CS221: Digital Design

QM Logic Minimization

Dr. A. Sahu

Dept of Comp. Sc. & Engg.

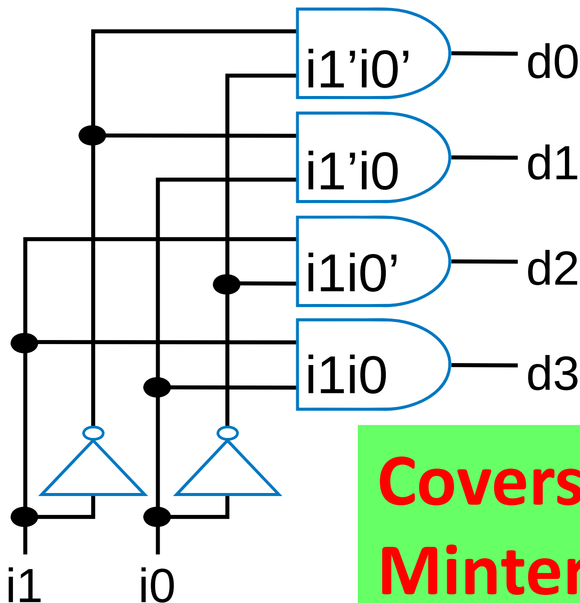
Indian Institute of Technology Guwahati

Outline

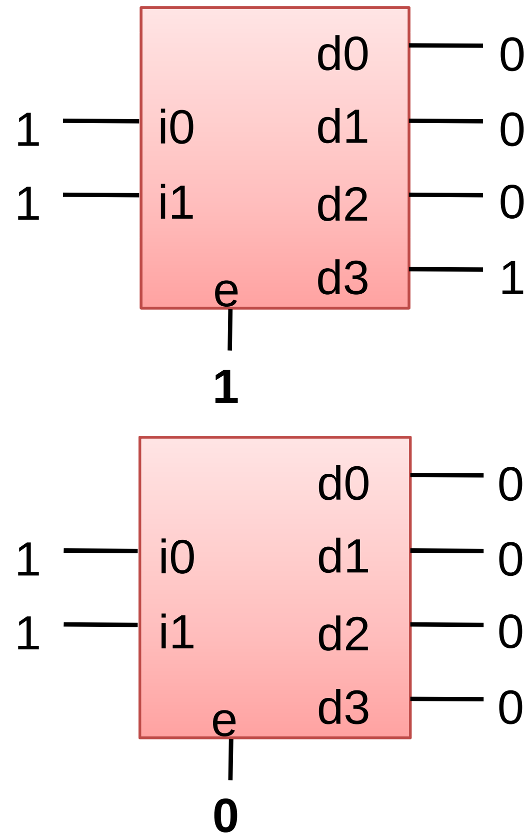
- Study of Components
  - Logic Implementation Using MUX & Decoder
  - 4 Bit Adder, BCD adder
- Quine-McCluskey (QM) Logic Minimization
- Examples
- Writing C/C++ program for QM Method

Decoders

- Internal design
  - AND gate for each output to detect input combination
- Decoder with enable e
  - Outputs all 0 if e=0, Regular behavior if e=1
- n-input decoder: 2^n outputs



Covers All Minterms



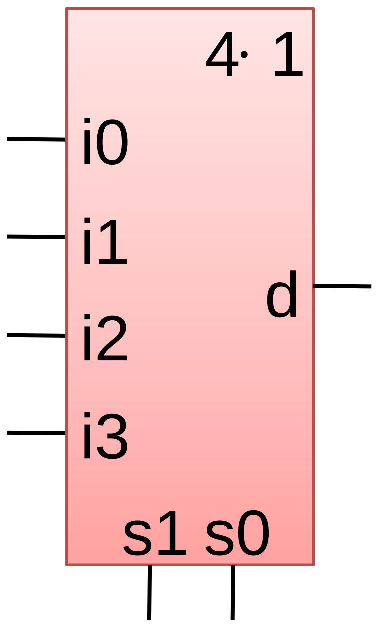
Boolean Function Implementation using Decoders

- Using a n-to-2n decoder and OR gates any functions of n variables can be implemented.
- Example:

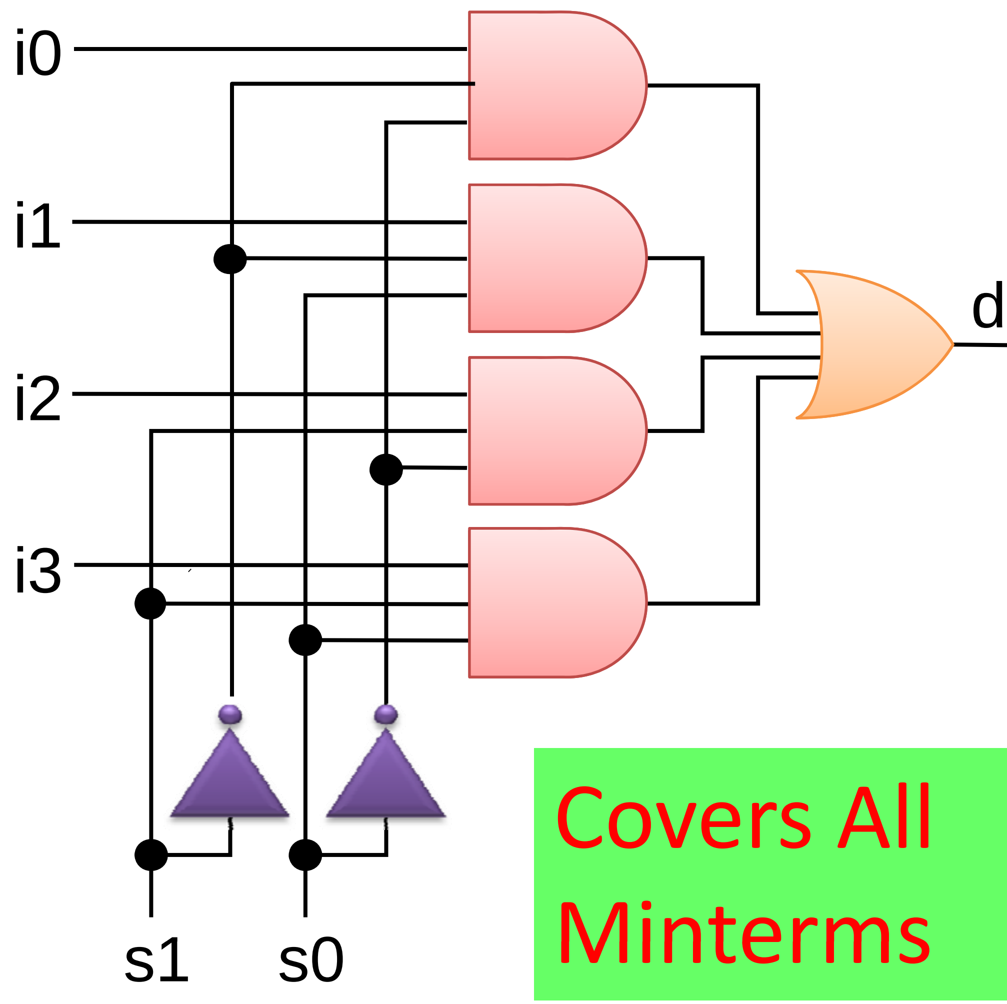
  $S(x,y,z)= \Sigma(1,2,4,7)$, $C(x,y,z)=\Sigma(3,5,6,7)$
- Functions S and C can be implemented using a 3-to-8 decoder and two 4-input OR gates

Decoder: Covers All Minterms

Mux

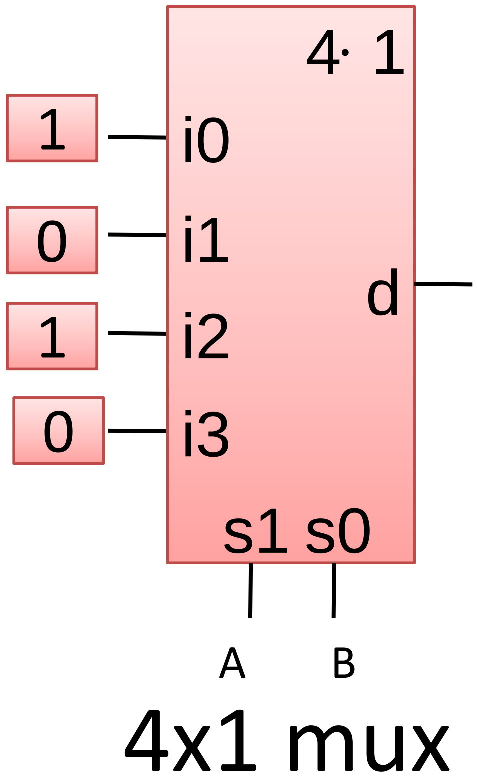


4x1 mux



Covers All Minterms

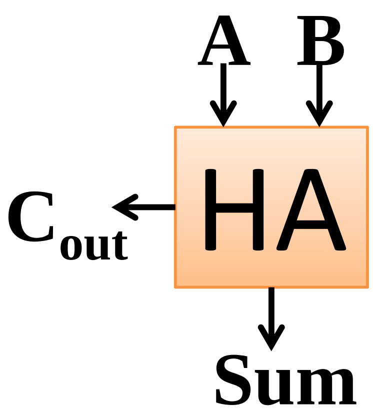
Implementing logic Function using MUX



$F\ (A, B) = \sum m(0,2)$

Adding Two One-bit Operands

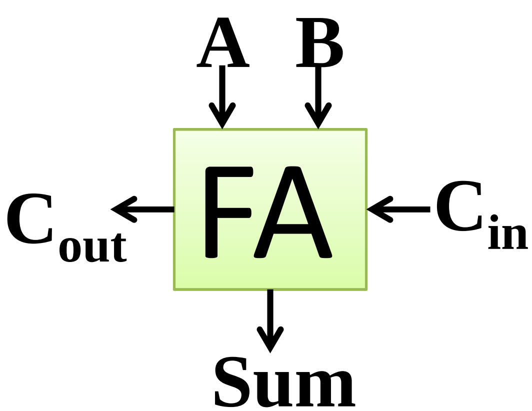
- One-bit Half Adder:



$$Sum = A \oplus B$$
$$Cout = A.B$$

A	B	Sum	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- One-bit Full Adder:

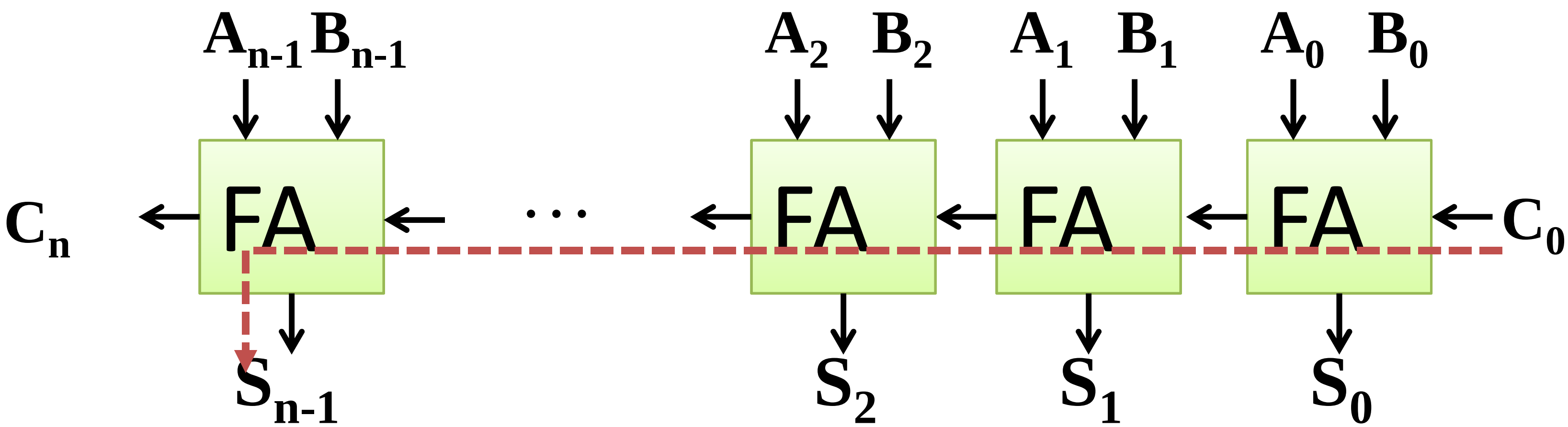


$$\mathbf{Sum = A \oplus B \oplus Cin}$$
$$\mathbf{Cout = A.B + B.Cin + A.Cin}$$

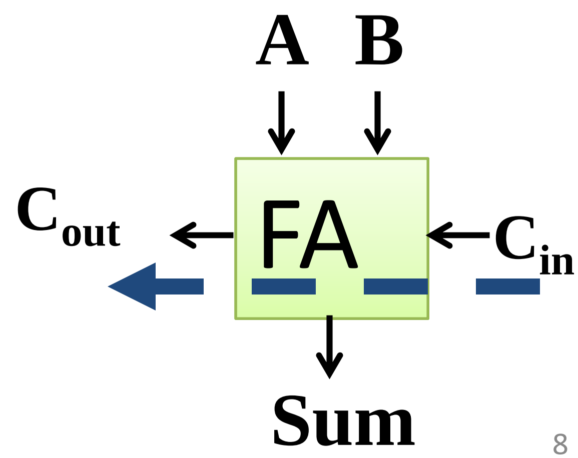
C_{in}	A	B	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

N-Bit Ripple-Carry Adder: Series of FA Cells

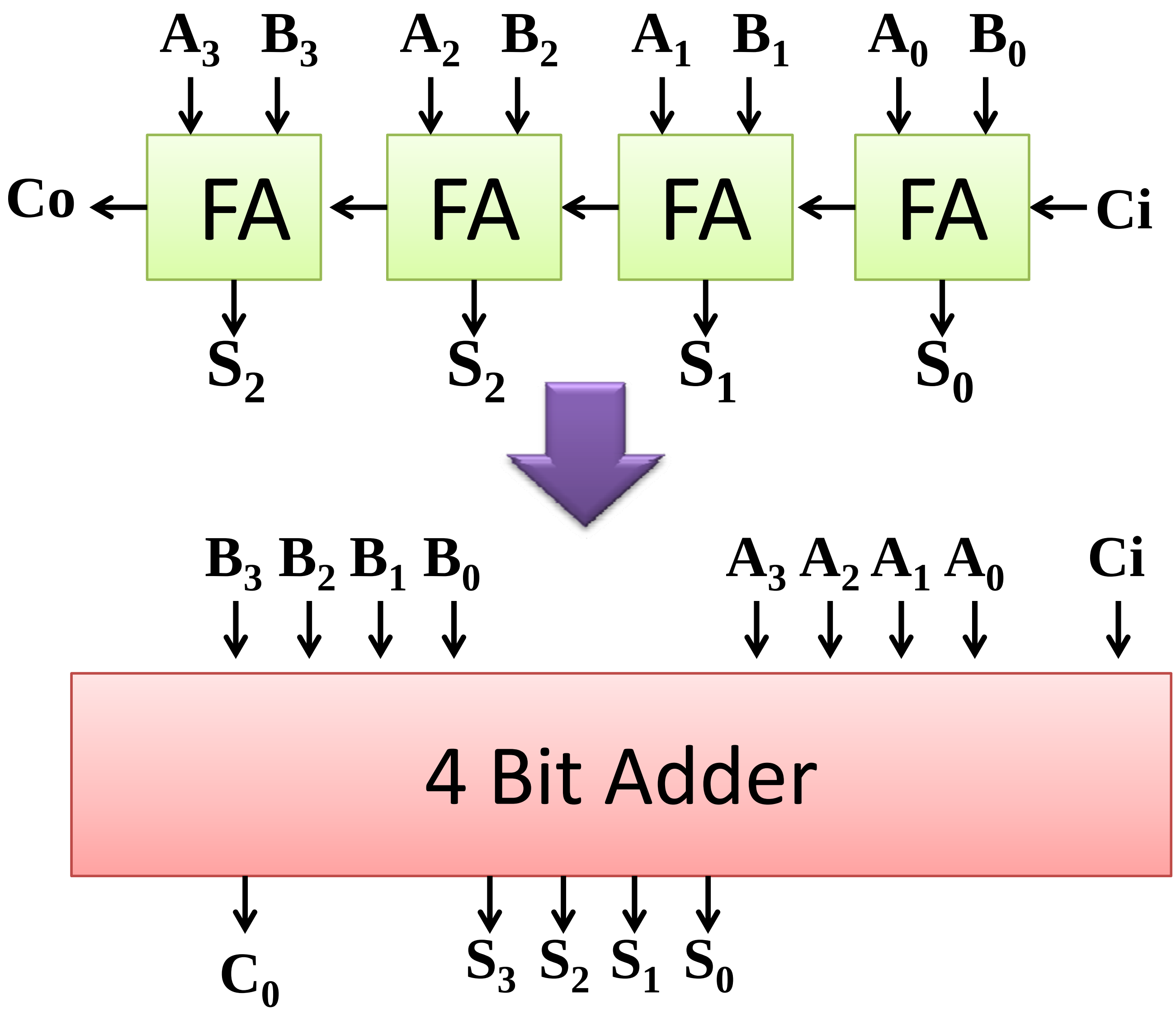
- To add two n-bit numbers



- Adder delay = Tc * n
- Tc = (C_{in} to C_{out} delay) of a FA

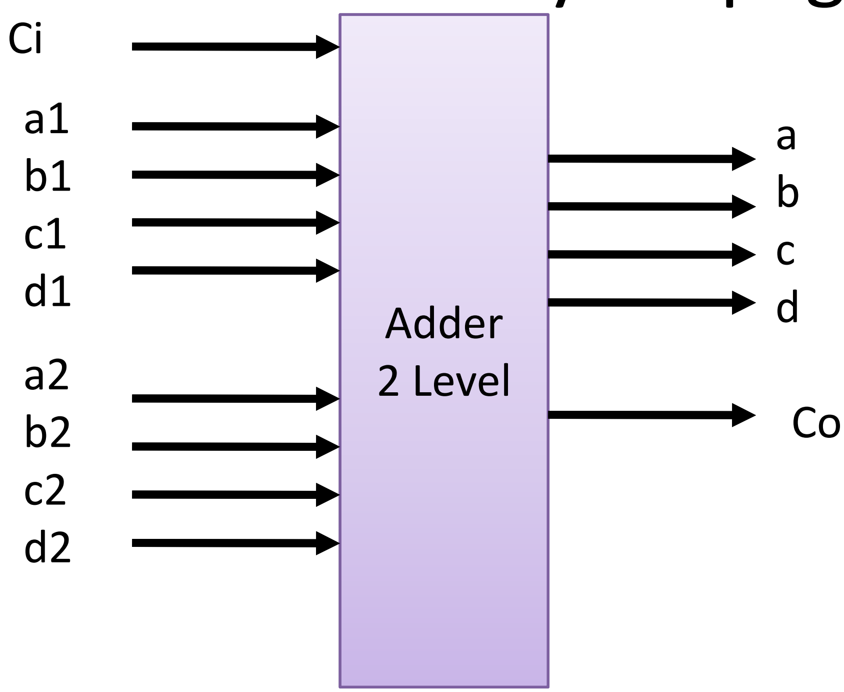


4 bit Binary Adder



Binary Adder (Two Level)

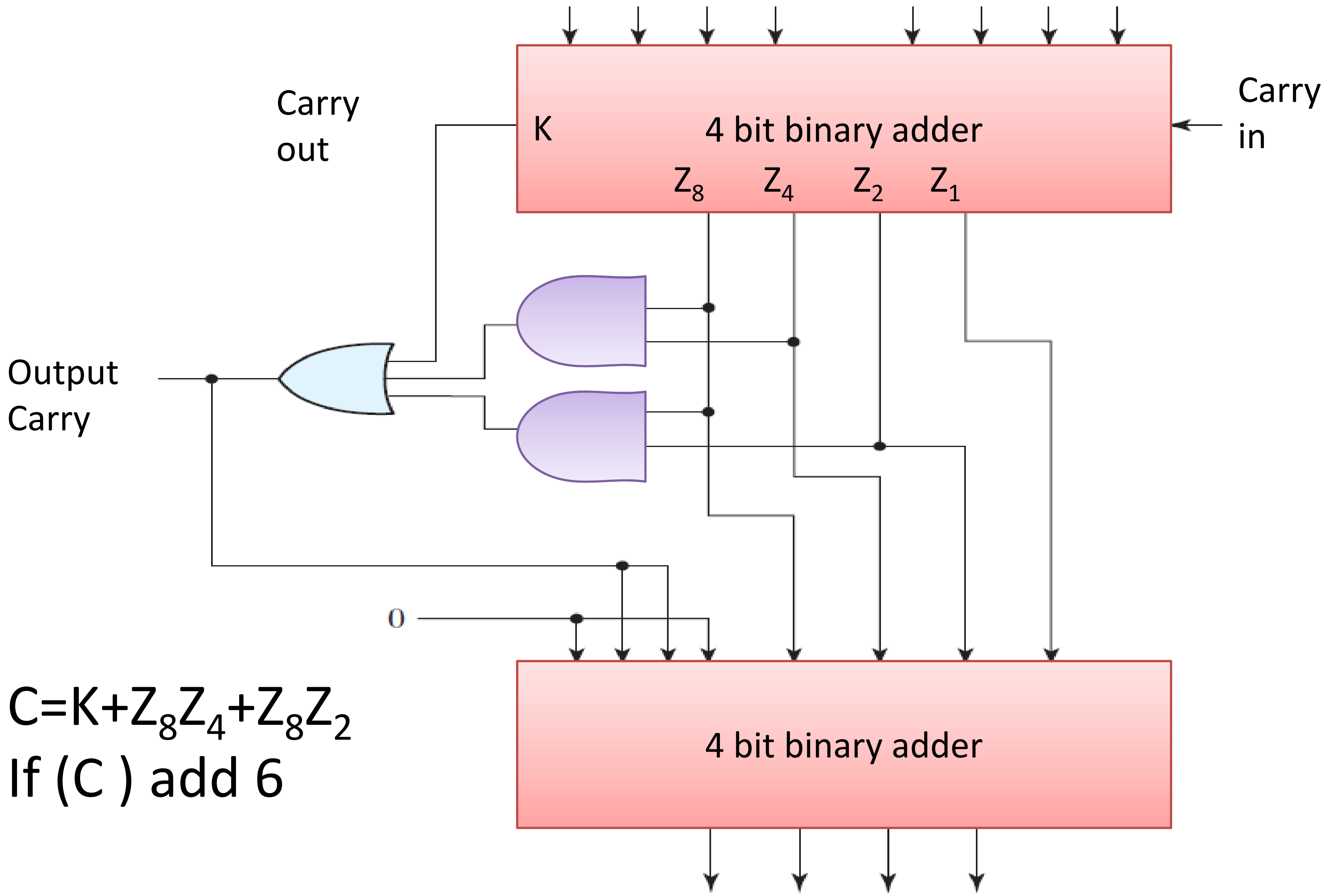
- Treat as 9 input & 5 output functions
- Generate Truth Table for each outputs
- Solve each function using KMAP/QM Method
- Only Two Level: No carry Propagation



<u>Decimal Adder</u>

- Decimal numbers are represented with BCD code.
- When two BCD digits A and B are added
  - if A+B<10 result is a valid BCD digit
  - if A+B>9 result will not be valid BCD digit. It must be corrected by adding 6 to the result
- If A+B >9 add 6 to solve this issue

Decimal Adder

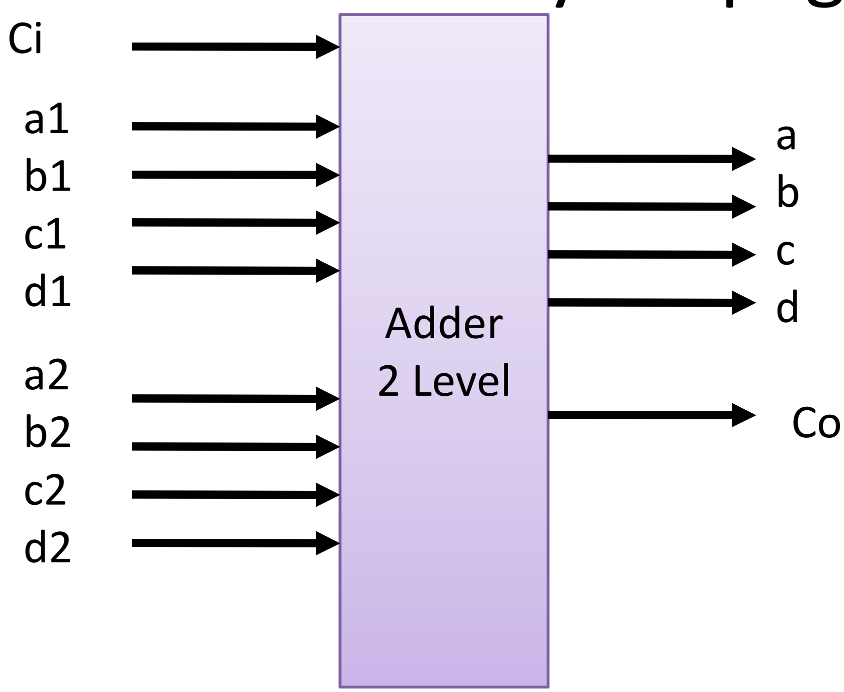


$C=K+Z_8Z_4+Z_8Z_2$

If (C) add 6

BCD Adder (Two Level)

- Treat as 9 input & 5 output functions
- Generate Truth Table for each outputs
- Solve each function using KMAP/QM Method
- Only Two Level: No carry Propagation



Quine-McCluskey Method for Minimization

- KMAP methods was practical for at most 6 variable functions
- Larger number of variables: need method that can be applied to computer based minimization
- **Quine-McCluskey** method

- For example:

$$\sum m(0,1,2,3,5,7,13,15)$$

QM Method

- Phase I : finding Pis
  - Tabular methods: Grouping and combining
- Phase II: Covers minimal PIs

QM Method

- Minterms that differ in one variable's value can be combined.
- Thus we list our minterms so that they are in groups with each group having the same number of 1s.
- So the first step is ordering the minterms according to their number of 1s (0-cube list)
- only minterms residing in adjacent groups have the chance to be combined.):

QM Method

$$\sum m(0,1,2,3,5,7,13,15)$$

0_Cube	
0	0000
1	0001
	0010
2	0011
	0101
3	0111
	1101
4	1111

QM Method: Combining Adjacent

- Compare minterms of a group with those of an adjacent one to form 1-cube list.
- When doing the combining, we put checkmark alongside the minterms in the 0-cube list that have been combined.

QM Method

0_Cube	
0	0000 √
1	0001 √
	0010 √
2	0011 √
	0101 √
3	0111 √
	1101 √
4	1111 √

1_Cube		
0	000X	1,2
	00X0	
1	00X1	2,3
	0X01	
	001X	
2	0X11	3,4
	01X1	
	X101	
3	X111	4,5
	11X1	

QM Method: Combining Adjacent

- Do same combination of comparing adjacent group minterms
  - To form 2-cubes, 3-cubes and so on.

- Only minterms of adjacent groups have the chance of being combined
  - **Which have an X in the same position.**

QM Method

1_Cube	
0	000X √
	00X0 √
1	00X1 √
	0X01 √
	001X √
2	0X11 √
	01X1 √
	X101 √
3	X111 √
	11X1 √

2_Cube	
0	00XX *
1	0XX1 *
2	X1X1 *

Q-M Method: Cover PIs

- PIs : terms left without checkmarks.
- After identifying our PIs, we list them against the minterms needed to be covered

$$\sum m(0,1,2,3,5,7,13,15)$$

	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 1	0 1 1 1	1 1 0 1	1 1 1 1
0 0 x x	√	√	√	√				
0 x x 1		√		√	√	√		
x 1 x 1					√	√	√	√
Func	√	√	√	√	√	√	√	√

QM Method : Covers

- To find a minimal cover, we first need to find essential PIs
- To do this we need to find columns that only have one checkmark in them, the according row will thus show the essential PI.
- After identifying essential PIs, that are necessarily part of the cover, we cover any remaining minterms using a minimal set of PIs.

In this example: $F(a,b,c,d) = \overline{a}\,\overline{b} + bd$

Thanks