CS2100 Assignment 2 Answer Book

Name:	Lim Jia Wei
Student ID:	A0255479M
Tutorial Group Number:	13

After completion, save this file as AxxxxxxxY.pdf and submit on Canvas together with your code as per the instructions given in the question paper.

If you do not fill your particulars above, or do not follow the submission instructions you will forfeit up to 3 marks.

Submission information: ______________ / 3

Part A

Question 1. (7 MARKS)
(Use the space to describe the working/thinking behind your code. Think of it as your "working". Marks will be deducted for failure to do so, or simple answers like "Go see the code yourself!" This applies for Questions 1 to 3.)

(a) (3 marks)
The multiplexor takes in several inputs sources, as well as a input which toggles which input source to output. In MIPS, each mux takes in only 2 input sources, so the output of the mux has to be either one of those inputs. We control this output stream simply by a Boolean variable, which I describe for each function as a ternary operator. Since each mux function's parameters already match the output of the function (uint8_t, uint32_t and int32_t) I simply return the result of the ternary operator.

(b) (4 marks)
The function decode simply separates the given input of 32 bits into its respective buckets (opcode, rs, rt, rd, shamt, funct, immed, address). Each of these buckets are nothing but a range of digits in the input of 32 bits, and we control the selection of bits using a combination of the shift left and shift right logical functions as well the AND function in order to isolate the desired digits, according to the size of bucket. I was debating as to whether I should only populated the insn variable's fields according to the type of opcode that we extract from instruction (R,I and J) type instructions, however I chose to go with populating all the fields in the variable insn, and only in the later parts of the code do I take the fields when they are necessary for each corresponding instruction type.

Q1 Total: _________ / 7

Question 2. (10 MARKS)
(a) (3 marks)
Simply here the function Control reads the opcode and sends out the appropriate signals for each of the various parameters, just like how the MIPS Control unit would do in the MIPS Processor, instead here the variable are global points which you simply change from a 1 to a 0. My solution analyses the general patterns of each instruction format, and narrows the cases down to those

which I have written in the if else or switch cases. Some cases overlap which is why there are 'or' conditions as well, whereas most of them are singular cases for only a few specific instructions.

(b) (3 marks)
In the same spirit as part (a), the function ALUControl looks at the ALUOp code as part of the multilevel decoding to simplify the circuit design. Here I have split the function into three possible states essentially, R-Format Instructions, BEQ and load/store word instructions. Although I'll note that the output of ALUControl should actually 4 bits, but C only provides the uint8_t as the smallest data type under the standard library, but effectively only 4 bits are being used.

(c) (4 marks)
The ALU takes in two inputs of 32 bits and is responsible for producing an arithmetic operation performed on both the two input streams depending on what control signal we send to the ALU unit. The possible operations are AND, OR, Addition, Subtraction, Set Less Than, and bitwise negation, which is nicely represented by a simple switch case. Lastly, we also consider the additional out signal which we consider when the Branch Signal is 1.

Q2 Total: _________ / 10

Question 3. (10 MARKS)

Here the task is to code the program as closely as possible to the full MIPS Datapath. I have broken down the stages into the 4 main stages – Decode, Execute, Memory and Writeback. Most of the code here are functions that were previously implemented so calling them is sufficient to output the appropriate control signals.

Here I update the PC counter by 4 at the start of the program to replicate how the MIPS Datapath updates its PC counter at every up clock cycle and is often achieved before the rest of the Datapath is executed.

I will also point out that I needed to create new variables to represent the signals of RD1 and RD2 as pointers for signed 32 bits which represents the 32-bit outs from the file register, since these signals would likely be changed every new instruction, they do need to be declared as global variables outside the program.

I used 2 mux functions to determine which signal to input for the RegFile Function in Stage 2 and in Stage 4 for the correct Write Register and the Write Data respectively. They are 8 bit and 32 bit long respectively as well. I had to call the RegFile twice, once in Stage 2 and another time in Stage 4 because they serve different purposes in each stage. In Stage 2, the purpose is to decide which Write Register to Write to depending on the instruction, and in Stage 4, the purpose is to write the appropriate Write Data into the Write Register set in Stage 2. According to the theoretical MIPS Datapath, this seems somewhat accurate as signals will pass through the Register File twice in the case of a Write Register Writeback such as in R-Formatted Instructions.

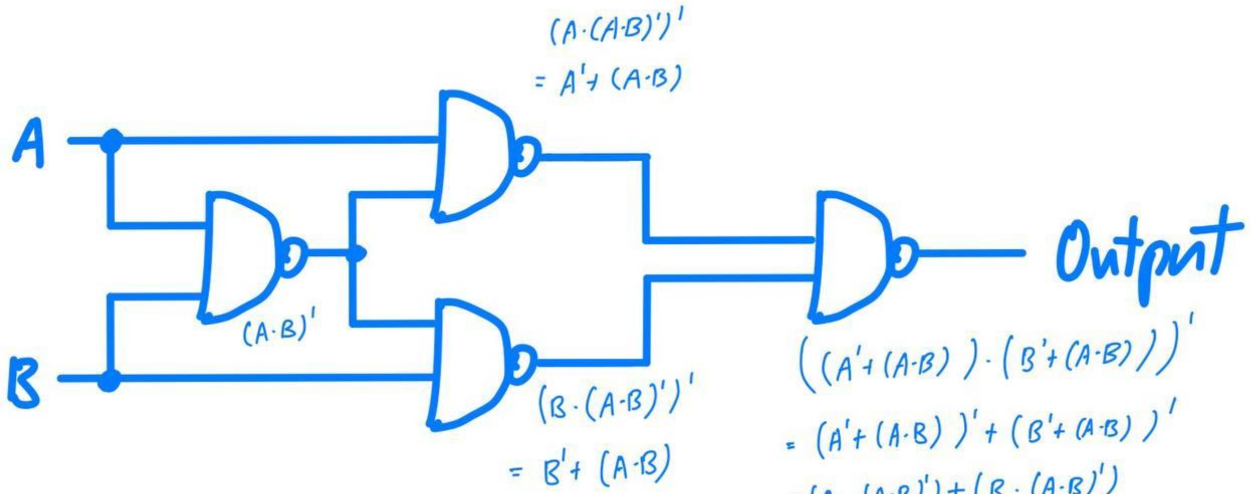
The remaining instructions are self-explanatory and simply reflect the nature of how the MIPS processor functions.

Q3 Total: _________ / 10

Part B

Question 4. (10 MARKS)

4a. Draw your logic diagram neatly below. (2 marks)



$$((A'+(A\cdot B))\cdot(B'+(A\cdot B)))'$$
$$= (A'+(A\cdot B))' + (B'+(A\cdot B))'$$
$$= (A\cdot(A\cdot B)') + (B\cdot(A\cdot B)')$$
$$= A\cdot(A'+B') + B\cdot(A'+B')$$
$$= A\cdot A' + A\cdot B' + B\cdot A' + B\cdot B'$$
$$\text{Output} = A\cdot B' + A'\cdot B$$

4b. $S_k = (A_k \cdot B_k) + (A_k{}' \cdot B_k{}')$ (1 mark)

4c. $F = S_0$ (1 mark)

4d. Prime implicants: $(P' \cdot Q), (Q \cdot S), (R \cdot S), (P' \cdot S'), (P' \cdot R)$ (1 mark)

4e. Essential prime implicants: $(P' \cdot Q) + (Q \cdot S) + (R \cdot S)$ (1 mark)

4f. Simplified POS for $G = (P' + S) \cdot (Q + R + S')$ (2 marks)

4g. Simplified SOP for $H = (C' \cdot D' \cdot E') + (A \cdot C' \cdot E') + (A' \cdot C \cdot E) + (B \cdot C \cdot E)$ (2 marks)

Q4 Total: _________ / 10

Total Marks: ____________ / 40 (To be filled by TA only)

=== END OF PAPER ===