Name: _____

CIS 351 Sample SS4 Problem

15 October 2025

SS4: Modify the Single-Cycle CPU

Consider this following code:

```
for(int x = 0; x < 1000; x++) {
  array[x] = x*x;
}
```

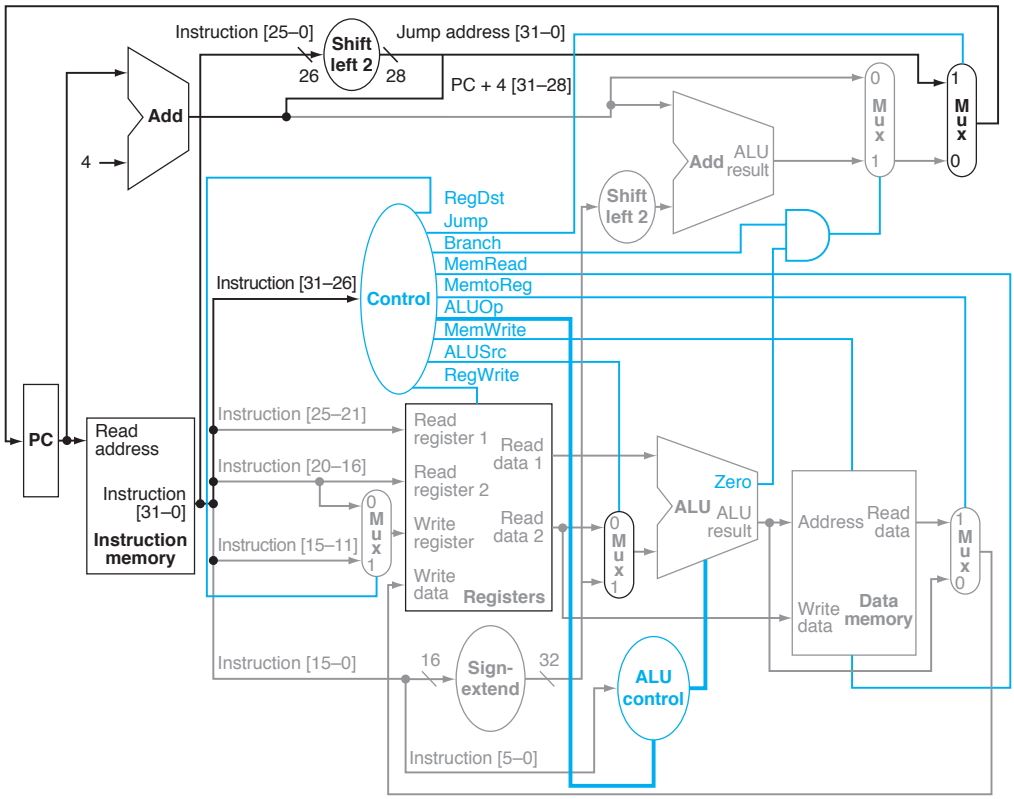
We could save one cycle per loop by combining the storing of data into `array` with the increment of `x` into a new instruction *store word and increment* (`swi R1, R2, increment`). This new instruction will do the following in a single cycle: (1) store the contents of R1 into the memory location contained in R2, and (2) add `increment` to R2 and store the result back in R2. In other words, the instruction does this: `M[R2] = R1; R2 += increment`.

1. Design this instruction. Specify how each bit if the instruction is used. For example, specify which bits contain R1, which contain R2, and which contain the `increment`.

2. Make any necessary changes to the CPU to support this new instruction. You may add additional muxes and control wires if necessary.

3. Specify how each control wire is set for this instruction (including any new control wires you added).

Name: _____



Click here for solutions: `ss4_sample_solutions.pdf`

CIS 351 Sample SS4 Problem Solutions

Sat 1st Nov, 2025

SS4: Modify the Single-Cycle CPU

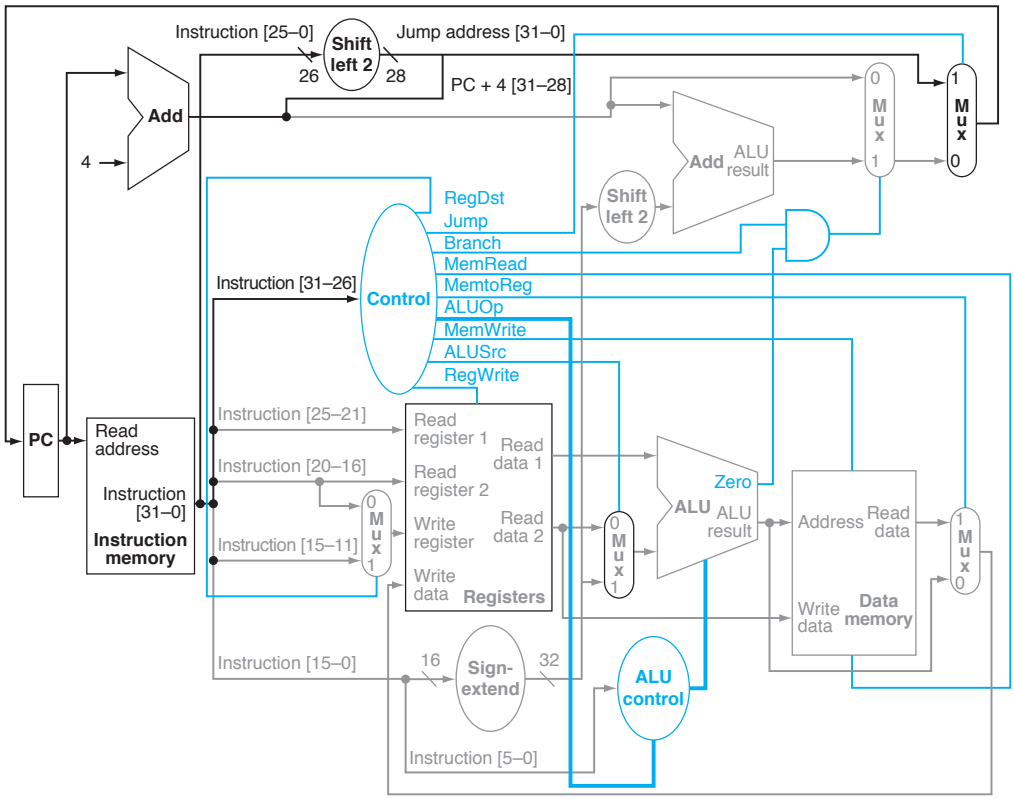
Consider this following code:

```
for(int x = 0; x < 1000; x++) {
  array[x] = x*x;
}
```

We could save one cycle per loop by combining the storing of data into `array` with the increment of `x` into a new instruction *store word and increment* (`swi R1, R2, increment`). This new instruction will do the following in a single cycle: (1) store the contents of R1 into the memory location contained in R2, and (2) add `increment` to R2 and store the result back in R2. In other words, the instruction does this: `M[R2] = R1; R2 += increment`.

1. Design this instruction. Specify how each bit if the instruction is used. For example, specify which bits contain R1, which contain R2, and which contain the `increment`.
2. Make any necessary changes to the CPU to support this new instruction. You may add additional muxes and control wires if necessary.
3. Specify how each control wire is set for this instruction (including any new control wires you added).

I'll check these individually upon request.


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
M u x
0
1
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory