

Midterm

CSEE W3827 - Fundamentals of Computer Systems
Spring 2021

Feb. 25, 2021
Prof. Rubenstein

This midterm contains 4 pages with 3 questions, and totals 100 points. To get full credit you must answer all questions. **BOOKS AND NOTES ARE PERMITTED, ELECTRONIC DEVICES CAN BE USED FOR NON-COMPUTATIONAL PURPOSES.** The time allowed is 75 minutes, plus an additional 15 minute grace period to deal with upload/download issues.

Please submit questions and their parts on separate pages (to facilitate the grading process).

Show all work! We are not just looking for the right answer, but also how you reached the right answer. Right answers without work get no credit!!!

Some advice:

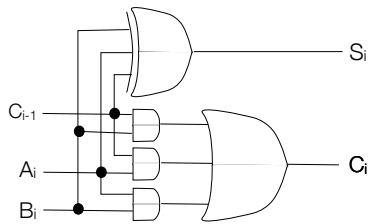
- Be sure to leave some time to work on each problem. The right answer to each problem does not require a very long answer.
- Be sure to start every problem. And take some time to think about how to set the problem up before you start writing.

1. (30 pts) In class, we showed how to use contraction to build an incrementer from a full-adder. Use contraction to build a (2's-complement) decrementer from a full adder. Recall that the algebraic expressions for a full adder are:

- $S = A_i \oplus B_i \oplus C_{i-1}$
- $C_i = AB + BC_{i-1} + AC_{i-1}$

where A_i and B_i are data input bits (i th most significant bit of numbers A And B) and C_{i-1} is the prior carry.

Recall that the following is the circuitry of a full adder:



$$B = 11 \dots 11$$

$$\forall i, B_i = 1$$

contraction

$$S = A_i \oplus \cancel{B_i} \oplus C_{i-1}$$

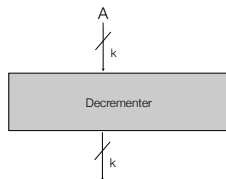
$$(A_i \oplus C_{i-1}) \oplus 1$$

$$\text{Note } a \oplus 1 = \bar{a}$$

$$\overline{A_i \oplus C_{i-1}} \quad (\text{for } C_i)$$

- (a) (20 pts) For a k -bit decrementer, show the **reduced** circuitry (or boolean expression), with a separate circuit diagram OR boolean expression for each S_i and C_i for each i . (Hint: When two i have the same design, you don't have to draw each case.) For each i , your solution may be a circuit diagram or simplified boolean expressions. Simplify as much as possible for full credit.

- (b) (10 pts) Suppose we wish to add a single-bit enable input E to the decrementer circuit such that when $E = 1$, the decrement is performed, but when $E = 0$, the circuit simply returns the input. Show how this can be implemented using the decrementer from part (a) and a 2-to-1 MUX. You may use the diagram below to represent your decrementer circuit:



$$C_{i-1}$$



$$A_i$$

$$B_i = 1$$

$$C_i = A_i \cancel{B_i} + \cancel{B_i} C_{i-1} + A_i C_{i-1}$$

$$= A_i + C_{i-1} + (A_i C_{i-1})$$

$$= \cancel{A_i} + \cancel{C_{i-1}}$$

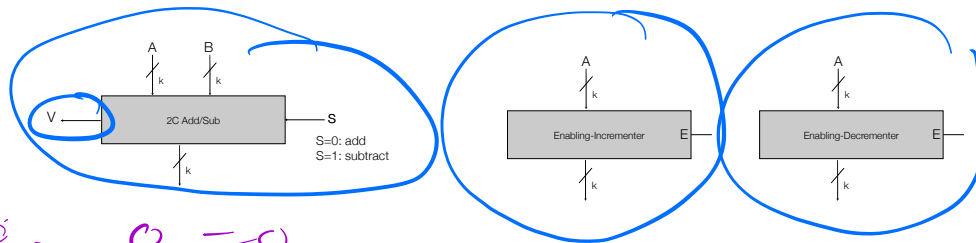
$$\cancel{B_i} (A_i + C_{i-1}) + A_i C_{i-1}$$

2. (30 pts) Build a k -bit 1's-complement adder-subtractor circuit that takes as input two k -bit values, interpreted in 1's-complement form, and a selector bit S that indicates whether to add ($S = 0$) or subtract ($S = 1$) and outputs the k -bit result in 1's-complement form, and a 1-bit overflow indicator, V .

(a) (20 pts) You may build the 1's-complement adder-subtractor using:

- 2's-complement adder-subtractor circuits
- enabling-decrementer circuits (as in problem 1)
- enabling-incrementer circuits
- any additional circuitry you think you need

The adder/subtractor, incrementer, and decrementer circuits are pictured below.



in 1's comp:

$$0000 \dots 0 = 0$$

$$1111 \dots 1 = 0$$

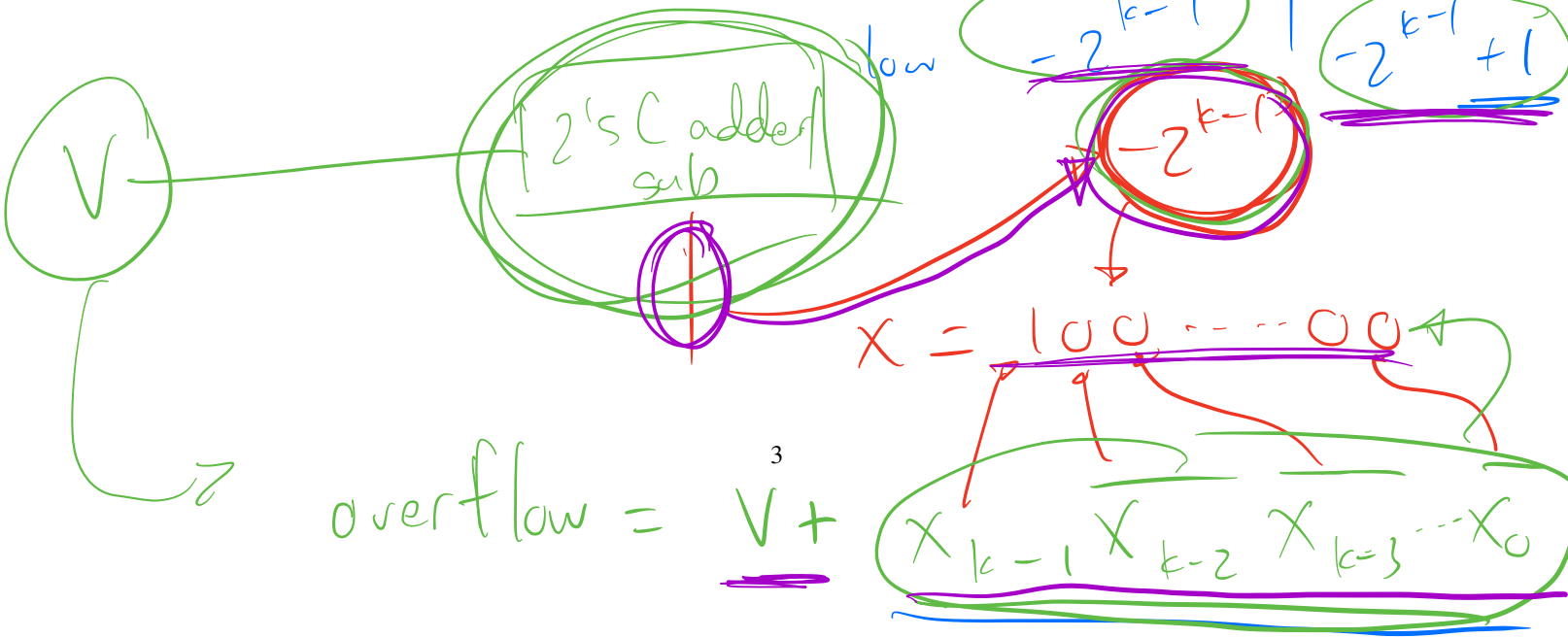
$$100 \dots 00 = \sum_{i=0}^{k-2} 2^i = (2^{k-1} - 1)$$

(b) (10 pts) Show the circuitry or provide algebraic expressions that compute the overflow for the 1's-complement adder/subtractor that you designed in part (a). (Hint: you may want to access intermediate results from your solution to part (a) - just make a notation on your part (a) diagram and explain here how you are using these signals.)

cannot represent -2^k in 1's comp.

circuitry

2's comp	1's comp
$2^{k-1} - 1$	$2^{k-1} - 1$
-2^{k-1}	$-2^{k-1} + 1$



3. (40 pts) Do you hate the ReopenCU app? Well, here's your chance to design something better... maybe. You are to build a sequential circuit that tells Columbia personnel when they should quarantine. The circuit is fed a clock that completes a cycle every 84 hours (that's 3.5 days). During each cycle, each person's current status is described by two boolean inputs, $X(t)$ and $Y(t)$, where $X = 1$ if the person is confirmed infected (otherwise $X = 0$), and where $Y = 1$ if the person had a recent close contact who is infected (else $Y = 0$). During clock cycle t , if

- $X = 1$ (infected), the person must quarantine for 2 weeks (4 clock cycles), starting immediately (i.e., clock cycles $t, t + 1, t + 2, t + 3$).
- $Y = 1$ (contact infection), the person must quarantine for 1 week, starting immediately (i.e., clock cycles $t, t + 1$).

The circuit should output 0 in any clock cycle where quarantine is not required, and 1 otherwise (quarantine required).

If in the middle of a quarantine period, a new quarantine is ordered ($X = 1$ or $Y = 1$), then to play it safe, the length of the remaining quarantine is the maximum over the remaining time of all previously existing quarantines and the time of the new quarantine.

The following table shows an example:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
X(t)	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0
Y(t)	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
O(t)	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0

- (a) (15 pts) Draw the state machine that depicts the above process
- (b) (25 pts) Design the circuit using JK Flip-Flops. Simplify your circuitry for maximum credit.

The following state/excitation tables may be of assistance while designing sequential circuitry with JK Flip-flops

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

$Q(t)$	$Q(t+1)$	$J_Q(t)$	$K_Q(t)$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

**** YOU HAVE REACHED THE END OF THE EXAM. HAVE A GREAT SPRING BREAK! ****