# COE 561, Term 091
# Digital System Design and Synthesis
# HW# 3 Solution

**Due date: Tuesday, Dec. 29**

**Q.1.** Consider the following function:

$$X = ACE + BCE + AC'D' + BC'D' + DE + F$$

**(i)** Compute all the kernels of *X* using the recursive kernel computation algorithm. Show all the steps.

**(ii)** Compute all the kernels of X based on matrix representation. Compare your answer to the result obtained in (i).

**(iii)** Find a quick factor of *X* by using the first level-0 kernel found. Assume that input variables are sorted in lexicographic order. Determine the number of literals obtained. Compare your solution with the result obtained by running the sis commands ***factor –q x; print_factor; print_stats –f.***

**Q.2.** Consider the following function:

$$X=AC+BC+AD+BD+A'B'C'+A'B'D'+AEF+BEF+AE'F'+BE'F'+A'B'EF'+A'B'E'F$$

**(i)** Compute all double-cube divisors of *X* along with their bases and their weights. Show only double-cube divisors that have non-empty bases.

**(ii)** Apply the fast extraction algorithm based on extracting double-cube divisors along with complements or single-cube divisors with two-literals. Show all steps of the algorithm. Determine the number of literals saved. Compare your solution with the result obtained by running the sis commands ***fx***.

**Q.3.** Consider the logic network defined by the following expressions:

$X = AB' + A'B;$
$Y = XC + A\ B;$
$Z = Y + A';$

Inputs are {A, B, C} and output is {Z}.

**(i)** Compute the SDC set for node X.

**(ii)** Compute the ODC set for node Y.

**(iii)** Simplify the function of Y using both its ODC and SDC of node X.

**(iv)** Compute the ODC set for node X based on the optimized network on (iii).

**(v)** Simplify the function of X using its ODC.

**(vi)** Apply the sis command ***full_simplify*** and compare the solution obtained with your obtained solution based in (iv).

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

e=a  b d

f =c  d

g = e + f

h = a d

i =a’ b’ d’

j = h + i

k= b d

l =j + k

x =g + l

Inputs are {a, b, c, d} and output is {x}. Assume that the delay of a gate is related to the number of its inputs. Also, assume that the input data-ready times are zero except for input d, which is equal to 2.

**(i)** Draw the logic network graph and compute the data ready times and slacks for all vertices in the network.

**(ii)** Determine the **maximum propagation delay** and the **topological critical path**.

**(iii)** Suggest an implementation of the function ***x*** to reduce the delay of the circuit. What is the **maximum propagation delay** after the modified implementation?

COE 561 Dr. Aiman EL-Maleh

# HW #3

Q1. $X = ACE + BCE + A\bar{C}\bar{D} + B\bar{C}\bar{D} + DE + F$

(I) Recursive Kernel Computation:

We assume that the variables are ordered in lexicographic order: $\{A, B, C, \bar{C}, D, \bar{D}, E, F\}$

$i = 1 : \{A\}$

Cubes containing A : $\{ACE, A\bar{C}\bar{D}\} \geq 2$

$C = A$

Kernel found : $CE + \bar{C}\bar{D}$

Recursive call on the Kernel with $i = 2$ $\{B\}$

- $i = 2 : \{B\}$ No Kernels found
- $i = 3 : \{C\}$ No Kernels found

The process will continue for all the variables and no Kernels will be found.

$i = 2 : \{B\}$

Cubes containing B : $\{BCE, B\bar{C}\bar{D}\} \geq 2$

$C = B$

Kernel found : $CE + \bar{C}\bar{D}$

Recursive call on the Kernel with $i = 3$ will not produce any Kernel.

$i = 3 : \{C\}$

Cubes containing C : $\{ACE, BCE\} \geq 2$

$C = CE$

Kernel found : $A + B$

Recursive call with $i = 4$ will not produce any Kernel

$i = 4 : \{\bar{C}\}$

Cubes containing $\bar{C} : \{ A\bar{C}\bar{D} , B\bar{C}\bar{D} \} \geq 2$

$C = \bar{C}\bar{D}$

Kernel found : $A + B$

Recursive call with $i = 5$ will not produce any kernel.

$i = 5 : \{D\}$

Cubes containing $D : \{DE\}$ No kernels found

$i = 6 ; \{\bar{D}\}$

Cubes containing $\bar{D} ; \{ A\bar{C}\bar{D} , B\bar{C}\bar{D} \} \geq 2$

$C = \bar{C}\bar{D}$

Since the cube contains literal $\bar{C}$, no kernel will be generated

$i = 7 : \{E\}$

Cubes containing $E : \{ ACE, BCE, DE \} \geq 2$

$C = E$

Kernel found : $AC + BC + D$

Recursive call with $i = 8$ will produce no kernels

$i = 8 ; \{F\}$

No kernels will be generated

| Kernel | Co-Kernel |
|---|---|
| $CE + \bar{C}\bar{D}$ | $A , B$ |
| $A + B$ | $CE , \bar{C}\bar{D}$ |
| $AC + BC + D$ | $E$ |
| $X$ | $1$ |

(ii) Kernel computation using matrix representatron:

|  | Vari | A | B | C | $\bar{C}$ | D | $\bar{D}$ | E | F |
|---|---|---|---|---|---|---|---|---|---|
| cube | R/C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| ACE | 1 | 1 |  | 1 |  |  |  | 1 |  |
| BCE | 2 |  | 1 | 1 |  |  |  | 1 |  |
| $A\bar{C}\bar{D}$ | 3 | 1 |  |  | 1 |  | 1 |  |  |
| $B\bar{C}\bar{D}$ | 4 |  | 1 |  | 1 |  | 1 |  |  |
| DE | 5 |  |  |  |  | 1 |  | 1 |  |
| F | 6 |  |  |  |  |  |  |  | 1 |

| Prime Rectangle | Cube | Kernel |
|---|---|---|
| ( {1, 3}, {1} ) | A | $CE + \bar{C}\bar{D}$ |
| ( {2, 4}, {2} ) | B | $CE + \bar{C}\bar{D}$ |
| ( {1, 2}, {3, 7} ) | CE | A + B |
| ( {3, 4}, {4, 6} ) | $\bar{C}\bar{D}$ | A + B |
| ( {1, 2, 5}, {7} ) | E | AC + BC + D |

(iii) Quick factor of X based on first level-0 kernel found

$L(A) = 2 > 1$

$C = A$

Then, we call the procedure one-level-0-kernel on $\frac{X}{A} = CE + \bar{C}\bar{D}$

Since all varrable occur only once, the first level-0 kernel found is $CE + \bar{C}\bar{D}$.

$(h, r) = \text{Divide}\ (X,\ K = CE + \bar{C}\bar{D})$

$h = A + B$ , $r = DE + F$

Then, Quick factor will be called on h & r but they can't be factored further. Thus,

$X = (A+B)(CE + \bar{C}\bar{D}) + DE + F$ 9 literals

SIS produces the same quick factor as shown below:

```
sis> read_eqn hw3q1.eqn
sis> print
    {X} = A C E + A C' D' + B C E + B C' D' + D E + F
sis> factor -q X; print_factor; print_stats -f
   {X} = (C E + C' D') (B + A) + D E + F
hw3q1.eqn  pi= 6  po= 1  node=  1  latch= 0  lits(sop)=  15  lits(ff)=   9
sis>
```

Q2. $X = AC + BC + AD + BD + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{D} + AEF + BEF$
$+ A\bar{E}\bar{F} + B\bar{E}\bar{F} + \bar{A}\bar{B}E\bar{F} + \bar{A}\bar{B}\bar{E}F$ 34 literals

(i) Double Cube Divisors

| Double-Cube divisor | Base | Weight |
|---|---|---|
| $A+B$ | $C, D, EF, \bar{E}\bar{F}$ | 4×2−4−2+6+4 = 12 |
| $C+D$ | $A, B$ | 2×2−2−2+2+0 = 2 |
| $C+EF$ | $A, B$ | 2×3−2−3+2 = 3 |
| $C+\bar{E}\bar{F}$ | $A, B$ | 2×3−2−3+2 = 3 |
| $D+EF$ | $A, B$ | 2×3−2−3+2 = 3 |
| $D+\bar{E}\bar{F}$ | $A, B$ | 2×3−2−3+2 = 3 |
| $\bar{C}+\bar{D}$ | $\bar{A}\bar{B}$ | 1×2−1−2+2+0 = 1 |
| $\bar{C}+E\bar{F}$ | $\bar{A}\bar{B}$ | 1×3−1−3+2 = 1 |
| $\bar{C}+\bar{E}F$ | $\bar{A}\bar{B}$ | 1×3−1−3+2 = 1 |
| $\bar{D}+E\bar{F}$ | $\bar{A}\bar{B}$ | 1×3−1−3+2 = 1 |
| $\bar{D}+\bar{E}F$ | $\bar{A}\bar{B}$ | 1×3−1−3+2 = 1 |
| $EF+\bar{E}\bar{F}$ | $A, B$ | 3×4−3−4+4 = 9 |
| $AF+\bar{A}\bar{B}\bar{F}$ | $E$ | 1×5−1−5+1 = 0 |
| $AE+\bar{A}\bar{B}\bar{E}$ | $F$ | 1×5−1−5+1 = 0 |
| $BF+\bar{A}B\bar{F}$ | $E$ | 1×5−1−5+1 = 0 |
| $BE+\bar{A}\bar{B}\bar{E}$ | $F$ | 1×5−1−5+1 = 0 |
| $A\bar{E}+\bar{A}\bar{B}E$ | $\bar{F}$ | 1×5−1−5+1 = 0 |
| $A\bar{F}+\bar{A}\bar{B}F$ | $\bar{E}$ | 1×5−1−5+1 = 0 |
| $B\bar{E}+\bar{A}\bar{B}E$ | $\bar{F}$ | 1×5−1−5+1 = 0 |
| $B\bar{F}+\bar{A}\bar{B}F$ | $\bar{E}$ | 1×5−1−5+1 = 0 |
| $E\bar{F}+\bar{E}F$ | $\bar{A}\bar{B}$ | 3×4−3−4+4 = 9 |

(ii) Fast Extraction

From part (i), we can see that $W_{dmax} = 12$ and the double cube divisor A+B has the highest weight.

The highest weight single cube divisor is $\bar{A}\bar{B}$ with a weight of 2.

Thus, the double cube divisor is selected and the resulting network is:

$$[1] = A + B$$

$$X = C[1] + D[1] + \bar{C}\,\overline{[1]} + \bar{D}\,\overline{[1]} + EF[1] + \bar{E}\bar{F}[1] + E\bar{F}\,\overline{[1]} + \bar{E}F\,\overline{[1]}$$

22 literals

Double cube dividers are updated in the same way and the double cube divider $EF + \bar{E}\bar{F}$ is extracted as it has the highest weight of $2 \times 4 - 2 - 4 + 2 = 4$.

Thus, the resulting network will be:

$$[1] = A + B$$

$$[2] = EF + \bar{E}\bar{F}$$

$$X = C[1] + D[1] + \bar{C}\,\overline{[1]} + \bar{D}\,\overline{[1]} + [2][1] + \overline{[2]}\,\overline{[1]}$$

18 literals

Since none of the remaining double cube divisors or single cube divisors has a positive weight, none of them will be extracted.

SIS has resulted in the same extracted network as shown below:

```
sis> read_eqn hw3q2.eqn
sis> print
    {X} = A C + A D + A E F + A E' F' + A' B' C' + A' B' D' + A' B' E F' + A' B
' E' F + B C + B D + B E F + B E' F'
sis> print_stats
hw3q2.eqn  pi= 6  po= 1  node=  1  latch= 0  lits(sop)=  34  lits(ff)=  16
sis> fx
sis> print
    {X} = C [1] + C' [1]' + D [1] + D' [1]' + [1] [2] + [1]' [2]'
    [1] = A + B
    [2] = E F + E' F'
sis> print_stats
hw3q2.eqn  pi= 6  po= 1  node=  3  latch= 0  lits(sop)=  18  lits(ff)=  14
sis>
```

Q3. $X = A\bar{B} + \bar{A}B$

$Y = XC + AB$

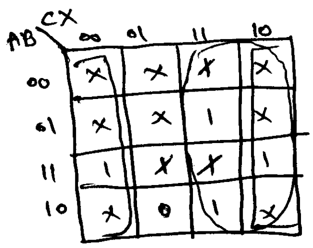
$Z = Y + \bar{A}$

(i) $SDC_X = X \oplus (A\bar{B} + \bar{A}B) = \bar{X}(A\bar{B} + \bar{A}B) + X(AB + \bar{A}\bar{B})$

$= \bar{X}A\bar{B} + \bar{X}\bar{A}B + XAB + X\bar{A}\bar{B}$

(ii) $ODC_Y = \bar{A}$

(iii) Simplification of Y



| AB \ CX | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | X | X | 1 | X |
| 11 | 1 | X | X | 1 |
| 10 | X | 0 | 1 | X |

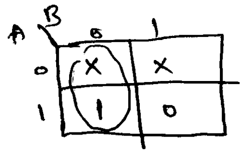
There are several possible solutions for Y.

one solution is :

$Y = \bar{X} + C$

(iv) $ODC_X = C + \bar{A}$

(v) Simplification of X



| A \ B | 0 | 1 |
|---|---|---|
| 0 | X | X |
| 1 | 1 | 0 |

$X = \bar{B}$

Thus, the optimized network is :

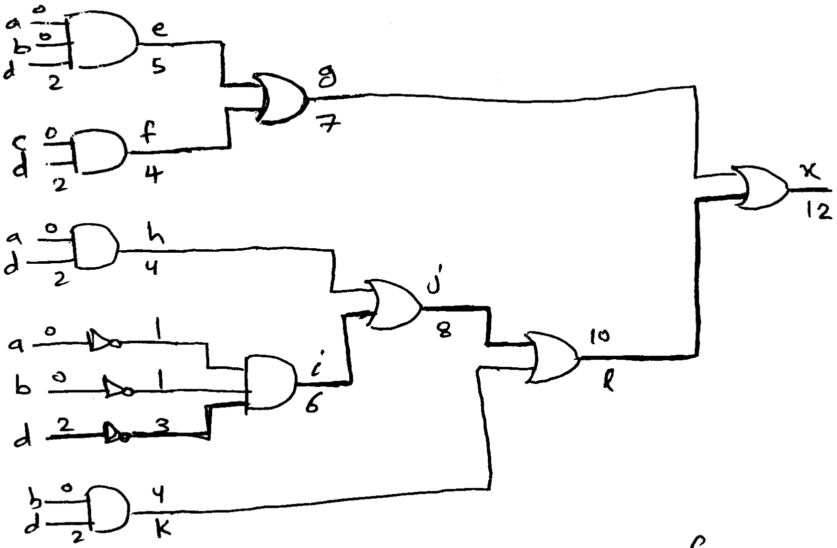
$X = \bar{B}$ $Y = \bar{X} + C$ $Z = Y + \bar{A}$

SIS has resulted in the same simplified network as shown below:

```
sis> read_eqn hw3q3.eqn
sis> print
    {Z} = A' + Y
    X = A B' + A' B
    Y = A B + C X
sis> print_stats
hw3q3.eqn  pi= 3  po= 1  node=  3  latch= 0  lits(sop)=  10  lits(ff)=  10
sis> full_simplify
sis> print
    {Z} = A' + Y
    X = B'
    Y = C + X'
sis> print_stats
hw3q3.eqn  pi= 3  po= 1  node=  3  latch= 0  lits(sop)=   5  lits(ff)=   5
sis>
```

Qu.

(1)



The data ready times are shown on the figure. The maximum propagation delay is 12. To compute the slack for each node, the required time for x is set to 12.

$\bar{t}_x = 12$ $\quad S_x = 12-12=0$

$\bar{t}_g = 12-2 = 10$ $\quad S_g = 10-7=3$

$\bar{t}_l = 12-2=10$ $\quad S_l = 10-10=0$

$\bar{t}_e = 10-2 = 8$ $\quad S_e = 8-5=3$

$\bar{t}_f = 10-2=8$ $\quad S_f = 8-4=4$

$\bar{t}_j = 10-2=8$ $\quad S_j = 8-8=0$

$\bar{t}_k = 10-2=8$ $\quad S_k = 8-4=4$

$\bar{t}_h = 8-2=6$ $\quad S_h = 6-4=2$

$\bar{t}_i = 8-2=6$ $\quad S_i = 6-6=0$

$\bar{t}_a = \min\{8-3, 6-2, 6-3-1\} = 2$ $\quad S_a = 2-0=2$

$\bar{t}_b = \min\{8-3, 6-3-1, 8-2\} = 2$ $\quad S_b = 2-0=2$

$\bar{t}_c = 8-2=6$ $\quad S_c = 6-0=6$

$\bar{t}_d = \min\{8-3, 8-2, 6-2, 6-3-1, 8-2\} = 2$ $\quad S_d = 2-2=0$

(ii) The maximum propagation delay is 12 and the topological critical path is: $\{d, i, j, l, x\}$.

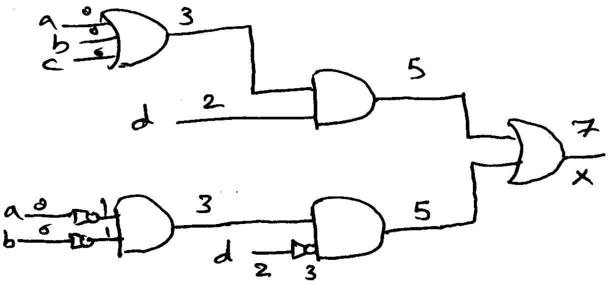
(iii) $x = g + l = e + f + j + k$

$= abd + cd + ad + \bar{a}\bar{b}\bar{d} + bd$

$= d\,[ab + c + a + b] + \bar{a}\bar{b}\bar{d}$

$= d\,[a + b + c] + \bar{a}\bar{b}\bar{d}$

Thus, to improve the delay of x, we can implement it as follows:



Thus, the delay of x is reduced from 12 to 7. At the same time, the area of the circuit was improved.