DC for Exam 3

Schedule

Week 1 (Aug 13-17)
Discussion of Exam 2

Week 2 (Aug 20-24)
Combinational Building Blocks

Week 3 (Aug 27-31)
Other Combinational Function Implementation

Exam #3: September 1, 2007

DC 3-1

Discussion of Exam 2

Problem #1

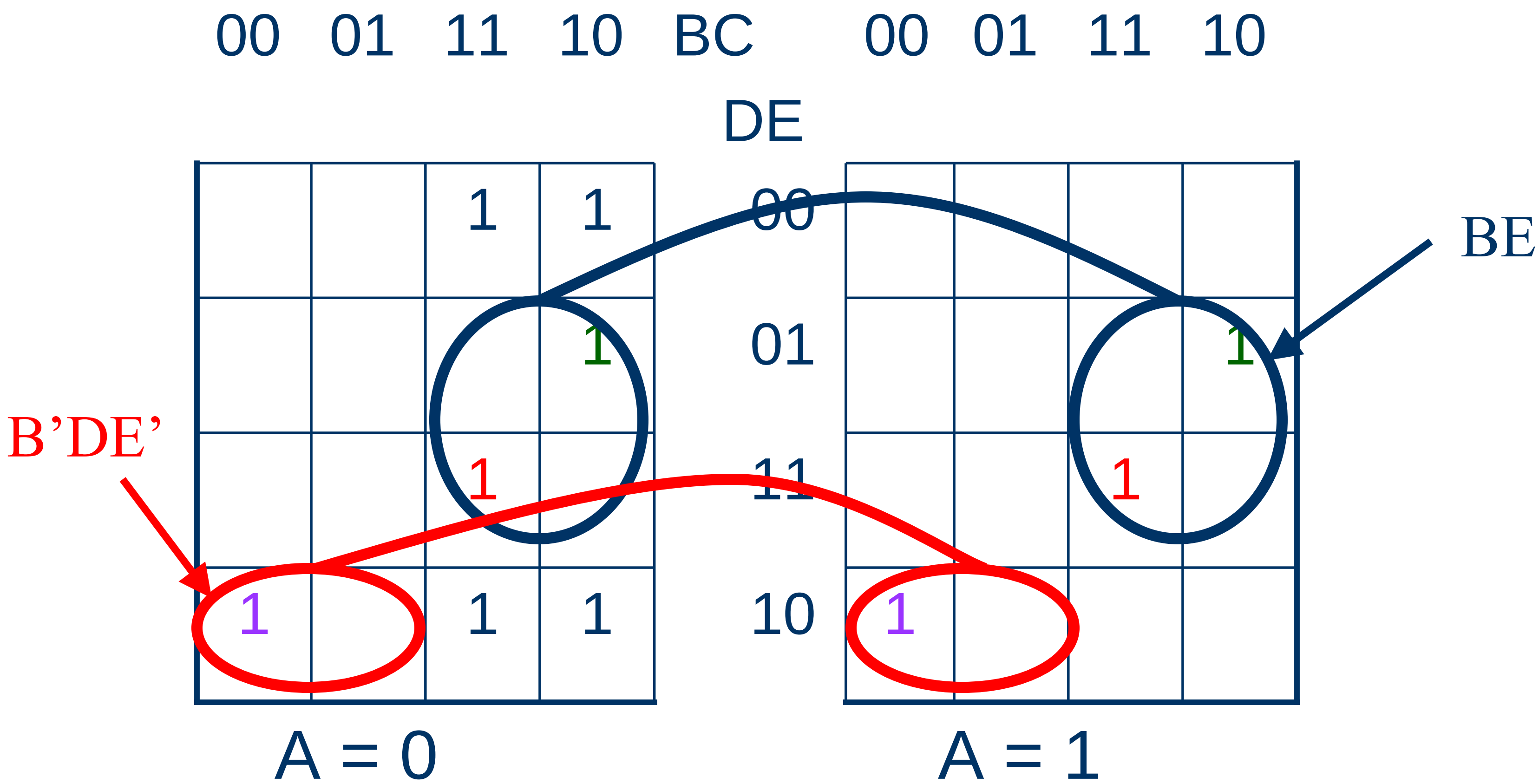
- The expression BE + B'DE' is a simplified version of

  A'BE + BCDE + BC'D'E + A'B'DE' + B'C'DE'

Are there are don't care conditions? If so, what are they?

Problem #1: Solution

A'BE + BCDE + BC'D'E + A'B'DE' + B'C'DE'



Problem #1: Solution (cont'd)

Therefore, don't care conditions are:

x1101, x1011, x0110

→ BCD'E + BC'DE + B'CDE'

→ d(00110,01011,01101,10110,11011,11101)

→ d(6, 11, 13, 22, 27, 29)

Problem #2

Given the PIT of function M(R,O,E,L)

	0	4	5	6	7	8	9	13
a	✔	✔						
b	✔					✔		
c						✔	✔	
d			✔		✔			
e							✔	✔
f			✔					✔
g		✔	✔	✔	✔			

Problem #2 (cont'd)

a.) Determine the product terms represented by the prime implicants (assuming no don't cares)

$a = \Sigma m(0,4) = r'o'e'l' + r'oe'l' = r'e'l'$

$b = \Sigma m(0,8) = r'o'e'l' + ro'e'l' = o'e'l'$

$c = \Sigma m(8,9) = ro'e'l' + ro'e'l = ro'e'$

$d = \Sigma m(5,7) = r'oe'l + r'oel = r'ol$

$e = \Sigma m(9,13) = ro'e'l + roe'l = re'l$

$f = \Sigma m(5,13) = r'oe'l + roe'l = oe'l$

$g = \Sigma m(4,5,6,7) = r'oe'l' + r'oe'l + r'oel' + r'oel = r'o$

Problem #2 (cont'd)

b.) Determine the essential prime implicants

	0	4	5	6	7	8	9	13
a	✔	✔						
b	✔					✔		
c						✔	✔	
d			✔		✔			
e							✔	✔
f			✔					✔
g		✔	✔	✔	✔			
		g	g	g	g			

g is an essential prime implicant

Problem #2 (cont'd)

Can proceed to RPIT

	0	8	9	13	
a	✔				Dominated
b	✔	✔			
c		✔	✔		
d					Redundant
e			✔	✔	
f				✔	Dominated

	0	8	9	13
b	✔	✔		
c		✔	✔	
e			✔	✔
	b	b	e	e

b and e are secondary essential prime implicants

Problem #2 (cont'd)

c.) Minimize M(R,O,E,L)

M(R,O,E,L) = essential prime implicants + secondary prime implicants

= g + b + e

= r'o + o'e'l' + re'l

Problem #4

Given F(A,B,C,D) = Σm(0,4,5,6,10,13) + d(2)

a.) Use K-map to simplify

AB / CD	00	01	11	10
00	1	1		
01		1	1	
11				
10		1		1

F(A,B,C,D)
= A'C'D' + A'BD' + AB'CD' + BC'D

Problem #4

b.) Using D as MEV

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

AB C	00	01	11	10
0	D’	1	D	
1		D’		D’

Problem #4

b.) Using D as MEV

AB / C	00	01	11	10
0	D’	1	D	
1		D’		D’

AB / C	00	01	11	10
0	0	x	0	
1		0		0

Phase 1: D’AB’C + D’A’B
+ D’A’C’ + DBC’

Phase 2: (nothing to add)

F = D’AB’C + D’A’B + D’A’C’ + DBC’

Problem #4

c.) Using A as MEV

AB / CD	00	01	11	10
00	1	1		
01		1	1	
11				
10		1		1

CD / B	00	01	11	10
0	A’			A
1	A’	1		A’

Problem #4

c.) Using A as MEV

CD / B	00	01	11	10
0	A’			A
1	A’	1		A’

CD / B	00	01	11	10
0	0			0
1	0	1		0

Phase 1: A’C’D’ + A’D’B + ACD’B’

Phase 2: C’DB

F = A’C’D’ + A’D’B + ACD’B’ + C’DB

DC 3-2

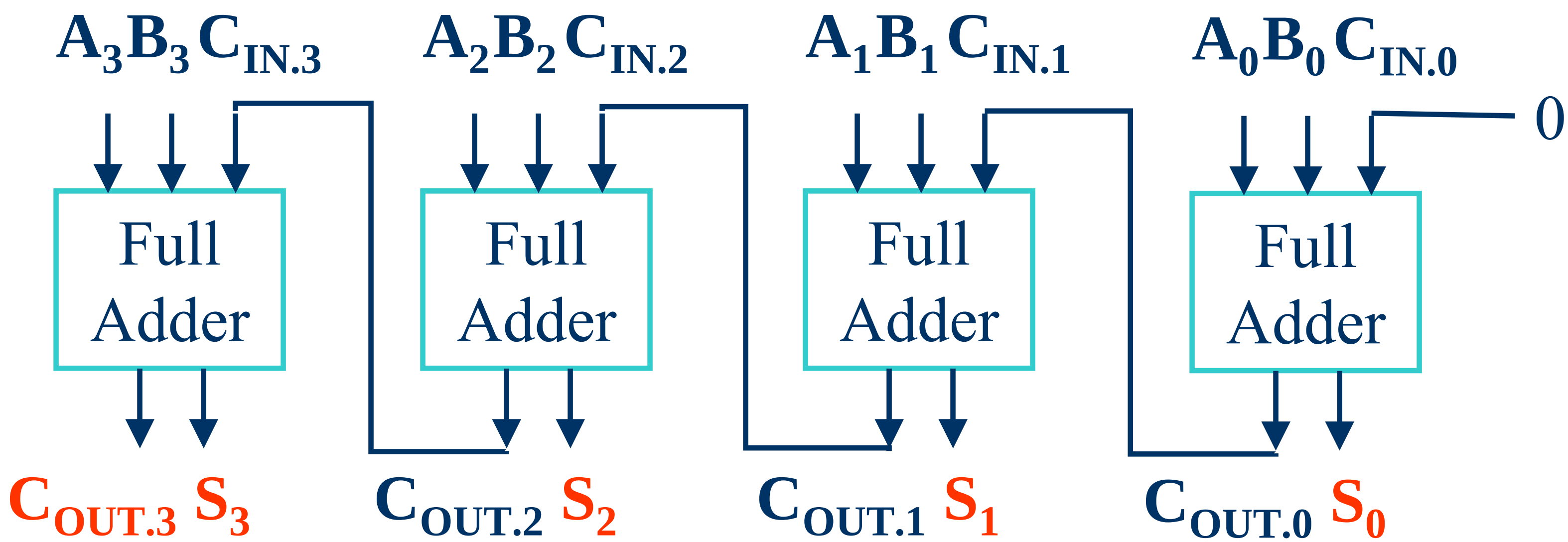
Adders

Comparators

Encoders

Decoders

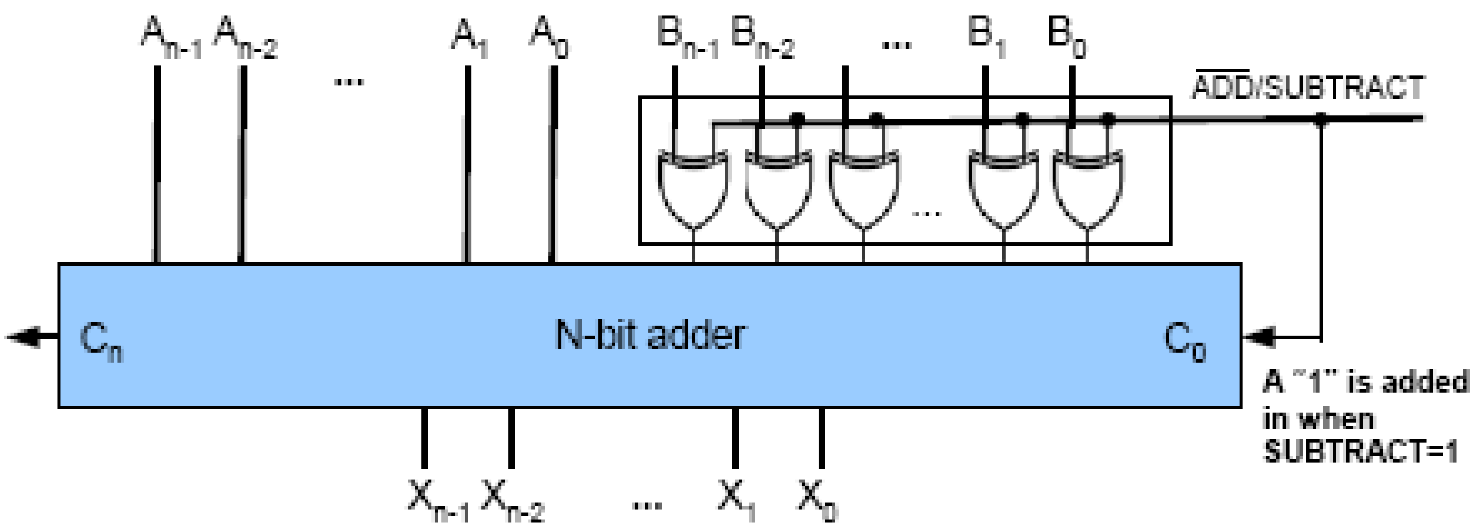
Recall: 4-bit adder from full adders



Result = $C_{OUT.3}\, S_3\, S_2\, S_1\, S_0$

Recall: 4-bit adder / subtractor

Subtraction



When $\overline{\text{ADD}}$/SUBTRACT = 0, addition is performed
Otherwise, subtraction

Example #1: BCD Addition

Recall: A Binary Coded Decimal (BCD) is a decimal number represented by a 4-bit code

example: $45_{10} \leftrightarrow$ 0100 0101

Note that when your sum of two number in BCD exceeds 9 (1001), the output is no longer in BCD as observed in the table given below.

$A_3\ A_2\ A_1\ A_0$	$B_3\ B_2\ B_1\ B_0$	S = A + B	Remarks
0 0 0 1 (1)	0 1 0 1 (5)	0 1 1 0 (6)	Sum in BCD
0 0 0 1 (1)	1 0 0 0 (8)	1 0 0 1 (9)	Sum in BCD
1 0 0 1 (9)	0 0 1 0 (1)	1 0 1 1 (11)	Not in BCD
1 0 0 1 (9)	0 1 1 1 (7)	1 0 0 0 0 (16)	Not in BCD
1 0 0 1 (9)	1 0 0 1 (9)	1 0 0 1 0 (18)	Not in BCD

Example #1: BCD Addition (cont'd)

A solution to this problem is to add 6 (0110) to the binary sum to produce the correct BCD format*.

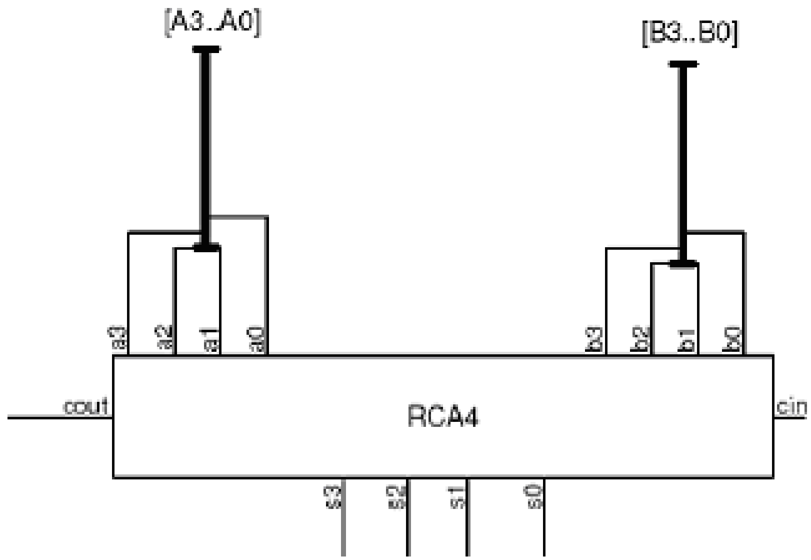
A_3 A_2 A_1 A_0	B_3 B_2 B_1 B_0	S = A + B	S + 6
1 0 0 1 (9)	0 0 1 0 (1)	1 0 1 1 (11)	1 0 0 0 1 (1 1 in BCD)
1 0 0 1 (9)	0 1 1 1 (7)	1 0 0 0 0 (16)	1 0 1 1 0 (1 6 in BCD)
1 0 0 1 (9)	1 0 0 1 (9)	1 0 0 1 0 (18)	1 1 0 0 0 (1 8 in BCD)

Q: How do we redesign our 4-bit RCA to create a BCD adder?

* source: Digital Design by Mano

Hint

Consider our normal 4-bit RCA



When do we know that the sum and cout are not in the correct BCD format?

Solution

carry_out = cout + s3 (s2 + s1)

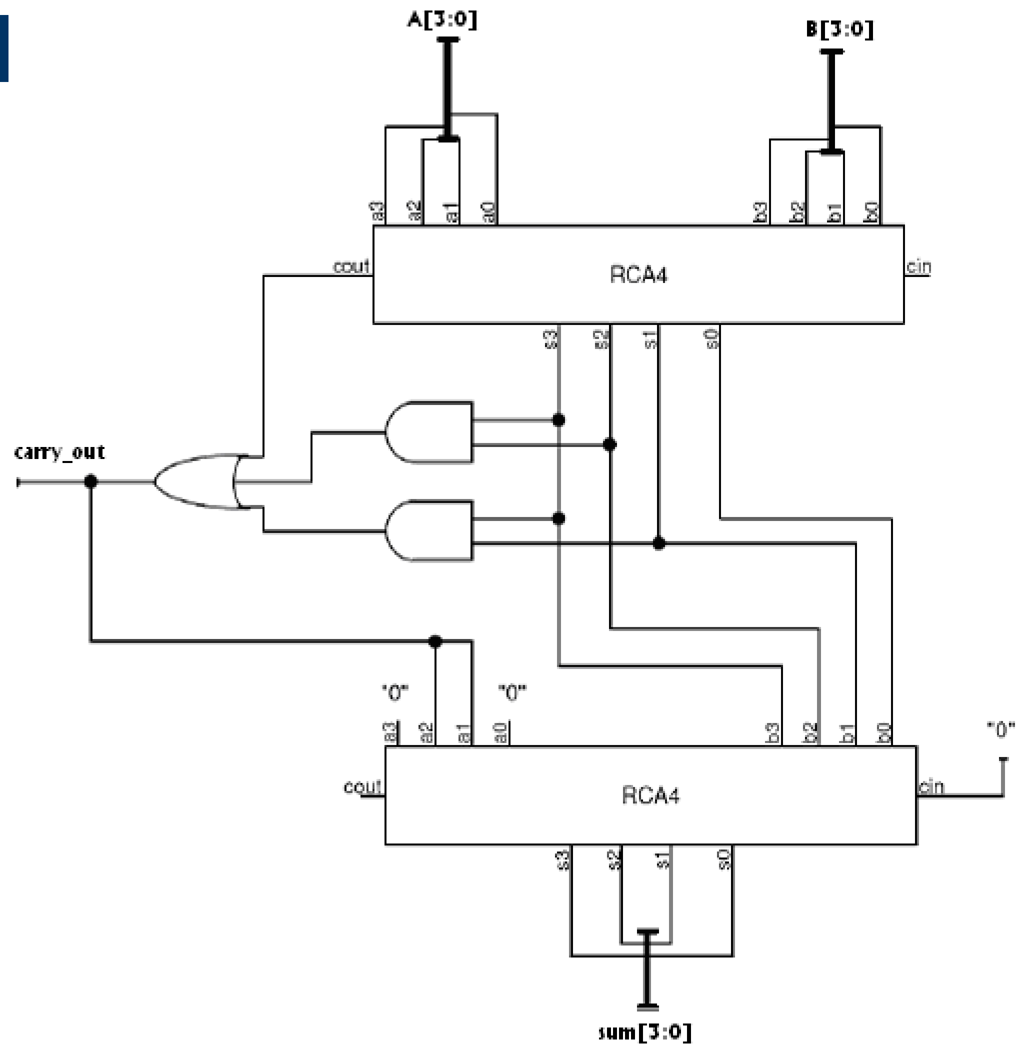
Note: when the BCD carry_out is ‘1’, then the new sum should be added a 6 (0110) otherwise, a 0 (0000)

Solution (cont'd)

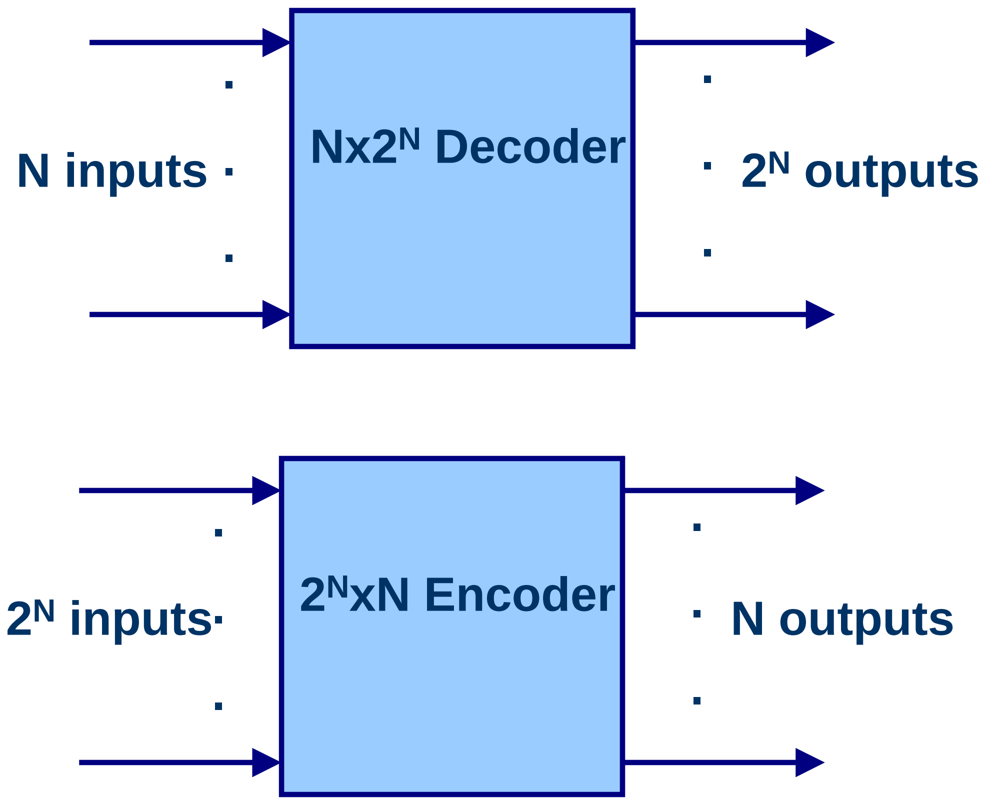
When carry_out is '0'; original sum is added to "0000"

When carry_out is '1'; original sum is added to "0110"

BCD carry is carry_out and BCD sum is sum[3:0]



Recall: Encoder and Decoder



N variables to 2^N minterms

Example #2

Given the functions

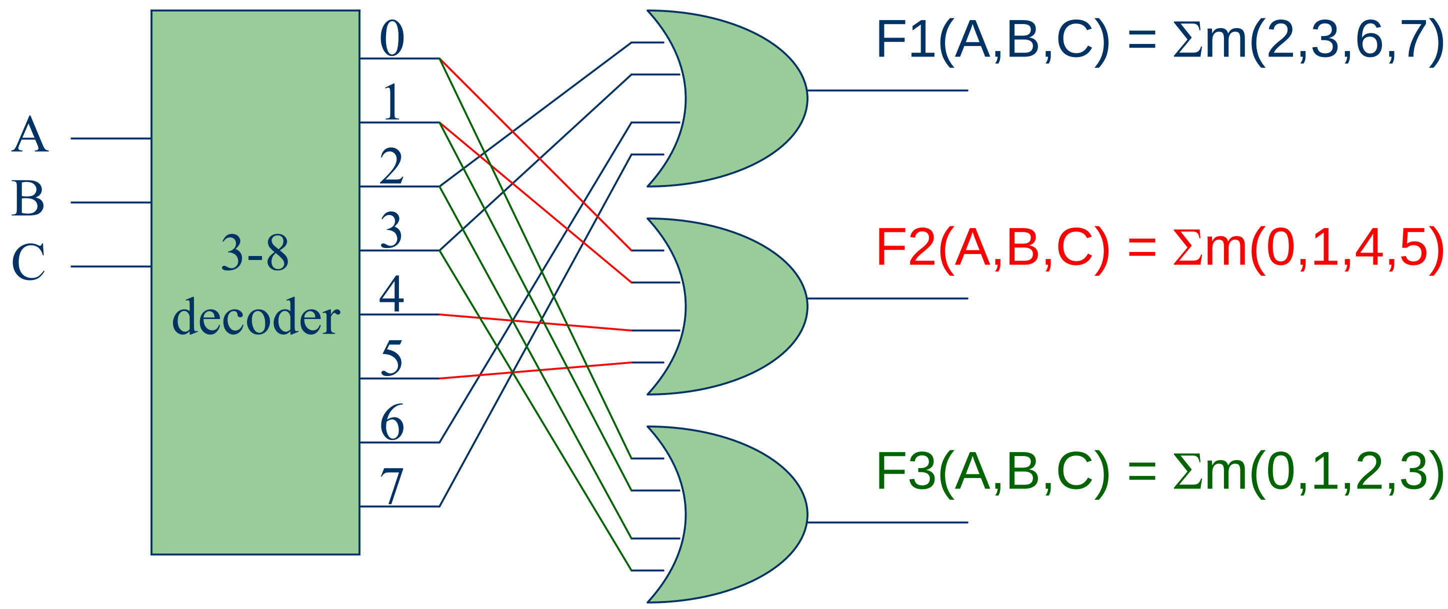
$$F1(A,B,C) = \Sigma m(2,3,6,7) + d(0)$$

$$F2(A,B,C) = \Sigma m(0,1,4,5) + d(6)$$

$$F3(A,B,C) = \Sigma m(0,1,2,3) + d(4)$$

Implement using a decoder and 3 OR gates

Example #2: Solution



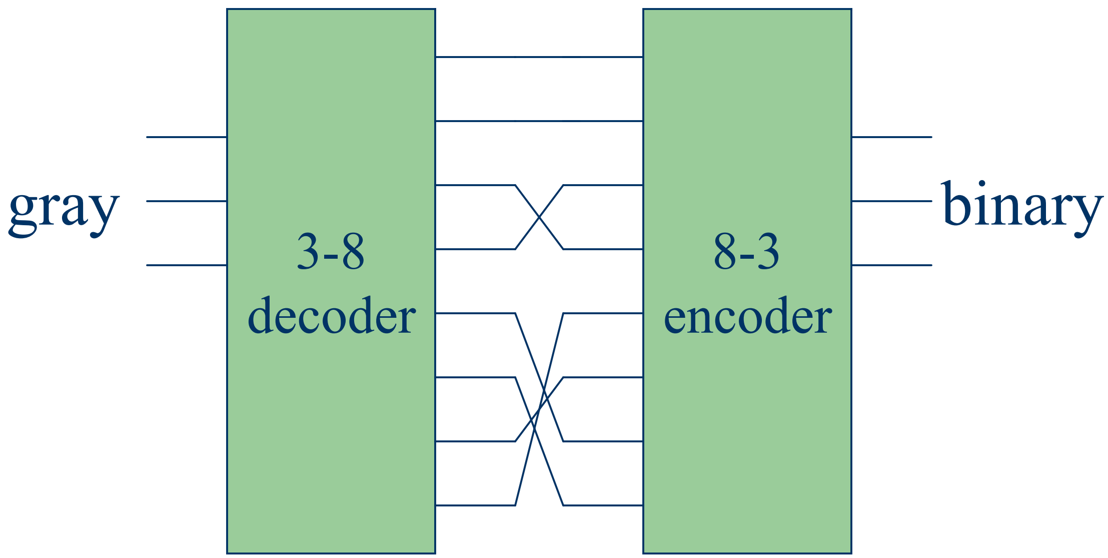
Note: don't cares no longer included as they will only increase the size of the OR gate

Example #3

Using a 3-input binary decoder and an eight-input binary encoder, implement a converter from the three-bit Gray code into the three-bit binary code.

Example #3: Solution

Gray		Binary
000		000
001		001
011		010
010		011
110		100
111		101
101		110
100		111



Quiz

A majority function is a function that outputs a ‘1’ when majority of its inputs is ‘1’ and outputs a ‘0’ when majority of its inputs is ‘0’. Design a 4-bit majority circuit using a decoder (specify the size) and a gate (specify type and size). Note: is there are equal number of 1’s and 0’s in the input, the circuit is free to choose to output a ‘0’ or a ‘1’ (good as having a don’t care)

Quiz: Solution

- $M(A,B,C,D) = \Sigma m(7,11,13,14,15)$
- Need a 4-to-16 decoder and a 5-input OR gate
- Use A,B,C and D as inputs of the decoder
- Route pins 7,11,13,14 and 15 of the 4-to-16 decoder to the inputs of the OR gate
- Output of OR gate is M(A,B,C,D)