Introduction to Computers II
Module 7

Basic problems:

1. Given the following RISC-V code fragment and assuming that the initial value of registers **t1** and **t2** is 11 and 22 respectively,

```
addi t1, t2, 5
add  t3, t1, t2
addi t4, t1, 15
```

Provide:

a) The final value of the registers after executing the code in the multicycle RISC-V.
b) The final value of the registers after executing the code in the pipelined RISC-V without hazard management.
c) A modified version of the program, inserting the necessary **nop** instructions so that the previous processor gets a correct result.
d) The final value of the registers after executing the code in the pipelined RISC-V.

2. Given the following RISC-V code fragment and assuming that the initial value of registers **t1** and **t2** is 11 and 22 respectively,

```
addi t1, t2, 5
add  t3, t1, t2
addi t4, t1, 15
add  t5, t1, t1
```

Provide:

a) The final value of the registers after executing the code in the multicycle RISC-V.
b) The final value of the registers after executing the code in the pipelined RISC-V without hazard management.
c) A modified version of the program, inserting the necessary **nop** instructions so that the previous processor gets a correct result.
d) The final value of the registers after executing the code in the pipelined RISC-V without hazard management, but with the capacity to write the Register File in the middle of the cycle and read it in the second half of the cycle.
e) A modified version of the program, inserting the necessary **nop** instructions so that the previous processor gets a correct result.
f) The final value of the registers after executing the code in the pipelined RISC-V.

3. Assuming that the following code is running on the pipelined RISC-V, indicate in which pipeline stage each instruction is during the first 5 execution cycles, as well as the read and written registers of the RF in each of the cycles. Draw the execution diagram of the program.

```
xor  s1, s2, s3
addi s0, s3, -4
lw   s3, 16(s7)
sw   s4, 20(s1)
```

```
or   t2, s0, s1
```

4. Assuming that the following code is running on the pipelined RISC-V, indicate in which pipeline stage each instruction is during the first 7 execution cycles. Draw the execution diagram of the program.

```
addi s1, zero, 11
lw   s2, 25(s0)
add  s3, s3, s4
or   s4, s1, s2
lw   s5, 16(s2)
```

5. Given the following RISC-V code fragment:

```
addi t1, t2, 5
add  t3, t1, t2
addi t4, t1, 15
add  t5, t3, t2
```

Insert the **nop** instructions that are necessary to get a correct result and draw the execution diagram for:

a) The pipelined RISC-V without hazard management.
b) The pipelined RISC-V without hazard management, but with the capacity to write the Register File in the middle of the cycle and read it in the second half of the cycle.
c) The pipelined RISC-V.

6. Given the following RISC-V code fragment:

```
add x7, x5, x8
lw  x6, 8(x7)
lw  x5, 0(x5)
or  x6, x7, x6
sw  x6, 0(x7)
```

Insert the **nop** instructions that are necessary to get a correct result and draw the execution diagram for:

a) The pipelined RISC-V without hazard management.
b) The pipelined RISC-V without hazard management, but with capacity to write the Register File in the middle of the cycle and read it in the second half of the cycle.
c) The pipelined RISC-V

For the last case, simulate the value of the following pipeline registers during the first 7 clock cycles: Rs1E, Rs2E, RdM, BRwrM, RdW y BRwrW, as well as the ForwardA and ForwardB signals.

7. Assuming that the following code is running on the pipelined RISC-V, indicate in which pipeline stage each instruction is during the first 5 execution cycles. Draw the execution diagram of the program.

```
lw   t1, 0(t0)
add  t5, t2, t1
addi t0, t0, 1
sub  t3, t5, t0
```

Repeat the exercise reordering the code to avoid the pipeline stall produced by the data hazard between the first and second instructions.

8. Given the following RISC-V code fragment:

```
lw  t1, 0(t0)
add t5, t2, t1
add t2, t1, t4
```

Insert the **nop** instructions that are necessary to get a correct result and draw the execution diagram for:

a) The pipelined RISC-V without hazard management.
b) The pipelined RISC-V without hazard management, but with the capacity to write the Register File in the middle of the cycle and read it in the second half of the cycle.
c) The pipelined RISC-V with partial hazard management (writing the RF in the middle of the cycle + forwarding unit).
d) The pipelined processor.
e) For the last case, indicate in which pipeline stage each instruction is during the first 5 execution cycles.

9. Assuming that the following code is running on the pipelined RISC-V, indicate in which pipeline stage each instruction is during the first 10 execution cycles. Draw the execution diagram of the program.

```
addi s1, zero, 11
lw   s2, 25(s1)
lw   s5, 16(s2)
add  s3, s2, s5
or   s4, s3, t4
and  s2, s3, s4
```

10. Given the following RISC-V code fragment:

```
addi s1, zero, 52
addi s0, s1, -4
lw   s3, 16(s0)
sw   s3, 20(s0)
xor  s2, s0, s3
or   s2, s2, s3
```

Draw the execution diagram and indicate in which pipeline stage each instruction is during the first 7 execution cycles for:

a) The pipelined processor.
b) The pipelined processor with optimized forwarding, i.e., with the capacity to forward data to the MEM stage from the WB stage, in order to handle **lw** → **sw** data hazards.

11. Given the following RISC-V code fragment:

```
lw  s3, 0(s4)
add s5, s4, s3
add s2, s3, s6
and s1, s1, s2
lw  s5, 0(t3)
sw  s5, 0(t3)
or  s2, s2, s5
```

Indicate the execution diagram for:

a) The pipelined processor.
b) The pipelined processor with optimized forwarding, i.e., with the capacity to

forward data to the MEM stage from the WB stage, in order to handle **lw** → **sw** data hazards.

12. Given the following RISC-V code fragment and assuming that the initial value of registers **t1** and **t2** is 11 and 22 respectively,

```
    beq  zero, s0, L1
    addi t1, t1, -1
    addi t2, t1, -1
L1:
    addi t3, t1, 1
    addi t4, t2, 1
```

Assuming that **s0** contains a value different from 0, calculate:

a) The final value of the registers after executing the code in the multicycle RISC-V.
b) The final value of the registers after executing the code in the pipelined RISC-V without control hazard management.
c) Insert the necessary **nop** instructions so that the previous processor gets a correct result. Draw the resulting execution diagram.
d) Draw the execution diagram and indicate in which pipeline stage each instruction is during the first 7 execution cycles, considering the pipelined RISC-V with control hazard management through pipeline stalling.
e) Draw the execution diagram and indicate in which pipeline stage each instruction is during the first 7 execution cycles, considering the pipelined RISC-V with control hazard management through not-taken branch prediction.

Assuming that **s0** contains a 0, calculate:

f) The final value of the registers after executing the code in the multicycle RISC-V.
g) The final value of the registers after executing the code in the pipelined RISC-V without control hazard management.
h) Insert the necessary **nop** instructions so that the previous processor gets a correct result. Draw the resulting execution diagram.
i) Draw the execution diagram and indicate in which pipeline stage each instruction is during the first 7 execution cycles, considering the pipelined RISC-V with control hazard management through pipeline stalling.
j) Draw the execution diagram and indicate in which pipeline stage each instruction is during the first 7 execution cycles, considering the pipelined RISC-V with control hazard management through not-taken branch prediction.

13. Assuming that the following code is running on the pipelined RISC-V, indicate in which pipeline stage each instruction is during the first 8 execution cycles, as well as the read and written registers of the RF in each of the cycles. Draw the execution diagram of the program.

```
    jal  x0, L1
    addi t1, x0, 5
    add  t3, t1, t2
L1:
    sw   t4, 0(t3)
```

14. Assuming that the initial value of register **t0** is 1, draw the execution diagram of the following code fragment running on the pipelined RISC-V:

```
    addi s0, x0, 0
L1:
    beq  t0, x0, L2
    add  s0, s0, t0
    addi t0, t0, -1
    jal  x0, L1
L2:
    add  s0, s0, s2
    sw   s0, 0(gp)
```

15. Assuming that the second instruction loads a 1 into register **t0**, draw the execution diagram of the following code fragment running on the pipelined RISC-V

```
    addi s0, x0, 0
    lw   t0, 4(gp)
L1:
    beq  t0, x0, L2
    add  s0, s0, t0
    addi t0, t0, -1
    jal  x0, L1
L2:
    lw   s1, 0(gp)
    add  s1, s0, s1
    sw   s1, 0(gp)
```

16. Given the following RISC-V code fragment:

```
    // for( sum=0, i=0; i!=N; i++ )
    //    sum = sum + a[i];
    //
    // a[]->s0, N->s1, i->s2, sum->s3
    //
    .equ N, ...
    ...
    addi s1, zero, N
    addi s2, zero, 0
    addi s3, zero, 0
for:
    beq  s2, s1, efor
    slli t0, s2, 2
    add  t0, s0, t0
    lw   t0, 0(t0)
    add  s3, s3, t0
    addi s2, s2, 1
    jal  x0, for
efor:
    sw   s3, 0(gp)
    ...
```

Assuming that it runs on the pipelined RISC-V, calculate:

a) The number of executed instructions.
b) The number of cycles needed to execute them.
c) The CPI.

If the code is reordered, could the execution time be smaller? If so, propose a more efficient code fragment and calculate the new CPI.

17. Assume that a RISC-V processor is running a program with 500 instructions, distributed as follows:

- 20% are **`lw`** instructions, half of which are followed by an arithmetic instruction that reads the register previously written by the **`lw`** instruction.
- 15% are **`sw`** instructions.
- 25% are **`beq`** instructions, being 70% of them taken branches.
- 5% are **`jal`** instructions.
- 35% are arithmetic logic instructions.

If the processor works with a 1.5 GHz frequency, calculate the CPI and the execution time of the program.

Additional problems:

18. Given the following RISC-V code fragment:

```
    addi s4, zero, 2
L1:
    lw   s0, 0(s2)
    lw   s1, 0(s2)
    add  s3, s0, s1
    sw   s3, 0(s2)
    add  s2, s3, s6
    and  s1, s1, s2
    beq  s4, zero, L1
    addi s2, s2, 1
    or   s3, s2, s4
```

Indicate the execution diagram for:

a) The pipelined processor.
b) A pipelined processor that solves all hazards by stalling the pipeline and with the capacity to write the RF in the middle of the cycle.

19. Assuming that the following code fragment runs on the pipelined RISC-V, draw the execution diagram when the **`beq`** instruction is taken and when it is not taken.

```
    lw  t1, 0(s1)
    lw  t2, 0(s2)
    beq t1, t2, else
    add t3, t1, t2
    jal zero, eif:
else:
    sub t3, t1, t2
eif:
    and t4, t3, t1
    or  t5, t3, t2
```

20. Given the followint RISC-V code fragement:

```
      addi s1, zero,1
L1:
     sub  s5, s5, s2
     addi s2, s2, 1
     addi s1, s1, -1
     beq  s1, zero, L1
     and  s5, s2, s3
     or   s3, s3, s4
     andi s4, s3, s2
```

Indicate the execution diagram for the first 2 iterations.

21. Discuss the modifications required in the data path of the pipelined RISC-V, in order to extend its ISA with a new R-type instruction that reads a word from memory using a base register and a variable offset stored in an index register, `lwi` ***`rd, rs1, rs2`***:

$$\{ \text{rd} \leftarrow \text{Mem[rs1+ rs2]} \}$$

Analyze the potential hazards and how to solve them.

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

$$\{ \text{rs1} \leftarrow \text{rs2}, \text{rs2} \leftarrow \text{rs1} \}$$

Analyze the potential hazards and how to solve them.

23. If you could reduce the delay of just one of the components of the pipelined RISC-V in the 90nm CMOS library, which one would you choose to get a smaller cycle time? How much reduction would it make sense? What would be the value of the new cycle time?

24. If you could divide one of the RISC-V stages in two with the same delay, which one would you choose to get a smaller cycle time? What would be the value of the new cycle time?

25. 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 pipelined RISC-V with the 90nm CMOS library), calculate:

a) The delay of the critical path in each pipeline stage.
b) The speed-up obtained when running a program with 100 million instructions.