**Delft University of Technology**

Faculty of Electrical Engineering, Mathematics and Computer Science

Computer Engineering Lab

**Computer Architecture and Organization (CAO) EE3D11**

**Assignment 4**

**Deadline March 18, 2025 before 08:45!**

**Note**:

* Upload a **single pdf file** having the name “YourFirstName-LastName-HW-4.pdf” to the corresponding assignment in Brightspace.
* Write your first name, last name, and student number on your assignment.
* When needed, the answer should be **justified**. Show clearly which theory is used to find your final answer.

**Exercise 1 [3/ 2/ 1/ 2/ 2]**

1. We examine how an instruction is executed in a single-cycle data path. Assume we have an instruction which results in the following 32 bits after it is fetched:

0000 00 **01 100** 0 0010 **0000 1** 000 00 **10 0101**

0000 00**01 100**0 0010 **0000 1**000 00**10 0101**

31 27 23 19 15 11 7 3 0

Use Figure 4.24 for this exercise and assume that the data memory is all zero and that the registers of the processor have the following values at the beginning of the cycle at which the above instruction is fetched:

R0=0 R1= 256 R2=-128 R3=19 R4=-32 R5=13

R6=-6 R8=-1 R12=16 R31=-2

1. What is the output of the ‘sign-extend’ unit, that of ‘shift left 2’ unit (given in the figure as Jump address) and that of the ‘shift left 2’ unit at the input of the Adder on the top right of the figure?

sign-extend: 0000 00**00 000**0 0000 **0000 1**000 00**10 0101**

**shift left 2:**  0000 **0110 0**000 10**00 001**0 0000 **1001 01**00

shift left 2 (adder): 0000 **0000 0**000 00**00 001**0 0000 **1001 0100**

1. What is the value of each control signal delivered by the main control unit?

The first 6 0s means that this is an R instruciton. With the function of “or” as specified by the last 6 bits.

RegDst 1

Branch 0

MemRead 0

MemtoReg 0

ALUOp 10

MemWrite 0

ALUSrc 0

RegWrite 1

1. What is the value of the new PC address after this instruction is executed?

PC + 4

1. What are the inputs and the outputs of the main ALU and the Adder on the top right of the figure.

ALU:

Inputs:

A: value\_of($0b01 100) = value\_of($12) = 16 =

= 0b0000 0000 0000 0000 0000 0000 0001 0000

B: value\_of($0b0 0010) = value\_of($2) = -128 =

= 0b1111 1111 1111 1111 1111 1111 1000 0000

Output: 0b1111 1111 1111 1111 1111 1111 1001 0000 = -112

ADD:

Inputs:

A: PC + 4

B: 0000 **0000 0**000 00**00 001**0 0000 **1001 0100**

**Output: PC + 4 + 8340 = Pc + 8344**

**Exercise 2 [2/ 2/ 1/ 2/ 3 ]**

Assume the following MIPS code will be executed in a 5-stage MIPS pipelined datapath.

1. lw t2, 0(t1)

2 and t1, t2, t1

3 lw t3, 0(t2)

4 lw t1 ,0(t1)

5 sw t1, 0(t2)

1. Give the data dependencies of this code by filling in the table below. The table has the following form:

“ x depends on y, type dependency, involved register, true-, anti- or out-dependency”.

For example: instruction x depends on instruction y, type RaW, register $ti, True dependency

where: x and y indicate the instruction number and $ti is the involved register. Note in our case x, y {1,2,3,4,5,}

Also note that there are three cases for the type dependency:

* + RaW: later instruction tries to read an operand before earlier instruction writes it
  + WaW: later instruction tries to write an operand before earlier instruction writes it
  + WaR: later instruction tries to write an operand before earlier instruction reads it

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x depends on y | | Type | Register | True versus anti-dependency |  |
| 2 | 1 | RaW | t2 | True |  |
| 3 | 1 | RaW | t2 | True |  |
| 4 | 2 | RaW | r1 | True |  |
| 5 | 4 | WaW | t1 | Out |  |
| 5 | 1 | RaW | t2 | True |  |
| 2 | 1 | WaR | t1 | Anti |  |
| 4 | 2 | WaW | t1 | Out |  |
| 4 | 2 | WaR | t1 | Anti |  |
| 5 | 4 | WaR | t1 | Anti |  |
|  |  |  |  |  |  |

1. Assume that the datapath has no forwarding unit and no hazard detection. In addition, assume that there is no structural hazard and that the register file can be written and read in the same cycle. Show how the execution of the code should look like in order to guarantee the correctness of the execution by inserting NOPs. What is the number of clock cycles needed? - 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| lw t2, 0(t1) | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |  |  |
| and t1, t2, t1 |  | IF | NOP | NOP | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |
| lw t3, 0(t2) |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |
| lw t1 ,0(t1) |  |  |  |  |  | IF | NOP | ID | EX | MEM | WB |  |  |  |  |  |  |
| sw t1, 0(t2) |  |  |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |

1. Is it possible to reorder the code in such way to reduce the number of NOPS (Stalls)? What will be the number of clock cycles needed for the execution of the code in this case? - 11

lw t2, 0(t1)

lw t3, 0(t2)

and t1, t2, t1

lw t1 ,0(t1)

sw t1, 0(t2)

1. Assume now that the above processor has a forwarding unit, but no hazard detection unit (e.g. we forget to implement it!). How many clock cycles are needed for the execution of the code? Is the code executing correctly? What will happen? – 9, “and” instruction will not wait for lw to finish therfore fetching a wrong value and creating a wrong result that will propagate and result in wrong lw later on,.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| lw t2, 0(t1) | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |  |
| lw t3, 0(t2) |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |
| and t1, t2, t1 |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |
| lw t1 ,0(t1) |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |
| sw t1, 0(t2) |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |

1. Assume now the above processor has a hazard detection and forwarding unit as shown in Figure 4.60.
   * Specify for the first five clock cycles how the execution of the instructions will look like and show for each cycle the value of the output signals of the hazard detection unit and the forwarding unit (see Figure 4.60).
   * What will be the required number of clock cycles to execute the code in this case?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Instructions | Clock cycles 1 to 5 | | | | |  | CC | Signals at each cycle |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |