CSE 490/590 Computer Architecture
Quiz 1
Friday, 2/11/11

DIRECTIONS

- Time limit: 45 minutes (12pm - 12:45pm)
- There are 20 points plus 5 bonus points.
- This is a closed-book, no calculator, closed-notes exam.
- Each problem starts on a new page.
- Please use a pen, not a pencil. If you use a pencil, it won't be considered for regrading.
- Each problem also explains its grading criteria. "Atomic" means that you get either all the points or no point, i.e., there are no partial points.

Name:	
UBITName:	

Problem #	Score
1	
2	
3	
4	
5	
6	
7	

1. List the two types of locality and explain what each of them means.
   (**Grading**: total 2 points; 1 point for each type; no point without a correct explanation)

   Answer:

   Temporal locality: if an item is referenced, it tends to be referenced again in the near future. (key point: “tendency” & the same item)

   Spatial locality: if an item is referenced, items whose addresses are close by tend to be referenced in the near future. (key point: “tendency” & items nearby)

2. Suppose that you currently have a computer system with the following two characteristics:

   - There are only three components that determine the overall performance — CPU, memory, and disk.
   - For any given computation, the system spends 30% of the time for CPU, 40% of the time for memory, and 30% of the time for disk.

   Now suppose that you want to purchase a new system that improves the performance in the following ways:

   - The memory performance is 4 times faster over your current system, i.e., the new system spends $\frac{1}{4}$ of the memory time compared to your current system.
   - The disk performance is 5 times faster over your current system, i.e., the new system spends $\frac{1}{5}$ of the disk time compared to your current system.
   - Everything else is the same.

   Using Amdahl's Law, calculate the overall speedup of the new system over your current system. (Note: give the answer in fraction form; no need for arithmetics)
   (**Grading:** atomic 2 points)

   Answer:

   $\frac{100}{46}$

3. Suppose that you have a CPU with the following characteristics:

   - 10 cycles per instruction (same for every instruction)
   - 1 ms per cycle

   Using the processor performance equation (i.e., the Iron Law), calculate the total CPU time when you execute a program that has 100 instructions.
   (**Grading:** atomic 2 points)

   Answer:

   1 sec

4. Consider the following sequence of instructions:

   (Note: #X is the integer X, e.g., #100 is the integer 100. Assume that jumps and branches all decide the next PC by using a jump label, which contains the absolute address of an instruction.)

Jump Label	Instruction Address	Instruction
	100	ADD R1, R2, R3
TARGET1:	104	ADDI R2, R3, #30
	108	J *TARGET2*
	112	BEQZ R1, *TARGET1*
TARGET2:	116	ADD R3, R1, R2

   The initial values are:

Register Name	Value
R1	#100
R2	#30
R3	#50

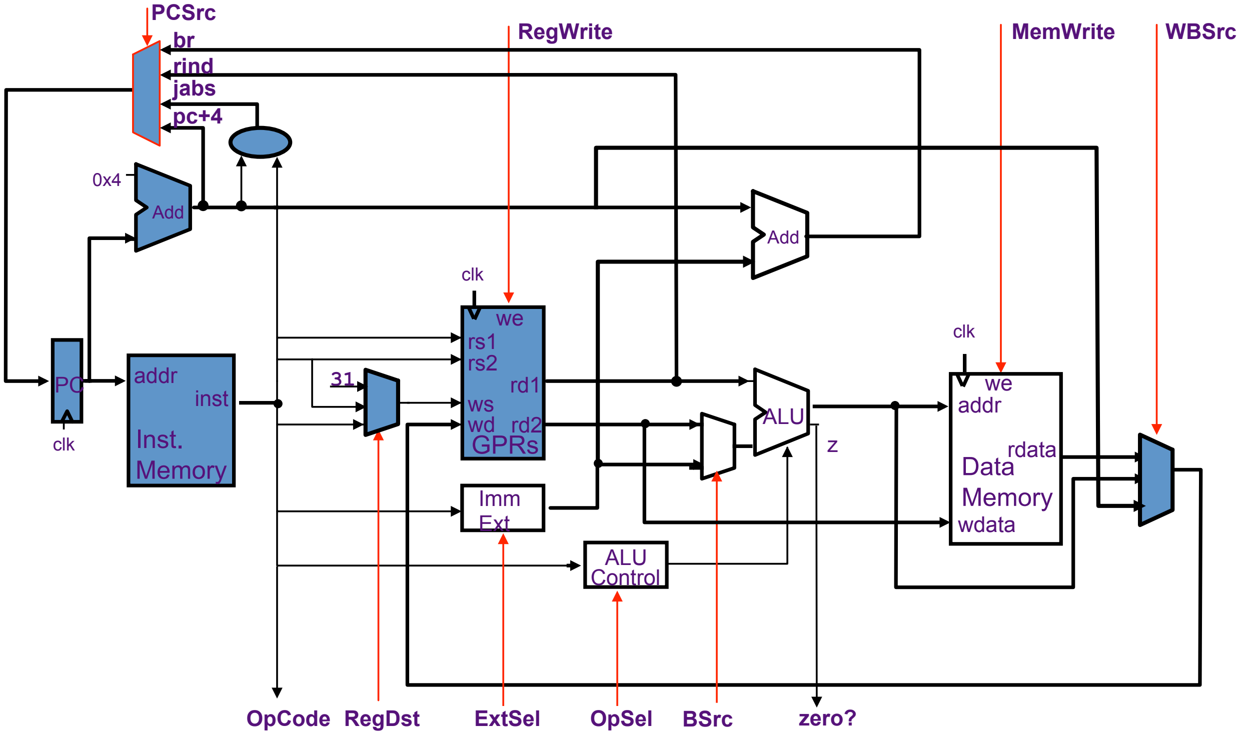
   What is the final value for each of R1, R2, and R3? You can use the following table for your answers. (**Grading:** total 3 points; 1 point for each correct register)

Register Name	Value
R1	#80
R2	#80
R3	#160

   Answer:

   Added above
   (Note: it is OK not to use the format #X.)

5. The following diagram shows the *single cycle* MIPS datapath with no support for pipelining:



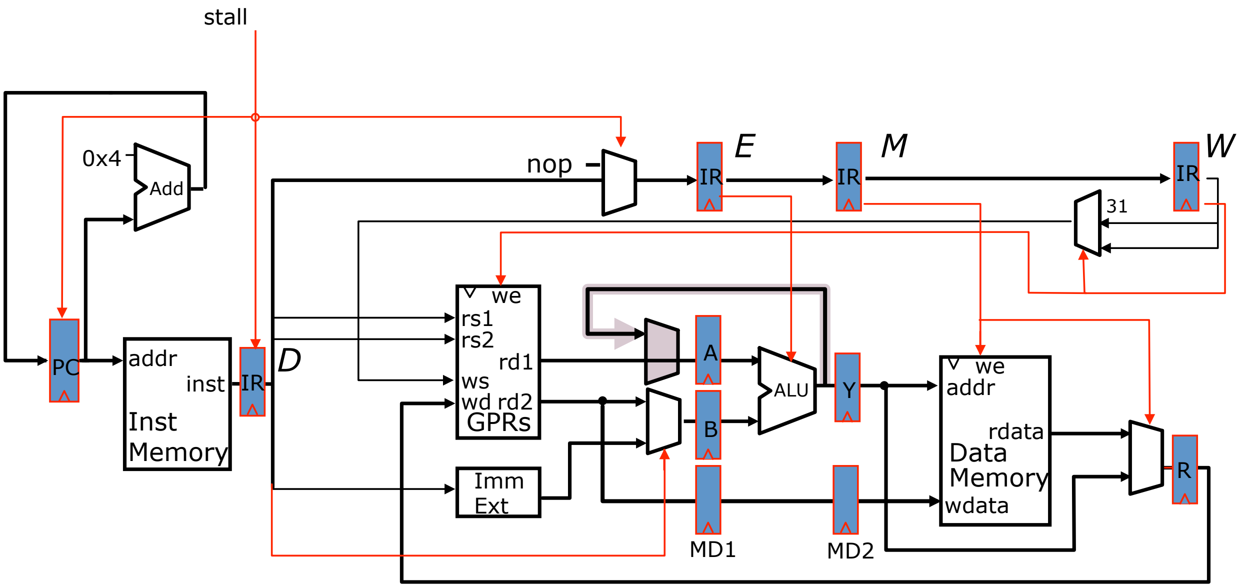
Highlight the components that are necessary for executing the instruction JAL. Components only include the shapes in the diagram *except the wires*, meaning, *do not* highlight the wires. (Hint: there are 8 total components necessary for JAL.)

(**Grading**: total 8 points; 1 point for each correct component)

Answer:

Highlighted above

6. The following diagram shows the *pipelined* MIPS datapath without considering jump and branch instructions:



Add a bypass (hint: a wire and a component) to eliminate pipeline bubbles in the following sequence of instructions:

Instruction Address	Instruction
100	ADDI R1, R0, #20
104	ADDI R4, R1, #80

(**Grading:** atomic 3 points)

Answer: Added above

7. Consider the following sequence of simplified instructions:

Instruction Address	Instruction
100	ADD
104	BEQZ +200
108	ADD
112	ADD
308	ADD

Assume the following:

- There are 5 pipeline stages — IF (Instruction Fetch), ID (Instruction Decode), EX (Execution), MA (Memory Access), WB (Write Back).
- The second instruction, "BEQZ +200", takes the branch and jumps to the instruction at the address 308.
- A branch instruction can determine the next PC only at the EX stage.
- The CPU always speculates that it will execute the instruction at (PC + 4) next.
- If a branch or jump needs to be taken, all instructions in the pipeline are killed.

Complete the time table below with the correct stages at each cycle for all instructions. Use *nop* for pipeline bubbles.

Instruction Address	Instruction	Time								
		t0	t1	t2	t3	t4	t5	t6	t7	t8
100	ADD	IF	ID	EX	MA	WB				
104	BEQZ +200		IF	ID	EX	MA	WB			
108	ADD			IF	ID	nop	nop	nop		
112	ADD				IF	nop	nop	nop	nop	
308	ADD					IF	ID	EX	MA	WB

(**Grading:** total 5 points; 1 point for each instruction with correct stages)

Answer:

Added above