Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Computer Engineering Lab
2022-2023

Computer Architecture and Organization (CAO) EE3D11

Assignment 4

Deadline Mar 21, 2023 before 08:00!

Note:
Upload a single file having the name "YourFirstName-LastName-HW-1" to the corresponding assignment in Brightspace.
When needed, the answer should be justified. Show clearly which theory is used to find your final answer.

Exercise 1 [5/2/1/2 pts]

Consider a simple single-cycle implementation of MIPS ISA as shown in the figure 4.17 of the book (Fifth edition). The latency (i.e., operation time) of each major functional components for this implementation is the following:

Component	Latency
PC register (read operation)	1 ns
PC register (write operation)	1 ns
Register file (read operation)	7 ns
Register file (write operation)	5 ns
Add	8 ns
2-1 MUX	2 ns
Control	4 ns

Component	Latency
Sign extender	3 ns
ALU	10 ns
ALU control	2 ns
Shifter	3 ns
Data memory access	15 ns
Instruction memory access	15 ns
Logic gate	1 ns

We assume that the MIPS single-cycle datapath implementation has the following basic instructions: R-type (op rd, rs, st), Load, Store, Branch.

1. Use the table format below to indicate the used hardware components when each instruction is executed. Components which are used in parallel should be included in the same column (e.g., of the Adder and the ALU are used simultaneously by an instruction). The components in the horizontal direction indicate the using order of such components (e.g., components included in the first column should be used before using the components in the second column, etc.). Note that all instructions begin by reading the PC register with a latency of 2ns (1ns + 1ns).

Instr.	Components/ Hardware element used by the instructions								
R-type									
Load									
Store									
Branch									

2. Calculate the latency of the critical path of each instruction. The table in question 1 can be used to derive the latency of critical paths.

3. What should be the clock frequency of this single-cycle datapath implementation?

4. What should be the clock frequency if we will restructure the design in order to create an optimal 5 pipeline stages in the implementation assuming an overhead of 10% of the critical path as consequence of pipeline design?

Exercise 2 [3/ 3/ 3/ 1]

Consider the following code and a 5 pipeline stage machine.

```
lw R1,  0(R3)
lw  R2,  0(R4)
beq R1, R2, target
add R3, R3, R4
addi R4, R3, 4
exit
target:   lw R5, 0(R4)
add  R5, R5, R6
……
```

1- Use the table below to show the pipeline execution diagram in order to guarantee the correctness of the execution. Use the following notation: F=IF; D=ID; E=EX; M=MEM; W=WB; insert NOPs and stalls (NOTE: these have different meanings) when needed. label stalls by "**" in the box, and NOP by "—". If the instruction is not in any pipeline stage, then do not include anything in the columns.

Assume:
A1. There is no structural hazard and the register file can be written and read in the same cycle.
A2. The datapath has no forwarding unit and no hazard detection.
A3. There is no branch prediction and no branch delay slot, and the branch decision is taken in the EX stage of the pipeline.
A4. R1 and R2 have *same* content when compared.

What is the number of clock cycles needed to correctly execute the code?

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
lw R1, 0(R3)																				
lw R2, 0(R4)																				
beq R1, R2, target																				
add R3, R3, R4																				
addi R4, R3, 4																				
exit																				
lw R5, 0(R4)																				
add R5, R5, R6																				

2- In a similar manner as for question 1, use the same table to show the pipeline execution diagram in order to guarantee the correctness of the execution, but now assumption A2 above is changed to:

- A2’ : The processor has data hazard and load-use hazard detection with its associated forwarding unit

What is the number of clock cycles needed to correctly execute the code?

3- In a similar manner as for question 1, use the same table to show the pipeline execution diagram in order to guarantee the correctness of the execution, but now A2 and A3 are changed to:

- A2’’ : The processor has data hazard and load-use hazard detection with its associated forwarding unit
- A3’’: The machine uses “Brach not taken” prediction and no branch delay slot, and the branch decision is taken in the ID stage of the pipeline

What is the number of clock cycles needed to correctly execute the code?

4- Assume now we would like to make use of “branch delay slot”. What are the possible ways to schedule this delay slot for the above code?