Midterm Solutions

CSEE W3827 - Fundamentals of Computer Systems — Feb. 25, 2021
Spring 2021 — Prof. Rubenstein

This midterm contains 9 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!!!

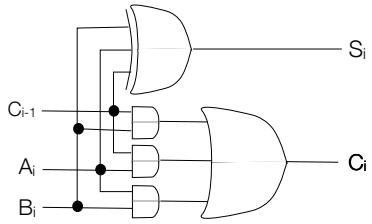
Some advice:

- 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 (ith 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:

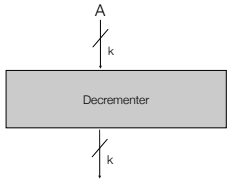


   (a) (20 pts) For a k-bit decrementer, show the **reduced** circuitry (or boolean expression), with a separate circuit diagram OR boolean expression for each S_i and C_i for each i. (Hint: When two i have the same design, you don't have to draw each case.) For each i, your solution may be a circuit diagram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

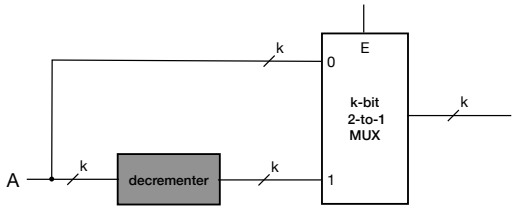
   Answer: since -1 in 2's-complement is the k-bit string of all 1's, we replace each B_i with 1. This yields:

   - Sum Circuitry:
     - For $i = 0, S_0 = A_0 \oplus 1 \oplus 0 = \overline{A_0}$
     - For $0 < i < k$, $S_i = A_i \oplus 1 \oplus C_{i-1} = \overline{A_i \oplus C_{i-1}}$. Note this is the same as $\overline{A_i} \oplus C_{i-1}$ and also $A_i \oplus \overline{C_{i-1}}$.
   - Carry circuitry:
     - For $i = 0$, the incoming carry should be treated as 0, hence: $C_0 = A_0(1) + 1(0) + A_0(0) = A_0$
     - For $0 < i < k, C_i = A_i(1) + 1C_{i-1} + AC_{i-1} = A_i + C_{i-1}$

(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:



Answer:

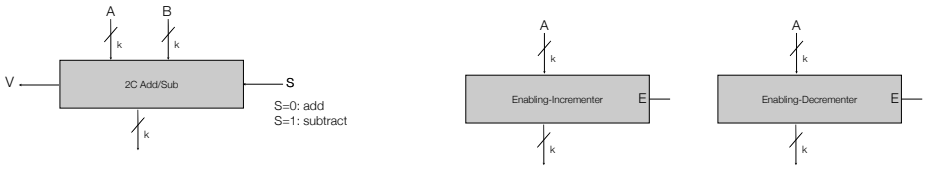


2. (30 pts) Build a k-bit 1's-complement adder-subtracter circuit that takes as input two k-bit values, interpreted in 1's-complement form, and a selector bit S that indicates whether to add ($S = 0$) or subtract ($S = 1$) and outputs the k-bit result in 1's-complement form, and a 1-bit overflow indicator,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

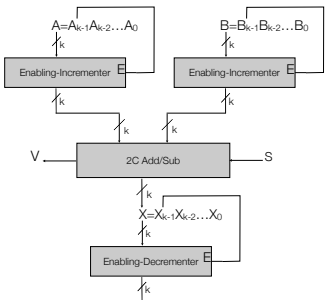
   (a) (20 pts) You may build the 1's-complement adder-subtracter using:

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

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



Answer: We begin by converting our inputs from 1's-Complement to 2's-Complement by performing an increment when the represented value is negative (when the highest order bits are 1). For instance, all 1's in 1's complement equals 0; from this we can see that we require an increment. Now we feed the values into the 2's-Complement Adder subtracter to get a 2's-Complement result of the addition or subtraction. If the value (written as $X = X_{k-1}X_{k-2}\cdots X_o$) is negative ($X_{k-1} = 1$), we now need to decrement by 1 to return the answer to 1's-Complement form.



(b) (10 pts) Show the circuitry or provide algebraic expressions that compute the overflow for the 1's-complement adder/subtracter 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.)

Answer: The result X, indicated in part (a), is a 2's complement overflow when and only when the 2's-Complement adder/subtracter's overflow $V = 1$. If it overflows in 2's-complement form, that means it is not representable via k bits in 2's-complement, so it is certainly not representable in 1's-complement form. Additionally, if the result is $100\cdots000$, this is a value that is representable in 2's-complement, but is not representable in 1's-complement. Thus, we must overflow in this case as well. Hence, a boolean equation for overflow is $V + X_{k-1} \cdot \overline{X_{k-2}} \cdot \overline{X_{k-3}} \cdot \ldots \cdot \overline{X_o}$.

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).

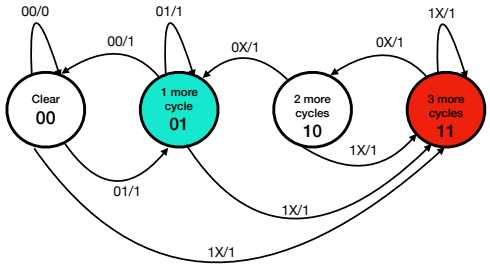
   If in the middle of a quarantine period, a new quarantine is ordered ($X = 1$ or $Y = 1$), then to play it safe, the length of the remaining quarantine is the maximum over the remaining time of all previously existing quarantines and the time of the new quarantine.

   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

Answer: A state machine describing the above system requires 4 states. The states can be numbered 0,1,2,3, where state i indicates how many clock cycles are required for quarantine prior to the current time t input. For instance, being in state 1 means that, without considering the current output, 1 more clock cycle is needed for quarantine. During the current clock cycle t, if both inputs are 0, the state machine simply counts down toward state 0, and if in state 0, may output the signal 0, whereas in other states it must output 1 since a previous quarantine is still counting down.
If $X(t) = 1$, the system outputs 1 (immediately) since quarantine takes effect immediately, and proceeds to state 3 so that a 1 will be output for the subsequent 3 clock cycles.
If $X(t) = 0$ and $Y(t) = 1$, then the system also outputs 1 (immediately), and transits to the maximum of state 1 or one less than the current state (i.e., not cutting a prior quarantine short but ensuring at least 2 clock cycles of quarantine).
There are a variety of ways to assign 2-bit boolean identifiers to the states, but in our example, we will use unsigned binary representations of 0,1,2,3 described above.



(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

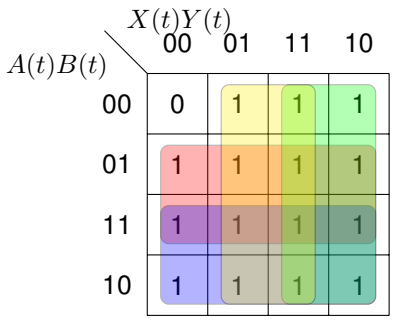
$J(t)$	$K(t)$	$Q(t+1)$
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

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

Answer: The state table, K-maps, and resulting boolean expressions follow:

$A(t)$	$B(t)$	$X(t)$	$Y(t)$	$O(t)$	$A(t+1)$	J_A	K_A	$B(t+1)$	J_B	K_B
0	0	0	0	0	0	0	X	0	0	X
0	0	0	1	1	0	0	X	1	1	X
0	0	1	0	1	1	1	X	1	1	X
0	0	1	1	1	1	1	X	1	1	X
0	1	0	0	1	0	0	X	0	X	1
0	1	0	1	1	0	0	X	1	X	0
0	1	1	0	1	1	1	X	1	X	0
0	1	1	1	1	1	1	X	1	X	0
1	0	0	0	1	0	X	1	1	1	X
1	0	0	1	1	0	X	1	1	1	X
1	0	1	0	1	1	X	0	1	1	X
1	0	1	1	1	1	X	0	1	1	X
1	1	0	0	1	1	X	0	0	X	1
1	1	0	1	1	1	X	0	0	X	1
1	1	1	0	1	1	X	0	1	X	0
1	1	1	1	1	1	X	0	1	X	0

$O(t)$:

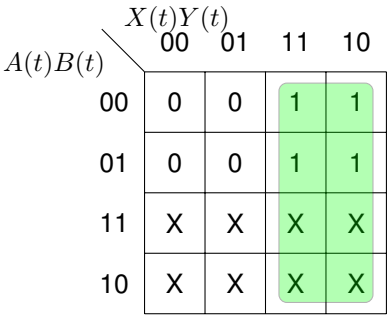


$$O(t) = A + B + X + Y$$

Answer continued on next page...

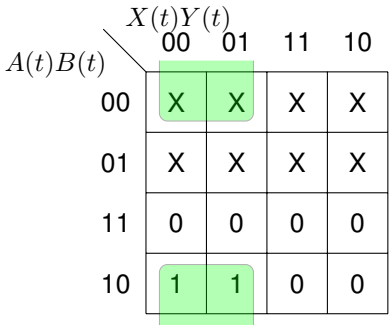
Answer:

$J_A(t)$:



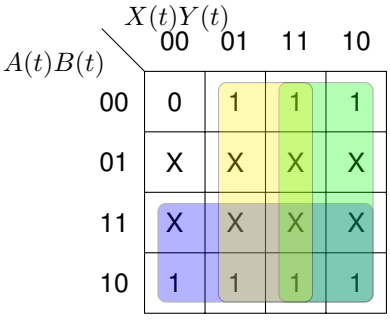
$J_A(t) = X$

$K_A(t)$:



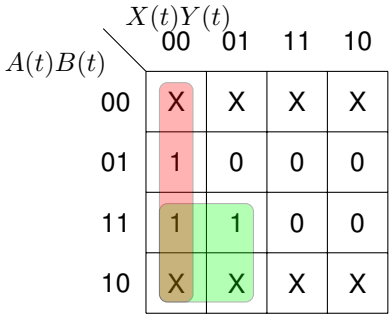
$K_A(t) = \bar{X}\bar{B}$

$J_B(t)$:



$J_B(t) = A + X + Y$

$K_B(t)$:



$K_B(t) = A\bar{X} + \bar{X}\bar{Y}$

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