CS 3330 Computer Architecture, Spring 2020
HW 2: ISA Tradeoffs and Single Cycle

Instructor: Prof. Samira Khan
TAs: Amel Fatima, Sihang Liu, Korakit Seemakhupt, Yasas Senevirathne, Yizhou Wei
Min Jae Lee, Nikita Semichev, Yuying Zhang.

Assigned: Feb 7, 2020
Due: **Feb 14, 2020**

1 Performance Evaluation [10 points]

Your job is to evaluate the potential performance of two processors, each implementing a different ISA. The evaluation is based on its performance on a particular benchmark. On the processor implementing ISA A, the best compiled code for this benchmark performs at the rate of 10IPC. That processor has a 500MHz clock. On the processor implementing ISA B, the best compiled code for this benchmark performs at the rate of 2 IPC. That processor has a 600 MHz clock.

- What is the performance in Millions of Instructions per Second (MIPS) of the processor implementing ISA A? Note that Instructions per Second (IPS) is calculated as $IPC * Frequency$.
- What is the performance in MIPS of the processor implementing ISA B?
- Which is the higher performance processor: A/B/Don't know. Briefly explain your answer.

2 ISA Tradeoff: Addressing Modes [40 points]

We covered the following addressing modes in class:

- Absolute
- Register indirect
- Based (base + displacement)
- Scale indexed (base + index * constant)
- Memory indirect
- Auto increment/decrement (by 1 byte)

Consider the following high-level programs:

- ```
  uint8_t a[150]; // a is allocated in memory
  for (i = 0; i < 150; i++) {
  a[i] = 5; }
  ```
- ```
  int a[150]; // a is allocated in memory
  for (i = 0; i < 150; i++) {
  a[i] = 5; }
  ```
- ```
  int *p; // *p is allocated in memory
  *p = 150;
  ```
- ```
  int **p; // *p and **p are allocated in memory
  **p = 150;
  ```

Assume that in the first two programs, a register contains the address of the start of the array, and in the last two programs, a register contains the value of p.

For each of the above four programs, which of the addressing modes, do you think, would lead to the minimum number of instructions? (Note that no addressing mode fits perfectly. You might require other instructions for address computation.) Explain your answer.

3 ISA Tradeoff: Number of Operands [40 points]

Your task is to compare the memory efficiency of five different styles of instruction sets for the code sequence below. The architecture styles are:

1. A zero-address machine is a stack-based machine where all operations are done using values stored on the operand stack. For this problem, you may assume that its ISA allows the following operations:
   - PUSH M - pushes the value stored at memory location M onto the operand stack.
   - POP M - pops the operand stack and stores the value into memory location M.
   - OP - Pops two values off the operand stack, performs the binary operation OP on the two values, and pushes the result back onto the operand stack. The popped values are NOT stored back to the memory.

   Note: To compute A - B with a stack machine, the following sequence of operations are necessary: PUSH A, PUSH B, SUB. After execution of SUB, A and B would no longer be on the stack, but the value A-B would be at the top of the stack.
2. A one-address machine uses an accumulator in order to perform computations. For this problem, you may assume that its ISA allows the following operations:
   - LOAD M - Loads the value stored at memory location M into the accumulator.
   - STORE M - Stores the value in the accumulator into memory location M.
   - OP M - Performs the binary operation OP on the value stored at memory location M and the value present in the accumulator. The result is stored into the accumulator (ACCUM = ACCUM OP M).
3. A two-address machine takes two sources, performs an operation on these sources and stores the result back into one of the sources. For this problem, you may assume that its ISA allows the following operation:
   - OP M1, M2 - Performs a binary operation OP on the values stored at memory locations M1 and M2 and stores the result back into memory location M1 (M1 = M1 OP M2).
4. A three-address memory-memory machine whose sources and destination are memory locations. For this problem, you may assume that its ISA allows the following operation:
   - OP M3, M1, M2 - Performs a binary operation OP on the values stored at memory locations M1 and M2 and stores the result back into memory location M3 (M3 = M1 OP M2).
5. A three-address load-store machine whose sources and destination are registers. Values are loaded into registers using memory operations (The MIPS is an example of a three-address load-store machine). For this problem, you may assume that its ISA allows the following operations:
   - OP R3, R1, R2 - Performs a binary operation OP on the values stored at registers R1 and R2 and stores the result back into register R3 (R3 = R1 OP R2).
   - LOAD R1, M - Loads the value at memory location M into register R1.
   - STORE R2, M - Stores the value in register R2 into memory location M.

To measure memory efficiency, make the following assumptions about all five instruction sets:

- The opcode is always 1 byte (8 bits).
- All register operands are 1 byte (8 bits).

- All memory addresses are 2 bytes (16 bits).
- All data values are 4 bytes (32 bits).
- All instructions are an integral number of bytes in length.

There are no other optimizations to reduce memory traffic, and the variables A, B, C, and D are initially in memory. You are only allowed to use the following instructions: LOAD, STORE, PUSH, POP, ADD, and SUB. As described above, PUSH and POP are only applicable to stack-based machines. For the two-address machines, only ADD and SUB are given to you for answering the questions.

1. Write the code sequences for the following high-level language fragment for each of the five architecture styles. Be sure to store the contents of A, B, and D back into memory, but do not modify any other values in memory.

```
A = B + C;
B = A + C;
D = A - B;
```

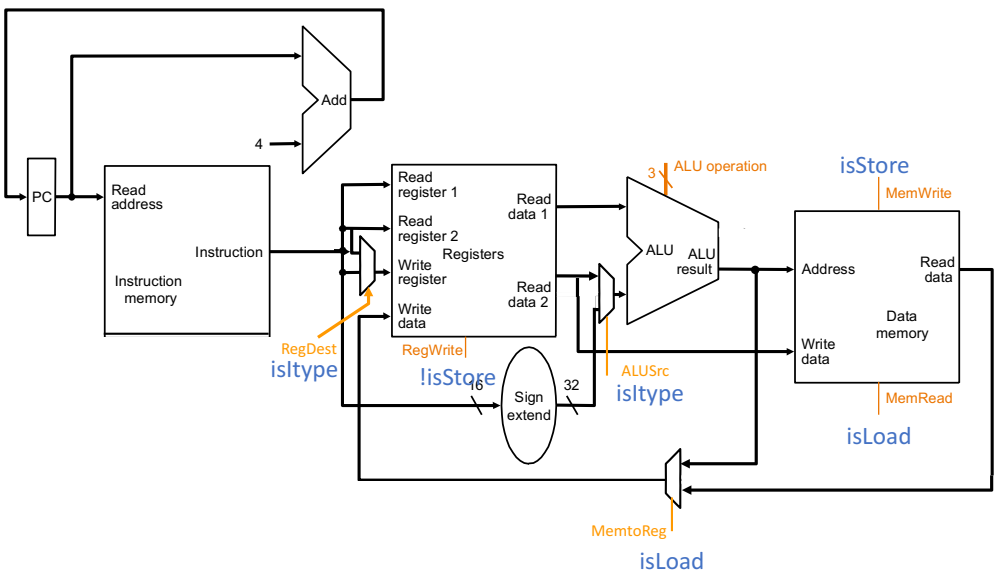
2. Calculate the instruction bytes fetched and the memory-data bytes transferred (read or written) for each of the five architecture styles.
3. Which architecture is most efficient as measured by code size?
4. Which architecture is most efficient as measured by total memory transfer (code+data)?

4 Single-Cycle Processor Datapath [30 points]

In this problem, you will modify the single-cycle datapath we built up in lecture to support the JAL instruction. The datapath that we will start with has been reproduced on the next page. Your job is to implement the necessary data and control signals to support the JAL instruction, which we define to have the following semantics:

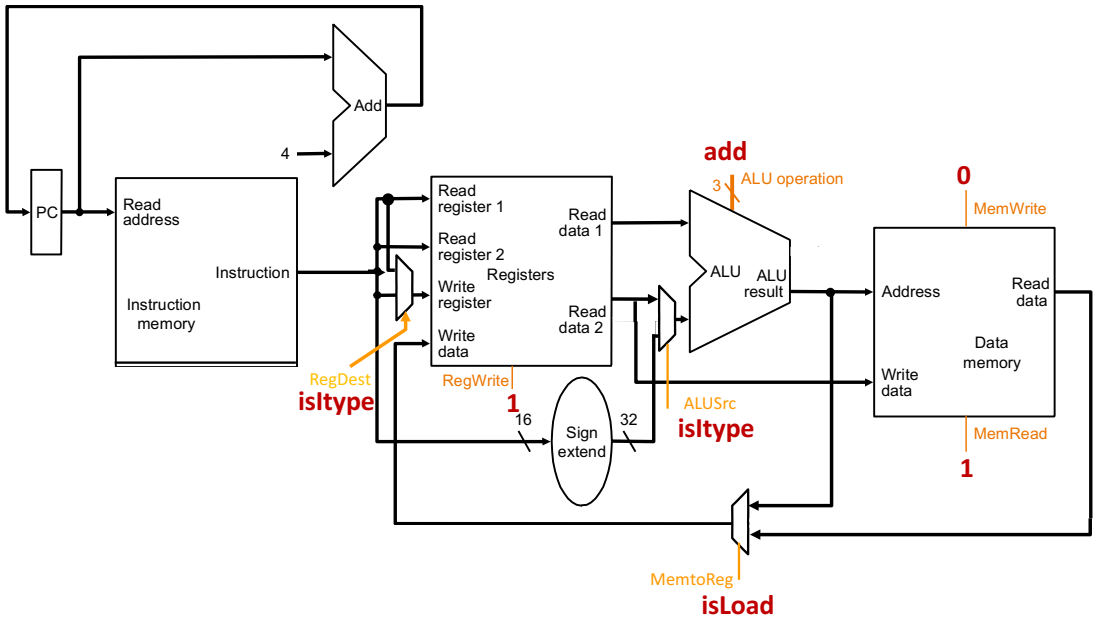
```
JAL : R31 ← PC+4
PC ← PC31...28 || Immediate || 0^2
```

Add to the datapath the necessary data and control signals to implement the JAL instruction. Draw and label all components and wires very clearly. (Hint: Add necessary wires/multiplexers/control signals/logic units to the existing datapath graph.)



5 Bonus: Mysterious Instruction [30 points]

A pesky engineer implemented a mystery instruction on the single cycle machine that we are learning in the class. It is your job to determine what the instruction does and what are the operands. Hint1: The control signals tell you what type of instruction it is. IsItype, MemRead, and RegWrite signals are on. Hint2: Where are the values for the WriteRegister address and data coming from?



6 Handin

You should electronically hand in your homework (in pdf format) to Collab.