LAB 2 STRUCTURAL REPORT

Academic Integrity (more info @ https://aisc.uci.edu/): You are encouraged to discuss the labs at a high level, but the code/equations/simulations you come up with should be your own. By typing "yes" at the end of this question and filling in your name, you certify that the work you are turning in is your own work. Is the work you are turning in your own? yes

If you worked on any portion of your report or vhdl code with other students (discussion at high level & debugging; if more, please describe), please list their names here: ____

Student Name:	Andy Tran
Student ID:	57422363
Date Completed:	5/3/2021
Time Spent:	Reviewing Digital Design Material: 1 hour
	Design/Preparation Work: 4 hours
	VHDL Coding & Debugging: 2 hours

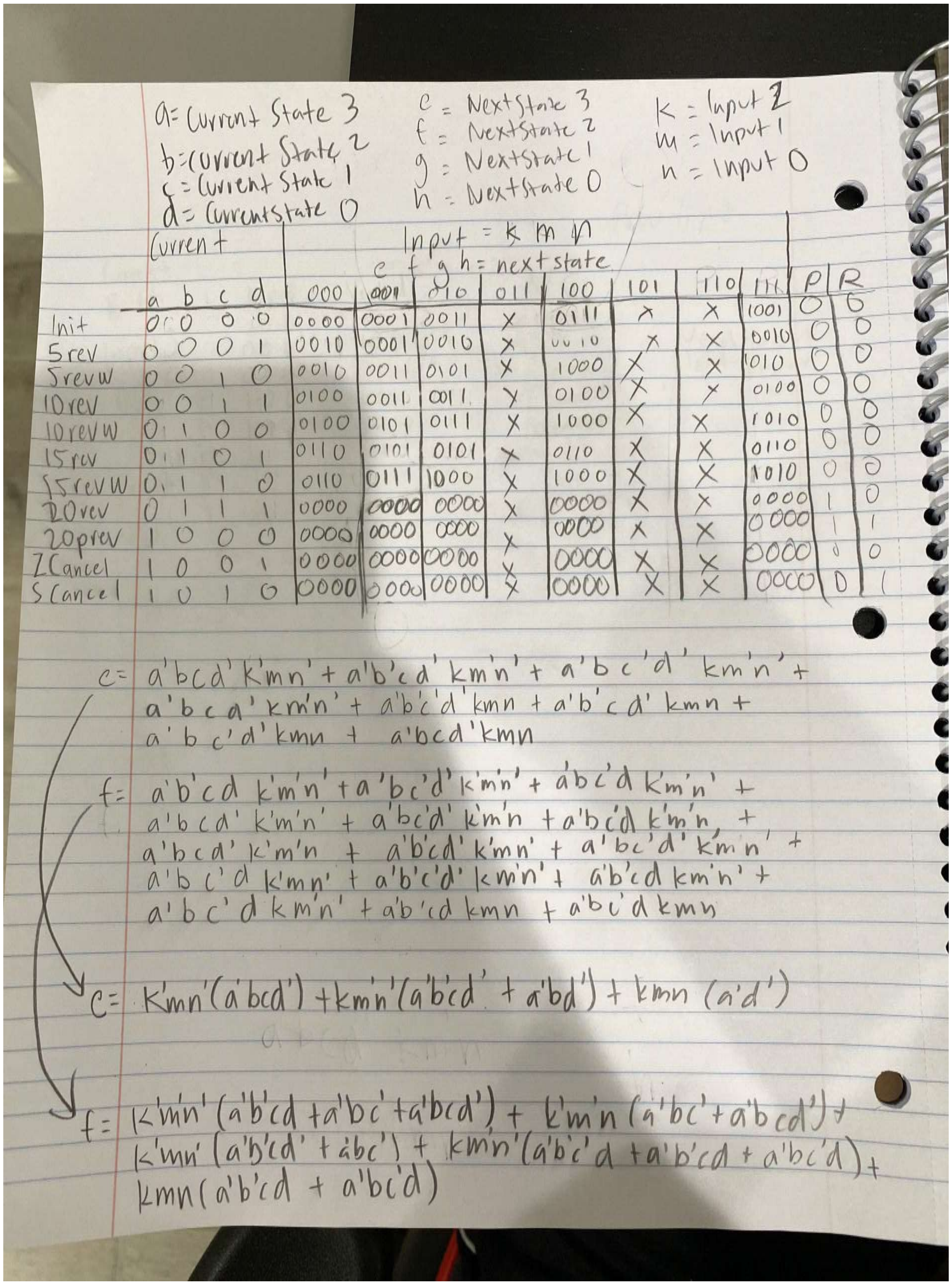
STRUCTURAL OVERVIEW

Replace this text w/ the % you feel you completed the lab. Be sure to list your general procedure of how you completed this lab & material (if any) you reviewed to help you complete this lab. Regardless of % stated, provide any details of difficulties (if any) you encountered during this lab. A few sentences are sufficient.

I think I did this lab with 100%. I spent around 2 hours doing the FSM state and truth tables to ge the boolean equations, becasue I have 12 states which had 4 variables each plus the 3 from the input. Then I spent 1 hour putting them in carefully, but missed something on my notes so spent an hour debugging to finally solve the problem.

LAB 2 TRUTH TABLE(S)

Create a truth table showing **Permit**, **ReturnChange**, and **NextState** based on your FSM. You can use Word, Excel, or even attach a picture of your truth table as long as it is legible. It must be in the correct orientation & legible for full credit.



a = Current State 3
b = Current State 2
c = Current State 1
d = Current State 0

e = Next State 3
f = Next State 2
g = Next State 1
h = Next State 0

k = Input 2
m = Input 1
n = Input 0

	Current				Input = k m n e f g h = next state									
	a	b	c	d	000	001	010	011	100	101	110	111	P	R
Init	0	0	0	0	0000	0001	0011	X	0111	X	X	1001	0	0
5rev	0	0	0	1	0010	0001	0010	X	0010	X	X	0010	0	0
5revW	0	0	1	0	0010	0011	0101	X	1000	X	X	1010	0	0
10rev	0	0	1	1	0100	0011	0011	X	0100	X	X	0100	0	0
10revW	0	1	0	0	0100	0101	0111	X	1000	X	X	1010	0	0
15rev	0	1	0	1	0110	0101	0101	X	0110	X	X	0110	0	0
15revW	0	1	1	0	0110	0111	1000	X	1000	X	X	1010	0	0
20rev	0	1	1	1	0000	0000	0000	X	0000	X	X	0000	1	0
20prev	1	0	0	0	0000	0000	0000	X	0000	X	X	0000	1	1
2Cancel	1	0	0	1	0000	0000	0000	X	0000	X	X	0000	0	0
5Cancel	1	0	1	0	0000	0000	0000	X	0000	X	X	0000	0	1

$$e = a'bcd'k'mn' + a'b'cd'km'n' + a'bc'd'km'n' + a'bca'km'n' + a'b'c'd'kmn + a'b'cd'kmn + a'bc'd'kmn + a'bcd'kmn$$

$$f = a'b'cd\,k'm'n' + a'bc'd'k'm'n' + a'bc'd\,k'm'n' + a'bcd'k'm'n' + a'bc'd'k'm'n + a'bc'd\,k'm'n + a'bcd'k'm'n + a'b'cd'k'mn' + a'bc'd'k'mn' + a'bc'd\,k'mn' + a'b'c'd'km'n' + a'b'cd\,km'n' + a'bc'd\,km'n' + a'b'cd\,kmn + a'bc'd\,kmn$$

$$e = k'mn'(a'bcd') + km'n'(a'b'cd' + a'bd') + kmn(a'd')$$

$$f = k'm'n'(a'b'cd + a'bc' + a'bcd') + k'm'n(a'bc' + a'bcd') + k'mn'(a'b'cd' + abc') + km'n'(a'b'c'd + a'b'cd + a'bc'd) + kmn(a'b'cd + a'bc'd)$$

LAB 2 FINAL EQUATIONS

a = CurrentState3

b = CurrentState2

c = CurrentState1

d = CurrentState0

k = Input2

m = Input1

n = Input0

Permit = a'bcd + ab'c'd'

ReturnChange = ab'd'

NextState3 = k'mn'(a'bcd') + km'n'(a'b'cd' + a'bd') + kmn(a'd')

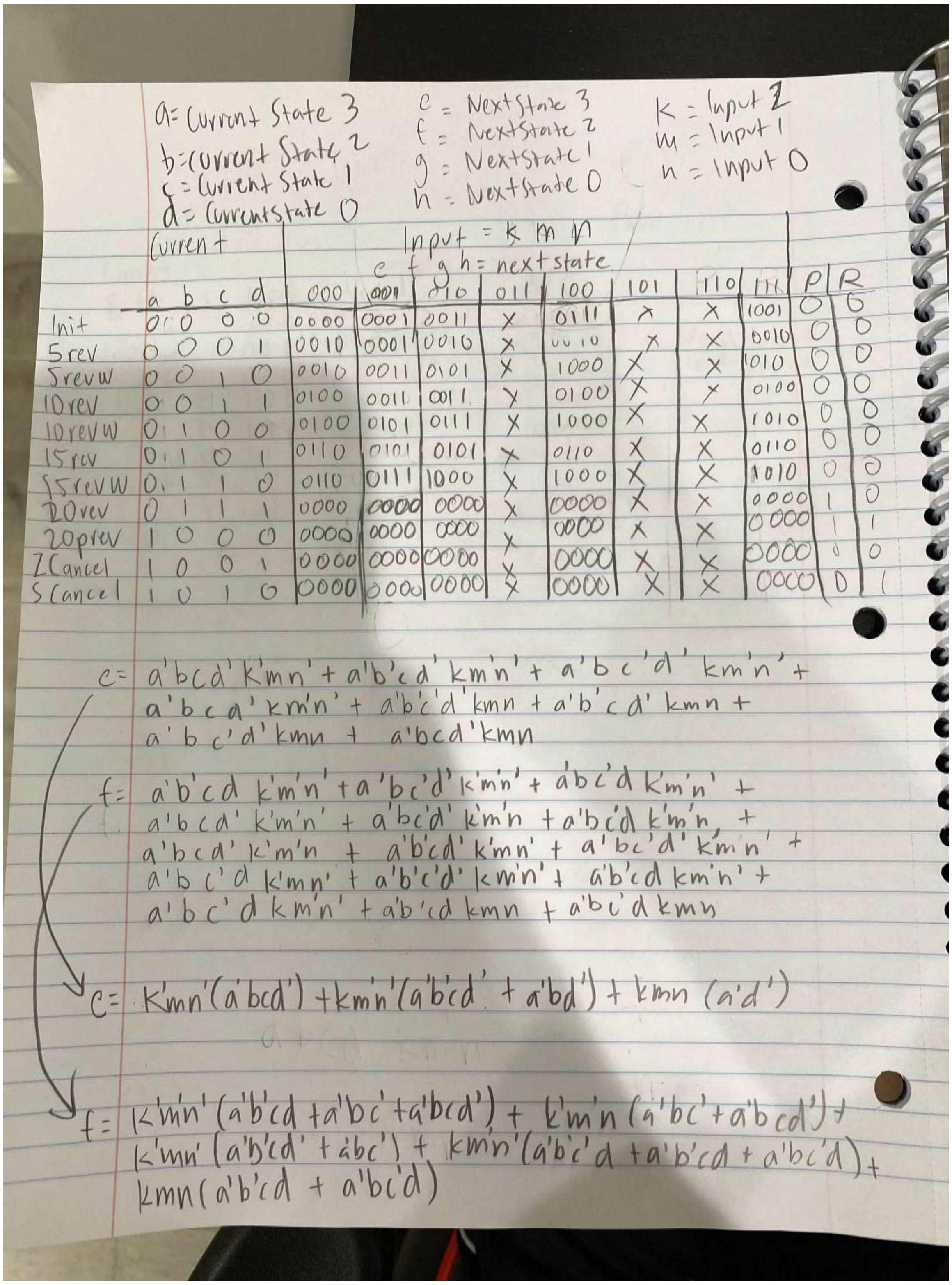
NextState2 = k'm'n'(a'b'cd + a'bc' + a'bcd') + k'm'n(a'bc' + a'bcd') + k'mn'(a'b'cd' + a'bc') + kmn(a'b'cd + a'bc'd)

NextState1 = k'm'n'(a'c'd + a'cd') + k'm'n(a'b'c + a'bcd') + k'mn'(a'c'd' + a'b'd) + km'n'(a'b'c' + a'bc'd) + kmn(a'c'd + a'b'cd' + a'bd')

NextState0 = k'm'n'(a'b' + a'bc' + a'bcd') + k'mn'(a'c'd' + a'b'c + a'bc'd) + km'n'(a'b'c'd') + kmn(a'b'c'd')

Explain any optimizations you may have made to your equations here.

I simplified the current state varable to make it a little easier to type in. The simplification will be on the minimum clock cycle pictures.



a = Current State 3
b = Current State 2
c = Current State 1
d = Current State 0

e = NextState 3
f = NextState 2
g = NextState 1
h = NextState 0

k = Input 2
m = Input 1
n = Input 0

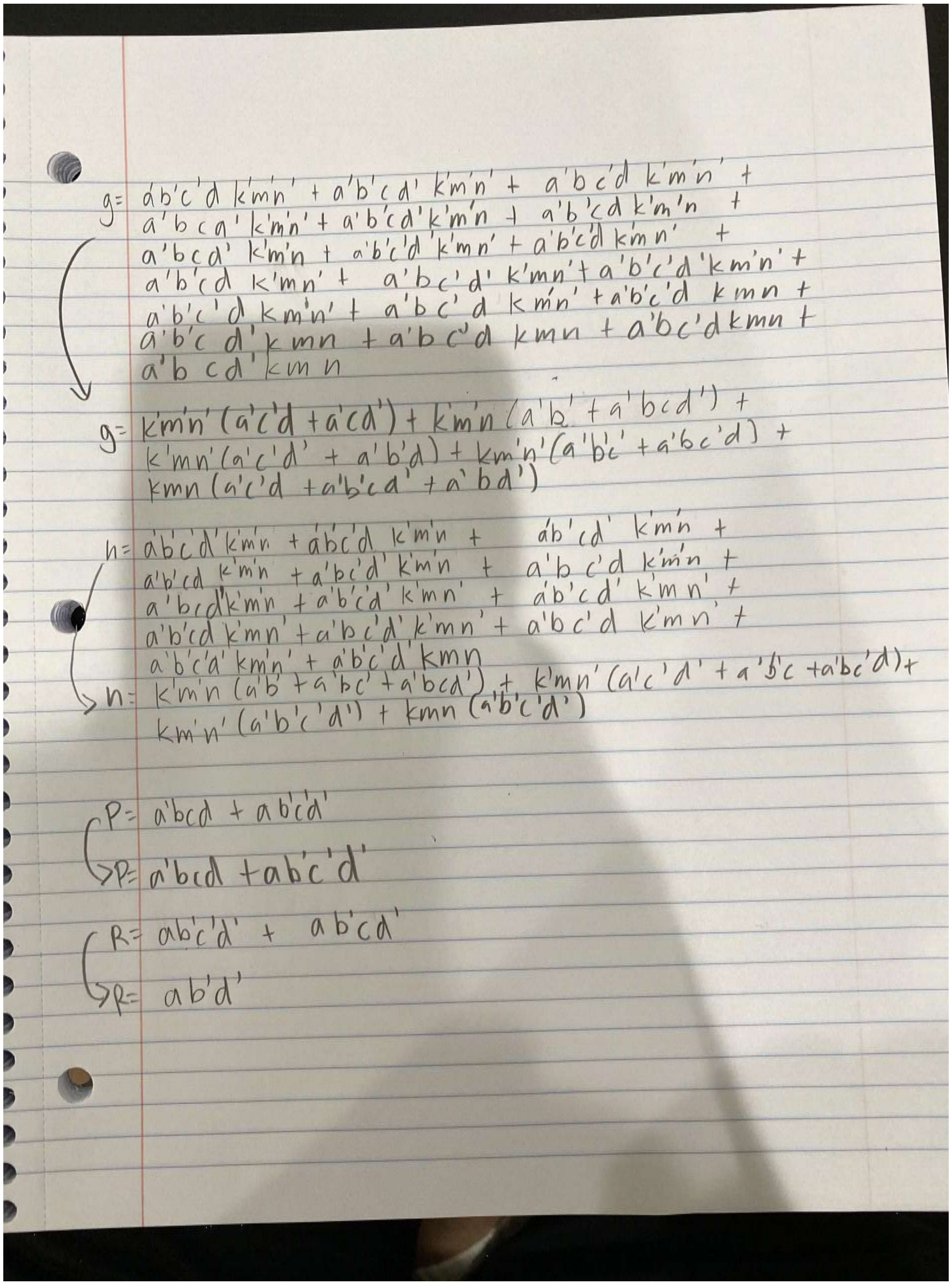
	Current				Input = k m n									
					e f g h = next state									
	a	b	c	d	000	001	010	011	100	101	110	111	P	R
Init	0	0	0	0	0000	0001	0011	X	0111	X	X	1001	0	0
5rev	0	0	0	1	0010	0001	0010	X	0010	X	X	0010	0	0
5revW	0	0	1	0	0010	0011	0101	X	1000	X	X	1010	0	0
10rev	0	0	1	1	0100	0011	0011	X	0100	X	X	0100	0	0
10revW	0	1	0	0	0100	0101	0111	X	1000	X	X	1010	0	0
15rev	0	1	0	1	0110	0101	0101	X	0110	X	X	0110	0	0
15revW	0	1	1	0	0110	0111	1000	X	1000	X	X	1010	0	0
20rev	0	1	1	1	0000	0000	0000	X	0000	X	X	0000	1	0
20prev	1	0	0	0	0000	0000	0000	X	0000	X	X	0000	1	1
ZCancel	1	0	0	1	0000	0000	0000	X	0000	X	X	0000	0	0
SCancel	1	0	1	0	0000	0000	0000	X	0000	X	X	0000	0	1

$$e = a'bcd'k'mn' + a'b'cd'km'n' + a'bc'd'km'n' + a'bca'km'n' + a'b'c'd'kmn + a'b'cd'kmn + a'bc'd'kmn + a'bcd'kmn$$

$$f = a'b'cd\,k'm'n' + a'bc'd'k'm'n' + a'bc'd\,k'm'n' + a'bcd'k'm'n' + a'bc'd'k'm'n + a'bc'd\,k'm'n + a'bcd'k'm'n + a'b'cd'k'mn' + a'bc'd'k'mn' + a'bc'd\,k'mn' + a'b'c'd'km'n' + a'b'cd\,km'n' + a'bc'd\,km'n' + a'b'cd\,kmn + a'bc'd\,kmn$$

$$e = k'mn'(a'bcd') + km'n'(a'b'cd' + a'bd') + kmn(a'd')$$

$$f = k'm'n'(a'b'cd + a'bc' + a'bcd') + k'm'n(a'bc' + a'bcd') + k'mn'(a'b'cd' + a'bc') + km'n'(a'b'c'd + a'b'cd + a'bc'd) + kmn(a'b'cd + a'bc'd)$$



$g = a'b'c'd\,k'm'n' + a'b'cd'\,k'm'n' + a'bc'd\,k'm'n' +$
$a'bca'\,k'm'n' + a'b'cd'\,k'm'n + a'b'cd\,k'm'n +$
$a'bcd'\,k'm'n + a'b'c'd\,k'mn' + a'b'c'd\,km'n' +$
$a'b'cd\,k'mn' + a'bc'd'\,k'mn' + a'b'c'd'\,km'n' +$
$a'b'c'd\,km'n' + a'bc'd\,km'n' + a'b'c'd\,kmn +$
$a'b'cd'\,kmn + a'bc'd\,kmn + a'bc'd\,kmn +$
$a'bcd'\,kmn$

$g = k'm'n'(a'c'd + a'cd') + k'm'n(a'b' + a'bcd') +$
$k'mn'(a'c'd' + a'b'd) + km'n'(a'b'c' + a'bc'd) +$
$kmn(a'c'd + a'b'cd' + a'bd')$

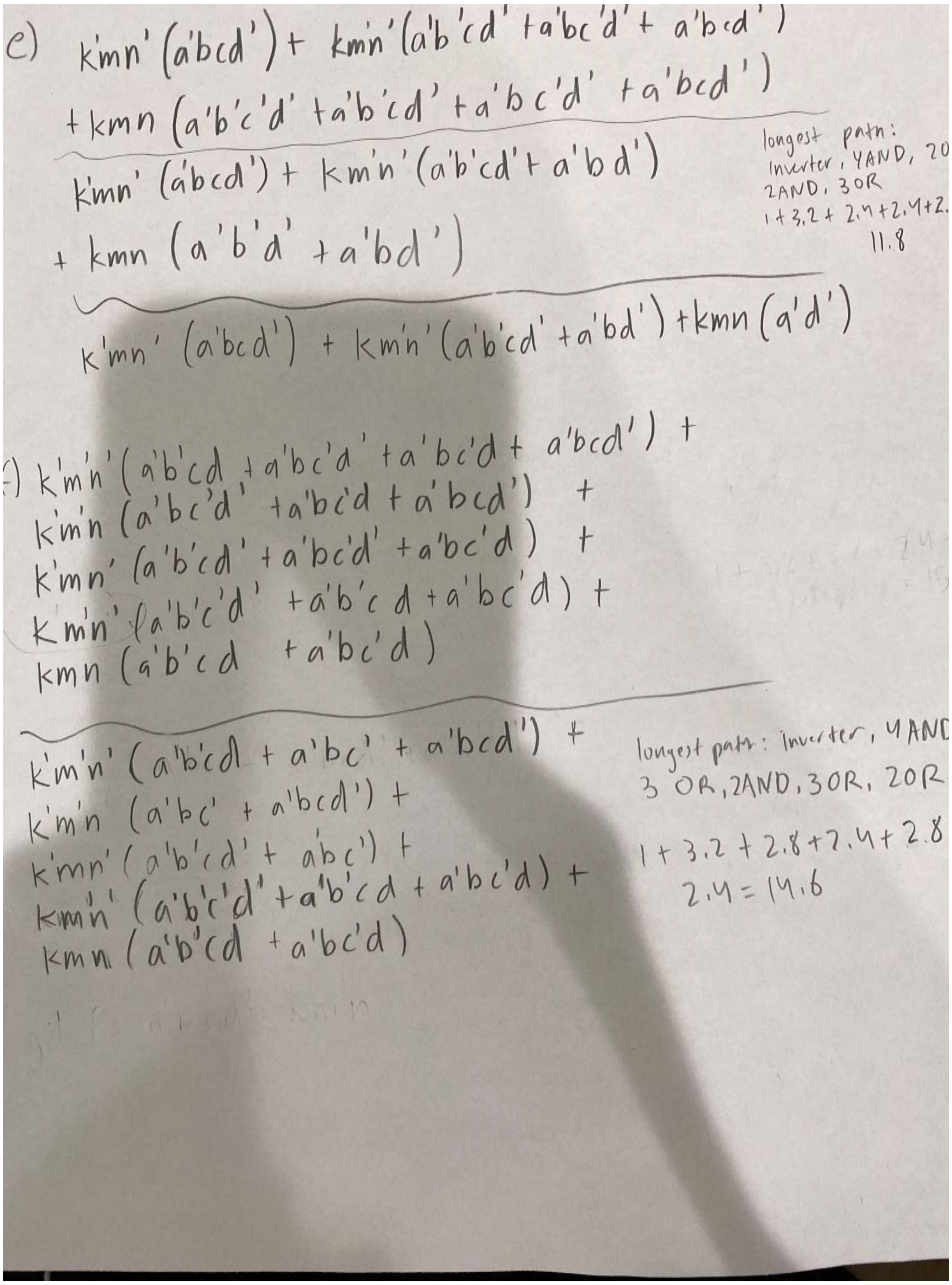
$h = a'b'cd'\,k'm'n + a'bc'd\,k'm'n + a'b'cd'\,k'm'n +$
$a'b'cd\,k'm'n + a'bc'd'\,k'm'n + a'bc'd\,k'm'n +$
$a'bcd\,k'm'n + a'b'cd'\,k'mn' + a'b'cd'\,k'mn' +$
$a'b'cd\,k'mn' + a'bc'd'\,k'mn' + a'bc'd\,k'mn' +$
$a'b'c'd'\,km'n' + a'b'c'd'\,kmn$
$h = k'm'n(a'b' + a'bc' + a'bcd') + k'mn'(a'c'd' + a'b'c + a'bc'd) +$
$km'n'(a'b'c'd') + kmn(a'b'c'd')$

$P = a'bcd + ab'cd'$

$P = a'bcd + ab'c'd'$

$R = ab'c'd' + ab'cd'$

$R = ab'd'$



e) $k'mn'(a'bcd') + km'n'(a'b'cd' + a'bc'd' + a'bcd')$

$+ kmn(a'b'c'd' + a'b'cd' + a'bc'd' + a'bcd')$

$k'mn'(a'bcd') + km'n'(a'b'cd' + a'bd')$

$+ kmn(a'b'd' + a'bd')$

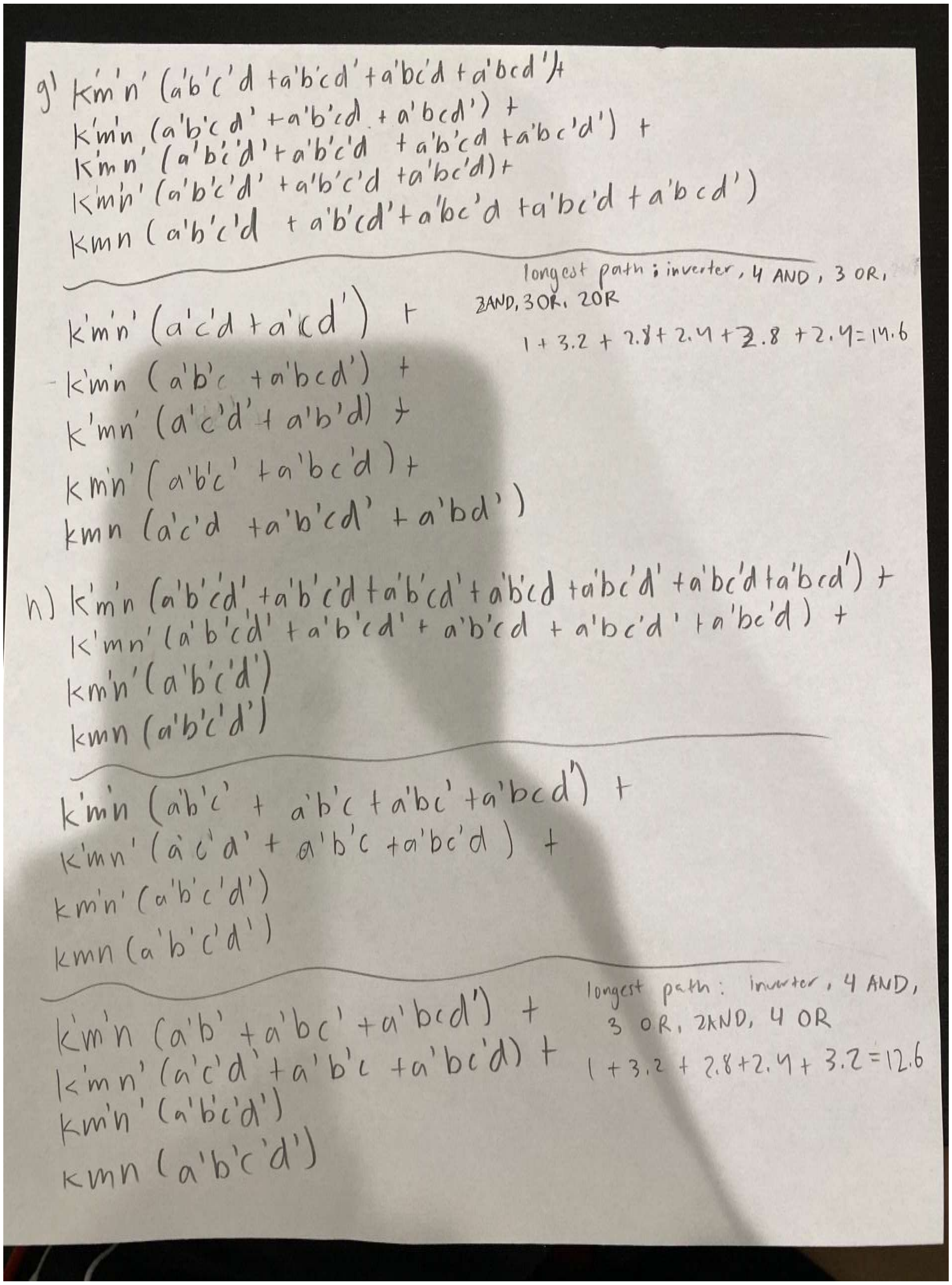
$k'mn'(a'bcd') + km'n'(a'b'cd' + a'bd') + kmn(a'd')$

longest path:
Inverter, 4AND, 2O
2AND, 3OR
1 + 3.2 + 2.4 + 2.4 + 2.
11.8

f) $k'm'n'(a'b'cd + a'bc'd' + a'bc'd + a'bcd') +$
$k'm'n(a'bc'd' + a'bc'd + a'bcd') +$
$k'mn'(a'b'cd' + a'bc'd' + a'bc'd) +$
$km'n'(a'b'c'd' + a'b'cd + a'bc'd) +$
$kmn(a'b'cd + a'bc'd)$

$k'm'n'(a'b'cd + a'bc' + a'bcd') +$
$k'm'n(a'bc' + a'bcd') +$
$k'mn'(a'b'cd' + a'bc') +$
$km'n'(a'b'c'd' + a'b'cd + a'bc'd) +$
$kmn(a'b'cd + a'bc'd)$

longest path: inverter, 4AND
3 OR, 2AND, 3OR, 2OR
1 + 3.2 + 2.8 + 2.4 + 2.8
2.4 = 14.6



g) $k'm'n'(a'b'c'd + a'b'cd' + a'bc'd + a'bcd') +$
$k'm'n(a'b'cd' + a'b'cd + a'bcd') +$
$k'mn'(a'b'c'd' + a'b'c'd + a'b'cd + a'bc'd') +$
$km'n'(a'b'c'd' + a'b'c'd + a'bc'd) +$
$kmn(a'b'c'd + a'b'cd' + a'bc'd + a'bc'd + a'bcd')$

$k'm'n'(a'c'd + a'cd') +$
$k'm'n(a'b'c + a'bcd') +$
$k'mn'(a'c'd' + a'b'd) +$
$km'n'(a'b'c' + a'bc'd) +$
$kmn(a'c'd + a'b'cd' + a'bd')$

longest path: inverter, 4 AND, 3 OR, 3AND, 3OR, 2OR

$1 + 3.2 + 2.8 + 2.4 + 2.8 + 2.4 = 14.6$

h) $k'm'n(a'b'cd' + a'b'c'd + a'b'cd' + a'b'cd + a'bc'd' + a'bc'd + a'bcd') +$
$k'mn'(a'b'cd' + a'b'cd' + a'b'cd + a'bc'd' + a'bc'd) +$
$km'n'(a'b'c'd')$
$kmn(a'b'c'd')$

$k'm'n(a'b'c' + a'b'c + a'bc' + a'bcd') +$
$k'mn'(a'c'd' + a'b'c + a'bc'd) +$
$km'n'(a'b'c'd')$
$kmn(a'b'c'd')$

$k'm'n(a'b' + a'bc' + a'bcd') +$
$k'mn'(a'c'd' + a'b'c + a'bc'd) +$
$km'n'(a'b'c'd')$
$kmn(a'b'c'd')$

longest path: inverter, 4 AND, 3 OR, 2AND, 4 OR

$1 + 3.2 + 2.8 + 2.4 + 3.2 = 12.6$

LAB 2 MINIMUM CLOCK CYCLE

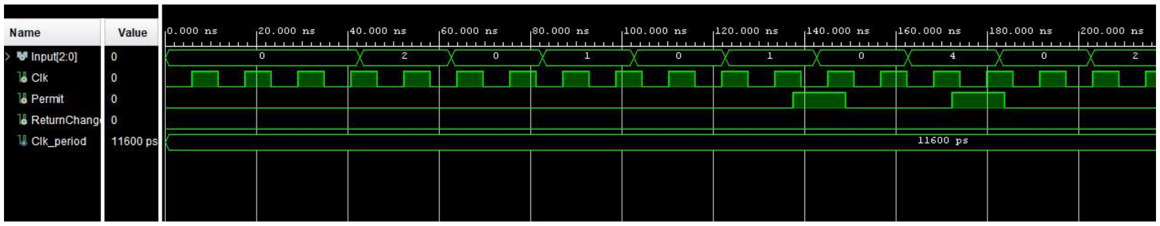
Minimum clock cyle: 10.6

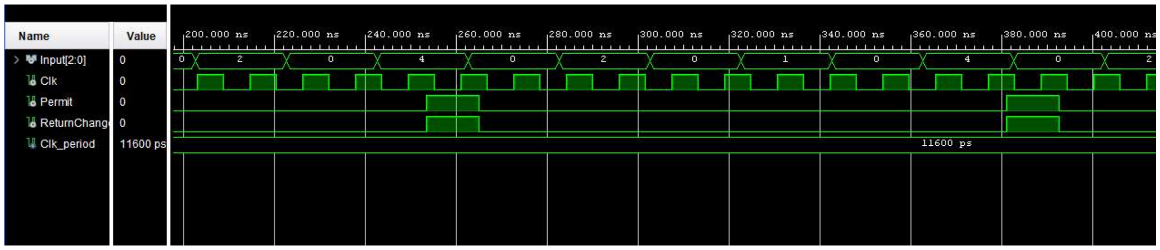
Explain how you derived your minimum clock cycle (as discussed in EECS 31) here.

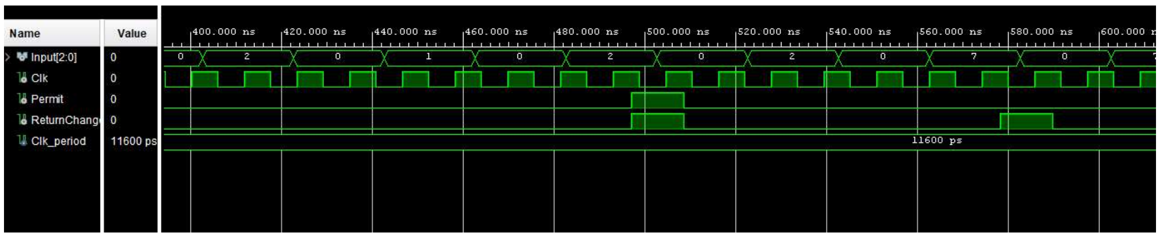
I derived the minumum clock delay throguht the dealy of the comblogic and state resister, the comblogis has a dealy of 5.6ns and state register has 4nd delay plus 1ns setup. To find the minimum clock syscle we have to look at the register to register longest path which in this case would be the register's 5ns plus the 5.6ns from the comb logic then back to the register, so it would be 5.6 + 5 = 10.6 ns.

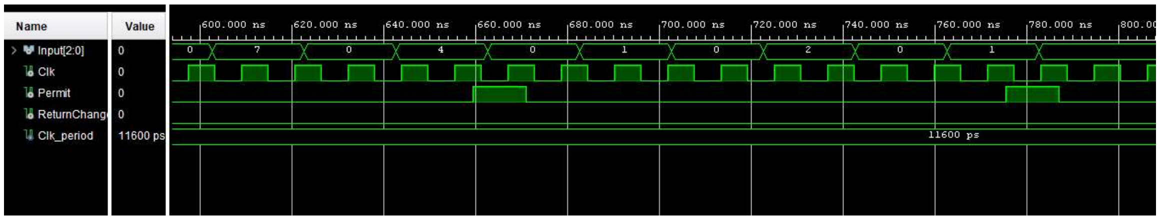
LAB 2 STRUCTURAL SIMULATION GRAPH

Show a screenshot of your final graph here. You should crop it to the appropriate size so that it is legible.









LAB 2 STRUCTURAL AND BEHAVIORAL SIMULATION GRAPH COMPARISONS

Compare your behavioral & structural graphs here. If there are any differences (delays, outputs, etc.), be sure to explain them here.

There is a difference in the graphs because now the Permit and ReturnChange are slightly delayed due to the delay of the State Register and the delay of the CombLogic. The clock cycles are also longer becasue they must not violate the setup and holdtime of the comblogic and state registers.