CS2100 Computer Organization

AY2023/24 Semester 2

Assignment 2

(Deadline: 18 March 2024, Monday, 1pm)

ANSWERS

Instructions

1. There are TEN questions in this assignment, totaling FORTY marks.

2. This assignment is due on **Monday, 18 March 2024, 1 pm.** The submission folder will be closed at 1:10 pm, **after which no submission will be accepted and you will receive ZERO for this assignment.**

3. Answer these questions on Canvas → Assignments.

4. You should do these assignments on your own. Do not discuss the assignment questions with others.

5. Please post on QnA “Assignments” topic if you have any queries.

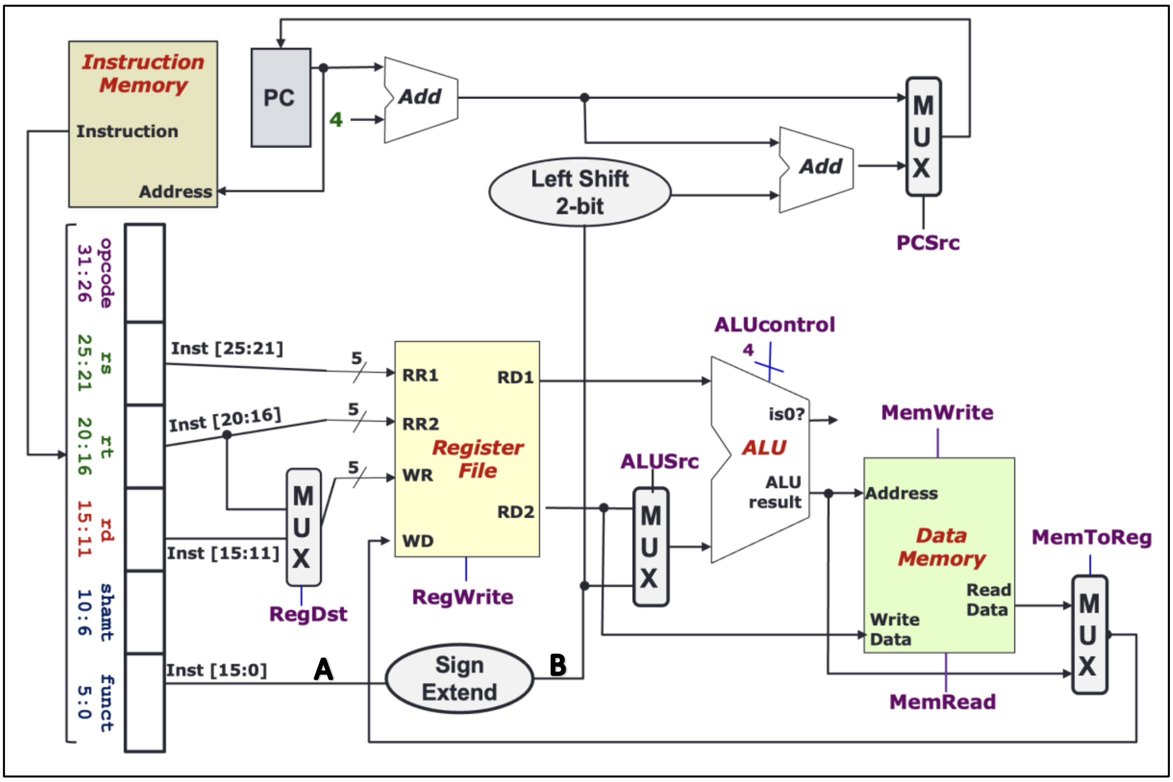
Part I: MIPS Datapath (20 marks)

Question 1. (20 marks)

Consider the MIPS Datapath with the initial values of the registers and memory as shown below. Values preceded by 0x are in hexadecimal.

```
MIPS Registers
R0  (r0) = 0x00000000 | R1  (at) = 0x00002000
R2  (v0) = 0x00000001 | R3  (v1) = 0x0000000a
R4  (a0) = 0x00000005 | R5  (a1) = 0x7ffff000
R6  (a2) = 0x7ffff004 | R7  (a3) = 0x000000b0
R8  (t0) = 0x00000001 | R9  (t1) = 0x00000c00
R10 (t2) = 0x0000c000 | R11 (t3) = 0xfffffff0
R12 (t4) = 0xf0000000 | R13 (t5) = 0x00000fff
R14 (t6) = 0x00006200 | R15 (t7) = 0x00000e00
R16 (s0) = 0x00300000 | R17 (s1) = 0x00000c00
R18 (s2) = 0x00040200 | R19 (s3) = 0x00011000
R20 (s4) = 0x00030200 | R21 (s5) = 0x00000000
R22 (s6) = 0x00055000 | R23 (s7) = 0xf0000000
R24 (t8) = 0x00000005 | R25 (t9) = 0x0000d000
R26 (k0) = 0x00000000 | R27 (k1) = 0x00000000
R28 (gp) = 0x10008000 | R29 (sp) = 0x7fffeff4
R30 (s8) = 0x1000000f | R31 (ra) = 0x00400018
```

Address	Memory
0xFFFFFF94	1
0xFFFFFF98	2
0xFFFFFF9C	3
0xFFFFFFA0	4
0xFFFFFFA4	5



The current PC value is **256** and the following instructions are being executed one by one:

```
lw  $s3, -100($s5)  //instruction 1
add $s6, $s3, $s4   //instruction 2
```

Fill in the values of the fields in the table below. You have to fill column 2 after executing instruction 1 and column 3 after executing instruction 2. For rows marked with *, your answers must follow the base given (0b for binary, 0x for hexadecimal), and you must write all the required digits, or it will be graded as wrong even if the value is correct. You do NOT need to include the prefix 0b or 0x in your answers. For rows without *, the answers are to be in decimal, and you are not to include leading zeroes in your decimal answers.

Fields	**Values for Instruction 1**	**Values for Instruction 2**
RegDst		
MemRead		
MemWrite		
ALUcontrol*	0b	0b
RegWrite		
Instruction[31-26]*	0b	0b
Instruction[25-21]*	0b	0b
Instruction[20-16]*	0b	0b
Instruction[15-0]*	0b	0b
A*	0b	0b
B*	0x	0x
RD1*	0x	0x
RD2*	0x	0x
ALU Result*	0x	0x
ALUSrc		
RR1		
RR2		
WR		
WD		
MemToReg		

Answers: The instructions are decoded as below for reference. Students are not expected to provide this solution or working. They just need to fill the values in the table below.

lw $s3, -100($s5) is decoded as:

100011 Op	10101 Rs	10011 Rt	1111111110011100 -100 in 2s Comp, 16bit rep

add $s6, $s3, $s4 is decoded as:

000000 Op	10011 Rs	10100 Rt	10110 Rd	00000 Shamt	100000 Funcode

Fields	Values for Instruction 1	Values for Instruction 2
RegDst	0	1
MemRead	1	0
MemWrite	0	0
ALUcontrol*	0b0010	0b0010
RegWrite	1	1
Instruction[31-26]*	0b100011	0b000000
Instruction[25-21]*	0b10101	0b10011
Instruction[20-16]*	0b10011	0b10100
Instruction[15-0]*	0b1111111110011100	0b1011000000100000
A*	0b1111111110011100	0b1011000000100000
B*	0xFFFFFF9C	0xFFFFB020
RD1*	0x00000000	0x00000003
RD2*	0x00011000	0x00030200
ALU Result*	0xFFFFFF9C	0x00030203
ALUSrc	1	0
RR1	21	19
RR2	19	20
WR	19	22
WD	3	197123
MemToReg	1	0

Part II. Boolean algebra, logic circuits and simplification (20 marks)

Note that in general, unless otherwise stated, complemented literals are not available.
Remember to write · for the AND operation, or mark will be deducted. If you are typing your answers, you may use the full stop (.) for AND and the single quote (‘) for complement.

Question 2 (4 marks)

The exclusive-or (XOR) binary operation is defined as follows:

$$x \text{ XOR } y = x'\cdot y + x\cdot y'$$

The exclusive-nor (XNOR) binary operation is the negation of XOR. Express the exclusive-nor (XNOR) binary operation, x XNOR y, in **sum-of-products expression**, by completing the derivation below using Boolean algebra and justifying the steps with appropriate laws and theorems of Boolean algebra.

x XNOR y = (x XOR y)' = …

For ease of typing, you may use the full stop (.) for AND and

Answer:

x XNOR y = (x XOR y)'

$= (x'\cdot y + x\cdot y')'$

$= (x'\cdot y)' \cdot (x\cdot y')'$ (De Morgan's)

$= ((x')' + y') \cdot (x'+(y')')$ (De Morgan's)

$= (x+y') \cdot (x'+y)$ (involution)

$= (x\cdot(x'+y)) + (y'\cdot(x'+y))$ (distributive law)

$= (x\cdot x' + x\cdot y) + (y'\cdot x' + y'\cdot y)$ (distributive law)

$= (0 + x\cdot y) + (y'\cdot x' + 0)$ (complement law: $A\cdot A' = A'\cdot A = 0$)

$= x\cdot y + y'\cdot x'$ (identity law: $A+0 = 0+A = A$)

Shorter:

x XNOR y = (x XOR y)'

$= (x'\cdot y + x\cdot y')'$

$= (x'\cdot y)' \cdot (x\cdot y')'$(De Morgan's)

$= ((x')' + y') \cdot (x'+(y')')$ (De Morgan's)

$= (x+y') \cdot (x'+y)$ (involution)

$= (x\cdot(x'+y)) + (y'\cdot(x'+y))$ (distributive law)

$= x\cdot y + y'\cdot x'$ (absorption theorem 2)

Question 3 (1 mark)

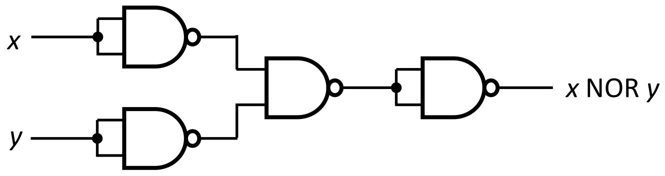
You want to build a logic circuit that performs the NOR operation (i.e. x NOR y) by using only 2-input NAND gates. What is the minimum number of NAND gates required?

Answer: 4 NAND gates are needed. (autograded)

Question 4 (2 marks)

You want to build a logic circuit that performs the NOR operation (i.e. x NOR y) by using only 2-input NAND gates and the fewest number of gates. Draw the circuit, labelling the inputs (x, y) and output (x NOR y) clearly. Complemented literals are not available.

Answer:



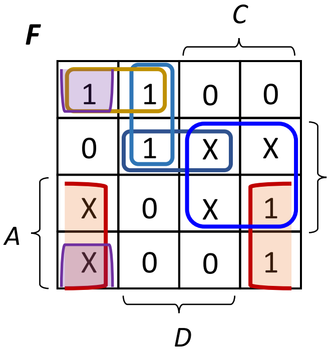
Question 5 (1 mark)

Given the 4-variable Boolean function $F(A,B,C,D) = \prod M(2,3,4,9,11,13) \cdot \prod X(6,7,8,12,15)$ where M are the maxterms and X the don't-cares, how many prime implicants are there on the K-map of F? You do not need to show your K-map.

Answer: 6 (autograded)

Working: The prime implicants are:

A·D', B·C, A'·B·D, A'·C'·D, A'·B'·C', and B'·C'·D'.



1	1	0	0
0	1	X	X
X	0	X	1
X	0	0	1

Question 6 (1 mark)

Given the 4-variable Boolean function $F(A,B,C,D) = \prod M(2,3,4,9,11,13) \cdot \prod X(6,7,8,12,15)$ where M are the maxterms and X the don't-cares, how many essential prime implicants are there on the K-map of F? You do not need to show your K-map.

Answer: 1 (autograded)

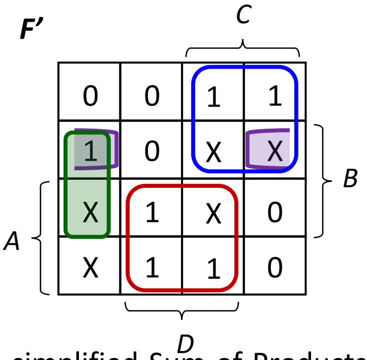
Working: The essential prime implicant is *A·D'*.

Question 7 (2 marks)

Given the 4-variable Boolean function $F(A,B,C,D) = \prod M(2,3,4,9,11,13) \cdot \prod X(6,7,8,12,15)$ where M are the maxterms and X the don't-cares, write out the simplified Product-of-Sums (POS) expression for F. If there are more than one answer, write out all answers.

Answer: Simplified POS expression for *F* = (*A'*+*D'*) · (*A*+*C'*) · (*B'*+*C*+*D*)
or (*A'*+*D'*) · (*A*+*C'*) · (*A*+*B'*+*D*)

Working: Simplified SOP expression for *F'* = *A·D* + *A'·C* + *B·C'·D'*
or *A·D* + *A'·C* + *A'·B·D'*



0	0	1	1
1	0	X	X
X	1	X	0
X	1	1	0

Question 8 (3 marks)

Given a Boolean function on 8 variables, *Z*(*A,B,C,D,E,F,G,H*), write the simplified Sum-of-Products (SOP) expression for each of the following.

(i) *m*123 · *m*4 (ii) *m*123 + *M*4 (iii) *m*123 · *M*4

Answers: (autograded)

(i) *m*123 · *m*4 = **0**

(ii) *m*123 + *M*4 = ***A+B+C+D+E+F'+G+H***

(iii) *m*123 · *M*4 = ***A'·B·C·D·E·F'·G·H***

123 = 01111011_2
4 = 00000100_2

Question 9 (3 marks)

For each of the following, write the minterm numbers in increasing order, separated by comma, without any space.

(i) Given the following functions:

$F(A,B,C) = A + B' \cdot C'$

$G(A,B,C) = B + A' \cdot C$

$H(A,B,C) = F(A,B,C) \cdot G(A,B,C)$

What is $H(A,B,C)$ in Σm notation? That is, $H(A,B,C) = \Sigma m(\ldots)$. Write out the list of minterm numbers in increasing order, separated by comma, without any space.

(Ii) Given the following functions:

$P(A,B,C) = A + B'$

$Q(A,B,C) = (A' + B) \cdot (A + C)$

$R(A,B,C) = P(A,B,C) \oplus Q(A,B,C)$ where $\oplus$ is the XOR operator

What is $R(A,B,C)$ in Σm notation? That is, $R(A,B,C) = \Sigma m(\ldots)$. Write out the list of minterm numbers in increasing order, separated by comma, without any space.

(iii) A circuit counts the number of '0' bits in an unsigned binary integer *ABCD* and outputs an unsigned binary integer *XYZ*. For example, if *ABCD* = 1010, then *XYZ* = 010 as *ABCD* contains two 0s; if *ABCD* = 0100, then *XYZ* = 11 as *ABCD* contains three 0s.

What is $Z(A,B,C,D)$ in Σm notation? That is, $Z(A,B,C,D) = \Sigma m(\ldots)$. Write out the list of minterm numbers in increasing order, separated by comma, without any space.

Answers: (autograded)

(i) $(A + B' \cdot C') \cdot (B + A' \cdot C) = A \cdot B = \Sigma m(\mathbf{6,7})$.

Or, $A + B' \cdot C' = \Sigma m(0,4,5,6,7)$; $B + A' \cdot C = \Sigma m(1,2,3,6,7)$; therefore $H(A,B,C) = \Sigma m(\mathbf{6,7})$.

(ii) $A + B' = \Sigma m(0,1,4,5,6,7)$;

$(A' + B) \cdot (A + C) = A' \cdot C + A \cdot B + B \cdot C = \Sigma m(1,3,6,7)$;

Therefore $R(A,B,C) = \Sigma m(\mathbf{0,3,4,5})$.

(iii) $Z(A,B,C,D) = \Sigma m(\mathbf{1,2,4,7,8,11,13,14})$.

A	*B*	*C*	*D*	*X*	*Y*	*Z*
0	0	0	0	1	0	0
0	0	0	1	0	1	1
0	0	1	0	0	1	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	0	0

Question 10 (3 marks)

The following Boolean function *F*(*a,b,c*) is given in sum-of-minterms expression using the Σm notation:

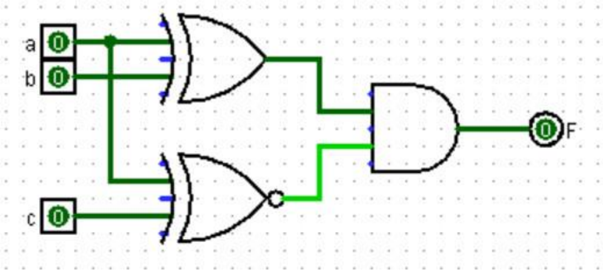
$$F(a,b,c) = \Sigma m(2,5).$$

Implement this function using the fewest number of logic gates. You are to draw the logic diagram clearly, labelling all inputs and the output of the circuit. Complemented literals are not available.

(Logic gates you may use are inverters, AND, OR, NAND, NOR, XOR and XNOR gates. Other than the inverters, the other gates are all 2-input gates.)

Answer:

Three logic gates are used, one XOR gate, one XNOR gate, and one AND gate:



Working:

$F(a,b,c) = (a \text{ XOR } b) \text{ AND } (a \text{ XNOR } c)$

$= (a'\cdot b + a\cdot b')\cdot(a'\cdot c' + a\cdot c)$

$= (a'\cdot b\cdot a'\cdot c' + a'\cdot b\cdot a\cdot c) + (a\cdot b'\cdot a'\cdot c' + a\cdot b'\cdot a\cdot c)$ (distributive law)

$= (a'\cdot b\cdot c' + a'\cdot b\cdot a\cdot c) + (a\cdot b'\cdot a'\cdot c' + a\cdot b'\cdot c)$ (idempotency)

$= (a'\cdot b\cdot c' + 0) + (0 + a\cdot b'\cdot c)$ (complement law: $a'\cdot a = a\cdot a' = 0$)

$= (a'\cdot b\cdot c') + (a\cdot b'\cdot c)$ (identity law: $x+0 = 0+x = x$)

$= m2 + m5$ (definition of minterms)

Other possible 3-gate answers:

- (*a* XOR *b*) AND (*b* XOR *c*)
- (*a* XNOR *c*) AND (*b* XOR *c*)
- (*a* XNOR *b*) NOR (*b* XNOR *c*)
- etc.

=== END OF PAPER ===