

CIS 240 Fall 2019: Midterm

Oct 23, 2019

Name :

Please write your name on the exam **and** the exam booklet and turn in both. You can answer the questions on this exam sheet or in the exam booklet. Please number the questions you are addressing clearly in the exam booklet.

Question 1 {25 pts}

Your job is to design a circuit that will take as input a 3-bit value, I and produce a 2-bit output O , which indicates the number of 1 bits in the input. For example, if $I = 101$ then O should be 10, if $I = 010$ then $O = 01$. In your diagram I_2 , I_1 and I_0 should indicate the 3 bits of the input and O_1 and O_0 the two bits of the output where O_1 is the MSB.

Remember that we cannot grade what we cannot read so please make your diagrams as neat and clear as possible.

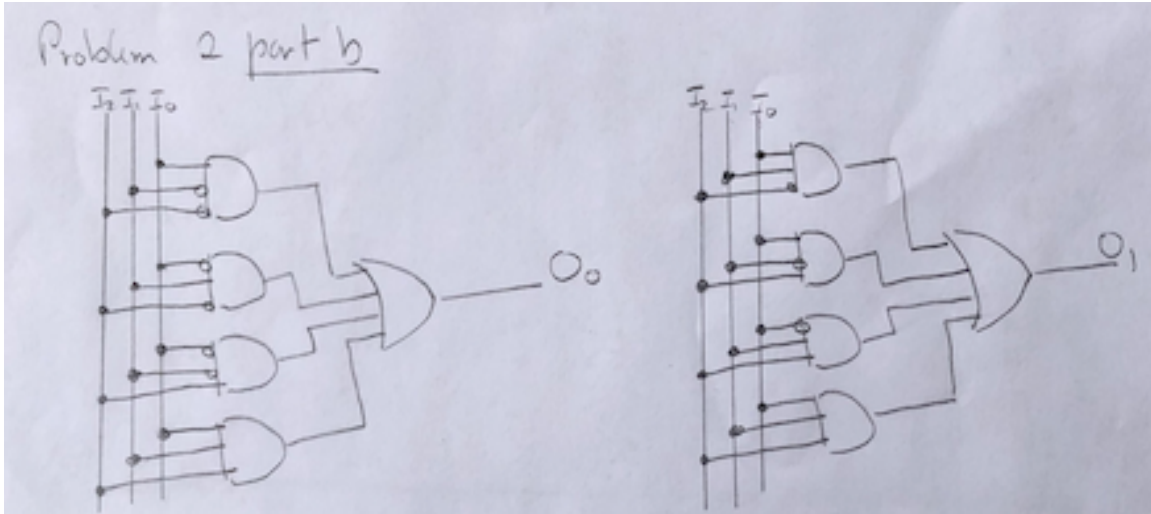
- {5 pts}** First provide a truth table indicating what the output O should be for every possible value of the input I .
- {5 pts}** Design 2 PLA circuits to produce the two output bits O_1 and O_0 .
- {10 pts}** Design 2 CMOS circuits to produce the two output bits O_1 and O_0 . You can assume that you have access to negated versions of all of the input bits. Your solution should consist of one CMOS network for each output bit. It should not involve cascading multiple CMOS circuits.
- {5 pts}** If the PLA circuit you provide for part b is ultimately implemented on the same kind of CMOS technology that you would use for your answer to part c which implementation do you think would have the lower overall delay? Explain your answer.

Answer

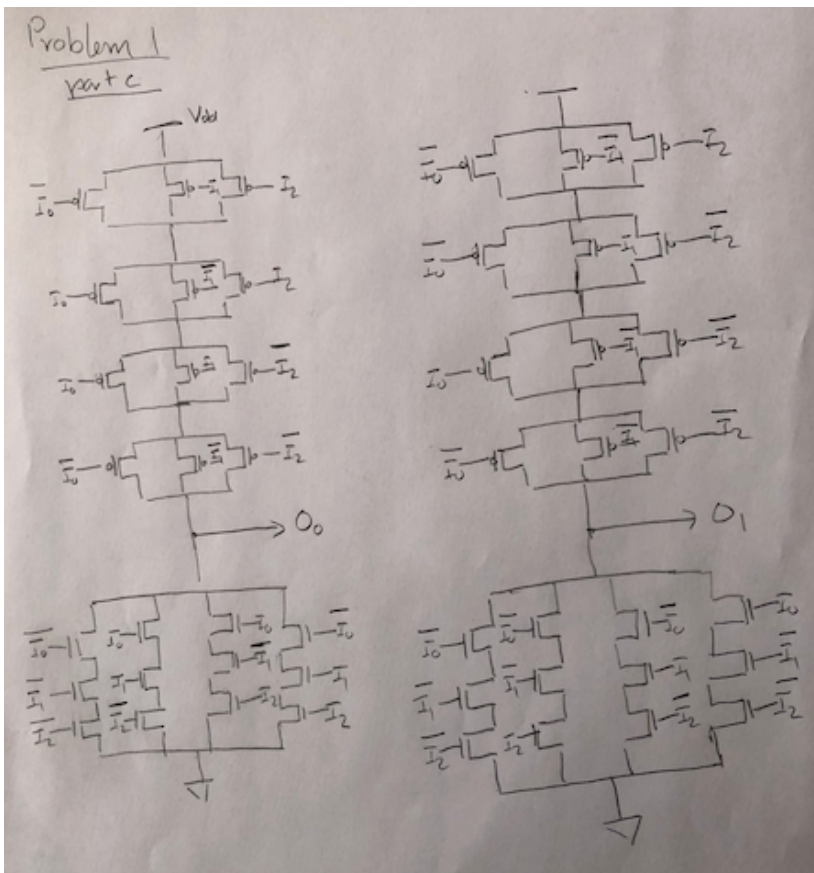
Part a

| I_2 | I_1 | I_0 | O_1 | O_0 |
|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Part b



Part c



Part d

The CMOS circuits from part c are implemented as single gates where the PLAs involve multiple levels of logic – a layer of and gates and an or gate. We would expect that this organization would add delay so the circuit for part c would have lower overall delay.

Question 2 {15 pts}

- a) List every LC4 instruction that requires the Privilege.CTL signal to be set to 1
 - a. TRAP – the only instruction that elevates the privilege bit to 1
- b) List every LC4 instruction that requires the ALU.CTL signal to be set to 5
 - a. ADD Immediate
- c) List every LC4 instruction that requires the ALU.CTL signal to be set to 6
 - a. LDR and STR
- d) List every LC4 instruction that requires the DATA.WE signal to be set to 1
 - a. STR
- e) List every LC4 instruction that requires the regInputMux.CTL signal to be set to 2
 - a. JSR, JSRR and TRAP all involve writing PC+1 into the register file and hence require this setting.

Question 4 {15 pts}

Your job in this question is to design part of the Decoder circuit for our Single Cycle LC4 implementation.

- a) **{10 pts}** Produce a PLA circuit that takes as input the relevant bits from the current instruction and produces as output the regFile.WE signal.
- b) **{5 pts}** Produce a second PLA circuit that generates the NZP.WE signal. Hint you can use the regFile.WE signal as an input to this second circuit.

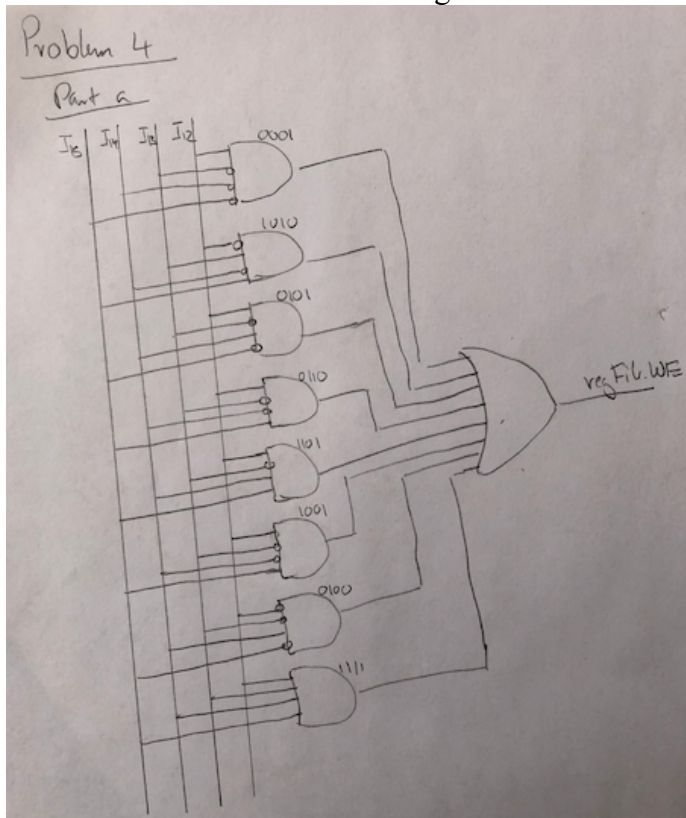
Please use the convention $I_{15}, I_{14}, \dots, I_0$ to refer to bits in the instruction word where I_{15} is the MSB and I_0 the LSB. Please note that we are asking for a PLA implementation specifically, alternative implementations will receive less points.

Answer:

Let's consider all of the instructions that write to the register file and their associated opcodes.

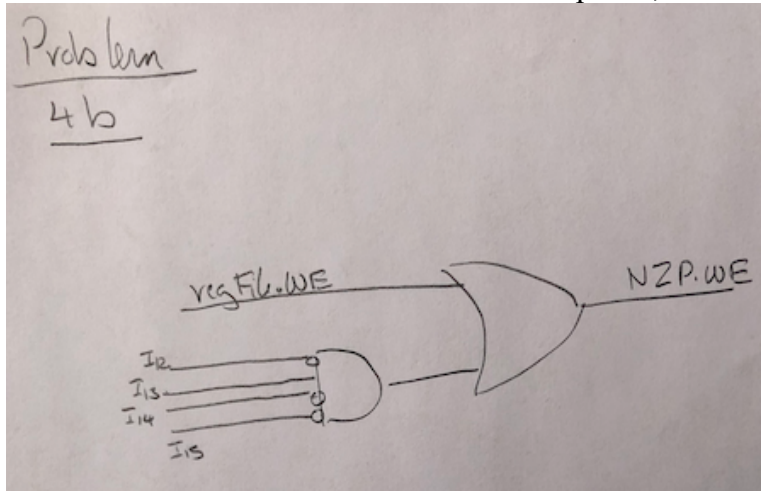
0001: ADD, SUB, MUL, DIV
1010: MOD, SLL, SRA, SRL
0101: AND, NOT, OR, XOR
0110: LDR
1001: CONST
1101: HICONST
0100: JSR and JSRR write to R7
1111: TRAP writes to R7

This leads to a PLA with 8 AND gates as shown below



Note that there are 3 opcodes that are not currently used in LC4 but available for future expansion. Note that it is entirely possible that the processor could encounter these unused opcodes during operation, and it would not be safe for the processor to write to the register file if this happened – solutions that did generate spurious write signals in these cases lost points.

To generate NZP.WE we recall that all operations that write to the register file write to NZP as do all the CMP instructions. So we just need to combine the regFile.WE signal with another term that checks for the CMP opcode, 0010.



Question 5 {5 pts}

One odd feature of the LC4 instruction set is that the MOD operation does not have the same opcode as the other arithmetic operations that it is most closely related to namely: ADD, MUL, SUB, DIV and ADD Immediate. Explain briefly why we cannot give MOD the same opcode as these other operations given the way that these other instructions are currently encoded.

Answer:

If we consider the way that the arithmetic instructions ADD, SUB, DIV and ADD Immediate are coded we notice that they all share the same opcode, 0001, and that they are distinguished by their 3 bit subopcode in bits 5:3. Now it would appear that with 3 bits you could support 8 different instructions which would allow room for MOD but the way that they are allocated if the first bit of the subopcode is 1 that signals an Immediate add and the remaining 5 bits in the instruction are viewed as part of the immediate operand. That means that we really only have 2 bits which can be used to code for different instructions and these are used to code the 4 fundamental operations, ADD, SUB, MUL and DIV, so there is no space to add the MOD instruction in this instruction coding.