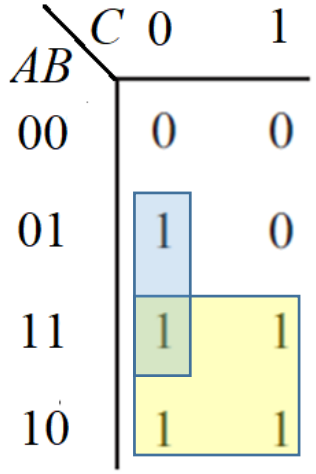
Digital Electronic Circuits
Section 1 (EE, IE)

Lecture 8

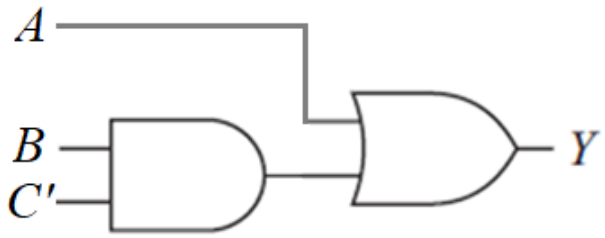
Simplest SOP and POS: An Example

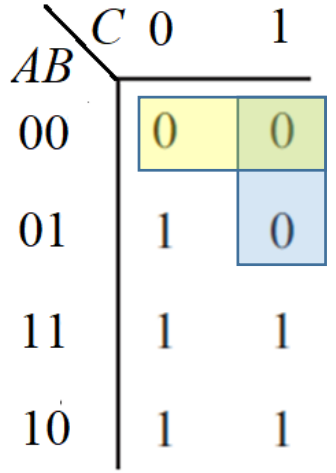
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$Y = F(A,B,C)$
$= \sum m(2,4,5,6,7)$

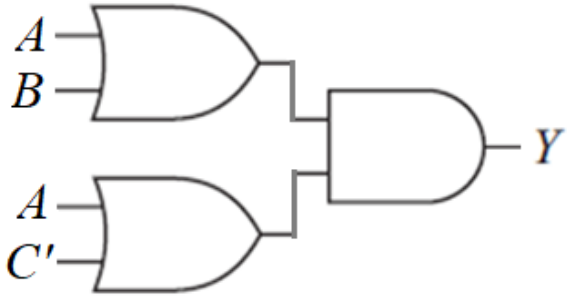


$Y = A + B.C'$





$Y = (A + B).(A + C')$



Here, realization of POS is costlier.

Cost Criteria

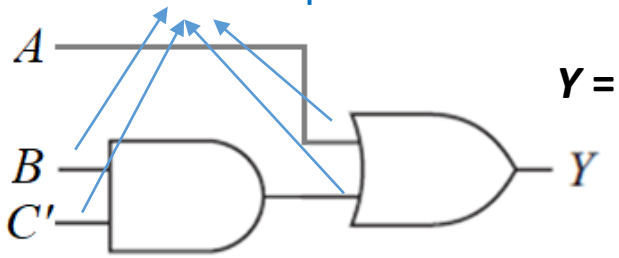
- **Literal Cost:** The number of times a literal appears in complemented or uncomplemented form (*L*).
- **Gate Input Cost:** The number of inputs to the gates in the implementation. It may not count NOT gates (*G*) or it may include NOT gate count (*GN*).

If the number of terms excluding single variable term is *T*, and the number of distinct inversions is *N*, then

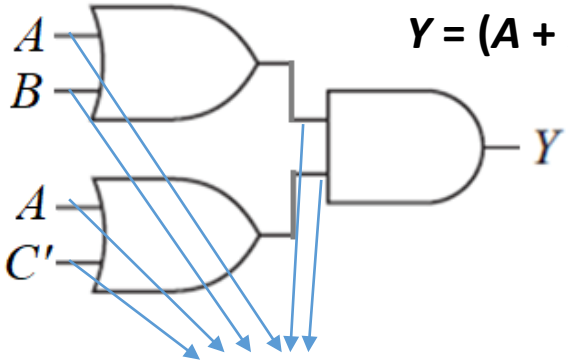
$$G = L + T$$

$$GN = L + T + N$$

SOP: 4 Gate inputs: G = 4



Y = A + B.C'



Y = (A + B) .(A + C')

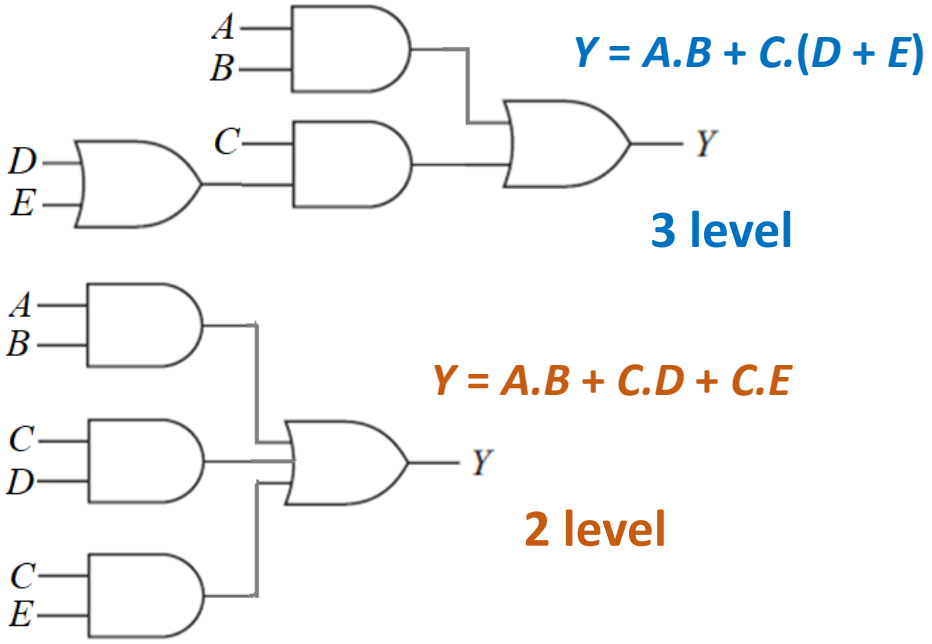
POS: 6 Gate inputs: G = 6

Implementation	L	T	N	G	GN
Y = A + B.C'	3	1	1	4	5
Y = (A + B) .(A + C')	4	2	1	6	7

Cost Criteria: Examples

Implementation	L	T	N	G	GN
$Y = A.B + B.C + C.A$	6	3	0	9	9
$Y = (A+B).(B+C).(C+A)$	6	3	0	9	9
$Y = A.B.C + A'.B'.C'$	6	2	3	8	11
$Y = (A'+B).(B'+C).(C'+A)$	6	3	3	9	12
$Y = A.B + B'.C.D + A'.B'.D + B.C'.D$	11	4	3	15	18
$Y = (A+C').(A+D).(B'+C'+D).(B+C+D')$	10	4	3	14	17

Considered: 2-Level implementation, no restriction in fan-in or kind of logic gates to be used (as in K-Map, QM simplification)



Total Cost (TC) = No. of gate i/p + No. of gates

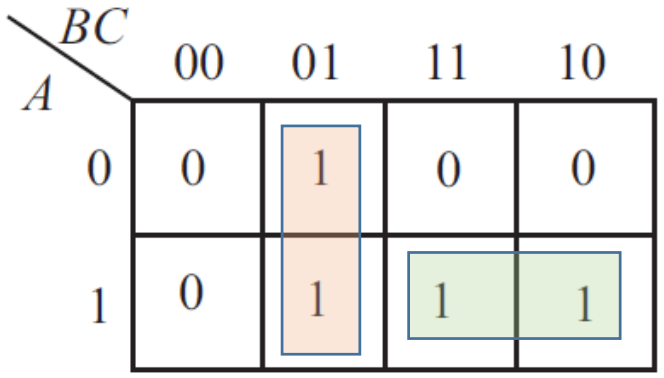
Implementation using only 2-input NAND gate: An Example

$Y = A.B + C.D.E = ((A.B)'.(C.D.E)')' = ((A.B)'.((C.D)')'.E)')'$

Cost: 5 Units
(10 Gate inputs)

Multiple Output Minimization

Common P.I.

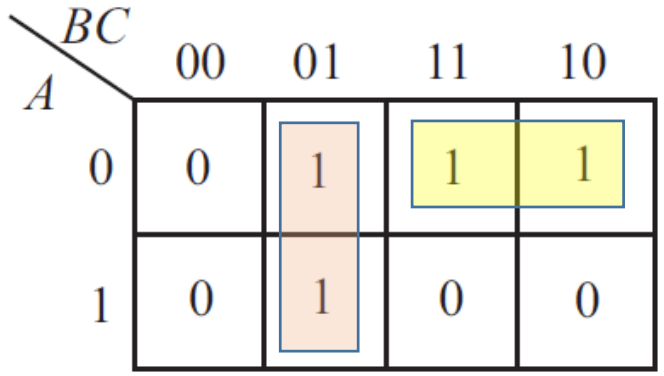


$F_1 = B'.C + A.B$

Individual implementation

G = 6 x 2 = 12

TC = 12 + 3 x 2 = 18

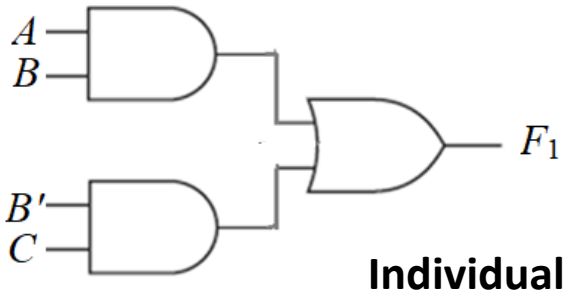


Using common P.I.

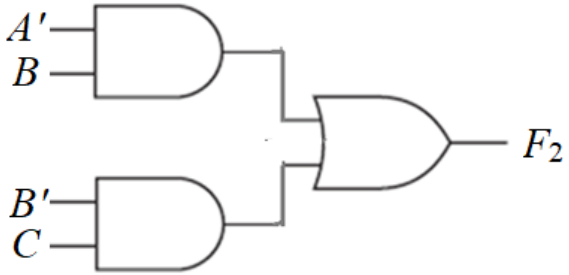
G = 6 + 4 = 10

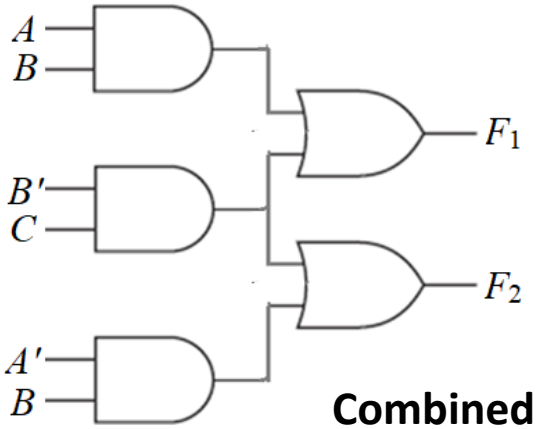
TC = 10 + 3 +2 = 15

$F_2 = B'.C + A'.B$



Individual

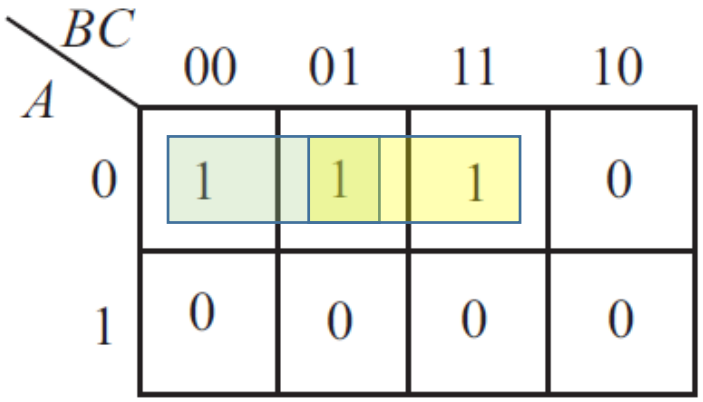


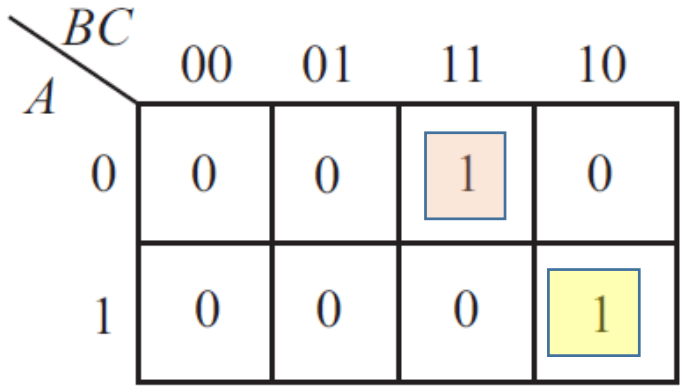


Combined

Multiple Output Minimization

Common term need to be generated

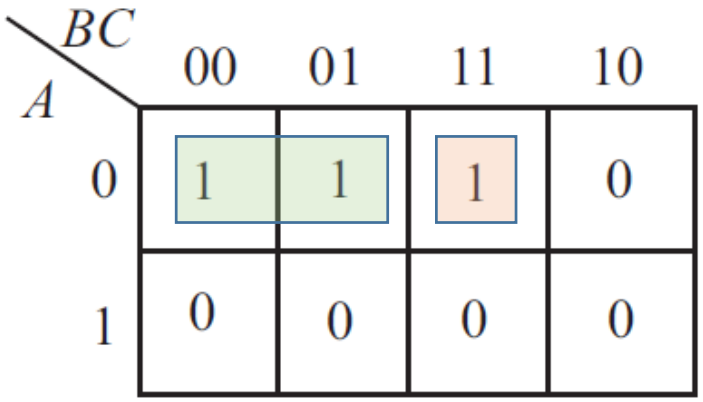




$F_1 = \sum m(0,1,3)$
$= A'.B' + A'.C$

Individual implementation
G = 6 + 8 = 14
TC = 14 + 3 + 3 = 20

$F_2 = \sum m(3,6)$
$= A'.B.C + A.B.C'$

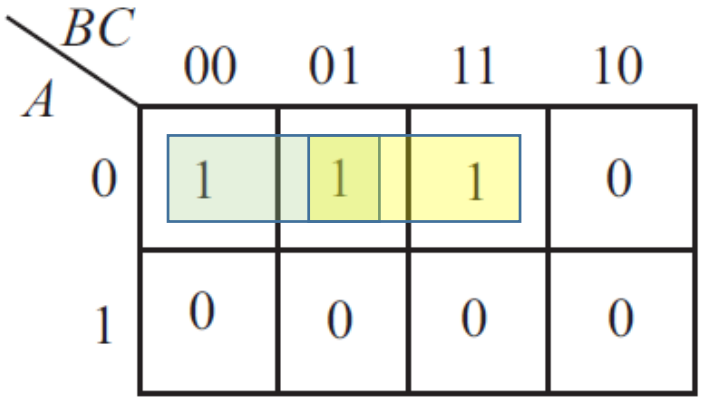


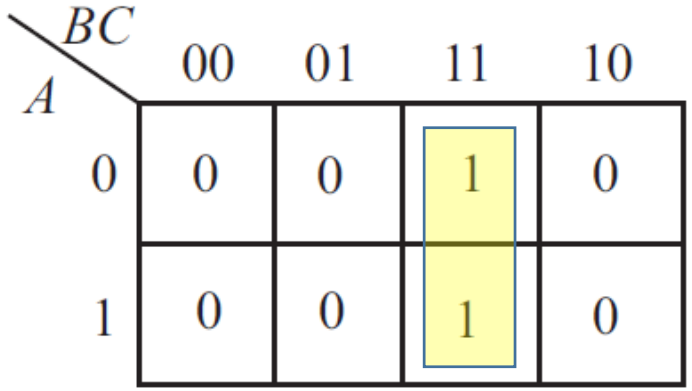
$F_1 = A'.B' + A'.B.C$

$F_2 = A'.B.C + A.B.C'$

Combined
G = 7 + 5 = 12
TC = 12 + 3 + 2 = 17

Multiple Output Minimization



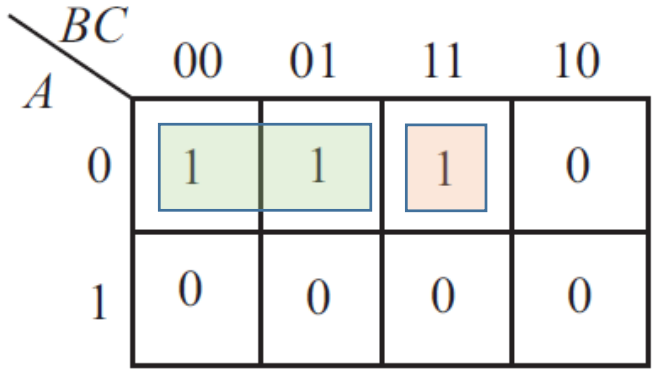


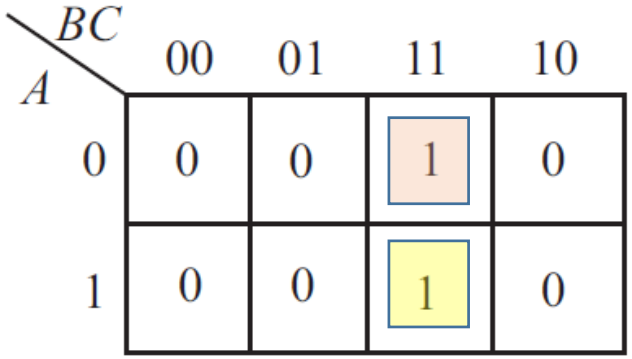
$F_1 = \sum m(0,1,3)$
$= A'.B' + A'.C$

Individual implementation
G = 6 + 2 = 8
TC = 8 + 3 + 1 = 12

$F_2 = \sum m(3,7)$
$= B.C$

Individual: Less cost





Common term part of individual bigger group

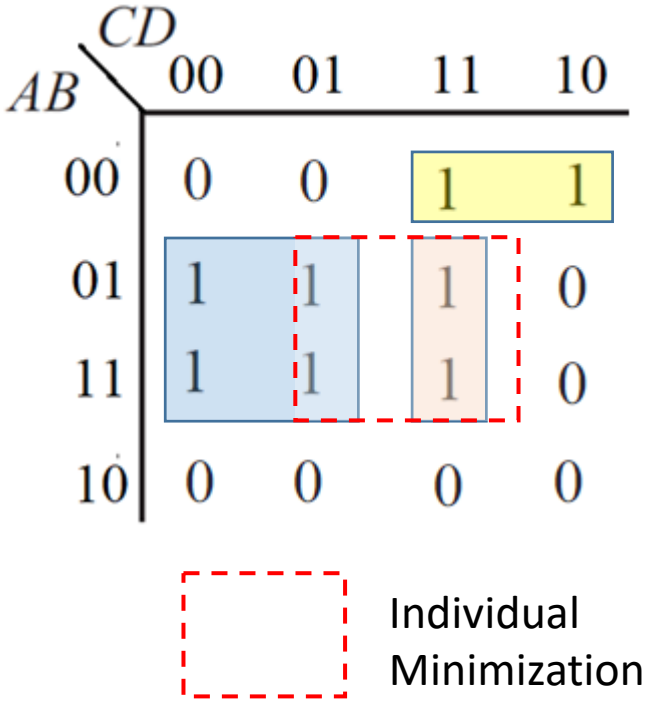
$F_1 = A'.B' + A'.B.C$

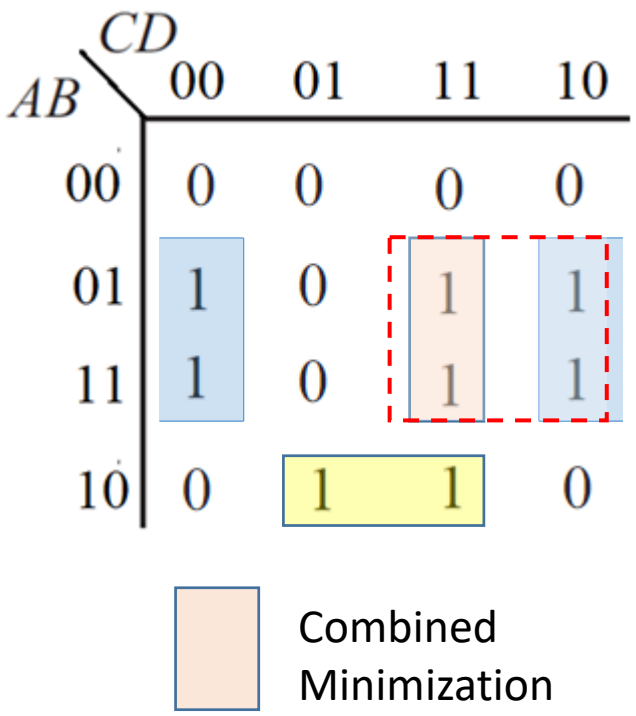
$F_2 = A'.B.C + A.B.C$

G = 7 + 5 = 12
TC = 12 + 3 + 2
= 17

Multiple Output Minimization

Common term part of bigger group in which other members already covered





Individual
G = 10 + 10 = 20
TC = 20 + 4 + 4 = 28

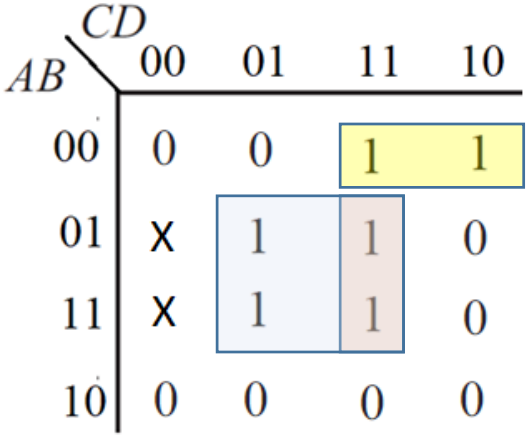
Combined
G = 11 + 8 = 19
TC = 19 + 4 + 3 = 26

$F_1 = B.C' + B.D + A'.B'.C$
$F_2 = B.D' + B.C + A.B'.D$

$F_1 = B.C' + B.C.D + A'.B'.C$
$F_2 = B.D' + B.C.D + A.B'.D$

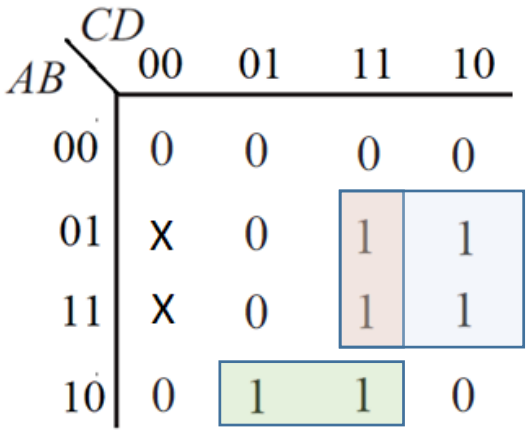
Any 3rd option?

Multiple Output Minimization



$F_1 = B.D + A'.B'.C$

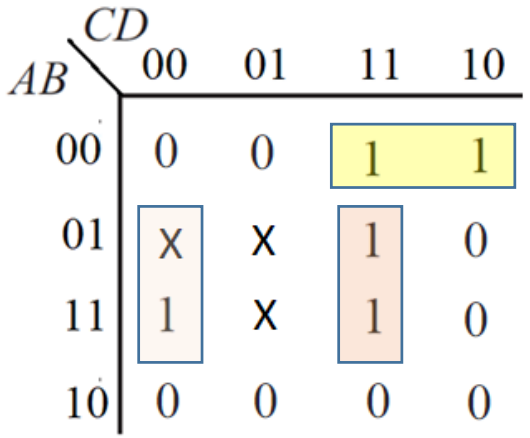
$F_2 = B.C + A.B'.D$



G = 7 + 7 = 14

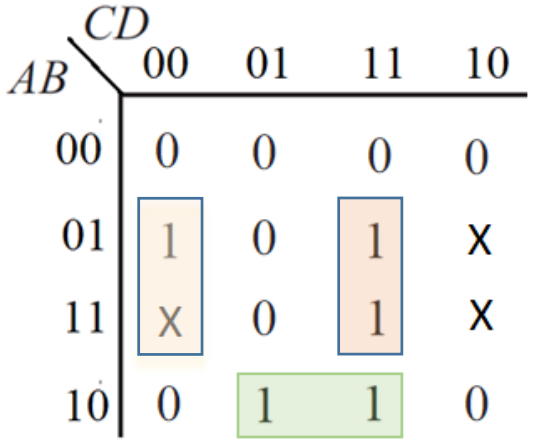
TC = 14 + 3 + 3 = 20

Use of Don't Care



$F_1 = B.C'.D' + B.C.D + A'.B'.C$

$F_2 = B.C'.D' + B.C.D + A.B'.D$



G = 12 + 6 = 18

TC = 18 + 4 + 2 = 24

Individual

G = 20, TC = 28

References:

- Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill