

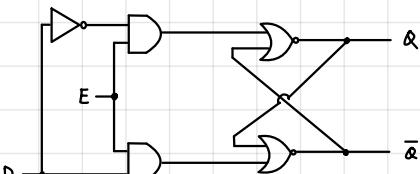
1. (5%) Given the following codes, which one will have the **most gate counts** after synthesis? Please briefly explain your reasons.

a)	b)	c)	d)
input en, D; reg Q; always@(en or D) if (en) Q = D; else Q = 0;	input en, D; reg Q; always@(en or D) if (en) Q = D; else Q = 0;	input en, D; reg Q; always@(en or D) Q = 0;	input en, D; output Q; always@(en or D) assign Q = en? D: 0;

Logic Synthesis - Part 1 P. 21

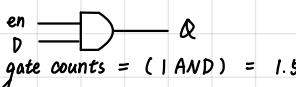
Gate	# Transistor	Gate counts
INV	2	0.5
NAND	4	1
NOR	4	1
AND	6	1.5
OR	6	1.5

a) D Latch



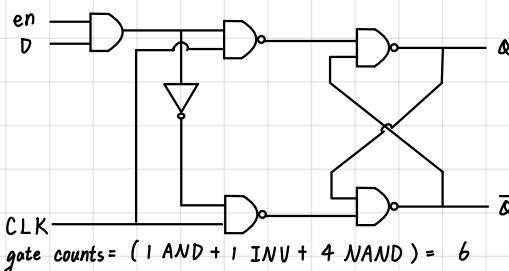
$$\text{gate counts} = (1 \text{ INV} + 2 \text{ AND} + 2 \text{ NOR}) = 5.5$$

b), c), d) Mux



So, a) will have the most gate counts after synthesis. *

[Hint] D-Flip Flop w/ enable



$$\text{gate counts} = (1 \text{ AND} + 1 \text{ INV} + 4 \text{ NAND}) = 6$$

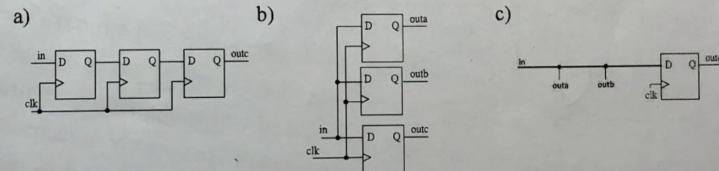


2. (5%) Given the Verilog code shown right, which one is most likely the corresponding circuit after synthesis? Please briefly explain your reasons.

```
input in;
output outa, outb, outc;
always @(posedge clk) begin
    outa = in;
    outb = outa;
    outc = outb;
end
```

Blocking dX

none of the above



<sol>

Logic Synthesis - Part 1 P. 17

D , because the code is blocking assignment , there are executed sequentially.
outa, outb & outc will be the same value , so they will be synthesized as a common wire. *

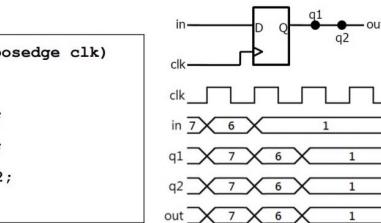
Assignment in Sequential Circuit

✓ Blocking assignment

- Evaluations and assignments are **immediate** and **in order**
- Syntax : <variable> = <expression>;

✓ Example

```
always @ (posedge clk)
begin
    q1 = in;
    q2 = q1;
    out = q2;
end
```



NCLAB NCTU Institute of Electronics



19

3. (5%) If a two-level circuit is implemented to a multi-level circuit, what are the possible advantages and disadvantages of this change?

<sol>

Logic Synthesis - Part 1 P. 78 (2-level circuit vs. multi-level circuit)

Multi-level circuit

• Advantages

- Has fewer gates (small circuit)
- Reduced fan-in
- Less wires

• Disadvantages

- More difficult design
(No simple method like K-map or QM)
- No global definition of "optimal"
- More gate delays (slower circuit)

*

4. Consider the logic network defined by the following expressions:

$$x = abc + abd + ac'd' + b'c'd'$$

$$y = ace + ade + b'ce + b'de$$

$$z = c + d$$

(a) Perform both x/z and y/z (weak division) and show the quotients and the remainders. (10%)

(b) Find the common sub-expression between x and y , and redraw the Boolean network. (6%)

Logic Synthesis - Part 1 P. 89 (weak division)

$$\begin{aligned} f &= pq + r \\ q &= \bigwedge_i V^{f_i} \\ r &= f - pq \end{aligned}$$

Logic Synthesis - Part 1 P. 93 (common sub-expression)

$\langle sol \rangle$

(a) x/z :

$$\begin{aligned} x &= \underline{\underline{abc + abd + ac'd' + b'c'd'}} \\ z &= c + d \end{aligned}$$

$$V^c = ab$$

$$V^d = ab$$

$$q = ab$$

$$zq = (c+d)(ab)$$

$$= \underline{\underline{abc + abd}}$$

$$r = ac'd' + b'c'd'$$

$$\begin{aligned} y/z: \\ y &= \underline{\underline{ace + ade + b'ce + b'de}} \\ z &= c + d \end{aligned}$$

$$V^c = ae + b'e$$

$$V^d = ae + b'e$$

$$q = ae + b'e$$

$$zq = (c+d)(ae + b'e)$$

$$= \underline{\underline{ace + b'ce + ade + b'de}}$$

$$r = 0$$

$$\therefore x = zq + r$$

$$abc + abd + ac'd' + b'c'd'$$

$$= (c+d)(ab) + (ac'd' + b'c'd') \quad *$$

$$= \underline{\underline{(c+d)(ab) + c'd'(a+b')}}$$

$$\therefore y = zq + r$$

$$ace + ade + b'ce + b'de$$

$$= (c+d)(ae + b'e) \quad *$$

$$= \underline{\underline{(c+d)(ae + b'e) + e(a+b')}}$$

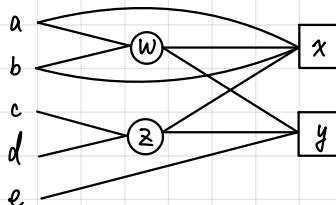
(b)

common sub-expression between x and y : $c + d$

$$z = c + d$$

$$z' = c'd'$$

$$w = a + b'$$



5. (6%) In static timing analysis, one of the possible factor that decreases the accuracy is called false path. Please explain what a false path is and discuss the possible solutions for this problem.

Logic Synthesis - Part 2 P. 49 (false path)

$\langle sol \rangle$

False Path: A false path is a path that topologically exists in the design but never sensitized in operation

- (1) is not functional
- (2) does not need to be timed #

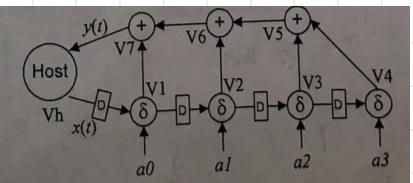
Possible Solution:

(1) Path blocking: User provides a list of nodes, or pair of nodes, that lie on paths of no interest.

(2) Case analysis: User provides a list of cases (nodes and values) analyze the longest path for every cases. #

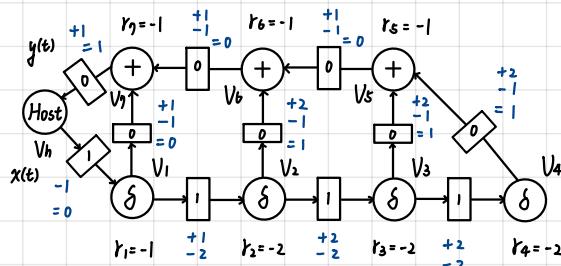
6. (10%) For the circuit shown right:

Given a retiming ($r_1=-1$, $r_2=-2$, $r_3=-2$, $r_4=-2$, $r_5=-1$, $r_6=-1$, $r_7=-1$), please redraw the new circuit after retiming.



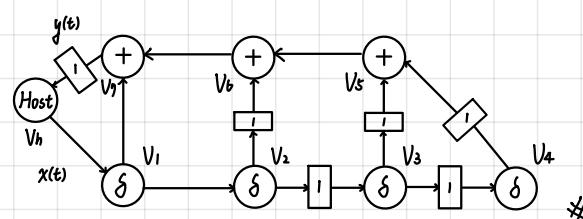
Logic Synthesis - Part 2 P. 77 (Retimed Correlator)

$\langle sol \rangle$



-1: register 往後移

+1: register 往前移



9. (5%) One popular technique to improve circuit timing is called “buffer insertion”. Please discuss why and when this approach can improve the circuit timing. In your opinions, can we always get timing improvement by this technique??

Logic Synthesis - Part 2 P. 60 (Buffer Insertion)

{sol}

- ① Buffers can reduce wire delay by restoring signal strength, in particular, for long wires. Moreover, buffers can be applied to shield capacitive load from timing-critical paths such that the interconnect delay along critical paths are reduced.

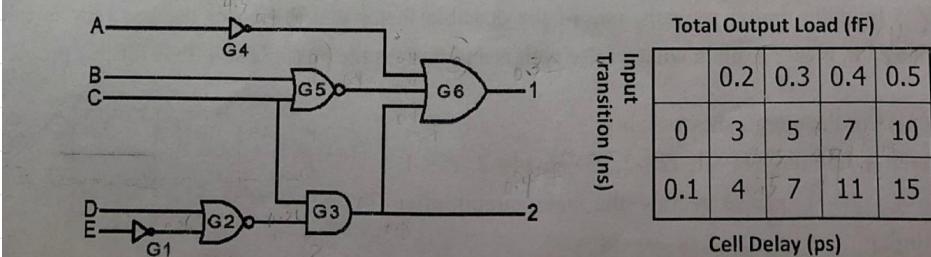
Buffer insertion divides the fanouts of a gate into critical and non-critical parts and drive the non-critical fanouts with a buffer.

Therefore, buffer insertion is to ensure timing closure and reduce the interconnection delay when the timing cannot meet the constraint. *

- ② No, we can't always get timing improvement by this technique. Because the buffer also has its own delay. *

10. Assume the delay time of each gate can be determined by the following lookup table. But the delay time of an inverter is only a half of the value obtained from the table. In order to simplify the calculation, the signal transition time is assumed as 50ps at each node in this problem. Assume the input capacitance of those gates is 0.25fF per input pin except G6, whose input capacitance is 0.4fF only. The output loading are 0.3fF and 0.4fF at node 1 and 2 respectively.

- (a) Please find the delay of each gate according to the lookup table. (12%)
- (b) Please find the longest path and its delay value for this circuit. (8%)



Logic Synthesis - Part 2 P. 29 (Tabular Delay Model)

{sol}

$$50 \text{ ps} = 50 \times 10^{-12} \text{ s} = 0.05 \text{ ns}$$

$$\text{Input transition time} = 0.05 \text{ ns}$$

	0.2	0.25	0.3	0.4	0.5	0.8
Input transition (ns)	0	3	4	5	7	10
0.05	3.5	4.75	6	9	12.5	19
0.1	4	5.5	7	11	15	23

$$\frac{Y_2 - Y_1}{X_2 - X_1} = \frac{Y - Y_1}{X - X_1}$$

(a)

$$\text{Delay}_{G1} = \frac{4.75}{2} = 2.375 \text{ ns}$$

$$\text{Delay}_{G2} = 4.75 \text{ ns}$$

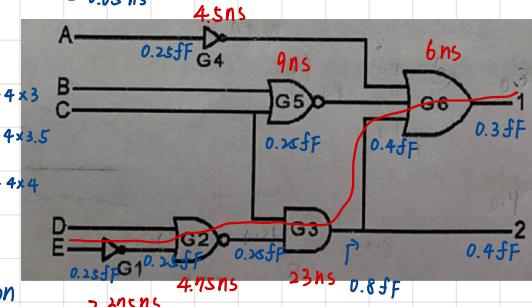
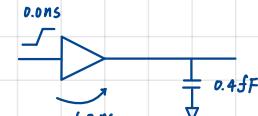
$$\text{Delay}_{G3} = 23 \text{ ns}$$

$$\text{Delay}_{G4} = \frac{9}{2} = 4.5 \text{ ns}$$

$$\text{Delay}_{G5} = 9 \text{ ns}$$

$$\text{Delay}_{G6} = 6 \text{ ns}$$

- (b) longest path: $E \rightarrow G1 \rightarrow G2 \rightarrow G3 \rightarrow G6 \rightarrow 1$
delay : $2.375 + 4.75 + 23 + 6 = 36.125 \text{ ns}$



1. (6%) Please explain the conditions for generating a latch and a flip-flop and in the RTL synthesis step and give an example for each case using Verilog descriptions.

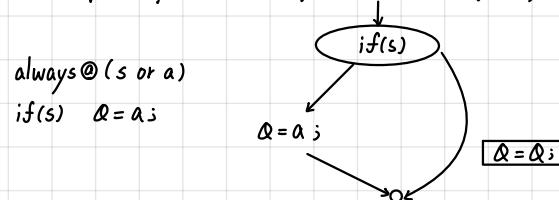
Logic Synthesis - Part 1 P. 21 ~ 28. (latch & flip-flop)

<sol>

Latch

Condition 1: There is no branch associated with the output of a conditional assignment for a value of the selector.

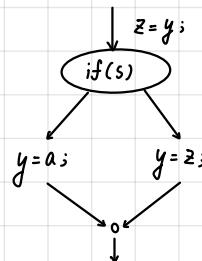
- Output depends on its previous value implicitly.



Condition 2: The output value of a conditional assignment depends on its previous value explicitly.

```

always @ (s or z or y or a)
begin
  z = y;
  if (s) y = a;
  else y = z;
end
  
```



Flip-Flop

Condition: Synthesizing the clocked statements (Sequential circuit)

```

always @ (posedge clk)
  Q = D;
  
```

2. (6%) If a two-level circuit, $f = ac + ad + bc + bd + e$, is implemented to a multi-level circuit, $f = (a+b)(c+d) + e$, what are the possible advantages and disadvantages of this change?

Logic Synthesis - Part 1 P. 78 (2-level circuit vs. multi-level circuit)

<sol>

Advantage:

- Has fewer gates (smaller area)
- Reduce fan-in

Disadvantage:

- More gate delay (slower circuit)

3. (6%) One popular technique to improve circuit timing is called "splitting", which splits the fanouts of a gate into several parts. Each part is driven with a copy of the original gate. Please discuss why and when this approach can improve the circuit timing.

Logic Synthesis - Part 2 P. 61 (Split)

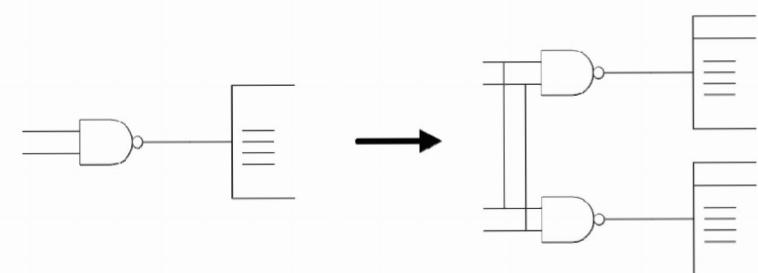
<sol>

When the fanouts of the previous gates are not very large, the circuit can be improved by splitting fanouts.

By splitting fanouts, output loading can be reduced. Thus, splitting method can improve circuit timing. *

Timing Optimization Techniques (3/8)

- **Split**: split the fanouts of a gate into several parts. Each part is driven with a copy of the original gate.



4. (a) Given a Boolean function $F(A,B,C,D) = \sum(0,4,6,8,13,14)$ with don't care $d(A,B,C,D) = \sum(2,9,12)$. Please use Quine-McCluskey algorithm to find all the prime implicants. (12%)
 (b) For the Boolean function in (a), please draw the corresponding ROBDD of it using the variable order $A > B > C > D$. (6%)
 (c) Can you generate the prime implicants from the BDD in (b)? Please briefly explain your reasons. (4%)

Logic Synthesis - Part 1 P.51 (Quine-McCluskey)

BDD

<sol>

(a)

$D = 0$	$0\ 0\ 0\ 0$	$q = 1\ 0\ 0\ 1$
$2 = 0\ 0\ 1\ 0$	$12 = 1\ 1\ 0\ 0$	
$4 = 0\ 1\ 0\ 0$	$13 = 1\ 1\ 0\ 1$	
$6 = 0\ 1\ 1\ 0$	$14 = 1\ 1\ 1\ 0$	
$8 = 1\ 0\ 0\ 0$		

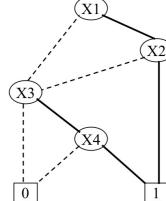
[Hint]

Ordering Effects

- The size of ROBDD depends on the ordering of variables

$$\text{ex: } x_1x_2 + x_3x_4$$

$$x_1 < x_2 < x_3 < x_4$$

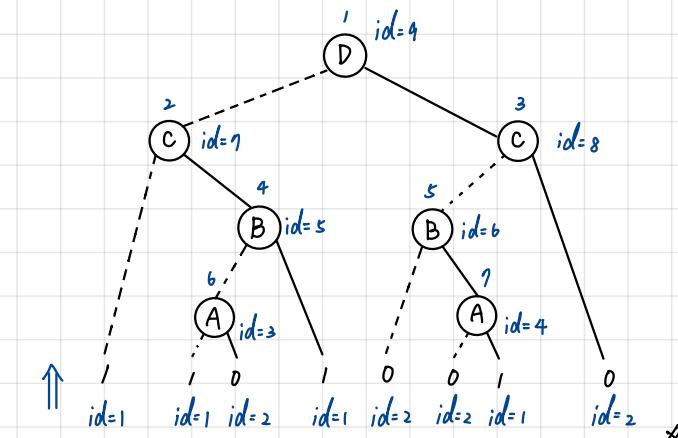
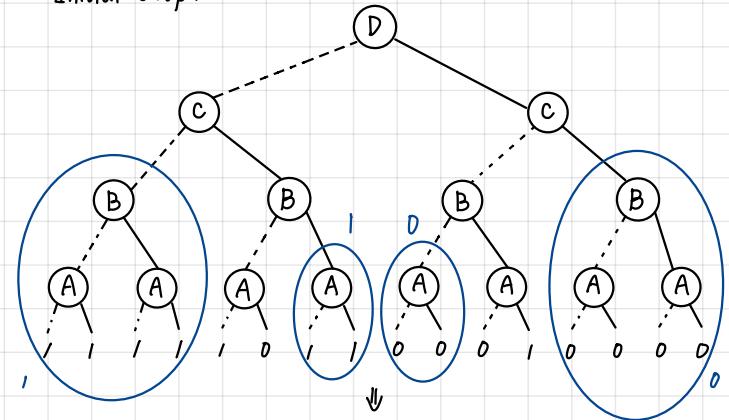


Group	Implication Table					
	Column I		Column II		Column III	
0	0	0 0 0 0 0	1	0, 2 0, 4	0 0 - 0 0, 4 D - 0 0	1 1
1	2	0 0 1 0	1	0, 8	- 0 0 0	1
	4	0 1 0 0	1			
	8	1 0 0 0	1	2, 6 4, 6	0 - 1 0 0 1 - 0	1 1
				8, 9	0 - 1 0 1 0 0 -	1
2	6	0 1 1 0	1	4, 12	- 1 0 0	1
	9	1 0 0 1	1	8, 9	1 0 0 -	1
	12	1 1 0 0	1	8, 12	1 - 0 0	1
3	13	1 1 0 1	1	6, 14	- 1 1 0	1
	14	1 1 1 0	1	9, 13	1 - 0 1	1
				12, 13	1 1 0 -	1
				12, 14	1 1 - 0	1

A B C D
 prime implicant:
 $0 - - 0$
 $- - 0 0$
 $- 1 - 0$
 $1 - 0 -$
 $\Rightarrow A'D', C'D', BD', AC' *$

(b)
 prime implicant: $A'D', C'D', BD', AC'$

Initial Graph

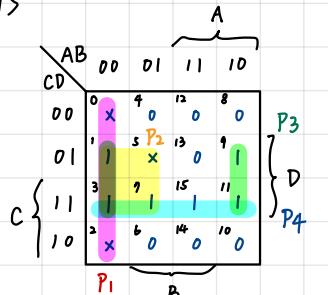


(c)
 No, BDD can only represent the boolean functions.
 It can not generate the prime implicants. *

5. (8%) In the Boolean function $F(A,B,C,D) = \sum(1,3,7,9,11,15)$ with don't care $d(A,B,C,D) = \sum(0,2,5)$, there are 4 possible prime implicants, $P1=A'B'$, $P2=A'D$, $P3=B'D$, $P4=CD$. Therefore, there are total 16 combinations of those 4 prime implicants as shown right. Assume we search the valid combinations with the least number of prime implicants from left to right by using the branch-and-bound algorithm. For each combination, please give a mark 'V' for valid case, a mark 'X' for invalid case, and a mark 'S' for the skipped case.

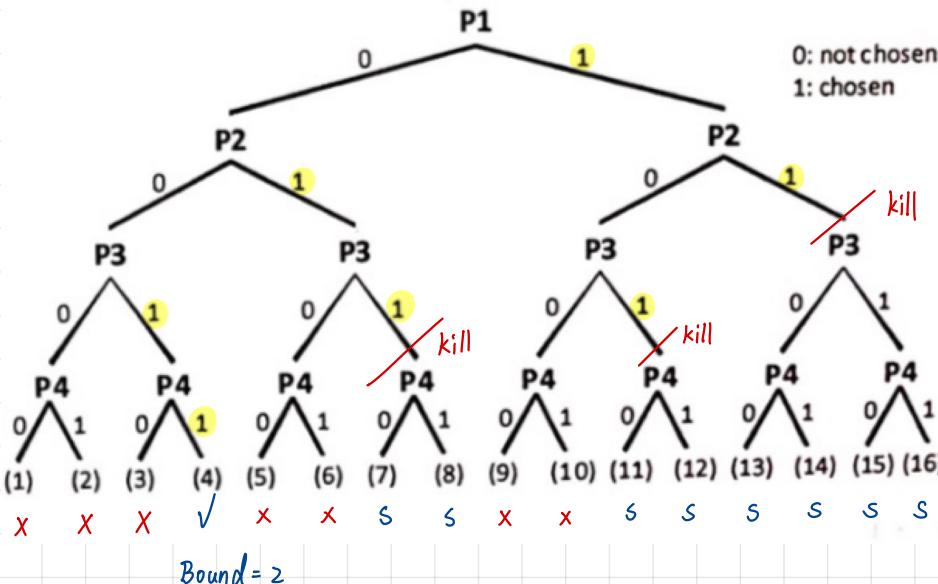
Logic Synthesis - Part 1 P.64 (Branch and Bound)

(sol)



[Hint] 看條件是否可以 cover 到全部的 on-set

⇒ essential prime implicants: P_3, P_4 (不足以 cover 全部的 1)



6. Consider the logic network defined by the following expressions:

$$x = ace + bce + de + g$$

$$y = ad + bd + cde + ge$$

$$z = a + b$$

- (a) Perform both x/z and y/z (weak division) and show the quotients and the remainders. (10%)
 (b) Substitute z into x and y and redraw the Boolean network graph. (4%)

Logic Synthesis - Part 1 P.89 (weak division)

Logic Synthesis - Part 1 P.93 (common sub-expression)

(sol)

(a)

x/z

$$\begin{aligned} x &= \underline{ace + bce + de} + g \\ z &= a+b \end{aligned}$$

y/z

$$\begin{aligned} y &= \underline{ad + bd + cde} + ge \\ z &= a+b \end{aligned}$$

$$V^a = ce$$

$$V^b = ce$$

$$g = d$$

$$\begin{aligned} z_B^a &= (a+b)(ce) \\ &= \underline{ace + bce} \end{aligned}$$

$$V^a = d$$

$$V^b = d$$

$$g = d$$

$$\begin{aligned} z_B^a &= (a+b)(d) \\ &= \underline{ad + bd} \end{aligned}$$

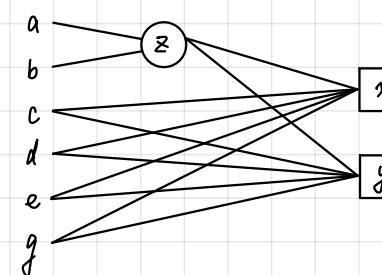
$$r = de + g$$

$$r = cole + ge$$

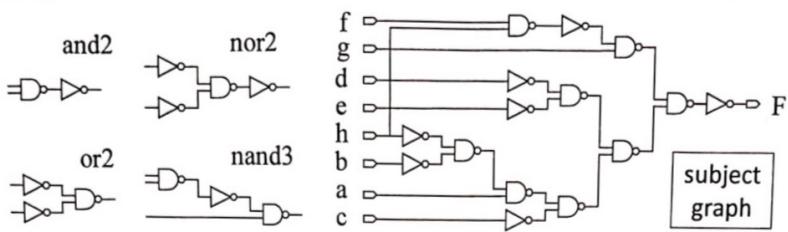
$$\begin{aligned} x &= z_B^a + r \\ &= \underline{(a+b)} ce + de + g \quad * \end{aligned}$$

$$\begin{aligned} y &= z_B^a + r \\ &= (a+b)d + cole + ge \\ &= \underline{(a+b)} d + e(cde + g) \quad * \end{aligned}$$

(b) $z = a+b$

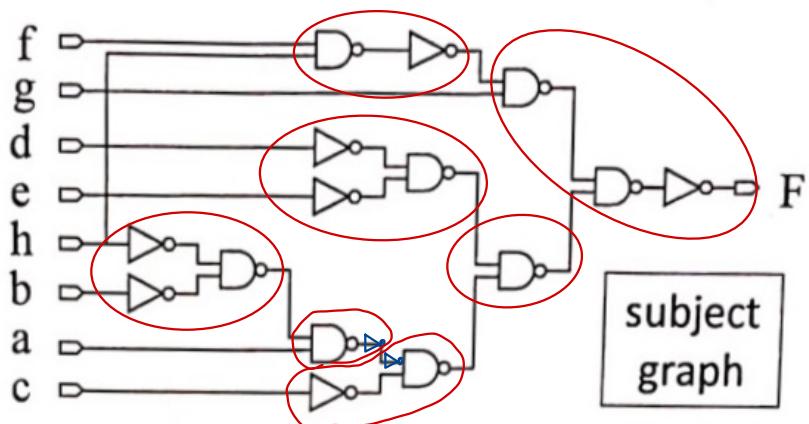


7. (12%) Assume the available gates in the cell library are INV, AND2, OR2, NOR2, NAND2, NOR2 and NAND3. Given the gate-level circuit shown below, what is the minimum gate count after mapped to the given cell library? In addition, please also redraw the best covering on your answer sheet.

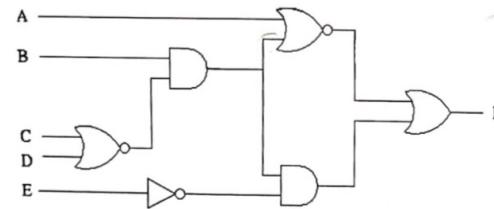


Logic Synthesis - Part 2 P. 11,12 (Sample covers)

`<sol>`



8. (12%) Assume the delay time of each gate can be determined by the following lookup table. But the delay time of an inverter is only a half of the value obtained from the table. In order to simplify the calculation, the signal transition time is assumed as 50ps at each node in this problem. Assume the input capacitance of those gates is 0.35fF per input pin except the inverter, whose input capacitance is 0.2fF only. The output loading at node F is 0.4fF. Please find the longest and shortest paths individually in the circuit shown below along with their delay values and detail calculation.



Input Transition (ns)	Total Output Load (fF)				
	0.2	0.3	0.4	0.5	
0	3	5	7	10	
0.1	4	7	11	15	

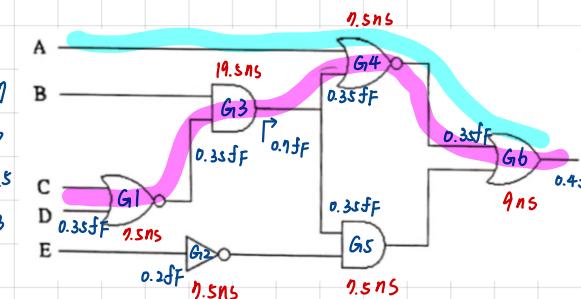
Cell Delay (ps)

Logic Synthesis - Part 2 P. 29 (Tabular Delay Model)

`<sol>`

Input Transition (ns)	Total Output Load				
	0.2	0.3	0.35	0.4	0.5
0	3	5	6	7	10
0.05	3.5	6	7.5	9	12.5
0.1	4	7	9	11	15

Cell Delay



$$\text{Delay}_{G1} = 7.5 \text{ ns}$$

$$\text{Delay}_{G2} = 7.5 \text{ ns}$$

$$\text{Delay}_{G3} = 19.5 \text{ ns}$$

$$\text{Delay}_{G4} = 7.5 \text{ ns}$$

$$\text{Delay}_{G5} = 7.5 \text{ ns}$$

$$\text{Delay}_{G6} = 9 \text{ ns}$$

Longest path : C or D \rightarrow G1 \rightarrow G3 \rightarrow G4 \rightarrow G6 \rightarrow F

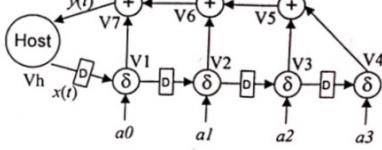
$$\text{Delay}_{\text{long}} : 7.5 + 19.5 + 7.5 + 9 = 43.5 \text{ ns} *$$

Shortest path : A \rightarrow G4 \rightarrow G6 \rightarrow F

$$\text{Delay}_{\text{short}} : 7.5 + 9 = 16.5 \text{ ns} *$$

9. For the circuit shown right:

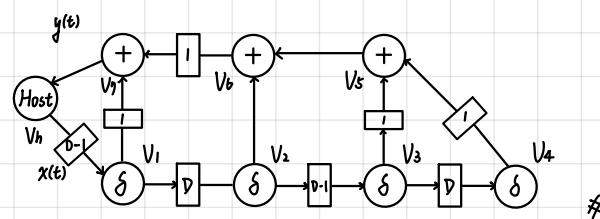
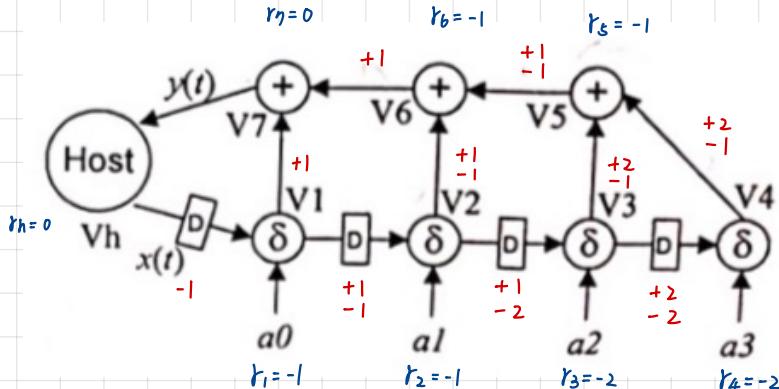
- (a) Given a retiming ($r_h=0$, $r_1=-1$, $r_2=-1$, $r_3=-2$, $r_4=-2$, $r_5=-1$, $r_6=-1$, $r_7=0$), please redraw the new circuit after retiming. (8%)
- (b) Assume the delay of each component is: Host=0, adder($+$)=6ns, comparator(δ)=3ns, flip-flop(D)=0. What is its minimum clock period of the retimed circuit? (6%)



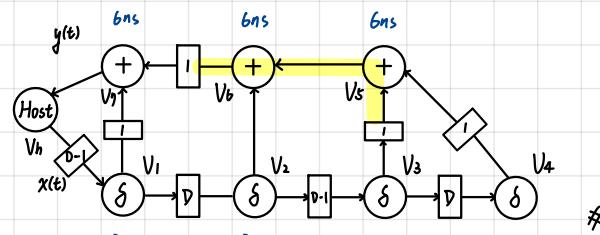
Logic Synthesis - Part 2 P. 77 (Retimed Correlator)

<sol>

(a)



(b)



$$\text{minimum clock period} = 6 + 6 = 12 \text{ ns} *$$

✓ 1. Please answer the following blank about comparison of design styles with the corresponding symbol. (8pts)

a. Density of {(A) FPGA (B) Full custom (C) Gate array} (2pts)

(the highest) _____ > _____ > _____ (the lowest)

b. Performance of {(A) FPGA (B) Full custom (C) Standard cell} (2pts)

(the highest) _____ > _____ > _____ (the lowest)

c. Flexibility of {(A) FPGA (B) Full custom (C) Gate array (D) Standard cell} (2pts)

(the highest) _____ > _____ > _____ > _____ (the lowest)

d. Manufacturing time of {(A) FPGA (B) Gate array (C) Standard cell} (2pts)

(the longest) _____ > _____ > _____ (the shortest)

Introduction P. 41

<sol>

a. Density

Full custom > Gate Array > FPGA
(B) (C) (A)

b. Per