DIGITAL SYSTEMS AND MICROPROCESSORS (ELE2002M)

LECTURE -4

COMBINATIONAL LOGIC – ADDERS AND SUBTRACTORS

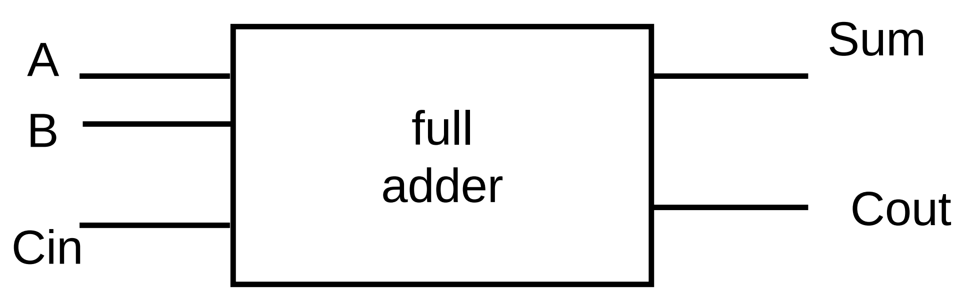
Instructor:

Assoc. Prof. Dr. Edmond Nurellari

Full Adder

Develop a truth table and Boolean expressions for the full adder, this circuit also includes a carry in.

Cin	A	B	S	C
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Truth table for full adder

Cin	A	B	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Exercise:

Complete the Karnaugh maps for the Sum and the Carry out columns

K-maps for Sum and Carry

Sum – 1 when odd number of inputs is 1 = XOR gate

Cin \ AB	00	01	11	10
0				
1				

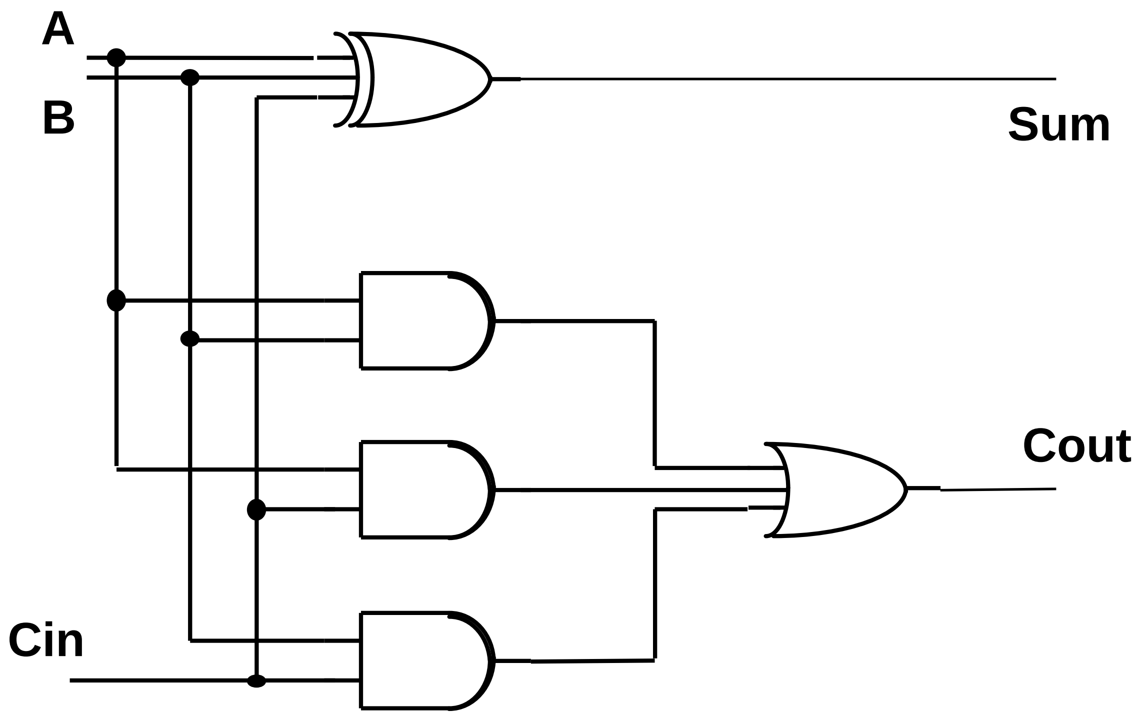
Sum = Cin xor A xor B

Carry out - simplifies to 3 pairs

Cin \ AB	00	01	11	10
0				
1				

Cout = A.B + A.Cin + B.Cin

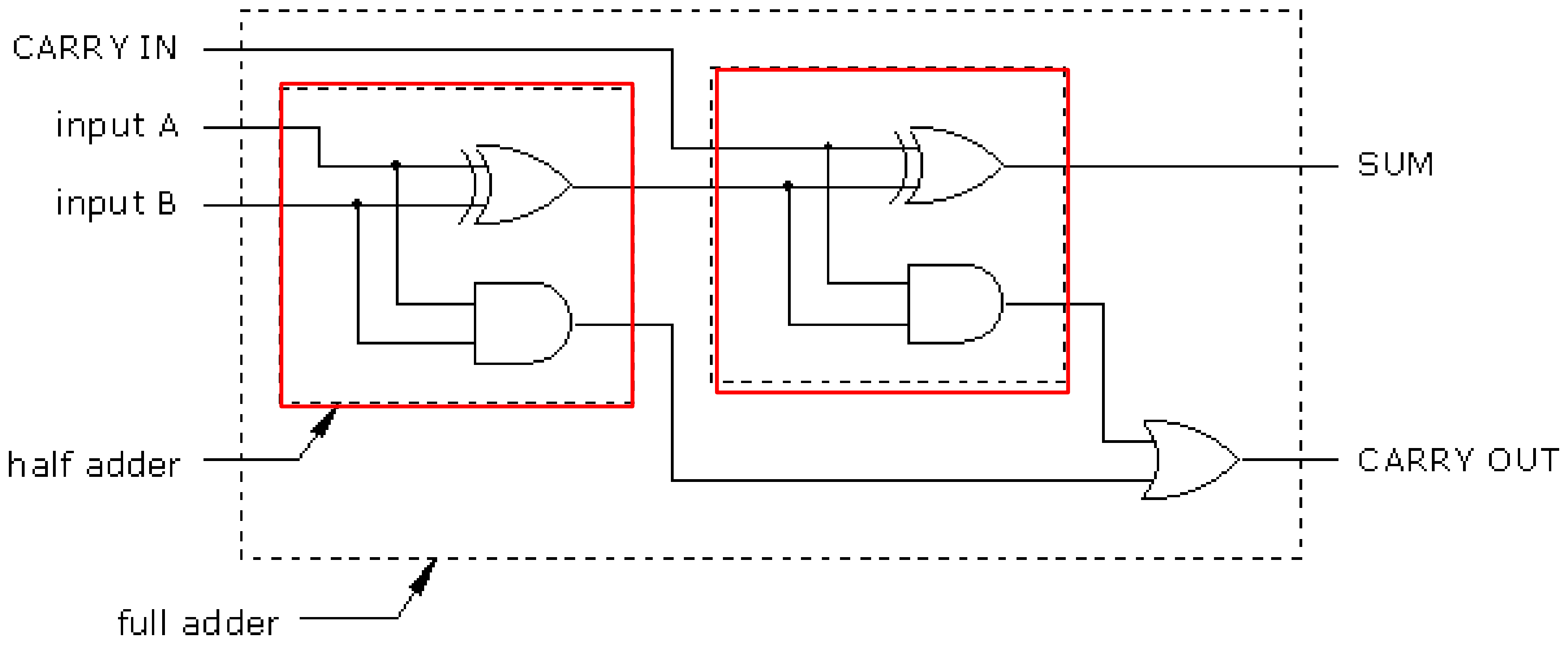
Full adder circuit



Sum = Cin xor A xor B

Cout = A.B + A.Cin + B.Cin

Full adder circuit with 2 half adders



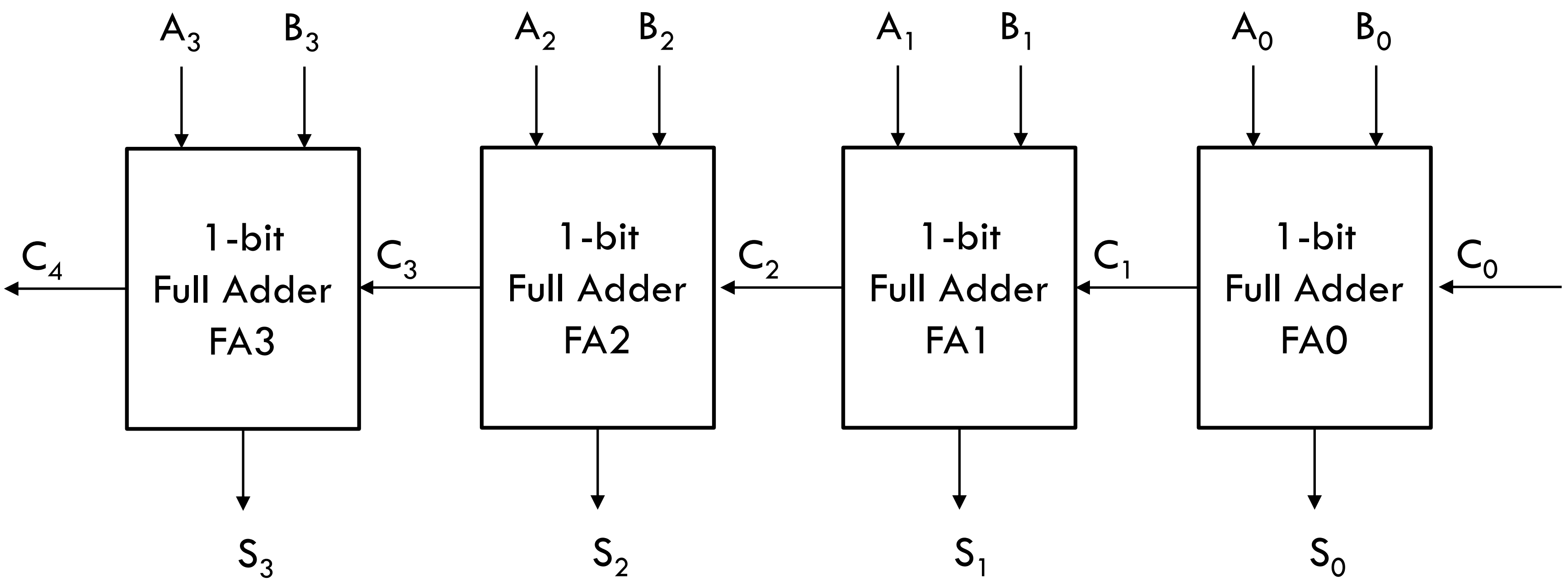
Binary Adder (Ripple Carry Adder)

- To add two multi-digit binary numbers.
- Consists of several full-adders connected in series.
- The carry-out from previous adder is the carry in to the next full adder.
- The carry out of the final adder (MSB) is the final carry out.
- The carry in of the first adder (LSB) is usually zero.

4-bit Addition with Ripple Carry Adder

$A = A_3A_2A_1A_0$

$B = B_3B_2B_1B_0$



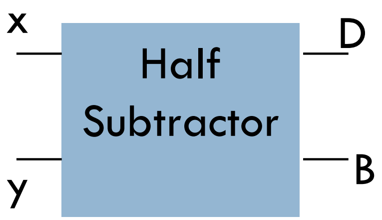
Carry Propagation

- Total propagation time for an addition operation.
- Total propagation time is equal to the propagation delay of a typical gat times the number of gate levels in the circuit.
- Propagation delay for full adders?
- How about generating carry values in advance and use them to calculate the sum?
- An adder that generates carry values in advance (using extra hardware) is known as "Carry Look-ahead Adder"

- Read "Carry Propagation", page 147 in the book.

Half Subtractor

- A simple 2-input half-subtractor consists of 2 inputs and 2 outputs.
- Two inputs x and y form the *minuend* and the *subtrahend*.
- D is the difference output
- B is the borrow output

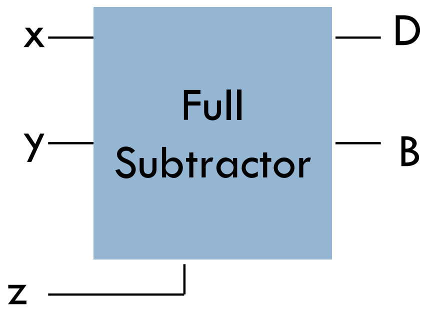


x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- $D = x'y + xy'$ and $B = x'y$
- Imlementation?

Full Subtractor

- Full subtractor has 3 inputs and 2 outputs.
- X, y and z are inputs to be subtracted, where z is the borrow from previous stage.
- D and B are the outputs.
- D =
- B =



x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Binary Subtractor

- How did we do Binary subtraction (A-B) earlier?
  - Input A added to 2’s complement of B.
- How to implement this?
  - Complement each data bit of B by placing an inverter between B and input of full adder.
- How to add a ‘1’ to get 2’complement?
  - Make C_0 as ‘1’

Adder-Subtractor Unit

- A control input 'M' is used to to determine whether the operation is addition or subtraction

- If M is '0' then addition is performed.

- If M is '1' then input B_i is inverted and C_i is taken as '1' to perform a subtraction operation.

M	A_i	B_i	x_i	y_i
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

M	xi	yi	Ci
0	A_i	B_i	0
1	A_i	B_i'	1