CIS 2400 Fall 2022: Midterm

Oct 26, 2022

First Name : Pyotr
Last Name : Perfect
Penn ID : c is a cool language

Please fill in your information above, read the following pledge, and sign in the space below:

I neither cheated myself nor helped anyone cheat on this exam. All answers on this exam are my own. Violation of this pledge can result in a failing grade.

Sign Here : ____

Exam Details & Instructions:

- There are 5 questions (and a short bonus 6th question) worth a total of 100 points.
- You have 90 minutes to complete this exam
- You will be provided with two pages containing references sheets. Do not put any answers on these pages, they will not be graded.
- The exam is closed book. This includes textbooks, phones, laptops, wearable devices, other electronics, and any notes outside of what is mentioned below.
- You are allowed one 8.5 x 11 inch sheet of paper (double sided) for notes.
- Any electronic or noise-making devices you do have should be turned off and put away.
- Remove all hats, headphones, and watches.

Advice:

- Remember that there are 5 questions (and a short bonus 6th question). Please budget your time so you can get to every question.
- Do not be alarmed if there seems to be more space than needed for answer, we try to include a lot of space just in case it is needed.
- **Try to relax and take a deep breath.** Remember that we also want you to learn from this. A bad grade on this exam is not the end of the world. This grade also can be overwritten by a better grade with the Midterm "Clobber" Policy (details in the course syllabus)

Please put your initials at the top of each page in case the pages become separated.

If you need extra space, the last page of this exam is blank for you as scratch space and to write answers. If you use it, please clearly indicate on that page and under the corresponding question prompt that you are using the extra page to answer that question. Please also write your full name and PennID at the top of the sheet.

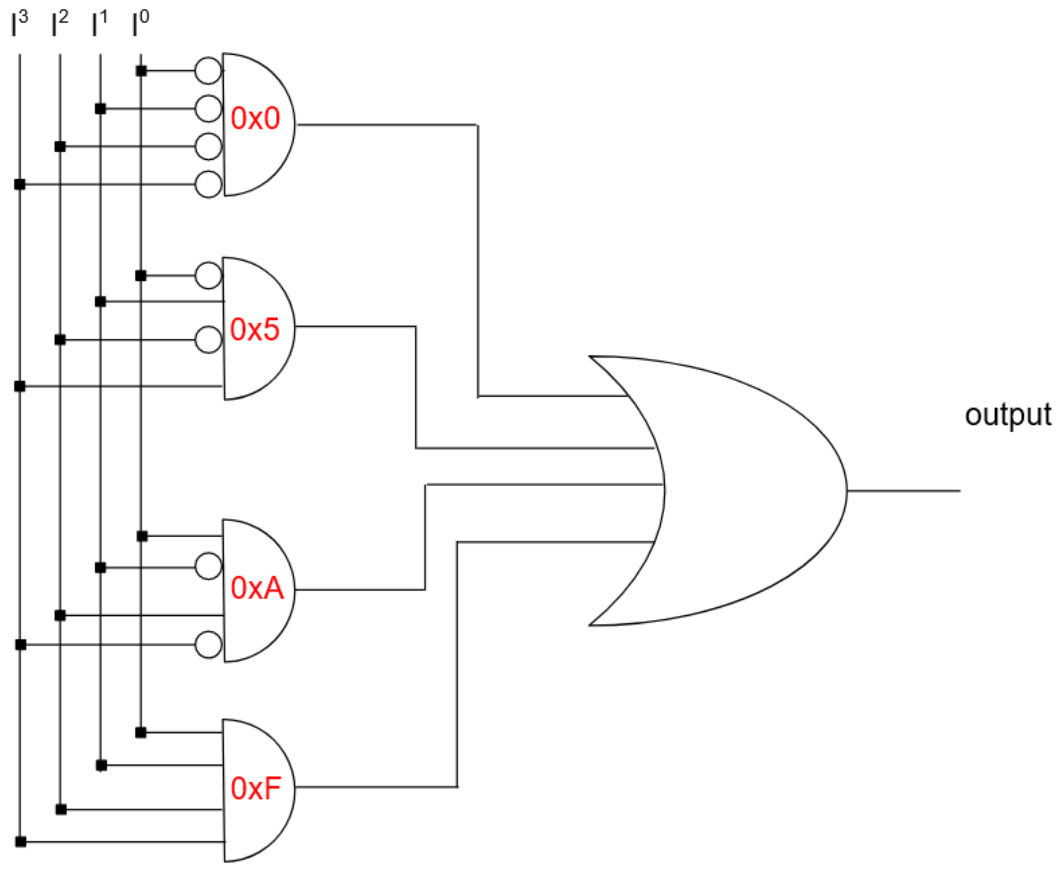
Question 1 {25 pts}

Your job is to design a circuit that will take as input a 4-bit value and produces the output 1 if and only if the first two bits of the 4-bit value are the same as the last two bits.
For example, if I = 1010 then the circuit should output 1. If I = 1001 then the circuit should output 0. In your diagram I_3 I_2, I_1 and I_0 should indicate the 4 bits of the input where I_3 is the MSB and I_0 is the LSB. Remember that we cannot grade what we cannot read so please make your diagrams as neat and clear as possible.

Part 1 {3 pts}: List all possible values for the input I that can result in the output being 1. <u>You do **not** have to list any input values that result in the output being 0.</u> Please circle your answer.

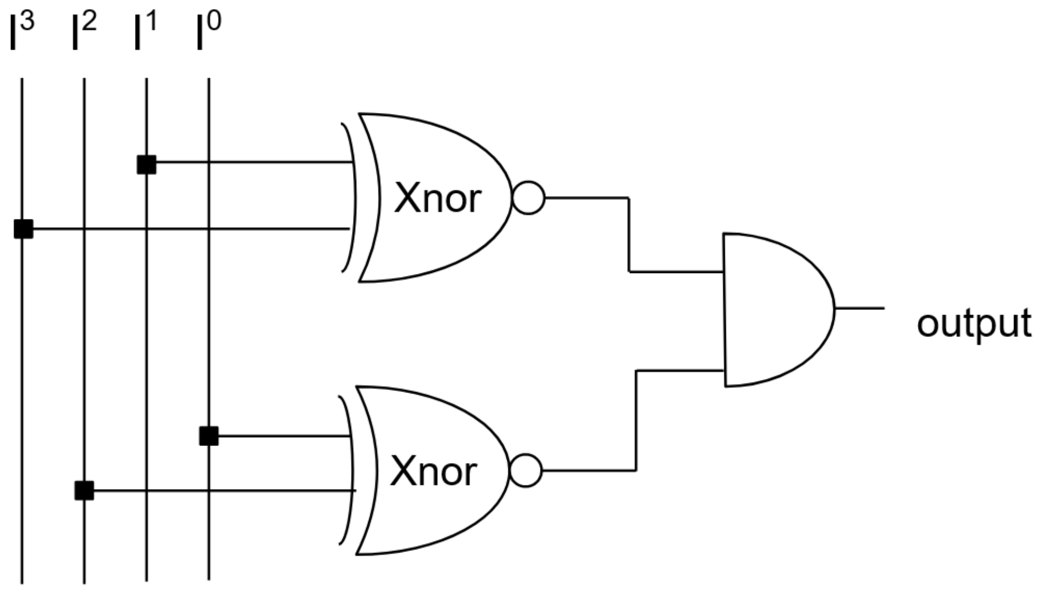
1111
1010
0101
0000

Part 2 {6 pts}: Design a **PLA** circuit that takes in the 4-bit input I and produces the correct output. Please label the inputs and outputs of your circuit clearly on your schematic.



Part 3 {6 pts}: Design a gate-level **non-PLA** circuit that takes in the 4-bit input I and produces the correct output. Please label the inputs and outputs of your circuit clearly on your schematic.

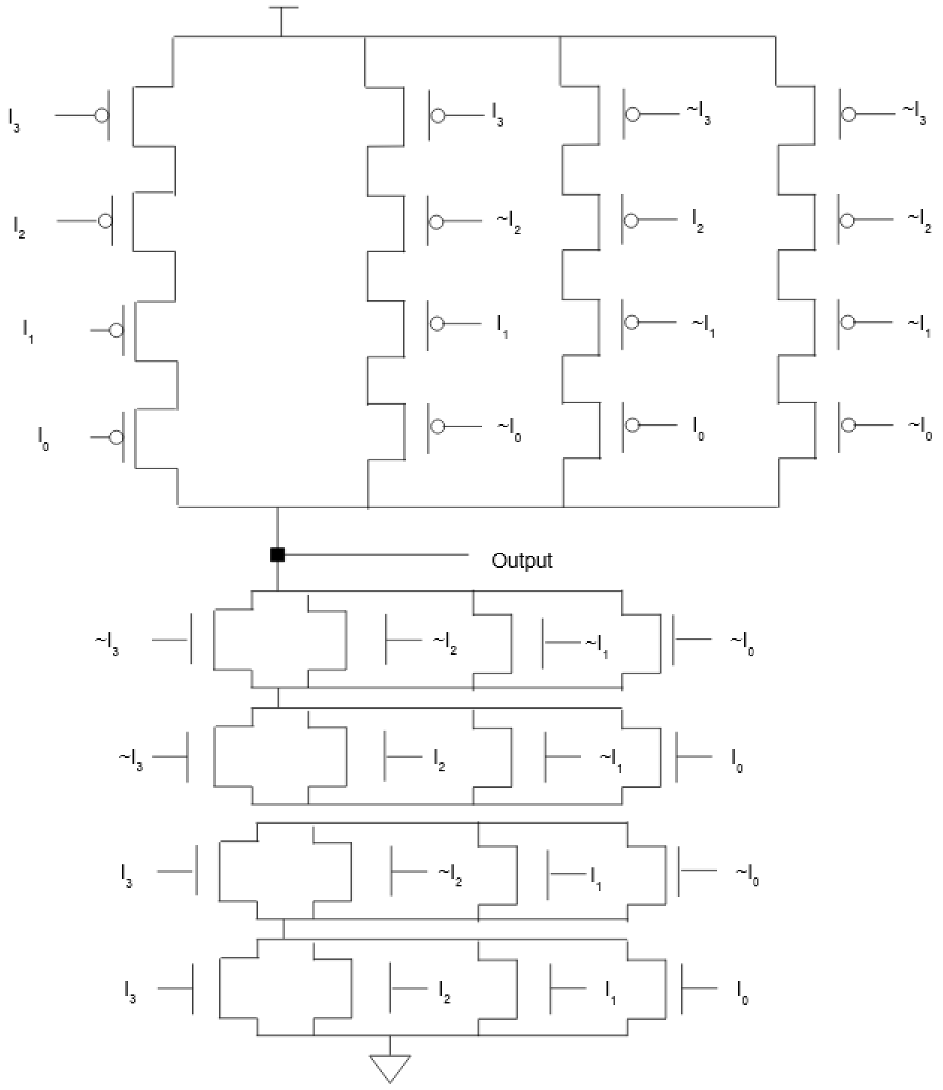
One possible solution



Part 4 {10 pts}: Design a proper **CMOS** circuit which takes in all 4 bits of the input I and produces the output bit. You can assume that you also have access to negated versions of all of the input bits. Your solution must be a single CMOS circuit consisting of complementary pull up and pull down transistor networks. It should not involve cascading multiple CMOS circuits. Please label the inputs and outputs of your circuit clearly on your schematic.

Answer:

One possible solution



Question 2 {14 pts}:

Suppose that we started designing an alternative version of LC4 that we will call LC5. In LC5, there is one instruction that chains many gates after another and so the propagated gate delay causes that instruction to execute in 12ns. Each other instruction in LC5 can finish execution in 3ns or less. LC5 also implements a single cycle processor similar to how LC4 does.

One thing that still needs to be decided for LC5 though is the clock period (how long a clock cycle should last). Noam thinks that the clock period should be 13ns, whereas Kanavi wants the clock period to be 5ns.

Part 1 {10 pts}:
Would you choose a 5ns clock cycle or a 13ns clock cycle? Please give one justification for why the clock cycle period you chose is better for our LC5 implementation.

Some accepted answers:

- 13ns would allow the long instruction to work properly in this computer
- 5ns would allow any program not using the long instruction to run much faster

Part 2 {4 pts}:
Give one justification for why the clock cycle period you didn't choose in Part 1 may be chosen by someone else to be used in LC5.

See above

Question 3 {12 pts}
When an LC4 program is assembled and loaded into memory, each instruction is stored as 16-bit pattern that represent that exact instruction. In this question, you will do this conversion in both directions

Part 1 {6 pts}:
Encode the following instructions into 16-bit patterns. You are free to leave spaces between the bits to make it easier to read.

Instruction	Binary representation
STR R6, R1, #7	0111 1100 0100 0111
ADD R5, R4, #-1	0001 1011 0011 1111
BRnp #3	0000 1010 0000 0011

Part 2 {6 pts}:
Decode the following 16-bit patterns into LC4 instructions. We have left spaces between some of the bits to make it easier to read.

Binary representation	Instruction
0101 0110 0101 1100	XOR R3, R1, R4
0010 0100 0010 1100	CMP R2, R4
1100 1000 0000 0111	JMP #7

Question 4 {25 pts}:
Your job is to write an LC4 assembly program that takes in an integer value in R0 and stores the factorial of that integer into R1. The factorial of an integer N, is the product of all integers between 1 and N. For example, the factorial of 4 is 1 * 2 * 3 * 4 = 24. We have provided the assembler directives, some labels and some comments to start the program. Your program should reach the label END when it is finished.
You can assume that:

- The integer value in R0 has been set for you, you do not need to set the input value of R0 in your program
- The input value in R0 is greater than or equal to 1
- You are free to modify any of the registers. You just need to make sure that R1 ends up with the correct value by the END of the program
- You are free to add additional labels and comments as needed

```
.CODE
.ADDR x0000

FACT
        ; start writing code here
        ; one possible answer
        CONST R1, #1
LOOP    CMPI R0 #1
        BRnz END
        MUL R1, R1, R0
        ADD R0, R0, #-1
        JMP LOOP

END     ; end of program
```

Question 5 {23 pts}
Consider the Single Cycle LC4 instruction set that we walked through in class. Suppose we wanted to add an additional variant on the LDR instruction that replaced the 5-bit integer immediate with another register as shown:

Mnemonic: LDR Rd, Rs, Rt
Semantics: Rd = dmemory[Rs + Rt]
Encoding: 0011 ddds ssxx xttt

The usual rules for updating the PC and NZP apply.

Part 1 {15 pts}: Indicate how each of the following control signals should be set to execute this instruction. If a control signal doesn't matter for this instruction, you must instead write an X.

Answer:

Control Singal Name	Value
PCMux.CTL	1
rsMux.CTL	0
rtMux.CTL	0
rdMux.CTL	0
regFile.WE	1
regInputMux.CTL	1
NZP.WE	1
DATA.WE	0
Privilege.CTL	2
ALUInputMux.CTL	0
ALU.CTL	0

Part 2 {8 pts}: While we can replace the integer immediate in LDR, we cannot do the same for STR without having to change the processor design or having to deal with weird semantics. Please explain why this is the case. (Advice: take a look at the LC4 Single Cycle Processor Diagram)

If we replaced the integer immediate in STR with a register, that would require a third source register.

Normally it is:

    dmem[Rs + sext(imm6)] = Rt

but we are trying to make it:

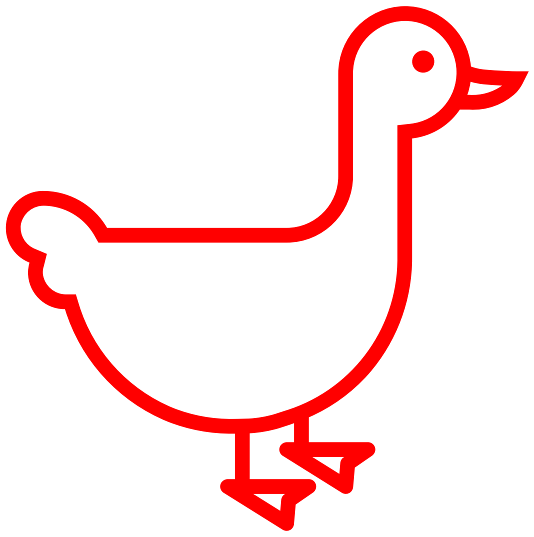
    dmem[Rs + R?] = Rt

We don't have a third "source" register, so we would have to re-use Rt as both the value we want to store and as part of the address calculation.

    dmem[Rs + Rt] = Rt

This would be very awkward to use.

Question 6 {1 pt; all exam submissions receive this point}
Select one member of the course staff. Create a piece of art (e.g. drawing, poem, anything you like) about that person.



quack