Logic Circuits

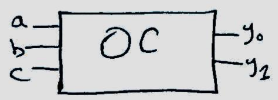
Computer Assignments #1 & #2

«تمرین‌ها/حل»

۸۱۰۱۰۱۴۷۶ کوروش علی‌نقی

810201476 Kaurosh Alinaghi

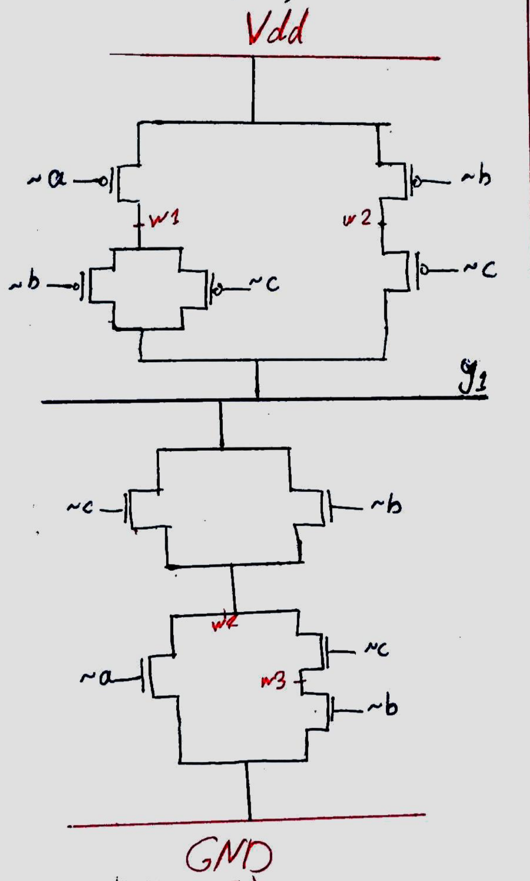
1



$$\Rightarrow y_0 = \underbrace{abc}_{3\text{ ones}} + \underbrace{\bar a\bar b c + \bar a b\bar c + a\bar b\bar c}_{1\text{ one}} = \underline{a(bc+\bar b\bar c)+\bar a(\bar b c + b\bar c)}$$

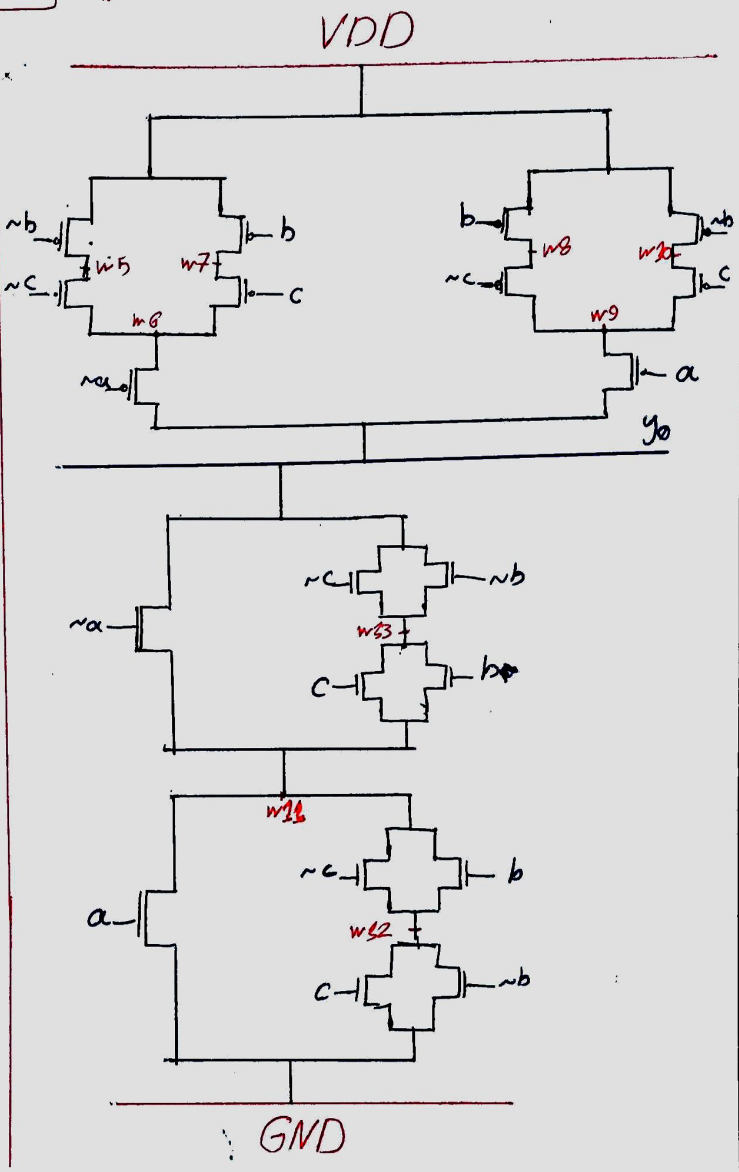
$$y_1 = \underbrace{ab+bc+ac}_{\text{more then 1 ones (2 or 3)}} = \underline{a(b+c)+ac}$$

* wires are mentioned as they are in Verilog desc.



pMOS #(5, 6, 7)

nMOS #(3, 4, 5)



2×5 + 2×10 = 30 Transistors

Worst Case Delay Values:

$Y_0 \Rightarrow$ To1 $\Rightarrow$ 3× ToZ:NMOS = 3×5, 15 ns

To0 $\Rightarrow$ 2× ToZ:pMOS = 2×7, 14 ns

Y_1: To1: 4× ToZ nMOS = 4×5 = 20 ns

To0: 3× ToZ pMOS, 3×7, 21 ns

#Y_0 (15, 14)

#Y_1 (20, 21)

1

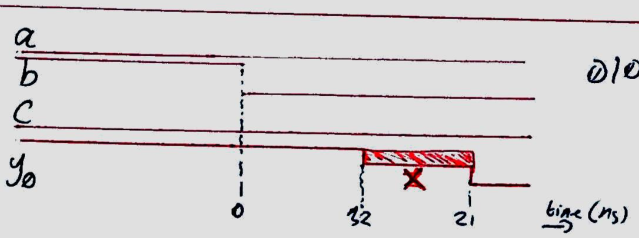
Transitions with worst case delays:

Y_0 : To1 : 011 ⟶ 010 : 20ns | Y_1 : To1 : 100 ⟶ 110 : 15 ns

To0 : 010 ⟶ 000 : 21 ns | To0 : 110 ⟶ 010 : 14 ns

2 Waveforms:

Y_0 To0:



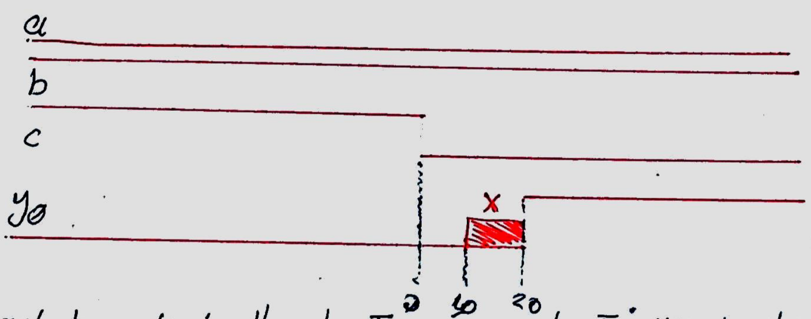
010 ⟶ 000

To2 pMOS

~~At~~ At t = 12 ns, the nMOS network conducts a path and sends 0 to output (12 = 3× nMOS To0 = 3×4) but the pMOS side's delay to stop ~~c~~ conducting is 3×7 = 21 so it sends 1 to output all the way until t = 21 ⟹ so at 12 < t < 21 ⟹ output is X.

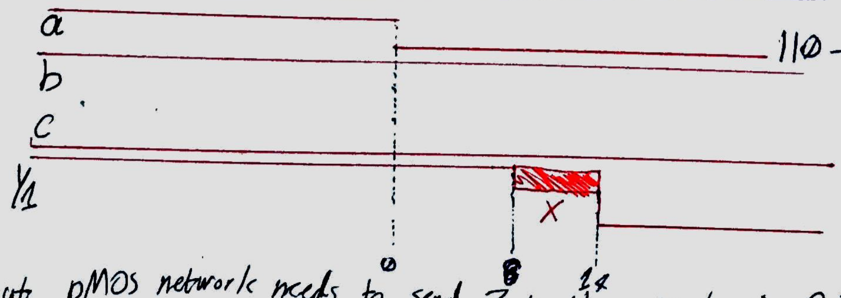
011 ⟶ 010

Y_0 To1 :



One should get to output through $\bar{b}ca$ ~~=~~ path. $\bar{b}$ pMOS is already conducting. so the delay of going 1 to output is 2×To1 for pMOS (for a and c pMOSes) = 2×5 = 10 ⟹ at t = 10 ⟶ one goes to y_0 but it takes nMOS networks ~~4~~ 4×ToZ nMOS = 4×5 = 20 ns to send Z to output ⟹ in 10 < t < 20 the output is X.

Y_1 To0



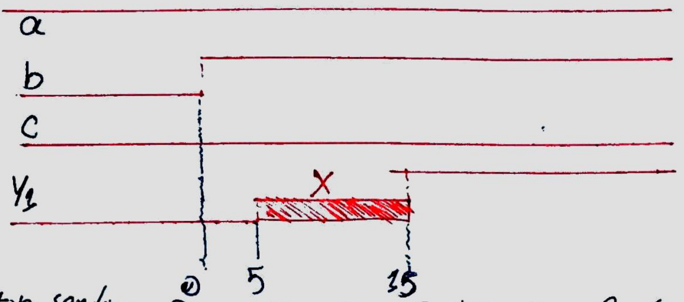
110 ⟶ 010

The time that pMOS network needs to send Z to the output is 2×ToZ pMOS = 2×7 = 14 ns. but nMOS network that was sending Z to the output, starts sending 0 to Y_1 after 2× To0 nMOS = 2×4 = 8 ns. ⟹ in 8 < t < 14 we have X.

Logic Circuits   ۸۱۰۱۰۱۴۷۶   کوروش علی‌نقی

CA 1&2   810101476   Kourosh Alinaghi

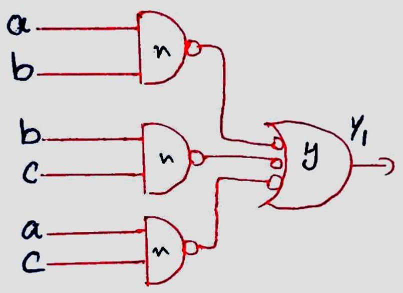
2 Y_1 To2:

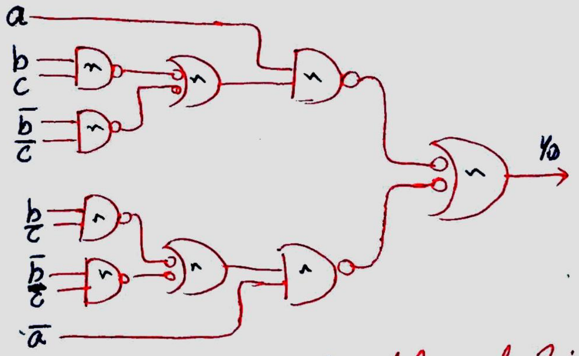


$100 \to 110$

nMOS network stop sending 0 to Y_1 after $3 \times$ ToZ nMOS $= 3 \times 5 = 15$ ns. But pMOS network sends 1 to Y_1 after 5ns.(becase ~a was already conducting ⇒ delay is the time the ~b pMOS needs to conduct ($1 \times$ To1 pMOS $= 5$ns) ⇒ output in $5 < t < 15$ is X

3 The questions wants us to define delays for gates such that the delay of our gate level circuits gets as close as possible to CMOS. (Without mentioning any minimization in gate level.) ⇒ I'm gonna describe my circuit using 2 and 3input NAND gates.





worst case delay for Y_0, Y_0 To1 $= 20 = 4 \times n \Rightarrow n = 5$ delay of 2input NAND

worst case delay for Y_1, Y_1 To2 $= 15$, $n + y = 5 + y \Rightarrow y = 10$ delay of 3-input NAND

Becaus we used 20 and 15 as our total delays, the worst case delay for our gate level circuit is 20 an 15 for same transition mentioned in part 1. Actually is 15 for Y_1 and 20 for Y_0 in any tranition. The point is that this is unrealistic. Even though we used small delays for our NAND gates (that cannot be achieved using the same pMOS and nMOS delays), our average delay in all transitions is way more than the CMOS we created.

4

Again, we use 20 ns for Y_0 and 15 ns for Y_1. This is exactly what we had in our gate level circuit. So there are no differences between expression level description and gate level. And the exact same differences mentiond in Part 3, hold between ExpLvL. and TrLvL.

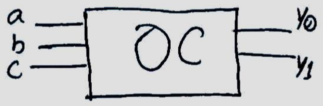
$$Y_0 = a(bc + \bar{b}\bar{c}) + \bar{a}(\bar{b}c + b\bar{c})$$

$$Y_1 = a(b+c) + bc \equiv ab + ac + ac$$

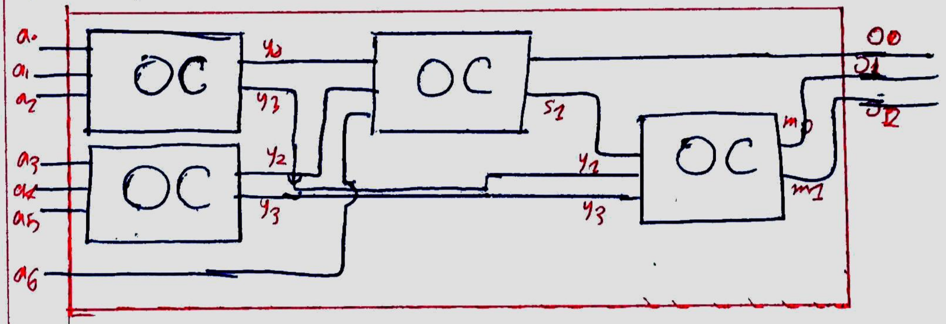
Section 2

5 So far, we've built an One Counter:

a	b	c	Y_0	Y_1
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



The 7input OC can be created in a structural fashion like this:



6 In assign statements, we used 20 ns for Y_0 and 15 ns for Y_1. this the delay that's gonna take for every transition. $\Rightarrow$ the delay of our 7input OC is

O0: 40 ns (20+20)
O1: 55 ns (20+15+20)
O2: 30 ns (15+15)

$\Rightarrow$ 7-input OC has a worst-case delay of 55 ns.

But this is because we used assign statement and took abstraction one step higher. If we implement or 7input OC using 3-input CMOS OC, we'll have less average delay.

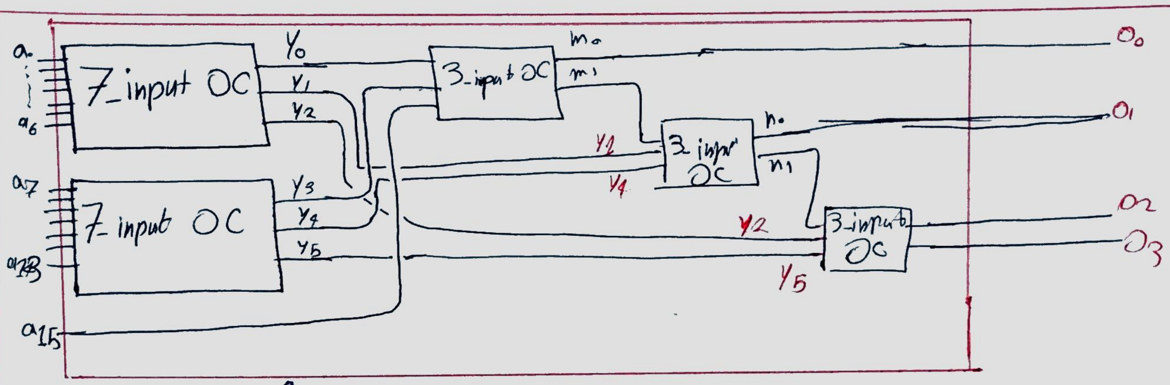
logic circuit

CA 1 and 2

۸۱۰۱۰۱۴۷۶ کوروش علینقی

810101476 Kourosh Alinaghi

7



Worst case delay for:

$O_0 : 40+20 = \underline{60\,ns}$

$O_1 : 20 + \max(15+40, 55) = 20+55 = \underline{75\,ns}$

$O_2 : 20 + \underbrace{\max(30, 30, 15+55)} = 20+70 = \underline{90\,ns}$

$O_3 : 15 + \quad\swarrow \quad = 15+70 = \underline{85\,ns}$

$\Longrightarrow$ Worst case delay for 15_input OC is 90ns

★ I've written a C++ script to test 7InputOC and 15InputOC. I'll upload that with Verilog files.