University of Cape Town
Computer Science Department

CSC2002S - Computer Architecture Assignment-4 (Architecture-2)
October 2020

Instructions:

- Submit a zip folder, with each question (Exercise 1, 2, 3, 4) as a separate PDF, named StudentID-Q#.pdf, e.g XYZABC001-Q1.pdf.
- Each PDF should indicate student id number, but NOT names
- All answers must include clear explanations/working.

Exercise 1

When implementing a logic expression in digital logic, one must use the available logic gates to implement an operator for which a gate is not available. Problems in this exercise refer to the following logic expressions:

	Control Signal 1	Control Signal 2
a.	(((A AND B) XOR C) OR (A XOR C)) OR (A XOR B)	(A XOR B) OR (A XOR C)
b.	(((A OR B) AND C) OR ((A OR C) OR (A OR B))	(A AND C) OR (B AND C)

I. For a.) and b.), implement the logic for the Control signal 1.Your circuits should directly implement the given expression (do not reorganize the expression to “optimize” it), using NOT gates and 2-input AND, OR, and XOR gates. **[10]**

II. When multiple logic expressions are implemented, it is possible to reduce implementation cost by using the same signals in more than one expression. Repeat (I) above, but implement both Control signal 1 and Control signal 2, and try to “share” circuitry between expressions whenever possible. **[10]**

Exercise 2

Refer to the loops a) and b) below. Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, and that the pipeline has full forwarding support. Also assume that many iterations of this loop are executed before the loop exits.

	Loop

a.	`Loop: ADD R1,R2,R1`
	`LW R2,0(R1)`
	`LW R2,16(R2)`
	`SLT R1,R2,R4`
	`BEQ R1,R9,Loop`

b.	`Loop: LW R1,0(R1)`
	`AND R1,R1,R2`
	`LW R1,0(R1)`
	`LW R1,0(R1)`
	`BEQ R1,R0,Loop`

For a) and b), show the pipeline execution diagram for the third iteration of this loop, from the cycle in which we fetch the first instruction of that iteration up to (but not including) the cycle in which we can fetch the first instruction of the next iteration. Show all instructions that are in the pipeline during these cycles (not just those from the third iteration). **[10]**

Exercise 3 [45 Marks]

In this exercise, we examine how pipelining affects the clock cycle time of the pro- cessor. Assume that individual stages of the datapath have the following latencies:

	IF	ID	EX	MEM	WB
a.	250ps	350ps	150ps	300ps	200ps
b.	200ps	170ps	220ps	210ps	150ps

I. What is the clock cycle time in a pipelined and non-pipelined processor? **[5]**

II. What is the total latency of an LW instruction in a pipelined and non-pipelined processor? **[10]**

III. Instead of a single-cycle organization, we can use a multi- cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare clock cycle times and execution times with single-cycle, multi-cycle, and pipelined organization. Assume that instructions executed by the processor are broken down as below. **[30]**

	ALU	BEQ	LW	SW
a.	45%	20%	20%	15%
b.	55%	15%	15%	15%

Exercise 4 [15 Marks]

For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

	Tag	Index	Offset
a.	31–10	9–5	4–0
b.	31–12	11–6	5–0

I. What is the cache line size (in words)? **[5]**

II. How many entries does the cache have? **[5]**

III. What is the ratio between total bits required for such a cache implementation over the data storage bits? **[5]**

END.