Arithmetic Circuits

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Computer Science & Engineering, and

Department of Electrical Engineering

Indian Institute of Technology Bombay

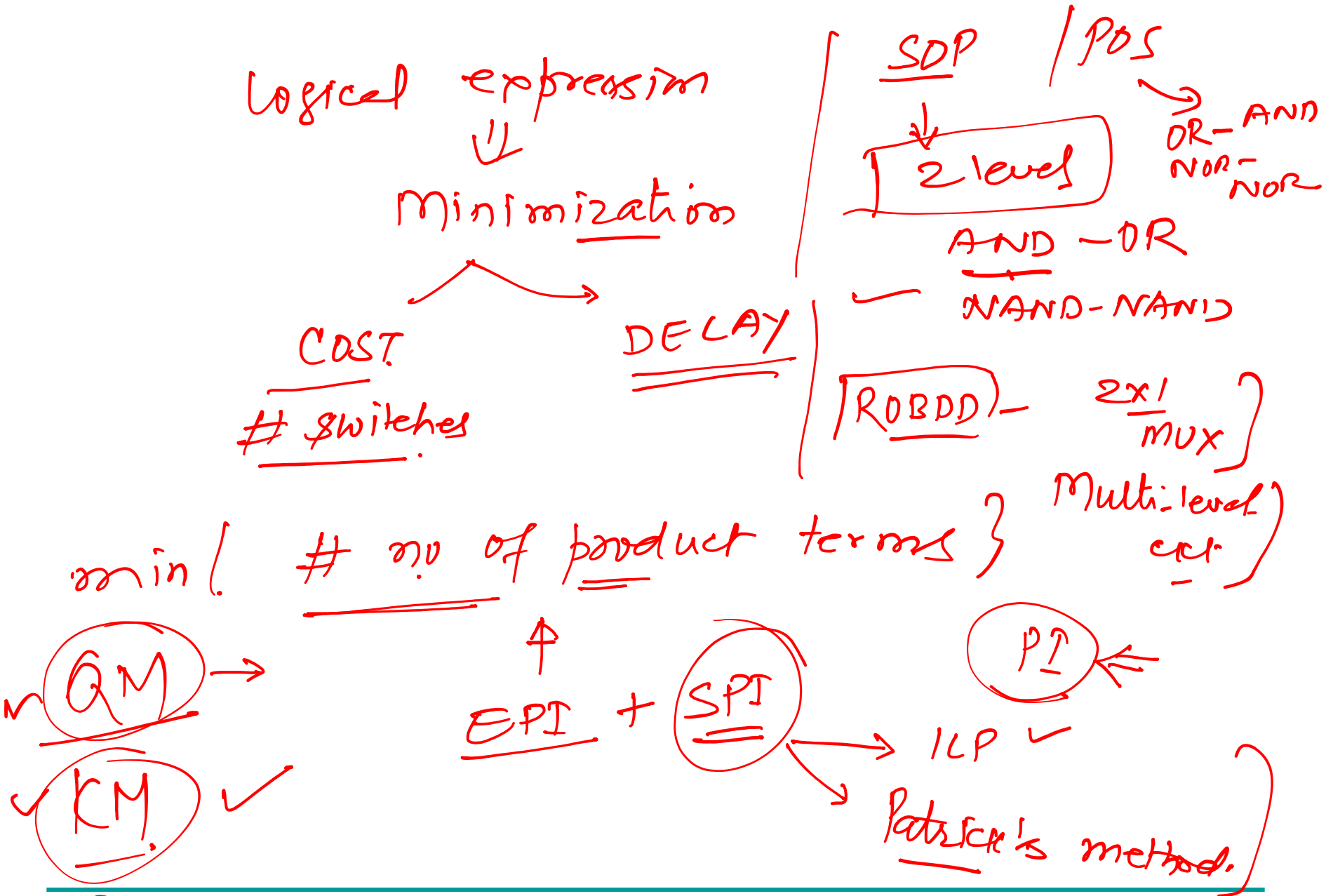
http://www.cse.iitb.ac.in/~viren/

E-mail: viren@{cse, ee}.iitb.ac.in

CS-230: Digital Logic Design & Computer Architecture

Lecture 12 (01 February 2022)

CADSL


Logical expression
Minimization
COST
switches
DELAY
SOP / POS
2 level
OR-AND
NOR-NOR
AND-OR
NAND-NAND
ROBDD - 2x1 MUX
Multi-level ckt
min (# no of product terms)
QM
KM
EPI + SPI
PI
ILP
Patrick's method.

Multiple-Level Optimization

- Multiple-level circuits are circuits that have more than two level (plus input and/or output inverters)
- For a given function, multiple-level circuits can have reduced gate input cost compared to two-level (SOP and POS) circuits
- Multiple-level optimization is performed by applying transformations to circuits represented by equations while evaluating cost

Transformations

- Factoring - finding a factored form from SOP or POS expression
  - Algebraic - No use of axioms specific to Boolean algebra such as complements or idempotence
  - Boolean - Uses axioms unique to Boolean algebra
- Decomposition - expression of a function as a set of new functions

Transformations (continued)

- Substitution of G into F - expression function F as a function of G and some or all of its original variables
- Extraction - decomposition applied to multiple functions simultaneously

Transformation Examples

- **Algebraic Factoring**

$F = \bar{A}\bar{C}\bar{D} + \bar{A}B\bar{C} + ABC + AC\bar{D}$ $\quad G = 16$

– Factoring:

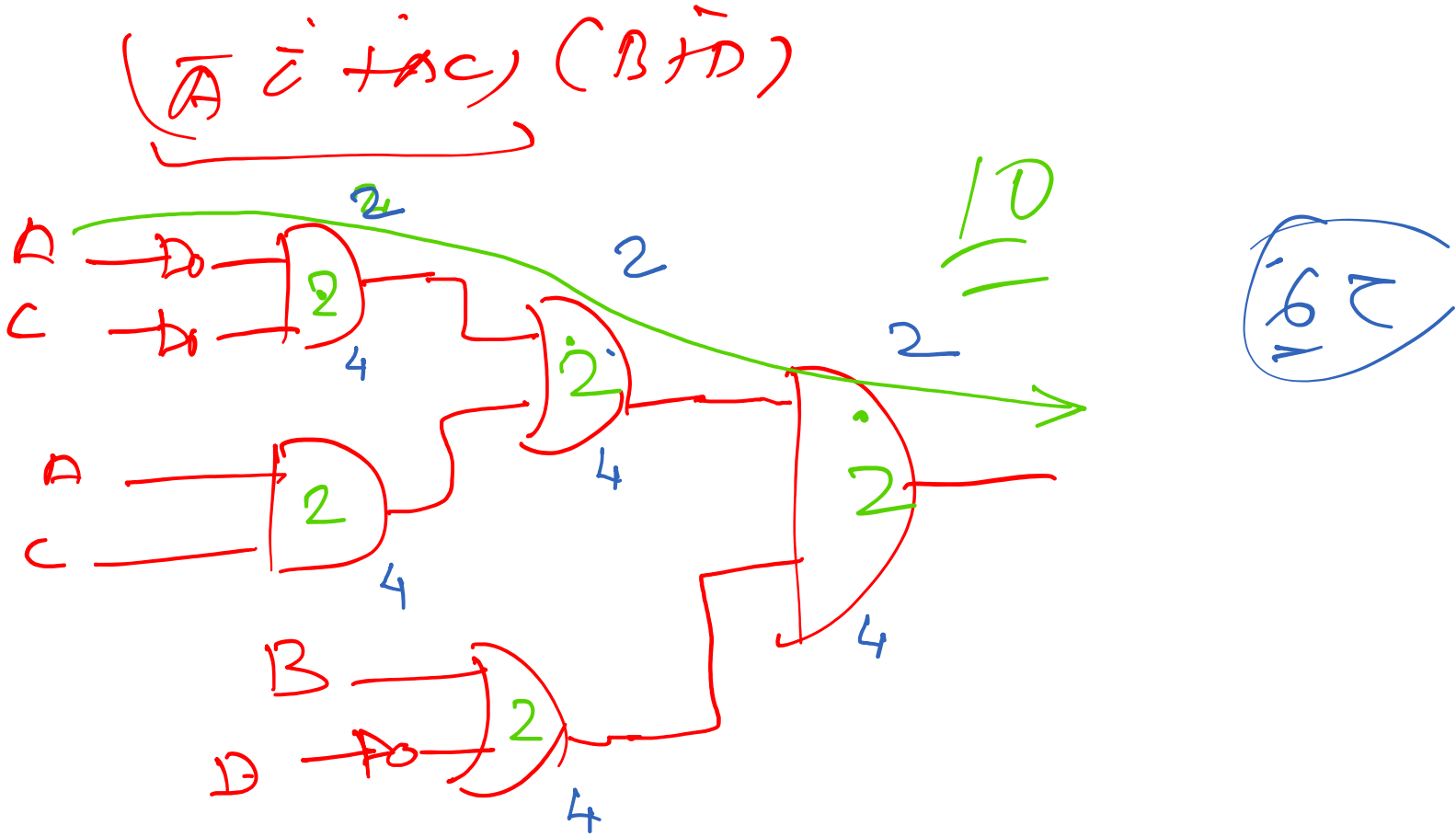
$F = \bar{A}(\bar{C}\bar{D} + B\bar{C}) + A(BC + C\bar{D})$ $\quad G = 16$

– Factoring again:

$F = \bar{A}\bar{C}(B + \bar{D}) + AC(B + \bar{D})$ $\quad G = 12$

– Factoring again:

$F = (\bar{A}\bar{C} + AC)(B + \bar{D})$ $\quad G = 10$


A
C
A
C
B
D
2
4
10
6

Transformation Examples

- Decomposition
  - F = A'C'D' + A'BC' + ABC + ACD' G = 16
  - The terms A'C' + AC and B + D' can be defined as new functions H and E respectively, decomposing F = (A'C' + AC)(B + D'):

  F = H E, H = A'C' + AC, E = B + D' G = 10

- This series of transformations has reduced G from 16 to 10, a substantial savings.
- The resulting circuit has three levels plus input inverters.

Transformation Examples

- **Substitution of E into F**
  - **Returning to F just before the final factoring step:**

$$F = \bar{A}\,\bar{C}\,(B + \bar{D}) + AC\,(B + \bar{D}) \qquad G = 12$$

  - **Defining $E = B + \bar{D}$, and substituting in F:**

$$F = \bar{A}\,\bar{C}\,E + AC\,E \qquad G = 10$$

  - **This substitution has resulted in the same cost as the decomposition**

Transformation Examples

Summary

- Multi-level Optimization
- Transformations
  - Factoring - find a factored form from SOP or POS expression
  - Decomposition - express a function as a set of new functions
  - Substitution - express function F as a function of G and some or all of its original variables

Arithmetic Circuits: Adders

2 input Adder ⇓ HA

3 input Adder ⇓ FA

TJHKPF

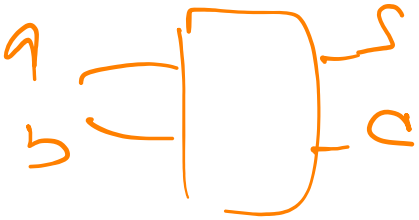
Half-Adder Adds two Bits

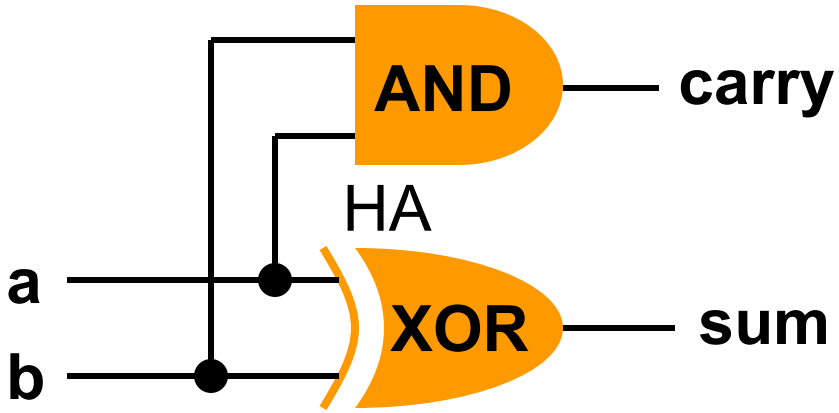
"half" because it has no carry input

- Adding two bits:

a	b	a + b
0	0	00
0	1	01
1	0	01
1	1	10

carry sum





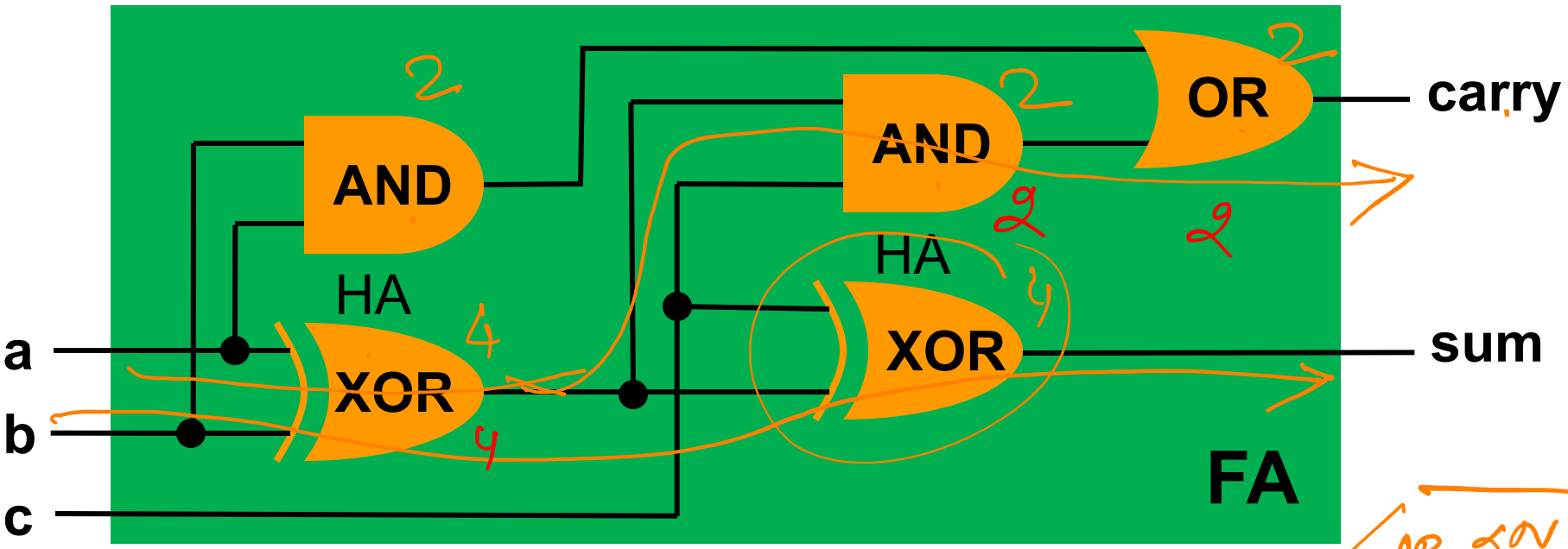
Full Adder: Include Carry Input

a	b	c	s = a + b + c	
			Decimal value	Binary value
0	0	0	0	00
0	0	1	1	01
0	1	0	1	01
0	1	1	2	10
1	0	0	1	01
1	0	1	2	10
1	1	0	2	10
1	1	1	3	11

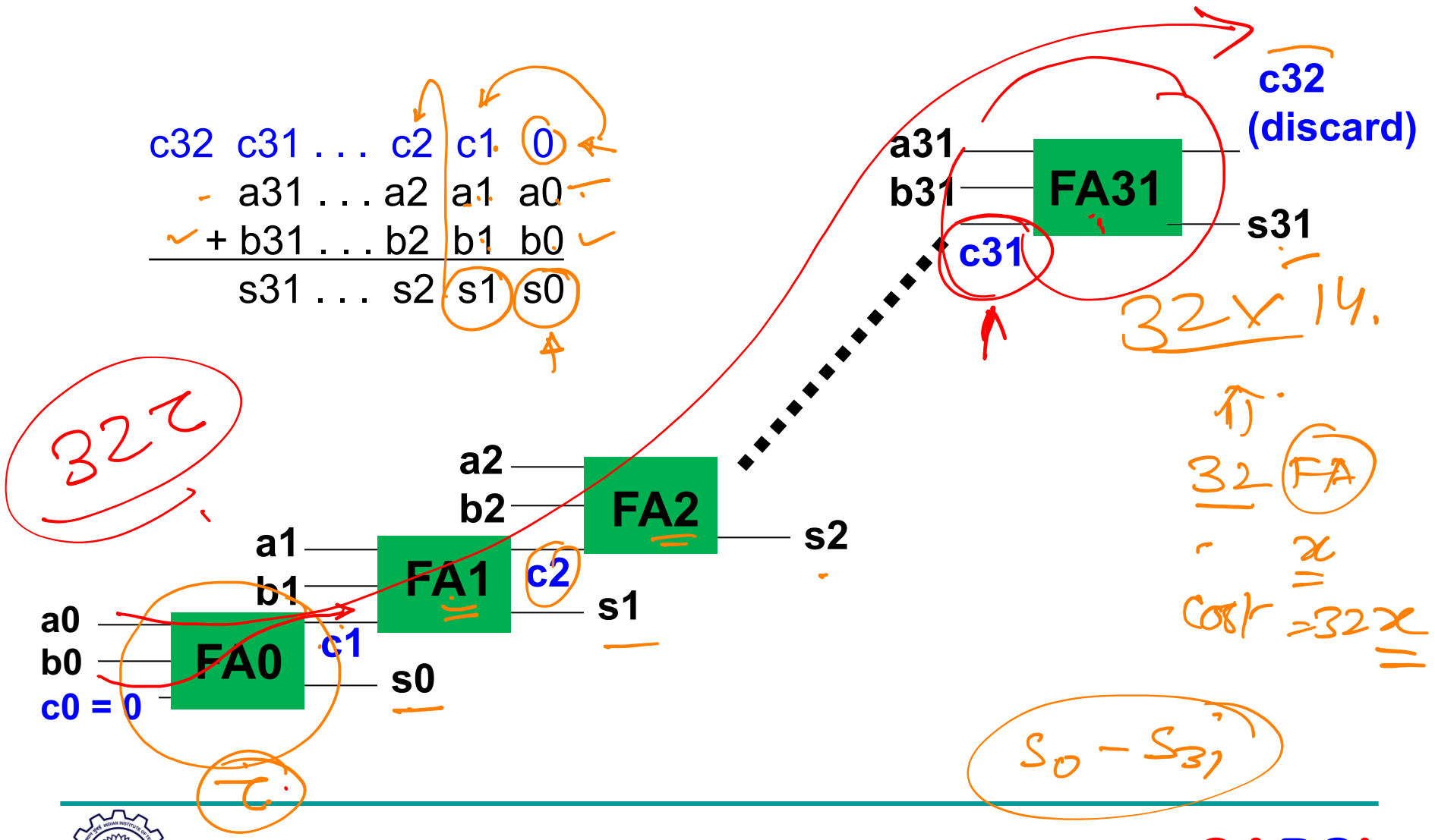
CARRY

SUM

Full-Adder Adds Three Bits

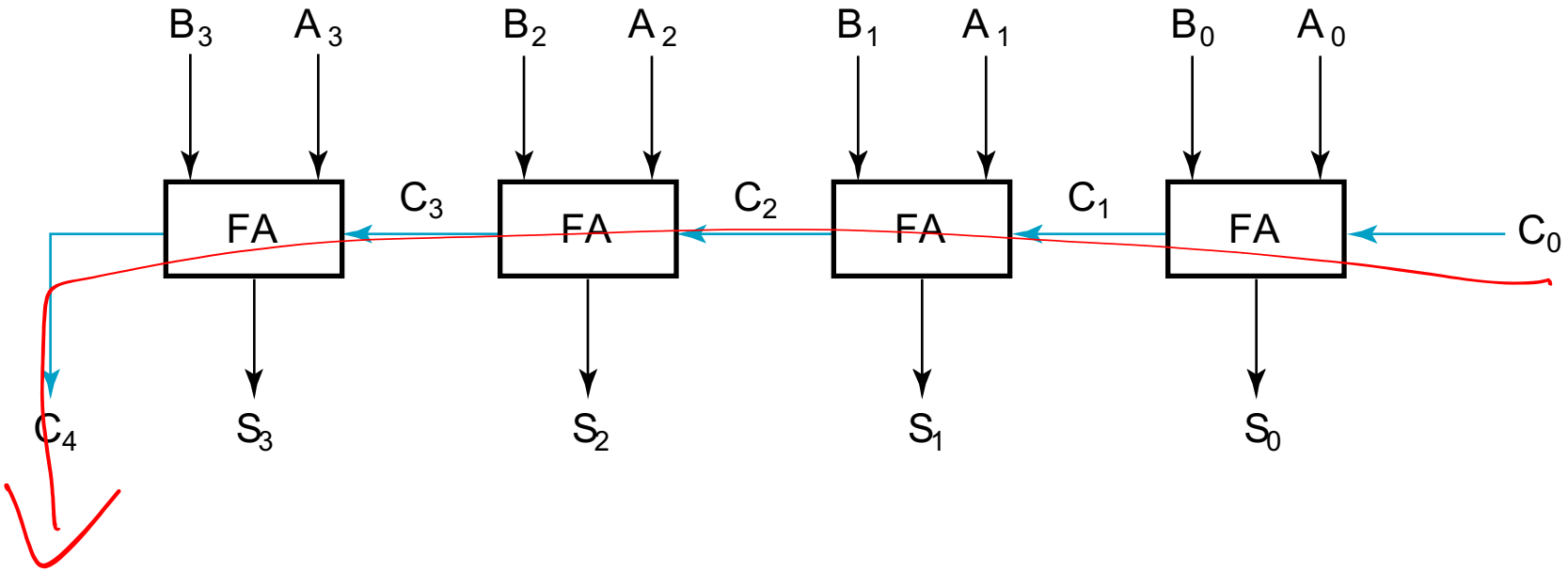


32-bit Ripple-Carry Adder



4-bit Ripple-Carry Binary Adder

- A four-bit Ripple Carry Adder made from four 1-bit Full Adders:

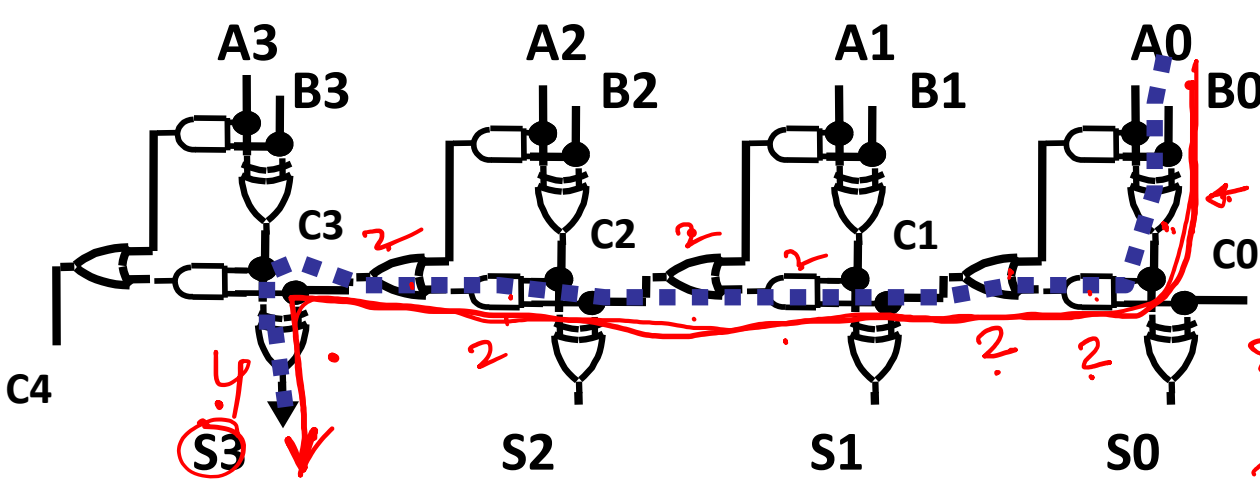


How Fast is Ripple-Carry Adder?

- Longest delay path (critical path) runs from (a0, b0) to sum31/C32
- Suppose delay of full-adder is 100ps
- Critical path delay = 3,200ps
- Must use more efficient ways to handle carry

Carry Propagation & Delay

- Propagation delay:
  - Carry must ripple from LSB to MSB.
- The gate-level propagation path for a 4-bit ripple carry adder of the last example:



- Too slow for many bits (e.g. 32 or 64)

Carry Propagation & Delay

32 bit

$= 4+4+6\times(3)$ n-1

$= 8 + 124 = 132$ ✓

10 ps

$= 1320$ ps $= 1.32$ ns

$= \frac{1}{1.32 \times 10^{9}} =$

Carry Propagation & Delay

Ripple Carry Adder

- Easy to create
  - Good hierarchy
  - Tileable structures
- Small hardware
- Slow!
  - Design is limited by the delays in propagating carry through all of the bitwise additions
  - The output bit at position m is not valid until after the carry out of the m-1 position is ready

32 FA

10 FA

$= 8 + 4 \times (n-1) \propto n$

Thank You