# SNS COLLEGE OF ENGINEERING

Kurumbapalayam (PO), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

## DEPARTMENT OF ECE

## COURSE NAME: 19IT301 COMPUTER ORGANIZATION AND ARCHITECTURE

## II YEAR/ III SEM

## Unit 2 : ARITHMETIC OPERATIONS

## Topic 1: Addition and subtraction of signed numbers

# Binary, signed interger representation

| $b_3$ $b_2$ $b_1$ $b_0$ | Sign and Magnitude | 2’s Complement |
|---|---|---|
| 0 1 1 1 | +7 | +7 |
| 0 1 1 0 | +6 | +6 |
| 0 1 0 1 | +5 | +5 |
| 0 1 0 0 | +4 | +4 |
| 0 0 1 1 | +3 | +3 |
| 0 0 1 0 | +2 | +2 |
| 0 0 0 1 | +1 | +1 |
| 0 0 0 0 | +0 | +0 |
| 1 0 0 0 | -0 | -8 |
| 1 0 0 1 | -1 | -7 |
| 1 0 1 0 | -2 | -6 |
| 1 0 1 1 | -3 | -5 |
| 1 1 0 0 | -4 | -4 |
| 1 1 0 1 | -5 | -3 |
| 1 1 1 0 | -6 | -2 |
| 1 1 1 1 | -7 | -1 |

# Logic specification for a stage of binary addition

At the $i^{th}$ stage:

Input:

$c_i$ is the carry-in

Output:

$s_i$ is the sum

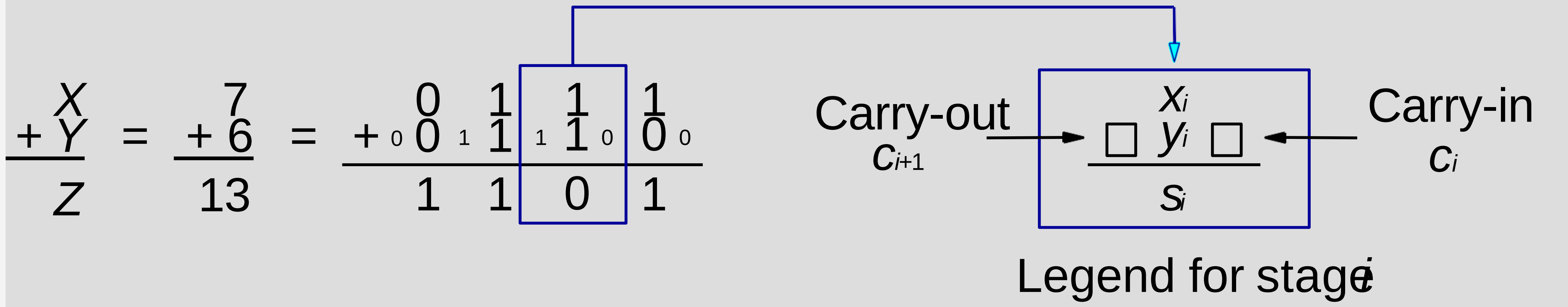
$c_{i+1}$ carry-out to $(i+1)^{st}$ state

| $x_i$ | $y_i$ | Carry-in $c_i$ | Sum $s_i$ | Carry-out $c_{i+1}$ |
|---|---|---|---|---|
| 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 |

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

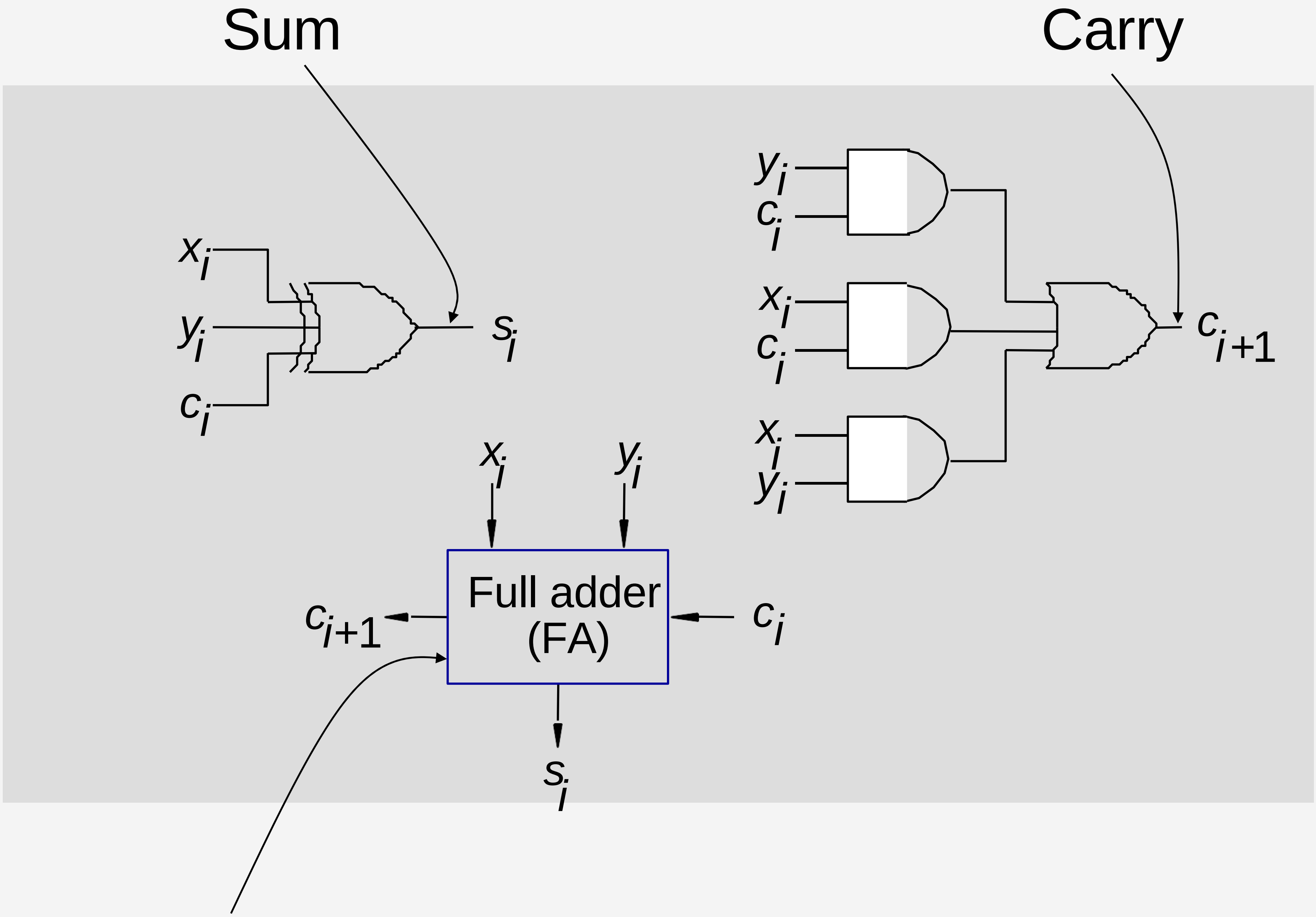
$$c_{i+1} = y_i c_i + x_i c_i + x_i y_i$$

Example:



Legend for stage $i$

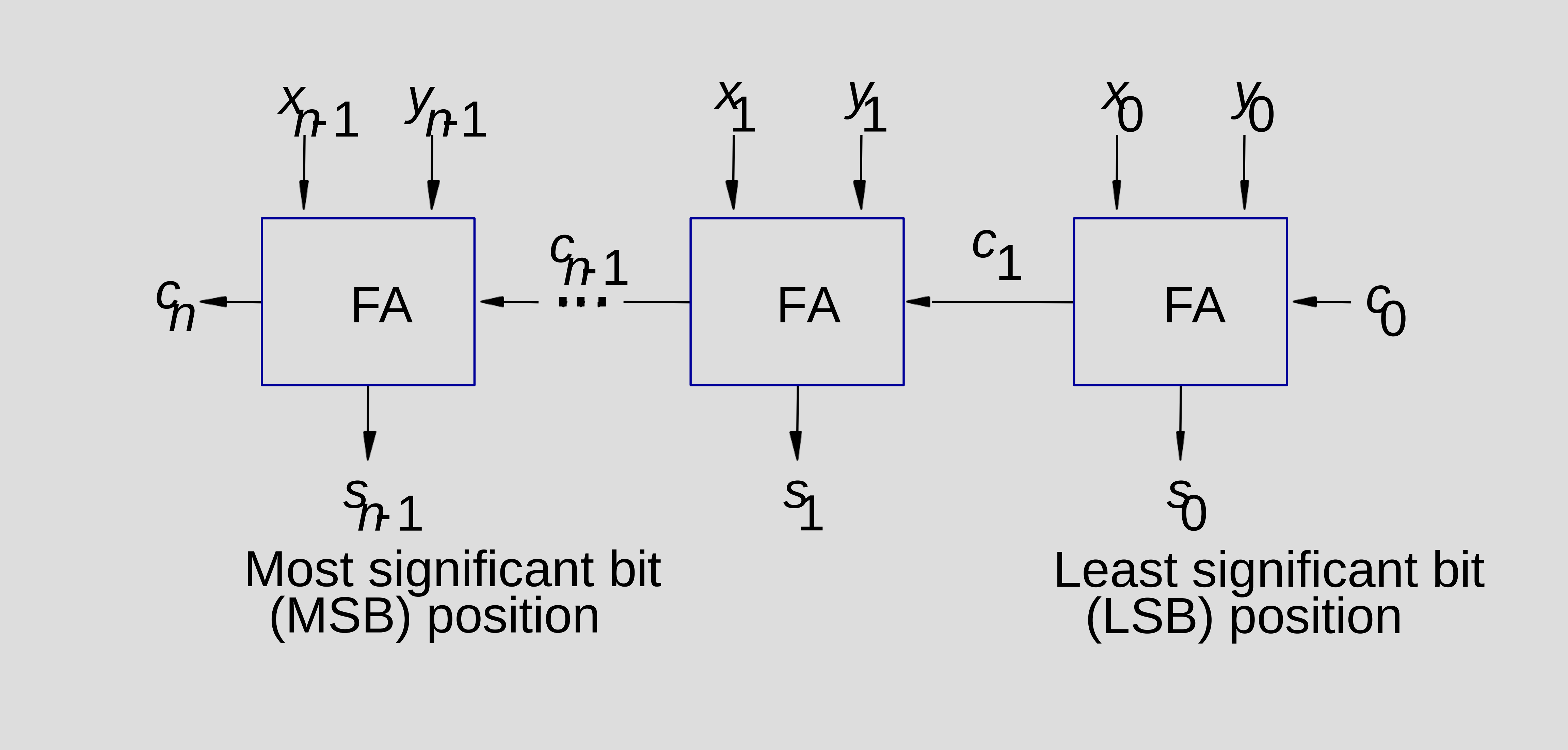
# Addition logic for a single stage



Full Adder (FA): Symbol for the complete circuit for a single stage of addition

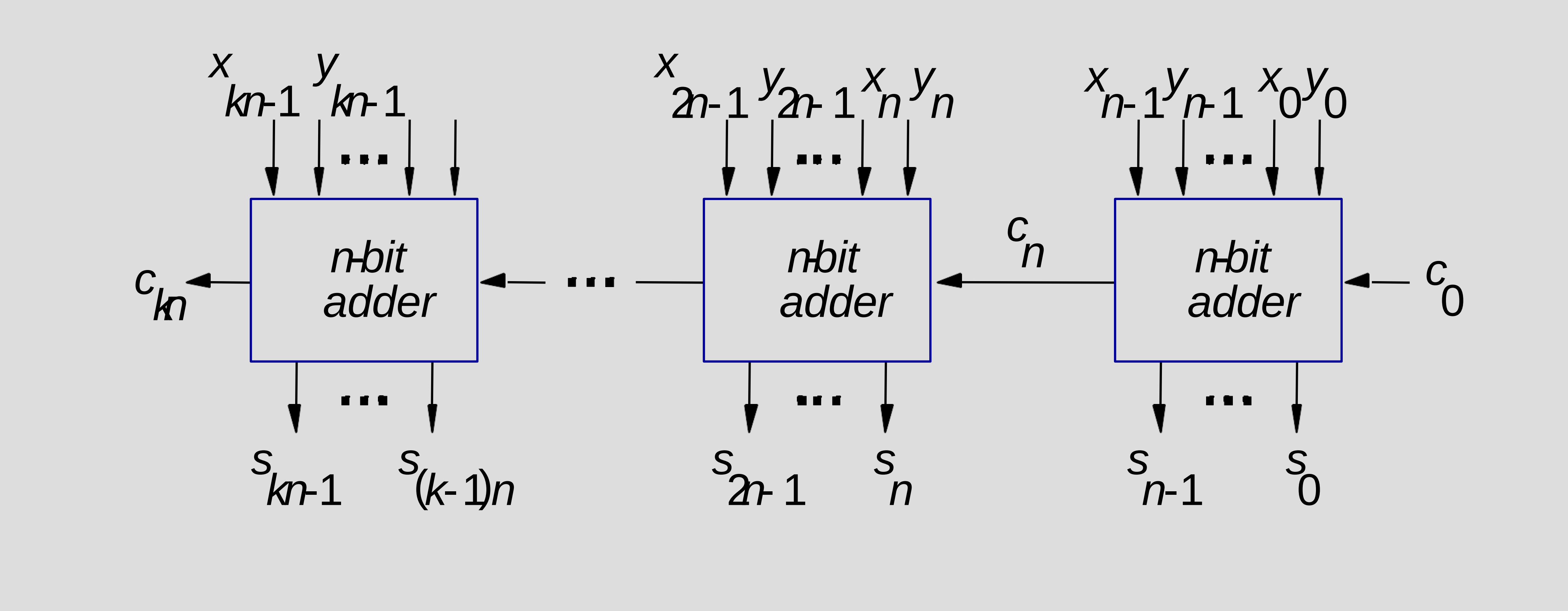
# n-bit ripple carry adder

- Cascade n-full adder (FA) blocks to form a n-bit adder.
- Carries propagate or ripple through this cascade, n-bit ripple carry adder.



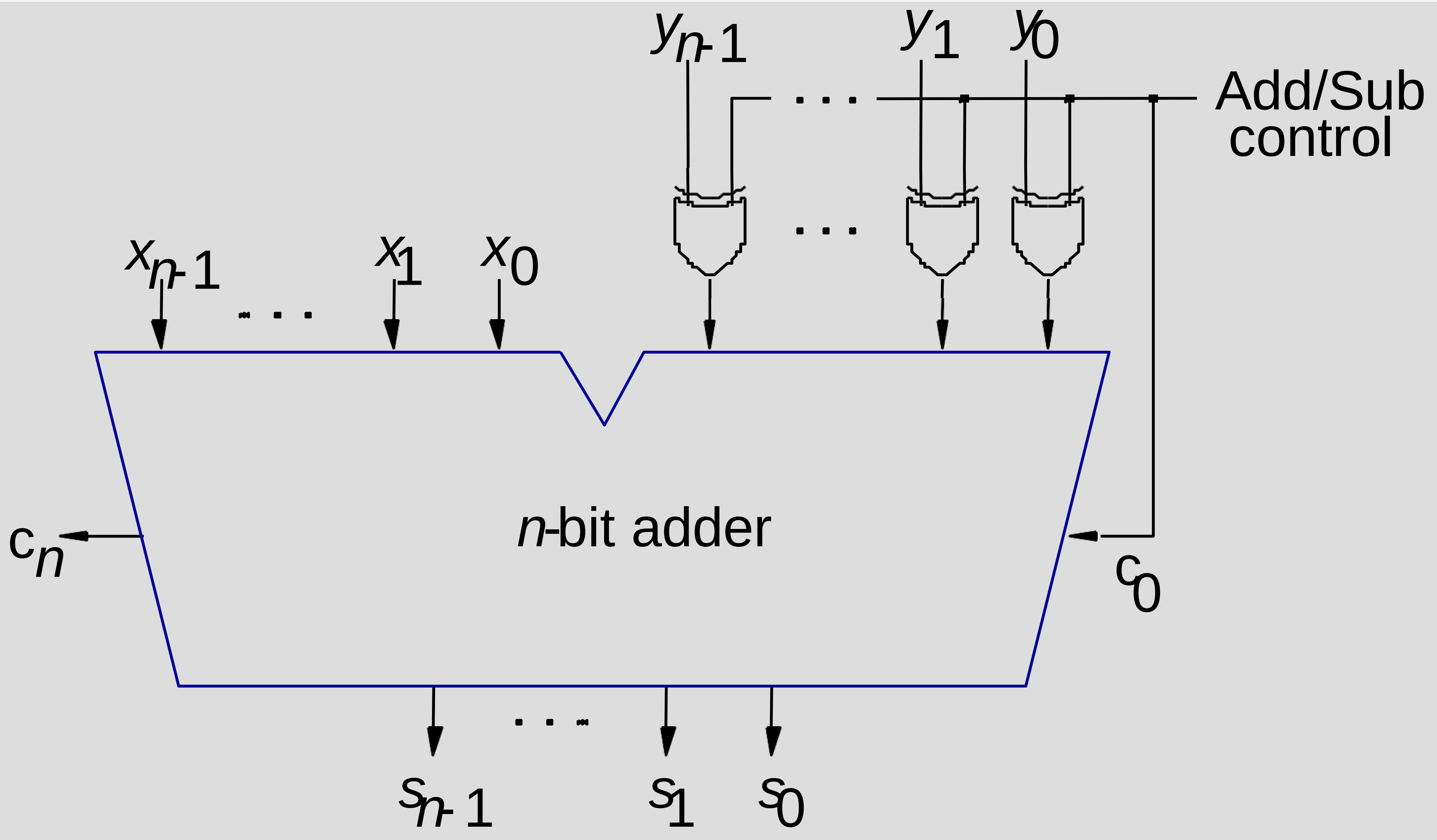
# K n-bit adder

- K n-bit numbers can be added by cascading k n-bit adders



- Carry-in $c_0$ into the LSB position provides a convenient way to perform subtraction
- Each n-bit adder forms a block, so this is cascading of blocks.
- Carries ripple or propagate through blocks, Blocked Ripple Carry Adder

# Binary addition- subtraction logic network



- Add/sub control = 0, addition
- Add/sub control = 1, subtraction

- X – Y is equivalent to adding 2’s complement of Y to X
- 2’s complement is equivalent to 1’s complement + 1
- X – Y = X + Y + 1

# Detecting overflows

- Overflows can only occur when the sign of the two operands is the same.
- Overflow occurs if the sign of the result is different from the sign of the operands.
- Circuit to detect overflow can be implemented by the following logic expressions:

$$Overflow = x_{n-1}y_{n-1}\bar{s}_{n-1} + \bar{x}_{n-1}\bar{y}_{n-1}s_{n-1}$$

$$Overflow = c_n \oplus c_{n-1}$$

# Thank You