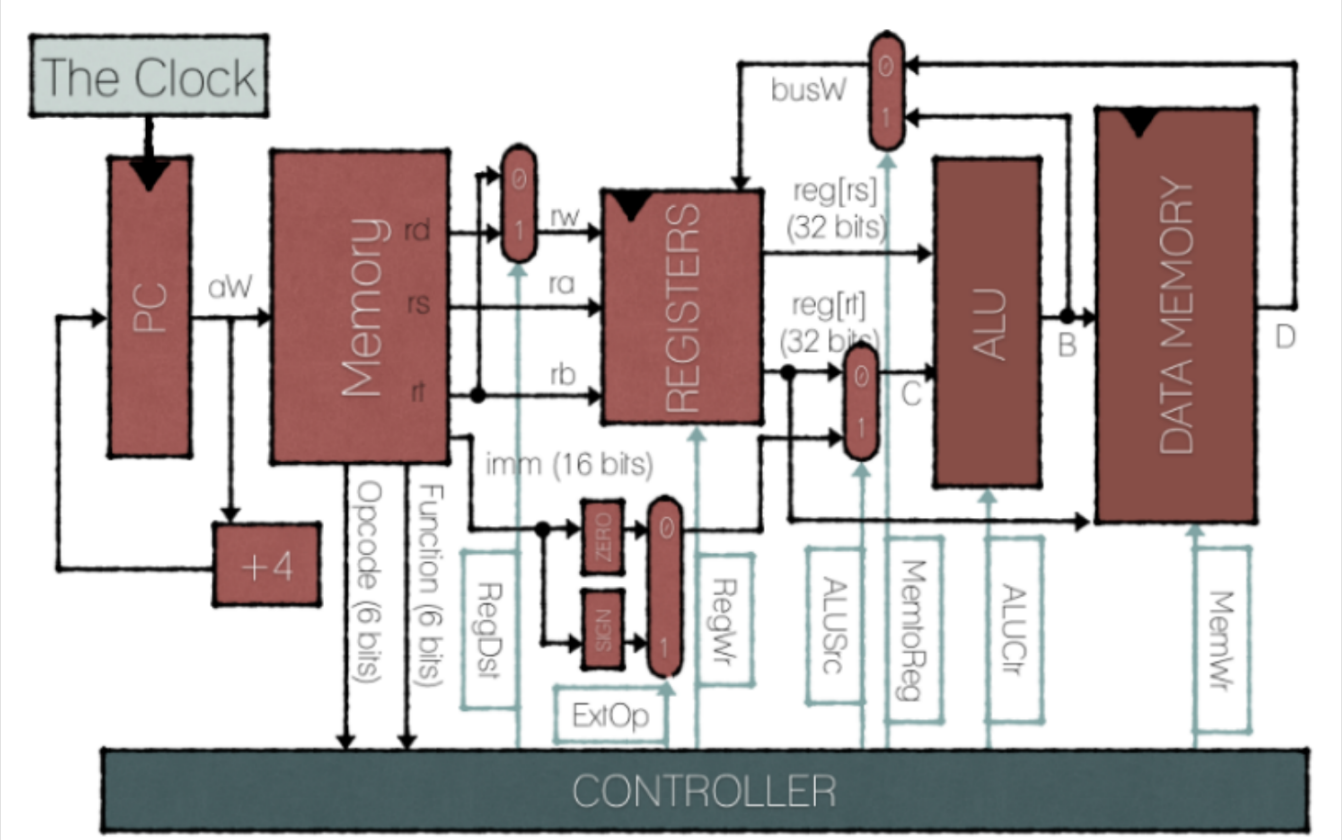
According to the diagram and table below:




| Operation | AND | OR | ADD | SUB |
|---|---|---|---|---|
| ALUCtr | 00 | 01 | 10 | 11 |

List the values for 'RegDst', 'ExtOp' ,'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'add' instruction.

RegDst = 1, ExtOp = X, ALUCtr = 10, MemWr = 0, MemtoReg = 1, RegWr = 1

According to the diagram and table shown in the first question. List the values for 'RegDst', 'ExtOp' ,'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'ori' instruction.

RegDst = 0, ExtOp = 0, ALUCtr = 01, MemWr = 0, MemtoReg = 1, RegWr = 1

According to the diagram and table shown in the first question. List the values for 'RegDst', 'ExtOp' ,'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'lw' instruction.

RegDst = 0, ExtOp = 1, ALUCtr = 10, MemWr = 0, MemtoReg = 0, RegWr = 1

According to the diagram and table shown in the first question. List the values for 'RegDst', 'ExtOp' ,'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'sw' instruction.

RegDst = X, ExtOp = 1, ALUCtr = 10, MemWr = 1, MemtoReg = X, RegWr = 0

According to the diagram and table shown in the first question. List the values for 'RegDst', 'ExtOp', 'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'sub' instruction.

RegDst = 1, ExtOp = X, ALUCtr = 11, MemWr = 0, MemtoReg = 1, RegWr = 1

According to the diagram and table shown in the first question. List the values for 'RegDst', 'ExtOp', 'ALUCtr', 'MemWr', 'MemtoReg' and 'RegWr' for the 'and' instruction.

RegDst = 1, ExtOp = X, ALUCtr = 00, MemWr = 0, MemtoReg = 1, RegWr = 1

Assume the times for CPU stages are as follows:

| Instruction fetch | Register Read / Instruction Decode | ALU | Memory Access | Register Write |
|---|---|---|---|---|
| 166ps | 82ps | 248ps | 201ps | 111ps |

a) Without pipelining, what's the maximum speed (Hertz) we can clock this CPU (according to the numbers above)?

b) With pipelining, what's the maximum speed (Hertz) we can clock this CPU (according to the numbers above)?

Without Pipelining = 1.24 GHz, With Pipelining = 4.03 GHz

Reorder the following sets of instructions to account for the branch delay slot. Only use a 'nop' if you must

```
addi $t0, $t1, 5
ori $t2, $t3, -20
beq $t0, $s0, label
lw $t4, 0($t0)
```

```
addi $t0, $t1, 5
beq $t0, $s0, label
ori $t2, $t3, -20
lw $t4, 0($t0)
```

Reorder the following sets of instructions to account for the branch delay slot. Only use a 'nop' if you must

```
addi $t0, $t1, 5
ori $t2, $t3, -20
beq $t0, $t2, label
lw $t4, 0($t0)
```

```
addi $t0, $t1, 5
ori $t2, $t3, -20
beq $t0, $t2, label
add $zero, $zero, $zero
lw $t4, 0($t0)
```

Using the circuit shown in the first question, and according to the following delays for the circuit elements:

| Element | Register CLK-to-Q | Register SETUP | MUX | ALU | (Instruction or Data) Memory Read | Memory Write | Register -file Read |
|---|---|---|---|---|---|---|---|
| Delays (PS) | 30 | 20 | 25 | 200 | 250 | 200 | 150 |

a) What instruction (store, add, load, sub, ...) exercises the critical path?

b) What is the duration (in pico seconds) for the critical path?

Critical path instruction = load word. Critical path duration = 950 ps