Digital Systems

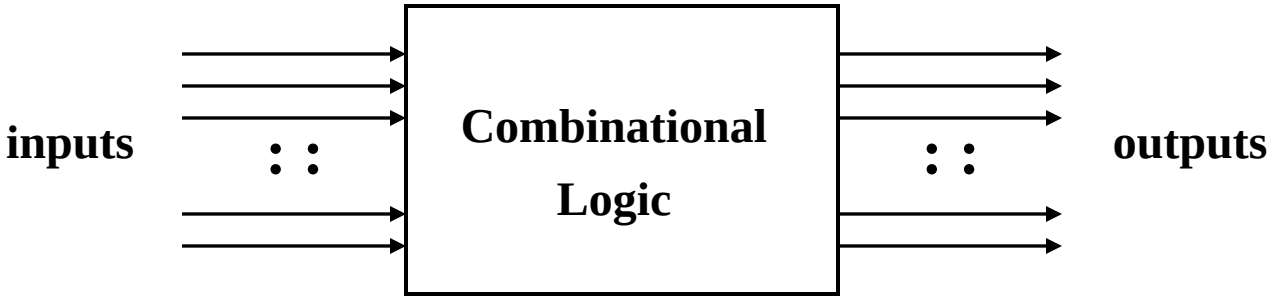
18B11EC213

Module 1: Boolean Function Minimization Techniques and Combinational Circuits-12

Dr. Saurabh Chaturvedi

Combinational Logic Circuits

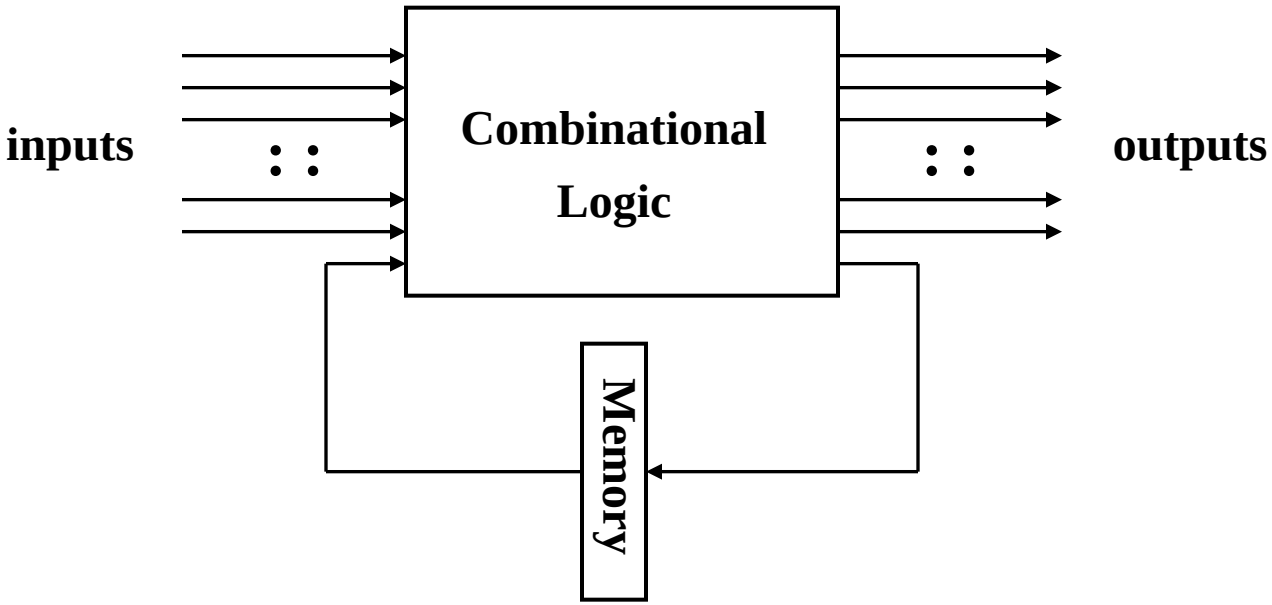
- Two classes of logic circuits:
  - combinational
  - sequential
- **Combinational Circuit**:



Each output depends entirely on the immediate (present) inputs.

Cont..

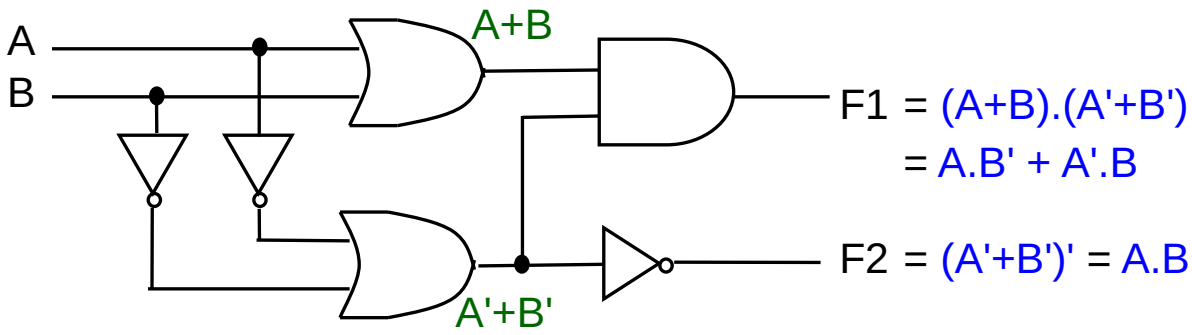
- **Sequential Circuit**:



Output depends on both present and past inputs.
Memory (via feedback loop) contains past information.

Analysis Procedure

- Analyze the function of the given combinational logic circuit:



Steps:

1. Label the inputs and outputs.
2. Obtain the functions of intermediate points and the outputs.
3. Draw the truth table.
4. Deduce the functionality of the circuit ➲ half adder

A	B	(A+B)	(A'+B')	F1	F2
0	0	0	1	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	1	0	0	1

Design Methods

- Different combinational circuit design methods:
  - Gate-level method (with logic gates)
  - Block-level design method
- Design methods make use of logic gates and useful functional blocks.
  - These are available as integrated circuit (IC) chips.

Gate-level Design: Half Adder

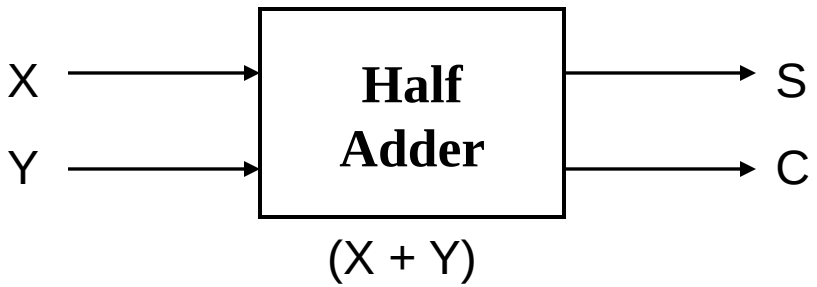
- Design procedure:

  1) State Problem

  Example: Build a Half Adder to add two bits

  2) Determine and label the inputs and outputs of circuit.

  Example: Two inputs and two outputs labelled, as follows:



S: Sum

C: Carry

3) Draw truth table.

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Cont..

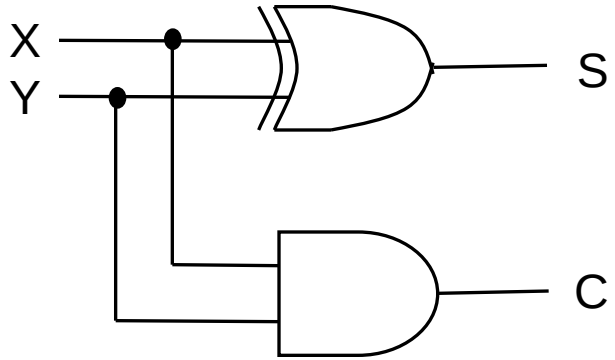
4) Obtain the simplified Boolean function from truth table:

C = X.Y

S = X'.Y + X.Y' = X⊕Y

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

5) Draw the logic diagram.



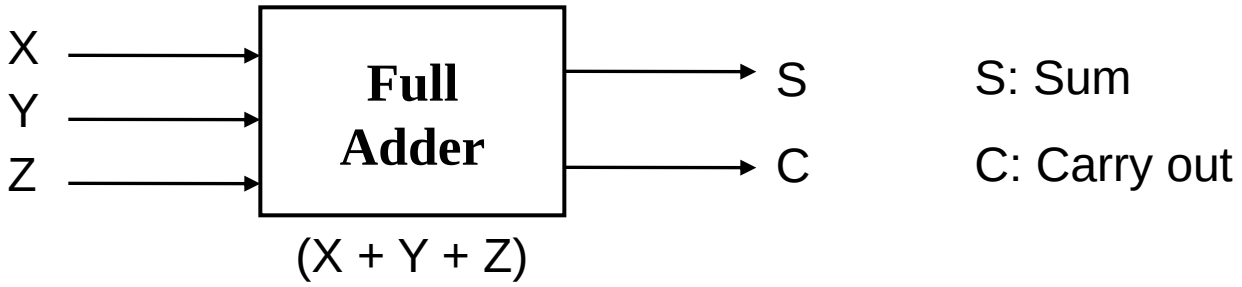
Half Adder

Gate-level Design: Full Adder

- Half adder adds up only two bits.
- To add two binary numbers, we need to add 3 bits (including the carry).
- Example:

	1	1	1		carry
	0	0	1	1	X
+	0	1	1	1	Y
	1	0	1	0	S

This addition operation requires a Full Adder circuit.
A full adder can be made using two half adders.



Cont..

- Truth table:

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Note:
Z: carry in (to the current position)
C: carry out (to the next position)

C

X \ YZ	00	01	11	10
0			1	
1		1	1	1

Using K-map, the simplified SOP forms:

S

X \ YZ	00	01	11	10
0		1		1
1	1		1	

C = X.Y + X.Z + Y.Z
S = X'.Y'.Z + X'.Y.Z' + X.Y'.Z' + X.Y.Z

Cont..

- Alternative formulae using algebraic manipulation:

$$C = X.Y + X.Z + Y.Z$$
$$= X.Y + X.(Y+Y').Z + (X+X').YZ$$
$$= X.Y + XYZ + XY'Z + XYZ + X'YZ$$
$$= X.Y\,(1+Z) + (X \oplus Y).Z$$
$$= X.Y + (X \oplus Y).Z$$

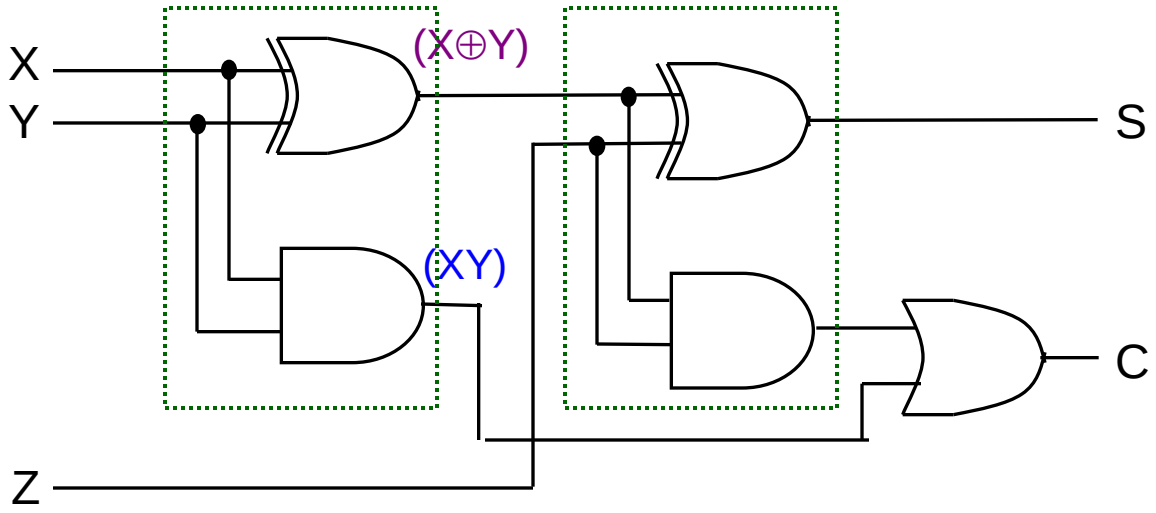
$$S = X'.Y'.Z + X'.Y.Z' + X.Y'.Z' + X.Y.Z$$
$$= X'.(Y'.Z + Y.Z') + X.(Y'.Z' + Y.Z)$$
$$= X'.(Y \oplus Z) + X.(Y \oplus Z)'$$
$$= X \oplus (Y \oplus Z)$$
$$= X \oplus Y \oplus Z$$

Cont..

- Full adder circuit using the following formulae:

  $C = X.Y + (X \oplus Y).Z$

  $S = (X \oplus Y) \oplus Z$



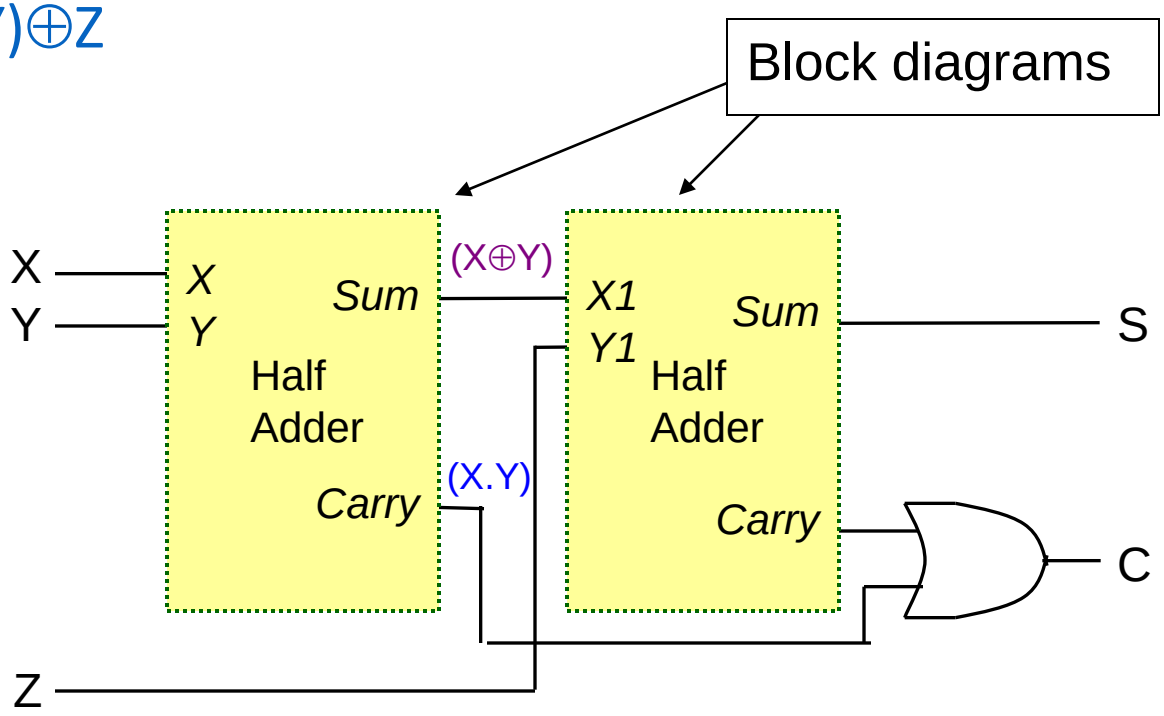
Full adder is made using two half adders and one OR gate.

Cont..

- Full adder circuit using the following formulae:

  $C = X.Y + (X \oplus Y).Z$

  $S = (X \oplus Y) \oplus Z$



Full adder is designed using two half adders and one OR gate.

Gate-level Design: Half Subtractor

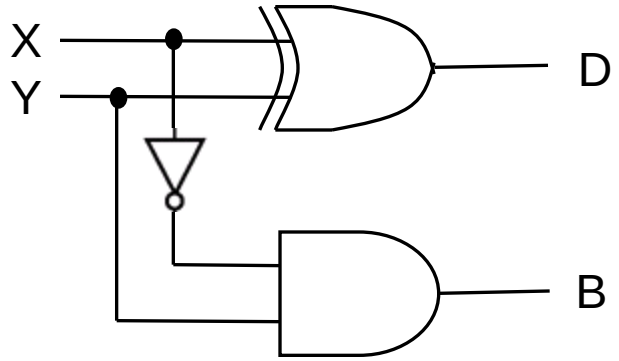
1) Obtain the simplified Boolean functions from truth table:

B = X'.Y

D = X'.Y + X.Y' = X⊕Y

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

2) Draw the logic diagram.



D: Difference

B: Borrow

Gate-level Design: Full Subtractor

- Full subtractor has 3 inputs:
  - Two data inputs
  - One borrow input

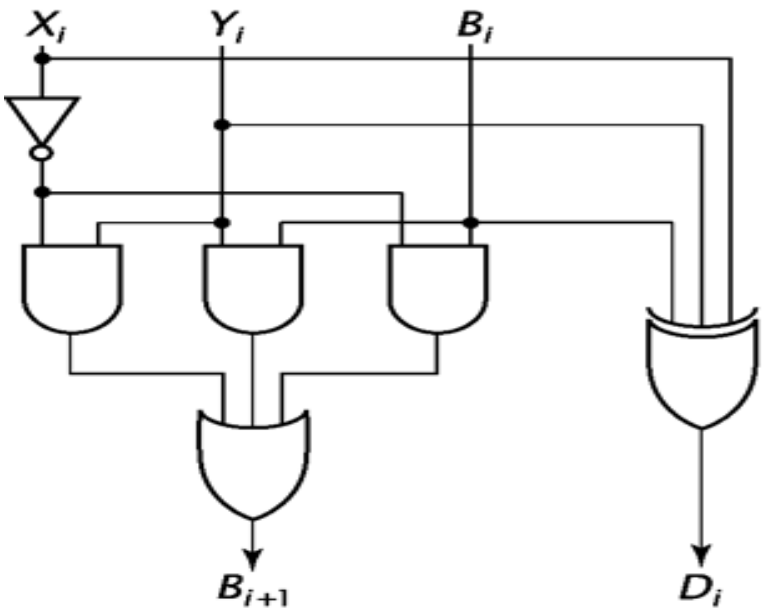
$B_{i+1} = X_i'Y_i + X_i'B_i + Y_iB_i$

$D_i = X_i \oplus Y_i \oplus B_i$

D_i: Difference

B_i: Borrow in

B_{i+1}: Borrow out



(a)

X_i	Y_i	B_i	B_{i+1}	D_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

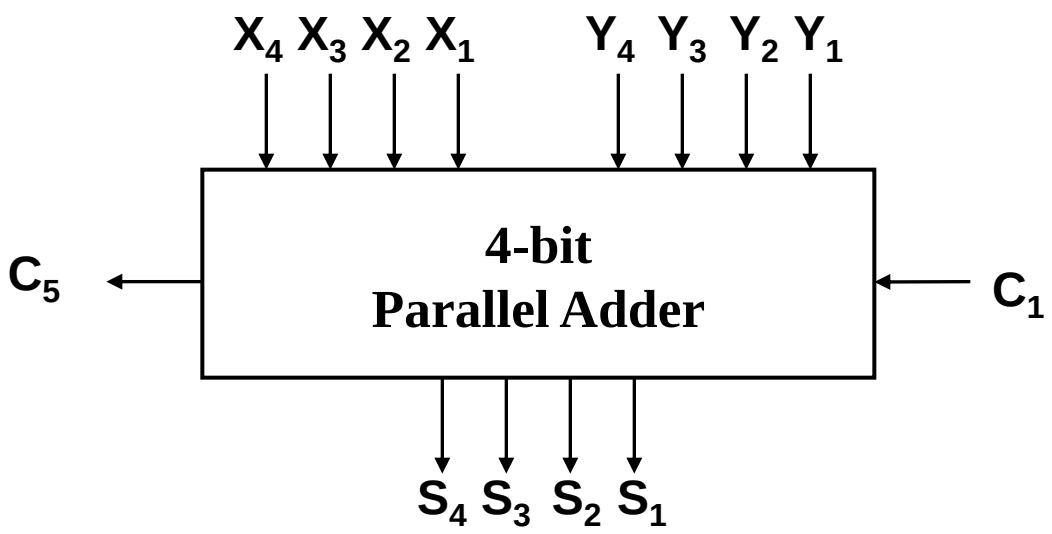
(b)

Block-Level Design Method

- More complex circuits can be built using the block-level method.
- In general, block-level design method (as opposed to gate-level design) relies on algorithms or formulae of the circuit, which are obtained by decomposing the main problem to sub-problems recursively (until small enough to be directly solved by blocks of circuits).

Cont.. 4-bit Parallel Adder

- Consider a circuit to add two 4-bit numbers together and a carry-in to produce a 5-bit result:



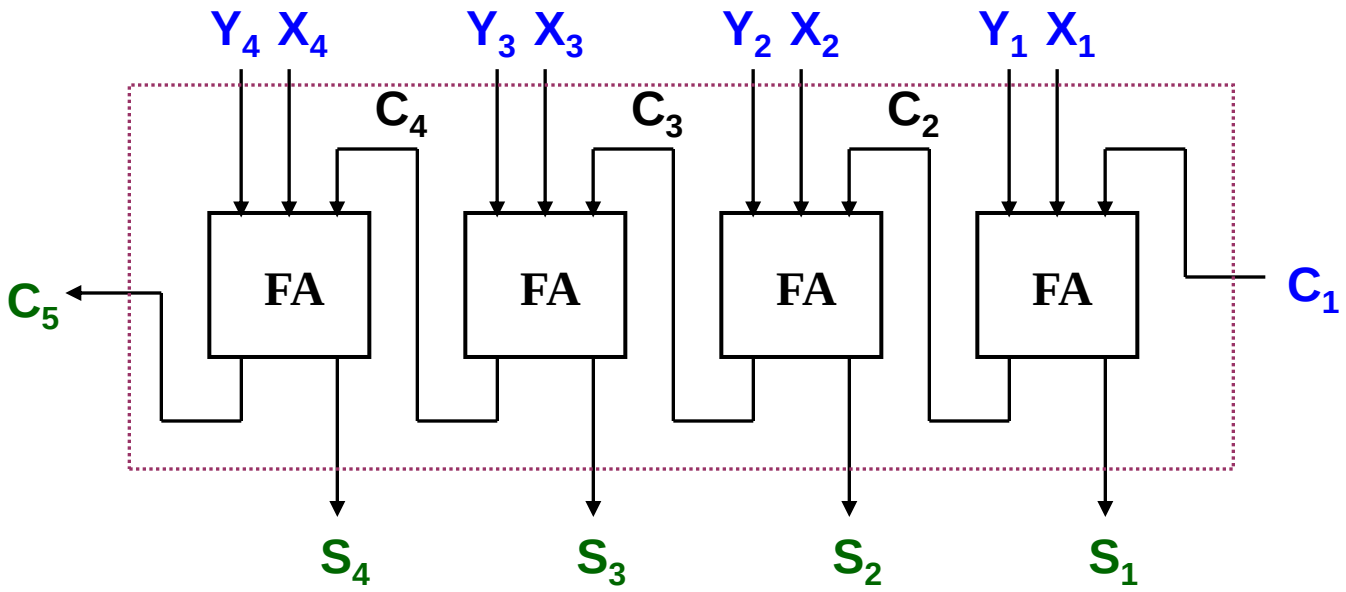
Black-box view of 4-bit parallel adder

5-bit result is sufficient because the largest result is:

$(1111)_2 + (1111)_2 + (1)_2 = (11111)_2$

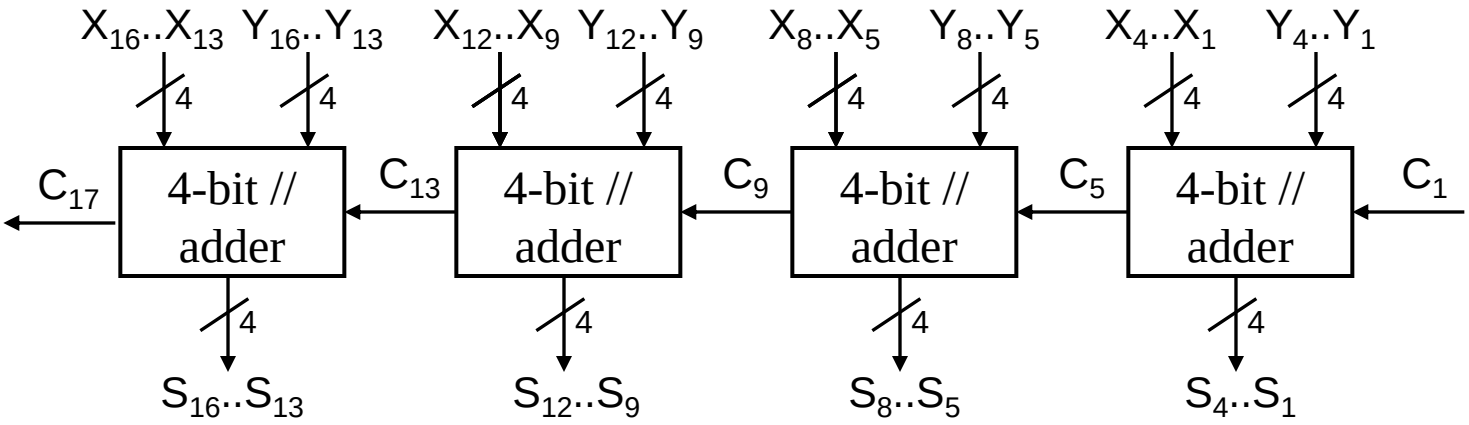
Cont.. 4-bit Parallel Adder

- Cascading four full adders via their carries:



Cont.. 16-bit Parallel Adder

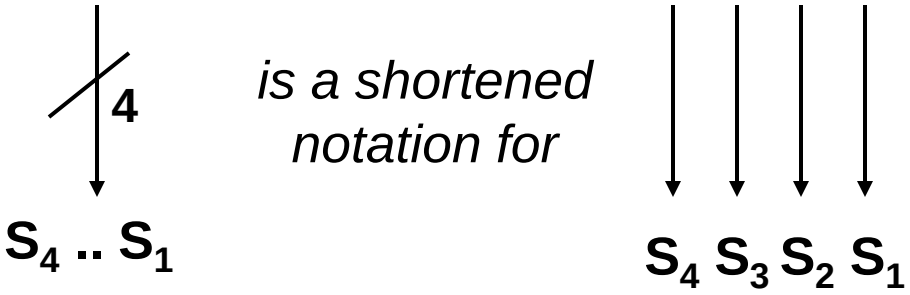
- Larger parallel adders can be built from smaller ones.
- Example: a 16-bit parallel adder can be constructed from four 4-bit parallel adders:



Block diagram of 16-bit parallel adder

Cont.. 16-bit Parallel Adder

- Shortened notation for multiple lines:



- 16-bit parallel adder ripples carry from one 4-bit block to the next.
- Such ripple-carry circuits are “slow” because of long delays needed to propagate the carries.

References

- M. M. Mano, *Digital Logic and Computer Design*, 5th ed., Pearson Prentice Hall, 2013.
- R. P. Jain, *Modern Digital Electronics*, 4th ed., Tata McGraw-Hill Education, 2009.