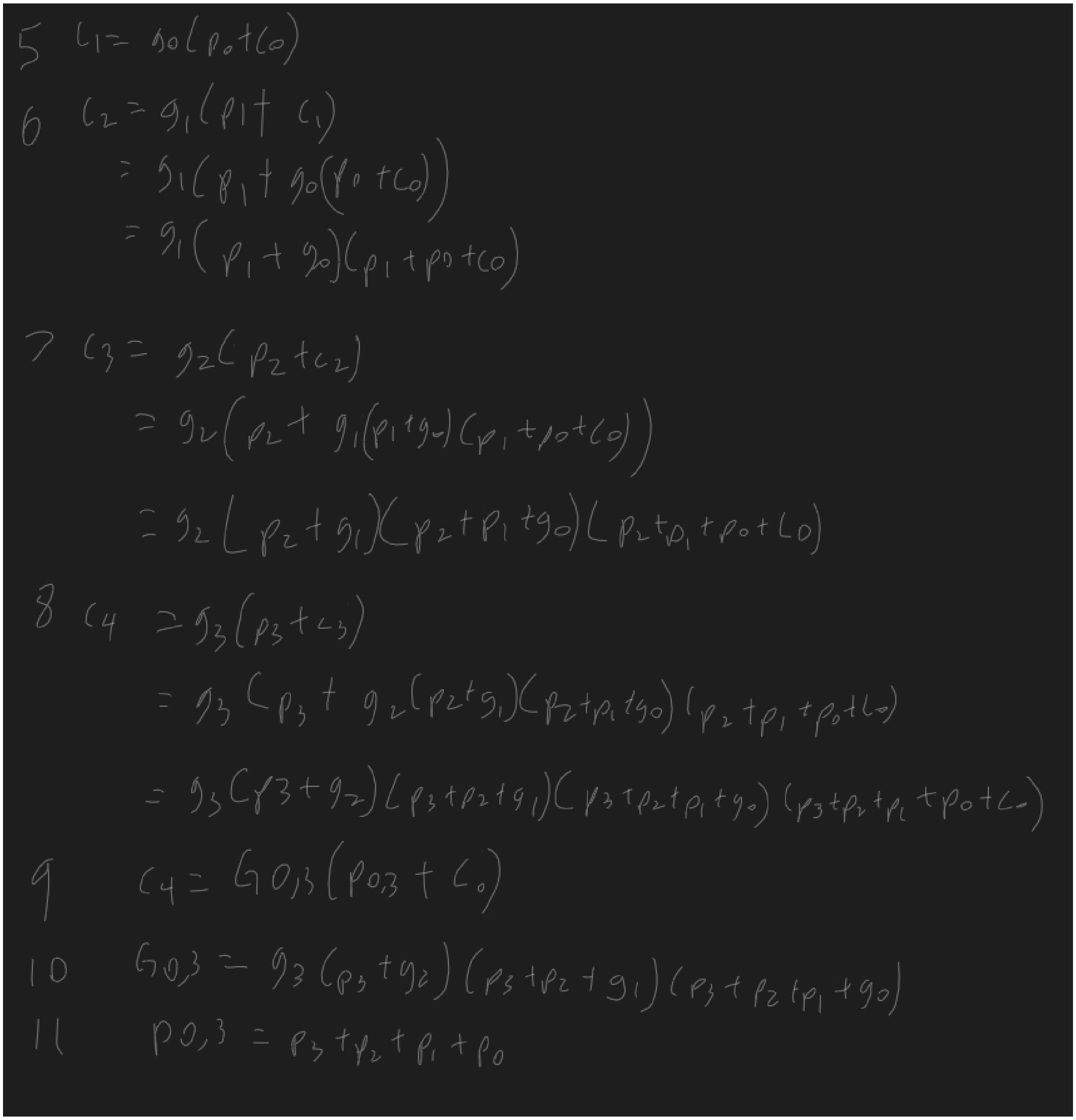
As a final step, you need to derive the new CLA equations (5)-(11) with the alternative sum and carry equations. You will be responsible for obtaining the equations through similar steps used in the derivation of (5)-(11) and constructing a 4-bit CLA adder with the alternative adder equations instead of the standard ones. Since the initial equations for the conventional form in Equations (3)-(4), and the equations in (16)-(17) are so similar, except for the interchange of the operations, you should find it quite straightforward to derive the alternative CLA block formulations.



5 $c_1 = g_0(p_0+c_0)$

6 $c_2 = g_1(p_1+c_1)$

$= g_1(p_1+g_0(p_0+c_0))$

$= g_1(p_1+g_0)(p_1+p_0+c_0)$

7 $c_3 = g_2(p_2+c_2)$

$= g_2(p_2+g_1(p_1+g_0)(p_1+p_0+c_0))$

$= g_2(p_2+g_1)(p_2+p_1+g_0)(p_2+p_1+p_0+c_0)$

8 $c_4 = g_3(p_3+c_3)$

$= g_3(p_3+g_2(p_2+g_1)(p_2+p_1+g_0)(p_2+p_1+p_0+c_0)$

$= g_3(p3+g_2)(p_3+p_2+g_1)(p_3+p_2+p_1+g_0)(p_3+p_2+p_1+p_0+c_0)$

9 $c_4 = G_{0,3}(p_{0,3}+c_0)$

10 $G_{0,3} = g_3(p_3+g_2)(p_3+p_2+g_1)(p_3+p_2+p_1+g_0)$

11 $p_{0,3} = p_3+p_2+p_1+p_0$

What is unique about these test vectors? Does any other test vector work to test the maximum delays in these paths? Please fill in Delay Tables 1-4 with your measured results (and fill in delay_tables.toml). Are they consistent with your expectations?

Path Delay of 4-bit CLA Binary Adder
Test Vector 1
X: 1000
Y: 0111
C0: 0 → 1

Delay Table 1

TOML Code	Path	Delay
d1.a	c0-c1	10
d1.b	c0-c2	11
d1.c	c0-c3	12
d1.d	c0-c4	10

Test Vector 2
X: 1000 → 1001
Y: 0111
C0 : 0

Delay Table 2

TOML Code	Path	Delay
d2.a	x0-c1	17
d2.b	x0-c2	18
d2.c	x0-c3	19
d2.d	x0-c4	24

Path Delay of 16-bit CLA Binary Adder
Test Vector 1
X: 1000100010001000
Y: 0111011101110111
C 0: 0 → 1

Delay Table 3

TOML Code	Path	Delay (Single-Level)	Delay (Two-Level)
d3.a	c0-c4	10	10
d3.b	c0-c8	20	11
d3.c	c0-c12	30	12
d3.d	c0-c16	40	10
d3.e	c0-s3	19	19
d3.f	c0-s7	29	29
d3.g	c0-s11	39	30
d3.h	c0-s15	49	31

Test Vector 2
X: 1000100010001000 → 1000100010001001
Y: 0111011101110111
C0: 0

Delay Table 4

TOML Code	Path	Delay (Single-level)	Delay (Two-level)
d4.a	x0-c4	24	24
d4.b	x0-c8	34	25
d4.c	x0-c12	44	26
d4.d	x0-c16	54	54
d4.e	x0–s3	26	26
d4.f	x0-s7	43	43
d4.g	x0-s11	53	44
d4.h	x0-s15	63	45

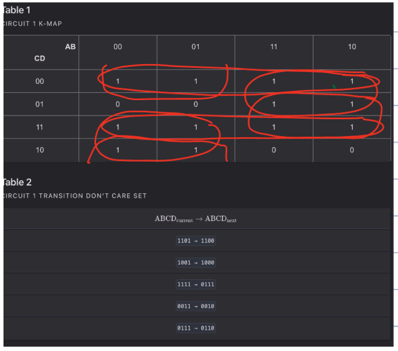
The unique thing about these test vectors is that they all sum up to the maximum value allowed by the current implementation of the adder. We then test the circuit by adding one, which makes it go over the maximum value by 1. Other test vectors that would also test the maximum delay of these adders would be having an X and a Y that sum up to 16. The delay tables are consistent with our expectation because the two layer adder is significantly faster than the one layer adder. This shows that having a look ahead to parallelize the calculations will speed it up.

All the steps from above must be clearly presented in your report Your report needs to clearly state how the Transition Don't Cares reduce the necessary redundant gates in each circuit, and to present the corresponding Karnaugh map representation according to the instructions outlined above.

Asdf

Part 4:

1)



able 1

IRCUIT 1 K-MAP

AB CD	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	1	1	1	1
10	1	1	0	0

able 2

IRCUIT 1 TRANSITION DON'T CARE SET

$ABCD_{current} \rightarrow ABCD_{next}$
1101 → 1100
1001 → 1000
1111 → 0111
0011 → 0010
0111 → 0110

2)

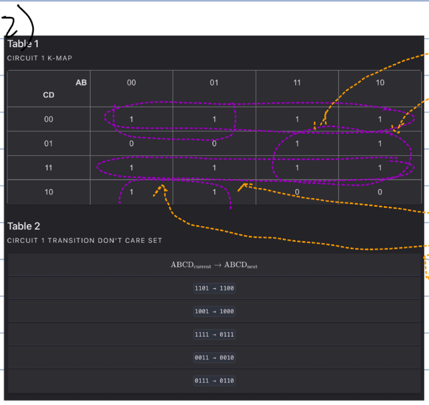


Table 1

CIRCUIT 1 K-MAP

AB CD	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	1	1	1	1
10	1	1	0	0

Table 2

CIRCUIT 1 TRANSITION DON'T CARE SET

$ABCD_{current} \rightarrow ABCD_{next}$
1101 → 1100
1001 → 1000
1111 → 0111
0011 → 0010
0111 → 0110

3)

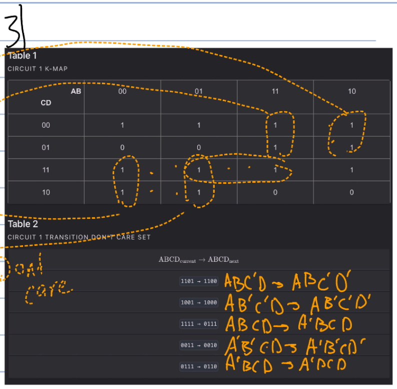


Table 1

CIRCUIT 1 K-MAP

AB CD	00	01	11	10
00	1	1	1	1
01	0	0	1	1
11	1	1	1	1
10	1	1	0	0

Table 2

CIRCUIT 1 TRANSITION DON'T CARE SET

Dont care

$ABCD_{current} \rightarrow ABCD_{next}$	
1101 → 1100	$ABC'D \rightarrow ABC'D'$
1001 → 1000	$AB'C'D \rightarrow AB'C'D'$
1111 → 0111	$ABCD \rightarrow A'BCD$
0011 → 0010	$A'B'CD \rightarrow A'B'CD'$
0111 → 0110	$A'BCD \rightarrow A'BCD$

$$A'D' + AD + C'D' + CD$$

1)

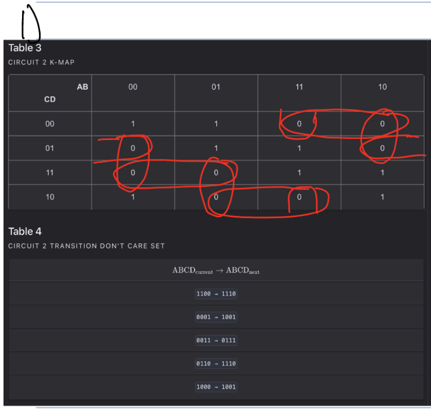


Table 3

CIRCUIT 2 K-MAP

AB CD	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	0	0	1

Table 4

CIRCUIT 2 TRANSITION DON'T CARE SET

$ABCD_{current} \rightarrow ABCD_{next}$
1100 → 1110
0001 → 1001
0011 → 0111
0110 → 1110
1000 → 1001

2)

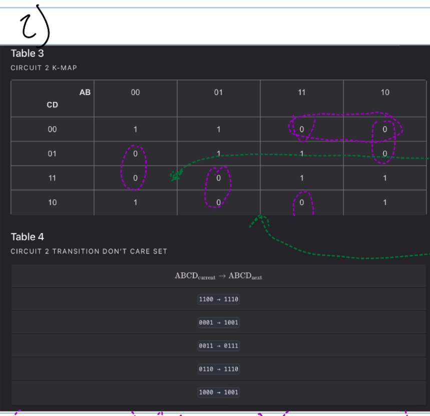


Table 3

CIRCUIT 2 K-MAP

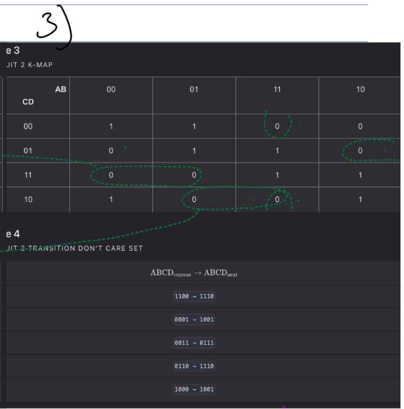
AB CD	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	0	0	1

Table 4

CIRCUIT 2 TRANSITION DON'T CARE SET

$ABCD_{current} \rightarrow ABCD_{next}$
1100 → 1110
0001 → 1001
0011 → 0111
0110 → 1110
1000 → 1001

3)



e 3

JIT 2 K-MAP

AB CD	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	0	0	1

e 4

JIT 2 TRANSITION DON'T CARE SET

$ABCD_{current} \rightarrow ABCD_{next}$
1100 → 1110
0001 → 1001
0011 → 0111
0110 → 1110
1000 → 1001

$$(A+B'+C')(A'+B+C)(A+B+D')(A'+B'+D)(A'+C+D)$$