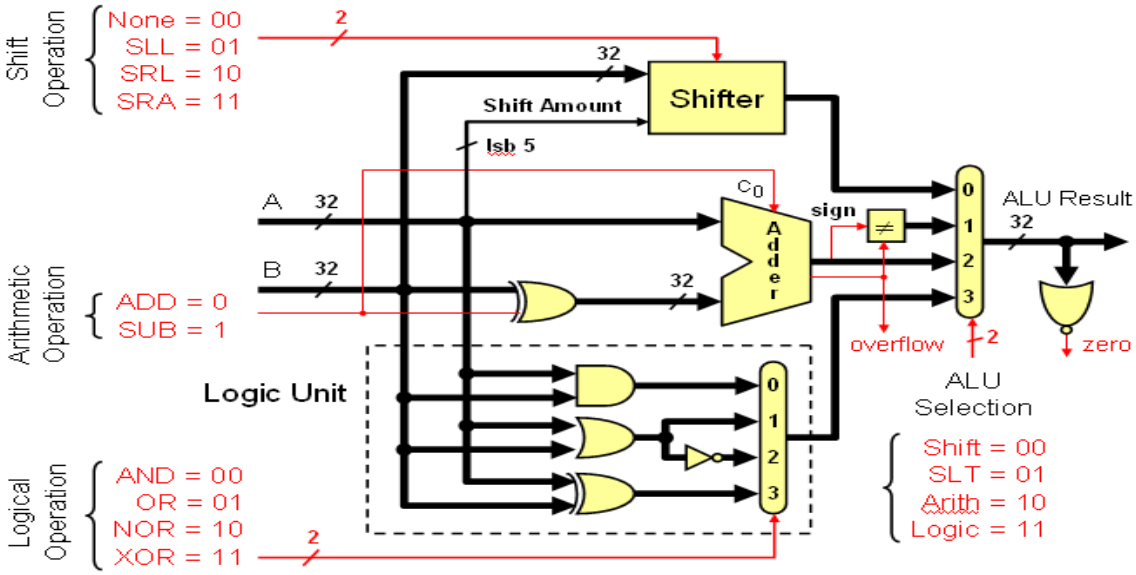
## Computer Architecture & Assembly Language

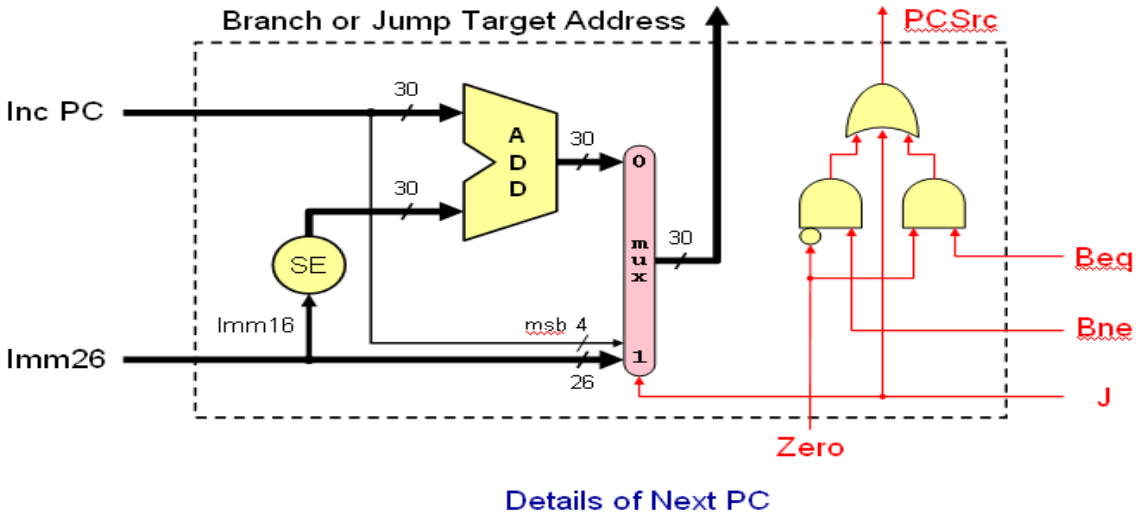
## Quiz# 6

Date: Monday, Dec. 22, 2014

**Q1.** Consider the single-cycle datapath and control given below along with ALU and Next PC blocks design for the MIPS processor implementing a subset of the instruction set:






Details of Next PC

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

| Op | RegDst | RegWrite | ExtOp | ALUSrc | ALUOp | Beq | Bne | J | MemRead | MemWrite | MemtoReg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R-type | 1 = Rd | 1 | x | 0=BusB | R-type | 0 | 0 | 0 | 0 | 0 | 0 |
| xori | 0 = Rt | 1 | 0=zero | 1=Imm | XOR | 0 | 0 | 0 | 0 | 0 | 0 |
| lw | 0 = Rt | 1 | 1=sign | 1=Imm | ADD | 0 | 0 | 0 | 1 | 0 | 1 |
| bne | x | 0 | x | 0=BusB | SUB | 0 | 1 | 0 | 0 | 0 | x |

The format of these instructions is given below for your reference:

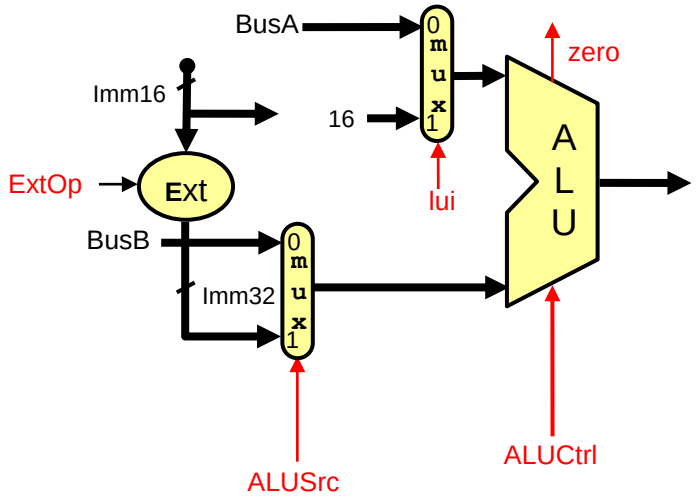
| Instruction | | Meaning | Format | | | | | |
|---|---|---|---|---|---|---|---|---|
| sub | rd, rs, rt | rd = rs - rt | $Op^6 = 0$ | $rs^5$ | $rt^5$ | $rd^5$ | 0 | 0x22 |
| xori | rt, rs, $imm^{16}$ | rt = rs ^ $imm^{16}$ | 0x0e | $rs^5$ | $rt^5$ | $imm^{16}$ | | |
| lw | rt, $imm^{16}$(rs) | rt = MEM[rs+imm16] | 0x23 | $rs^5$ | $rt^5$ | $imm^{16}$ | | |
| bne | rs, rt, label | branch if (rs != rt) | 0x05 | $rs^5$ | $rt^5$ | $imm^{16}$ | | |

**(ii)** 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 **<u>modified</u>** and **<u>added</u>** components to the datapath. Show the values of the control signals to control the execution of each instruction.

**a.** lui

| Instruction | | Meaning | Format | | | |
|---|---|---|---|---|---|---|
| lui | rt, $imm^{16}$ | rt= $imm^{16}$ << 16 | $Op^6$ = 0xf | 0 | $rt^5$ | $imm^{16}$ |

For this instruction, the shift amount is 16 and the operand to be shifted is the immediate value selected on the B-input of the ALU. Thus, we need to add another MUX to select the shift amount as 16 for this instruction. The modified parts of the datapath to support the execution of this instruction is given below:



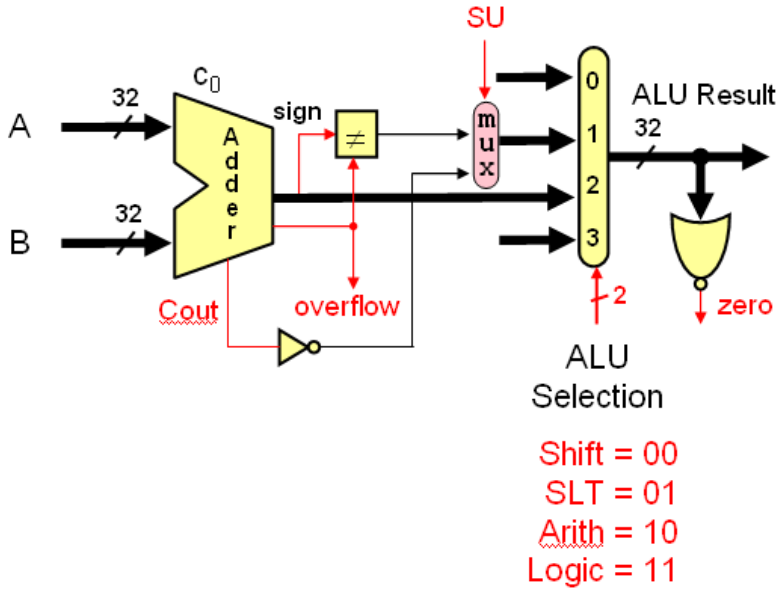

The values of the control signals to control the execution of this instruction are given below:

| Op | lui | RegDst | RegWrite | ExtOp | ALUSrc | ALUOp | Beq | Bne | J | MemRead | MemWrite | MemtoReg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| lui | 1 | 0 = Rt | 1 | x | 1=Imm | SLL | 0 | 0 | 0 | 0 | 0 | 0 |

**b.** sltiu

| Instruction | Meaning | Format | | | |
|---|---|---|---|---|---|
| sltiu rt, rs, $imm^{16}$ | rt=(rs<imm?1:0) | $Op^6$ = 0xb | $rs^5$ | $rt^5$ | $imm^{16}$ |

To execute this instruction the ALU needs to be modified such that when there is a borrow i.e. carry out = 0 then the result is 1. We can add a MUX to select between SLT and SLTU instructions. This MUX is controlled by the signal SU. When SU=1, it will assume it signed comparison otherwise it will assume it unsigned. The modified ALU is shown below:




The values of the control signals to control the execution of this instruction are given below:

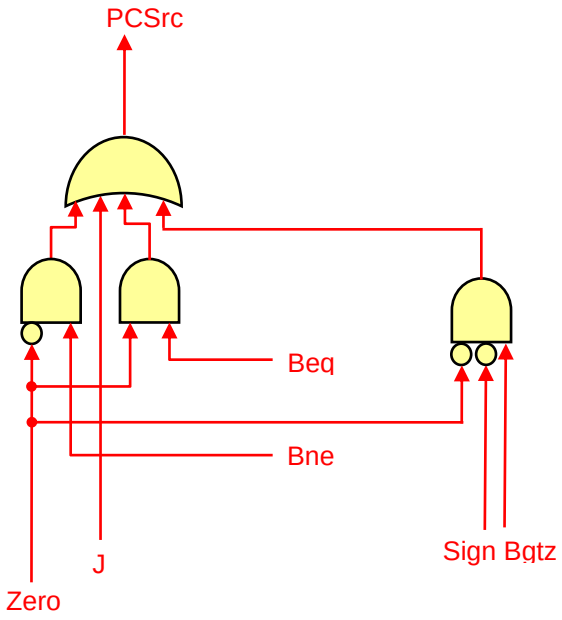
| Op | SU | RegDst | RegWrite | ExtOp | ALUSrc | ALUOp | Beq | Bne | J | MemRead | MemWrite | MemtoReg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sltiu | 0 | 0 = Rt | 1 | 0 | 1=Imm | SLT | 0 | 0 | 0 | 0 | 0 | 0 |

**c.** bgtz

| Instruction | Meaning | Format | | | |
|---|---|---|---|---|---|
| bgtz rs, label | branch if (rs>0) | $Op^6 = 7$ | $rs^5$ | 0 | $imm^{16}$ |

Since the first source operand specified by RS comes on BusA and the second operand which is the Zero register specified by the RT filed comes on BusB, all we need is to get the operand on BusA to appear at the output of the ALU as we just need to check the sign bit (i.e. most significant bit of the result). Performing an addition, subtraction, xoring, oring operations will work. Let us assume that we will do an ALU addition operation.

To check that the result is greater than 0, we need to check that the sign bit is 0 and that the result is not equal to zero. Thus, the changes needed to be done are in the NextPC block as shown below:




The values of the control signals to control the execution of this instruction are given below:

| Op | RegDst | RegWrite | ExtOp | ALUSrc | ALUOp | Bgtz | Beq | Bne | J | MemRead | MemWrite | MemtoReg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bgtz | x | 0 | x | 0= BusB | ADD | 1 | 0 | 0 | 0 | 0 | 0 | x |

**(iii)** Assume that the propagation delays for the major components used in the datapath are as follows:

- Instruction and data memories: 150 ps
- ALU and adders: 100 ps
- Register file access (read or write): 60 ps
- Main control: 20 ps
- ALU control: 20 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?

Cycle Time = IM + max(Main Control+ALU Control, Register Reading) + ALU + DM + Register Writing
= 150 ps + 60 ps + 100 ps + 150 + 60 ps = 520 ps