Department of Computer Science & Engineering (CSE)

# Course Title: Digital Logic Design

Shahriar Rahman Khan
Lecturer
Dept. of CSE, MIST

Course Code: CSE 103
Credit Hr: 3.00
Contact Hr: 3.00

# Overview

- What is MSI and PLD?
- Binary parallel Adder
- Binary Adder-Subtractors
- Carry Propagation
- Decimal Adder
  - BCD Adder
- Magnitude Comparator
- Decoders and Encoders
- Priority Encoders
- De-multiplexers and Multiplexers

# MSI Components

- Scale of Integration = Complexity of the Chip
  - SSI: small-scale integrated circuits, 1-10 gates
  - MSI: medium-scale IC, 10-100 gates
  - LSI: large scale IC, 100-1000 gates
  - VLSI: very large-scale IC, 1000+ gates
  - Today’s chip has millions of gates on it.

- A combinational circuit designed with individual gates can be implemented with SSI circuits that contain several independent gates.
- The number of gates in an SSI circuit is limited by the number of pins in it, (generally 14 – 16 pins).
- Medium Scale Integration components perform specific digital functions commonly needed in the design of digital systems.
- MSI components: adder, subtracter, comparator, decoder, encoder, multiplexer.

# PLD Components

- LSI technology introduced highly generalized circuit structures known as programmable logic devices (PLDs).
- Can consist of an array of and-gates and an array of or-gates. Must be modified for a specific application.
- Modification involves specifying the connections using a hardware procedure. Procedure is known as programming.
- Three types of programmable logic devices:
  - Programmable read-only memory (PROM)
  - Programmable logic array (PLA)
  - Programmable array logic (PAL)

# Full-Adder

You remember this combinational circuit named Full-Adder from chap 4. As this is a combinational circuit that adds three 1 bit binary digits, so there will be 3 input variables and 2 output variables.

C  10

    01

  + 01

    10

| $X_i$ | $Y_i$ | $C_i$ | $C_{i+1}$ | $S_i$ |
|---|---|---|---|---|
| 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 |

# Full-Adder

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |

Corresponding minimal sums:

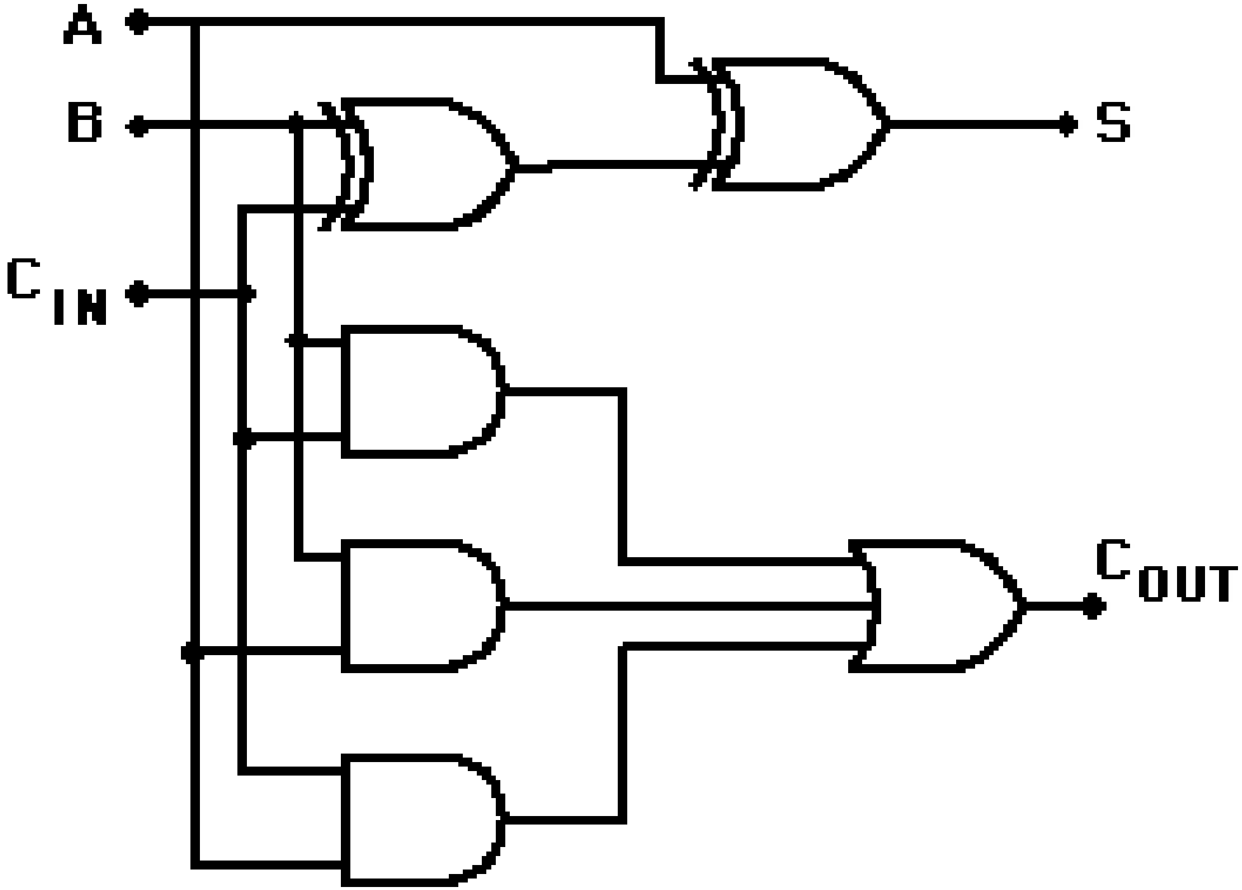
$$s_i = \overline{x}_i\overline{y}_i c_i + \overline{x}_i y_i \overline{c}_i + x_i \overline{y}_i \overline{c}_i + x_i y_i c_i$$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

We can simplify the sum for $s_i$ by using xor:

$$s_i = c_i \oplus x_i \oplus y_i$$

# Full-Adder



# Full-Adder

- So far, we have seen that, we can add 2 binary numbers, each consisting of 1 bit. For example,

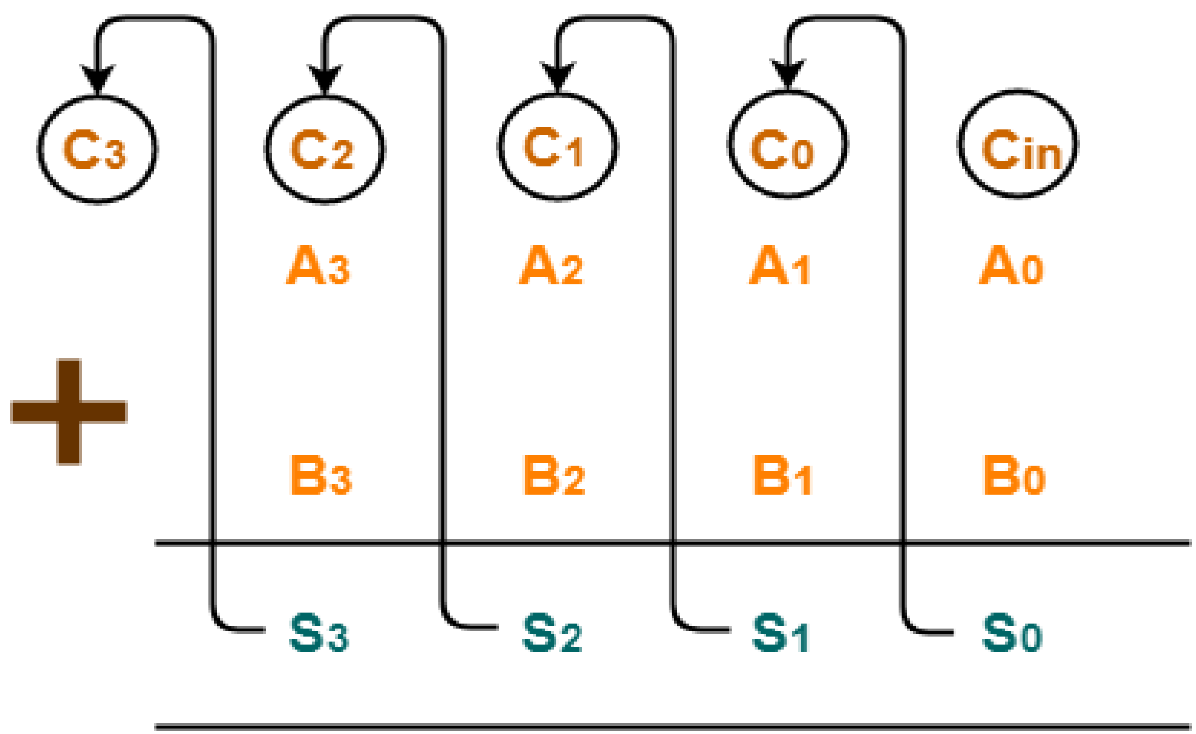
```
  0
 +1
 ---
  1
```

  - But consider this scenario, we want to add two binary numbers, each consisting of n bits.

```
  0110101
 +1011001
 --------
```

- One direct approach, write a truth table with 2^2n rows corresponding all the combinations of values and also specifying the values of the sum bits. But, that’s really time consuming.

# What About Many Bits?




Adding two 4-bit Numbers

# What About Many Bits?

| Subscript $i$ | 4 | 3 | 2 | 1 | | Full-adder of Fig. 4-5 |
|---|---|---|---|---|---|---|
| Input carry | 0 | 1 | 1 | 0 | $C_i$ | $z$ |
| Augend | 1 | 0 | 1 | 1 | $A_i$ | $x$ |
| Addend | 0 | 0 | 1 | 1 | $B_i$ | $y$ |
| Sum | 1 | 1 | 1 | 0 | $S_i$ | $S$ |
| Output carry | 0 | 0 | 1 | 1 | $C_{i+1}$ | $C$ |

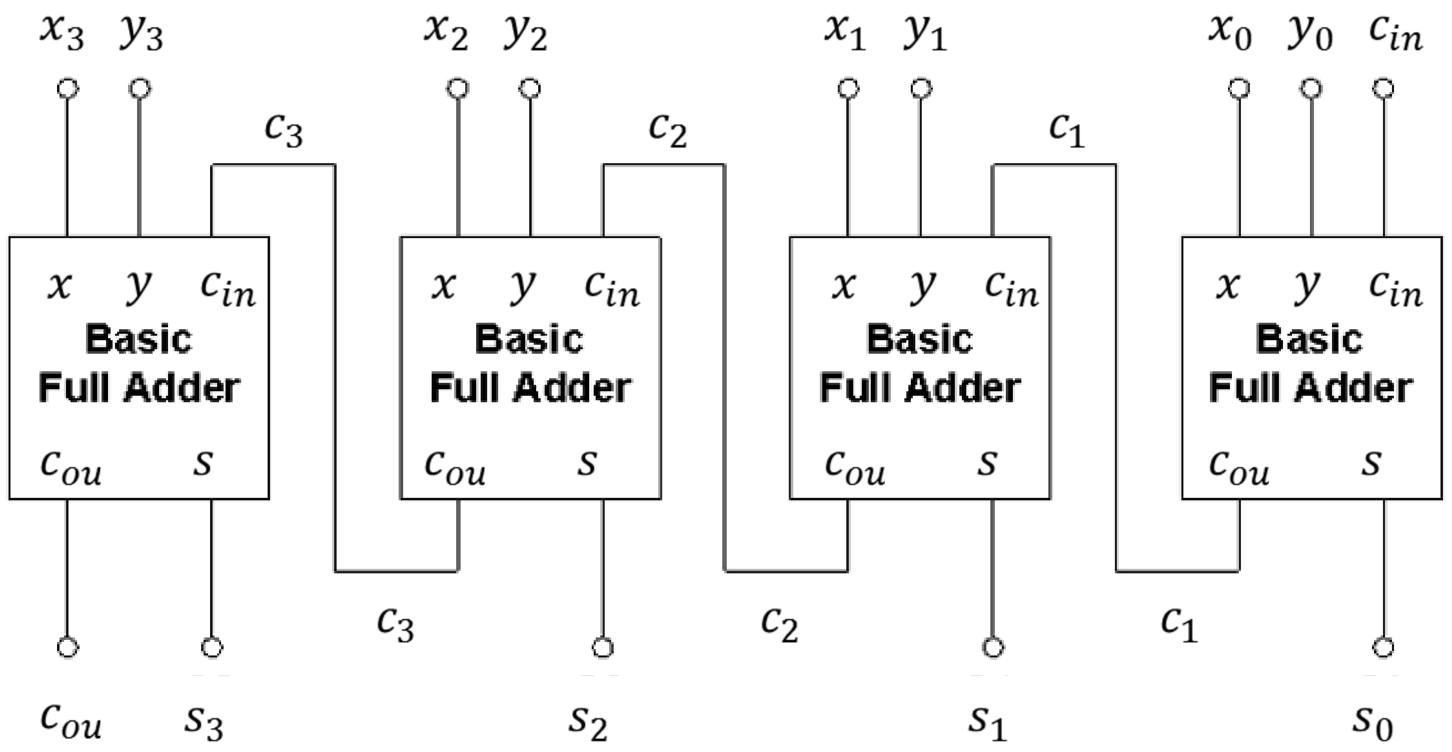
- Here, Let A=1011 and B=0011. The bits are added with full adders, starting from the least significant position. This will form a sum bit and a carry bit.
- The input carry C1 must be 0. The value of $C_{i+1}$ in a given significant position is the output carry of the full adder.

# What About Many Bits?

| Subscript $i$ | 4 | 3 | 2 | 1 | | Full-adder of Fig. 4-5 |
|---|---|---|---|---|---|---|
| Input carry | 0 | 1 | 1 | 0 | $C_i$ | $z$ |
| Augend | 1 | 0 | 1 | 1 | $A_i$ | $x$ |
| Addend | 0 | 0 | 1 | 1 | $B_i$ | $y$ |
| Sum | 1 | 1 | 1 | 0 | $S_i$ | $S$ |
| Output carry | 0 | 0 | 1 | 1 | $C_{i+1}$ | $C$ |

- There are two ways to implement this: 1. Serial addition 2. Parallel addition.
- Serial addition: It uses only one full adder and a storage device to hold the generated output carry.
- Parallel addition: It uses n full adder and all bits of A and B are applied simultaneously. The output carry of one full adder is connected to the input carry of next full adder.

# Parallel (ripple) Binary Adder




Why is it called “ripple” adder?


GOOD NEWS!
THE CLASS IS OVER…
THANK YOU!