Facultad de Informática
Universidad Complutense de Madrid

INTRODUCTION TO COMPUTERS II

MODULE 5

Basic problems:

1. Assume that the instructions of a program running in a single-cycle RISC-V follow this distribution:

	`add`-like	`addi`-like	`lw`	`sw`	`beq`	`jal`
frequency	24%	28%	25%	10%	11%	2%

Provide:

- The percentage of instructions that use the data memory.
- The percentage of instructions that use the instruction memory.
- The percentage of instructions that use the sign extension module.
- The percentage of instructions that use the ALU.

2. Provide the value of the control signals produced in a single-cycle RISC-V when executing a **`lw`** instruction.

3. Provide the value of the control signals produced in a single-cycle RISC-V when executing an **`andi`** instruction.

4. Provide the value of the control signals produced in a single-cycle RISC-V when executing a **`beq`** instruction.

5. Suppose that a single-cycle RISC-V reads from the address 0x00001000 of the instruction memory the 0x00c6aa23 word, that every processor register xi contains xi (ie. The register x1 contains 0x00000001, x2 contains 0x00000002, etc.) and that every address of the data memory contains 0x00000000. Provide:

- The value of the ALUctr signal generated by the ALU DEC.
- The value of the ImmSrc signal generated by the sExt DEC.
- The new value of the PC after the execution of the given instruction.
- The values of the data inputs and outputs of the multiplexers.
- The values of the register file outputs.
- The values at the inputs of the ALU and the adders.

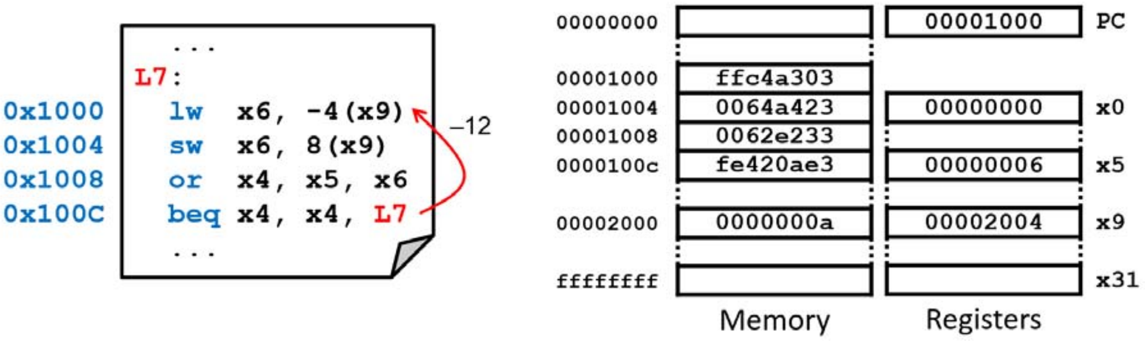
Additional problems:

6. When silicon chips are fabricated, defects in materials and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get damaged and always register a logical 0 (stuck-at-0) or a logical 1 (stuck-at-1). Indicate what single-cycle RISC-V instructions would fail if the following control signals suffered a stuck-at-0 defect:

a) BRWr
b) ALUop$_1$
c) ALUop$_0$
d) MemWr
e) ImmSrc$_1$
f) ImmSrc$_0$
g) ResSrc$_1$
h) ResSrc$_0$
i) PCSrc
j) ALUSrc

7. Repeat the previous exercise, but now with a stuck-at-1 defect.

8. Assume that the following program is running on a single-cycle RISC-V, with the initial value of the memory and registers shown in the picture. Represent the execution diagrams for the registers and the memory, as well as for the control and status signals, so that their values are shown for each clock cycle.



```
            ...
        L7:
0x1000    lw    x6, -4(x9)
0x1004    sw    x6, 8(x9)
0x1008    or    x4, x5, x6
0x100C    beq   x4, x4, L7
            ...
```

	Memory	Registers	
00000000		00001000	PC
00001000	ffc4a303		
00001004	0064a423	00000000	x0
00001008	0062e233		
0000100c	fe420ae3	00000006	x5
00002000	0000000a	00002004	x9
ffffffff			x31

9. Discuss the modifications required in the data path of the reduced architecture single-cycle RISC-V, in order to extend its ISA with each of the following instructions:

a) **`xor`** / **`xori`**
b) **`sll`** / **`slli`**
c) **`bne`** / **`blt`** / **`bge`**
d) **`slt`** / **`slti`**
e) **`lb`** / **`lh`** / **`lbu`** / **`lhu`**
f) **`sb`** / **`sh`**
g) **`jarl`**
h) **`lui`**
i) **`auipc`**

10. Discuss the modifications required in the data path of the reduced architecture single-cycle RISC-V, in order to extend its ISA with a new I-type instruction that reads a word from memory, pre-incrementing the base register, **`lwpreinc`** ***`rd, imm`*****`(`*****`rs1`*****`)`**:

{ rd ← Mem[rs1 + sExt(imm)], rs1 ← rs1 + sExt(imm) }

11. Discuss the modifications required in the data path of the reduced architecture single-cycle RISC-V, in order to extend its ISA with a new I-type instruction that reads a word from memory, post-incrementing the base register, **`lwpostinc`** ***`rd, imm`*****`(`*****`rs1`*****`)`**:

{ rd ← Mem[rs1], rs1 ← rs1 + sExt(imm) }

12. Discuss the modifications required in the data path of the reduced architecture single-cycle RISC-V, in order to extend its ISA with a new R-type instruction that swaps the value of 2 registers, **`swap`** ***`rs1, rs2`***:

{ rs1 ← rs2, rs2 ← rs1 }

13. Discuss the modifications required in the data path of the reduced architecture single-cycle RISC-V, in order to extend its ISA with a new U-type instruction that loads a data read from a certain memory position into a register, **`lwa`** ***`rd, imm`***:

{ rd ← Mem[zExt(imm)] }

14. If you could reduce the delay of just one of the components of the single-cycle RISC-V in the 90nm CMOS library by half, which one would you choose to get a smaller cycle time? What would be the value of the new cycle time?

15. If the ALU delay is reduced a 30%, the extension sign module delay a 10% and the register file read/write delays a 20% (referred to the single-cycle RISC-V with the 90nm

CMOS library), calculate the critical path delay of each instruction and how much time this processor would take to execute 100 million instructions.

16. Suppose that adding a multiplication instruction to the single-cycle RISC-V ALU doubles its delay, but this reduces the count of executed instructions a 5% (because multiplications do not have to be emulated by SW now).

a) Calculate the clock frequency of the new processor.
b) Discuss whether this change makes sense.
c) Calculate the maximum delay increment for the ALU so that this change keeps making sense.
d) Calculate the minimum percentage of the instruction count reduction so that this change makes sense.

17. The **`lw`** instruction is the slowest of the single-cycle RISC-V because it needs to use all the resources of the data path. If the behavior of the **`lw`** and **`sw`** instructions is modified so that the address that is sent to memory is the one stored in the rs1 base register (skipping the addition with the offset), the cycle time can be reduced. However, this forces to replace the old **`lw`** and **`sw`** instructions with **`lw`**/**`addi`** and **`sw`**/**`addi`** pairs, in all the programs. Assuming that the original processor executes a program with 10^8 instructions, following this distribution:

	`add`-like	**`addi`**-like	**`lw`**	**`sw`**	**`beq`**	**`jal`**
frequency	24%	28%	25%	10%	11%	2%

a) Calculate the total number of instructions that the new program will have.
b) Calculate the cycle time of the new processor.
c) Calculate the ratio between the execution times of the program running on both processors (speedup).
d) Discuss whether this change makes sense.

18. Suppose that doubling the number of general purpose registers, from 32 to 64, in the single-cycle RISC-V reduces the number of the executed **`lw`** and **`sw`** instructions a 12%. However, this increases the register file access delay a 25% and its cost a 50%. Assuming that the original processor executes a program with 10^8 instructions, following this distribution:

	`add`-like	**`addi`**-like	**`lw`**	**`sw`**	**`beq`**	**`jal`**
frequency	24%	28%	25%	10%	11%	2%

a) Calculate the total number of instructions that the new program will have.
b) Calculate the cost and the cycle time of the new processor.
c) Calculate the time of execution ratio between both processors.