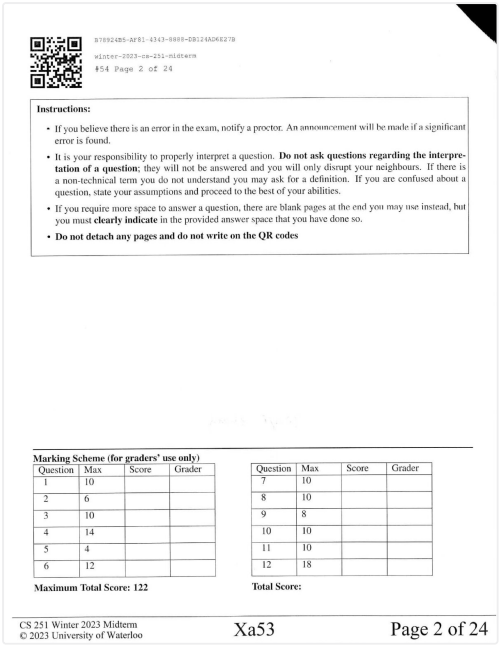
My grades for **Winter 2023 CS 251 Midterm**



B78924B5-AF81-4343-8888-DB124AD6E270
winter-2023-cs-251-midterm
#54 Page 2 of 24

Instructions:

- If you believe there is an error in the exam, notify a proctor. An announcement will be made if a significant error is found.
- It is your responsibility to properly interpret a question. **Do not ask questions regarding the interpretation of a question**; they will not be answered and you will only disrupt your neighbours. If there is a non-technical term you do not understand you may ask for a definition. If you are confused about a question, state your assumptions and proceed to the best of your abilities.
- If you require more space to answer a question, there are blank pages at the end you may use instead, but you must **clearly indicate** in the provided answer space that you have done so.
- **Do not detach any pages and do not write on the QR codes**

Marking Scheme (for graders' use only)

Question	Max	Score	Grader
1	10		
2	6		
3	10		
4	14		
5	4		
6	12		

Maximum Total Score: 122

Question	Max	Score	Grader
7	10		
8	10		
9	8		
10	10		
11	10		
12	18		

Total Score:

Q1 8

Points on this page: 10

1. (10 points) True/False.

Which of the following statements are true/false. (circle one of True or False to the left of each statement).
Note: +1 for each correct answer, −1 for each incorrect answer, 0 for each question left unanswered.
Maximum points on this question is 10.

1. True False A PMOS transistor generates a weak 0 and strong 1.
2. True False A CMOS NAND gate has more transistors than a CMOS AND gate.
3. True False IEEE 754 floating point representation stores the exponent in sign magnitude notation.
4. True False Flip-flops are used in the implementation of finite state machines to store inputs and state bits.
5. True False The 64-bit ALU performs subtraction of $A - B$ as $A + \bar{B} + 1$
6. True False The address stored in Program Counter (PC) register is a 2's complement address.
7. True False Static RAM is a form of memory that remembers its values when power is turned off.
8. True False Four bit two's complement numbers have a larger range than four bit numbers using Sign Magnitude representation.
9. True False A D-latch improved upon the SR-latch by incorporating a clock pulse and removing the possibility of two zero inputs on S and R.
10. True False A tri-state buffer can have the following three possible output values: 0, 1, and Don't Care 'X'.

2's: 1111 = -1
1000 = -8
0111 = 7

Sign: 1111 = -7
0111 = 7

Q2 3

Points on this page: 6

2. (6 points) CS251 ARM Instruction Set Architecture.

Consider the following ARM code that starts execution at memory address 000:

```
000: ADDI   X1, XZR, 10
004: SUBI   X1, X1, #0
008: CBZ    X1, #3
012: SUBI   X1, X1, #1
016: ADDI   X0, X0, #2
020: ADD    X2,X1,X0
024: B      #-4
028: ADDI   X5, XZR, #1
```

What are the final values in the following registers?

This is an infinite loop

X0 = $+\infty$

X1 = $-\infty$

X2 = $+\infty$

X5 = (~~uninitialized~~) (un-initialized)

XZR = 0

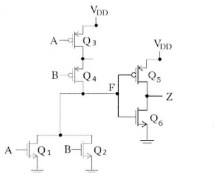
PC = {008, 024, 4} PC jumps between 008, 012, 016, 020, and 024

Q3 10

Points on this page: 10

3. (10 points) Transistors.

(a) (8 points)
Given the circuit below, complete the truth table, determining the internal resistances for Q_1 to Q_6 and the final outputs F and Z. The transistors Q_1 to Q_6 should be High or Low (or 'H' or 'L'). The outputs F and Z may be 0, 1, or float '–'.



A	B	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	F	Z
0	0	H	H	L	L	H	L	1	0
0	1	H	L	L	H	L	H	0	1
1	0	L	H	H	L	L	H	0	1
1	1	L	L	H	H	L	H	0	1

(b) (2 points) Give a simplified boolean formula for Z in terms of A and B.

$Z = A + B$

Q4abc 8

Points on this page: 8

4. (14 points) Digital Logic.

(a) (2 points) Give the *unreduced, minterm* expression for F based on the following truth table.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

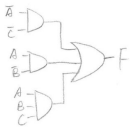
$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC$

(b) (2 points) Minimize the boolean equation in part (a).

$F = \bar{A}\bar{C} + A\bar{B} + ABC$

$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC$
$= \bar{A}\bar{C}(\bar{B}+B) + A\bar{B}(\bar{C}+C) + ABC$
$= \bar{A}\bar{C} + A\bar{B} + ABC$

(c) (4 points) Sketch a combinational circuit implementing the function F.

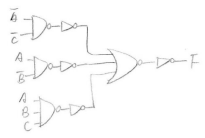


Q4d 6

Points on this page: 6

6DD5D7A3-9EC9-4A36-870C-D48EAFE4A62A
winter-2023-cs-251-midterm
#54 Page 7 of 24

(d) (6 points) Sketch a combinational circuit implementing the function F using only NOT, NAND and NOR gates.

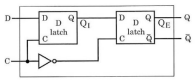


Q5 4

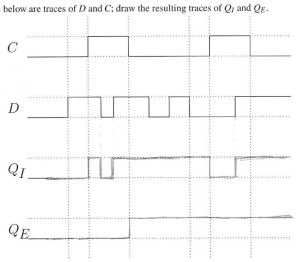
8D62D3FC-6EB0-4056-B4E7-630ED2EC56A36
winter-2023-cs-251-midterm
#54 Page 8 of 24

Points on this page: 4

5. (4 points) Sequential Logic.
Consider a D flip flop:



In the figure below are traces of D and C; draw the resulting traces of Q_I and Q_E.



Q6abcd 8 Points on this page: 8

6. (12 points) Data Representation and Manipulation.

(a) (2 points) Sign extend the following 4-bit 2's complement number to an 8-bit 2's complement number, and give the signed decimal value of the extended number.

Given number : 1010

Sign extension : 1111 1010 **Signed Decimal :** −6

(b) (3 points) Given the 8-bit binary number 1001 1010, give its decimal equivalent if these eight bits are interpreted as

1. an 8-bit unsigned number: $128+16+8+2 = 154$
2. an 8-bit signed magnitude number: $-(16+8+2) = -26$
3. an 8-bit 2's complement number: $-128+16+8+2 = -102$

(c) (1 point) Negate the following number, expressed in 8-bit 2's complement representation.

1010 1001

0101 0110
+ 1
0101 0111

0101 0111

(d) (2 points) Compute the following bitwise logical operations.

1101 0110	1101 1010
AND 0000 1111	XOR 0000 1111
0000 0110	1101 0101

Q6e 4

Points on this page: 4

(e) (4 points) Using 8-bit 2's complement **addition**, perform the following computations. For each computation, circle if an overflow occurred or not. **Use addition only, no subtraction**; you should take the 2's complement of any of the operands needed to convert the subtraction problem into an addition problem. Show your work.

$$\begin{array}{r} 1101\,0111 \\ +1011\,1101 \\ \hline 1\,1001\,0100 \end{array}$$

Overflow: YES (NO)

$$\begin{array}{r} 1010\,0111 \\ -0101\,1010 \\ \hline \end{array}$$

(YES) NO

~~0101~~
1010 0101
+ 1
1010 0110
+ 1010 0111
0100 1101

Q7a 4

Points on this page: 4

83F50FC5-5A15-4FDC-9797-E220236AA280
winter-2023-cs-251-midterm
t54 Page 11 of 24

7. (10 points) IEEE 754 Floating Point Representation.

(a) (4 points) The following number is given as a 32-bit, IEEE normalized floating point number with biased exponent. Convert it to a signed decimal number.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

sign: −
exponent: 128+4−127 = 5
fraction: 0.10101
signed binary: -1.10101×2^5
$= -110101$
signed decimal: $-(1+4+16+32)$
$= -53$

CS 251 Winter 2023 Midterm
© 2023 University of Waterloo
Xa53
Page 11 of 24

Q7bcd 6

Points on this page: 6

(b) (1 point) How can we detect whether this floating point number has generated an overflow or underflow?

Compare the exponent of the normalized binary floating point number with -126 and 127.

See if the exponent bits are all 1s or 0s.

(c) (1 point) What is an overflow and what is an underflow?

Overflow is the normalized binary floating point number has a bigger exponent than the biggest exponent that can be represented. In this case, it is >127.

Underflow is the normalized binary floating point number has a smaller exponent than the smallest exponent that can be represented. In this case, it is <-126.

(d) (4 points) Calculate the normalized binary floating point number that represents the value 6.375_{10}. You do not need to translate the number to IEEE format.

$6 \rightarrow 110$

$0.375 \rightarrow \frac{3}{8} = \frac{1}{4} + \frac{1}{8} = (0.011)_2$

$(6.375)_{10} = (110.011)_2$

$= (1.10011 \times 2^{2})_2$

Q8 10

Points on this page: 10

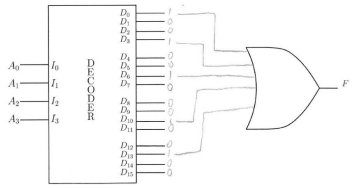
8. (10 points) Digital Design.

Design a circuit that takes a **4-bit 2's complement** number and whose output is 1 if the number is a multiple of 3.

(a) (5 points) Fill in the truth table for the output F of the circuit. Note that 0 is a multiple of 3.

	A_3	A_2	A_1	A_0	F(Mult-of-3)
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
-8	1	0	0	0	0
-7	1	0	0	1	0
-6	1	0	1	0	1
-5	1	0	1	1	0
-4	1	1	0	0	0
-3	1	1	0	1	1
-2	1	1	1	0	0
-1	1	1	1	1	0

(b) (5 points) Implement your circuit using only the 4−to−16 line decoder and OR gate shown below.

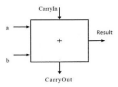


Q9ab 3

Points on this page: 4

9. (8 points) Full Adder.

(a) (2 points) A *1-bit Full Adder* as discussed in class is a combinational circuit which operates on three binary inputs, *a*, *b* and *CarryIn* (C_{in}). It computes the sum of the inputs and produces as outputs the *Result* (S) bit of the sum and a *CarryOut* (C_{out}) bit.



Complete the truth table for a Full Adder:

a	b	C_{in}	C_{out}	S
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

(b) (2 points) Give a *reduced* sum of products expression for C_{out}.

$C_{out} = ab + (a \oplus b)C_{in}$

$\bar{a}bC_{in} + a\bar{b}C_{in} + ab\bar{C}_{in} + abC_{in}$

$= C_{in}(\bar{a}b + a\bar{b}) + ab$

part(b)- not in sum of products form

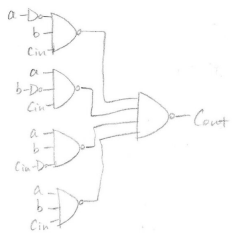
-1

Q9c 0

Points on this page: 4

C1D8E035-4098-431D-9709-1AC73D63A19E
winter-2023-cs-251-midterm
#54 Page 15 of 24

(c) (4 points) Implement a circuit for the output C_{out} from part(b) using only NAND gates.



$(\overline{a}\overline{b})\,(\overline{(a \oplus b)C_{in}}) = ab + (a \oplus b)C_{in}$

$\overline{a}\overline{b}$

$\overline{(a \oplus b)C_{in}}$

$(a \oplus b)\,C_{in}$

a	b	$\overline{a}$	$a \oplus b$	$\overline{ab}$	$\overline{\overline{a}\overline{b}}$	$\overline{\overline{a}\overline{b}}$
0	0	1	0	1	0	1
0	1	1	1	1	1	0
1	0	0	1	1	1	0
1	1	0	0	0	0	

CS 251 Winter 2023 Midterm
© 2023 University of Waterloo

Xa53

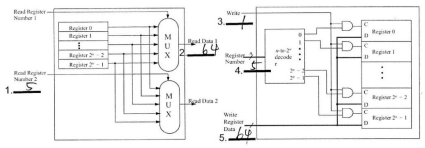
Page 15 of 24

Q10a 5

Points on this page: 5

10. (10 points) Memory.

(a) (5 points) Below is the diagram for the Register File showing both read and write ports. Suppose we have an ARM 64-bit architecture (i.e, each register stores 64 bits of data) and there are 32 registers in the Register File. In the figure, state the number of bits on the input or output indicated on the dark lines numbered 1-5.



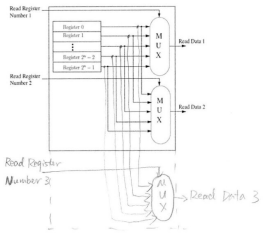
Q10b 5

Points on this page: 5

(b) (5 points) Below is the diagram for the Register File showing the read operation for two registers. Suppose we now have an ARM architecture that **requires reading from three registers in a single instruction.**

```
add3 X1, X2, X3, X4
```

This instruction reads from X2, X3 and X4, adds the values together and stores the result in the destination register X1.
Make any necessary changes to the diagram below to perform the read operation from three registers. You may add any logic gates or additional components discussed in class. **Be sure to clearly label your changes and any new inputs or outputs.**



Q11a 6

EAF7F552-DB7D-4300-A527-A941056281CF
winter-2023-cs-251-midterm
#54 Page 18 of 24

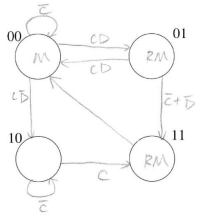
Points on this page: 6

11. (10 points) Finite State Machines for Sequential Systems.
Consider the following next-state table and corresponding output table:

S_1	S_0	C	D	S_1'	S_0'
0	0	0	X	0	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	X	0	1	1
0	1	1	1	0	0
1	0	0	X	1	0
1	0	1	X	1	1
1	1	X	X	0	0

S_1	S_0	R	M
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	1

(a) (6 points) Draw the finite state machine for this next state and output table using the states given below. Simplify the arc labels where possible.



Q11b 4

Points on this page: 4

2EFAFF0E-C434-4C32-83EB-3A9BEA17CCCA
winter-2023-cs-251-midterm
#54 Page 19 of 24

(b) (4 points) Complete the table below tracing the above finite state machine on a particular sequence of input bits for C and D. Note that the Next State of one column is the Current State for the next column. The first column has been filled in for you. The outputs are for the Current State.

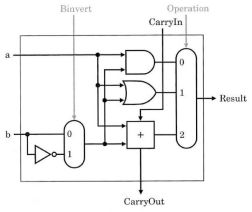
Current State	00	10	11	00	00	01
C	1	1	0	0	1	1
D	0	1	0	1	1	1
M	1	0	1	1	1	1
R	0	0	1	0	0	1
Next State	10	11	00	00	01	00

Q12a 5

Points on this page: 5

12. (18 points) ALU Design.

(a) (5 points) Consider the 1-bit ALU studied in class:



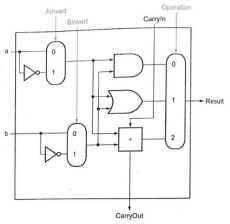
Fill in the values of CarryOut and Result in the following table (Operation is specified in binary):

a	b	CarryIn	Operation	Binvert	CarryOut	Result
0	1	0	01	0	0	1
1	1	0	00	0	1	1
1	1	0	10	0	1	0
0	1	1	01	1	0	0
1	1	1	10	1	1	0

Q12b 4

Points on this page: 4

(b) (4 points)
Consider the 1-bit ALU studied in class that has Ainvert:



We would like to perform the **NOR** of **a** and **b** using the 1-bit ALU given above. In the table below, list the values of the ALU inputs needed to perform NOR.

CarryIn	Operation	Ainvert	Binvert
X	00	1	1

NOR =

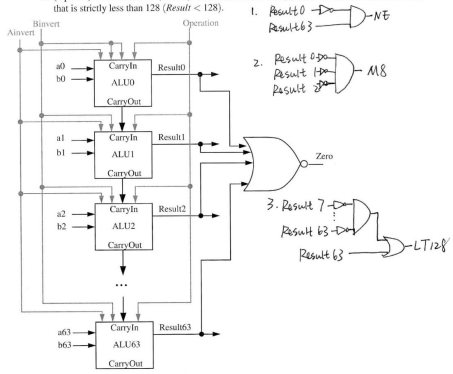
$\overline{A+B} = \bar{A} \cdot \bar{B}$

Q12c 9

Points on this page: 9

(c) (9 points) Below is the diagram for a 64-bit ALU. Modify this diagram to have the following three additional outputs. In the figure, label each output with the name given in bold in the question.

1. (3 points) **Negative Even (NE)**: This should be HIGH (1) when the output is a negative even number.
2. (3 points) **Multiple-of-8 (M8)**: This should be HIGH (1) when the output is a multiple of 8.
3. (3 points) **Less-Than-128 (LT128)**: This should be HIGH (1) when the 64-bit Result is a value that is strictly less than 128 (*Result* < 128).



0118D6E2-A801-456D-9F67-C9DE9032EC82

This page is intentionally left blank for your use. Do not remove it from your booklet. If you require more space to answer a question, you may use this page, but you must **clearly indicate** in the provided answer space that you have done so.

193920AA-F1E0-42D2-B72E-507334864AA8
winter-2023-cs-251-midterm

This page is intentionally left blank for your use. Do not remove it from your booklet. If you require more space to answer a question, you may use this page, but you must **clearly indicate** in the provided answer space that you have done so.