EE671: VLSI Design
Spring 2024/25

Laxmeesha Somappa

Department of Electrical Engineering

IIT Bombay

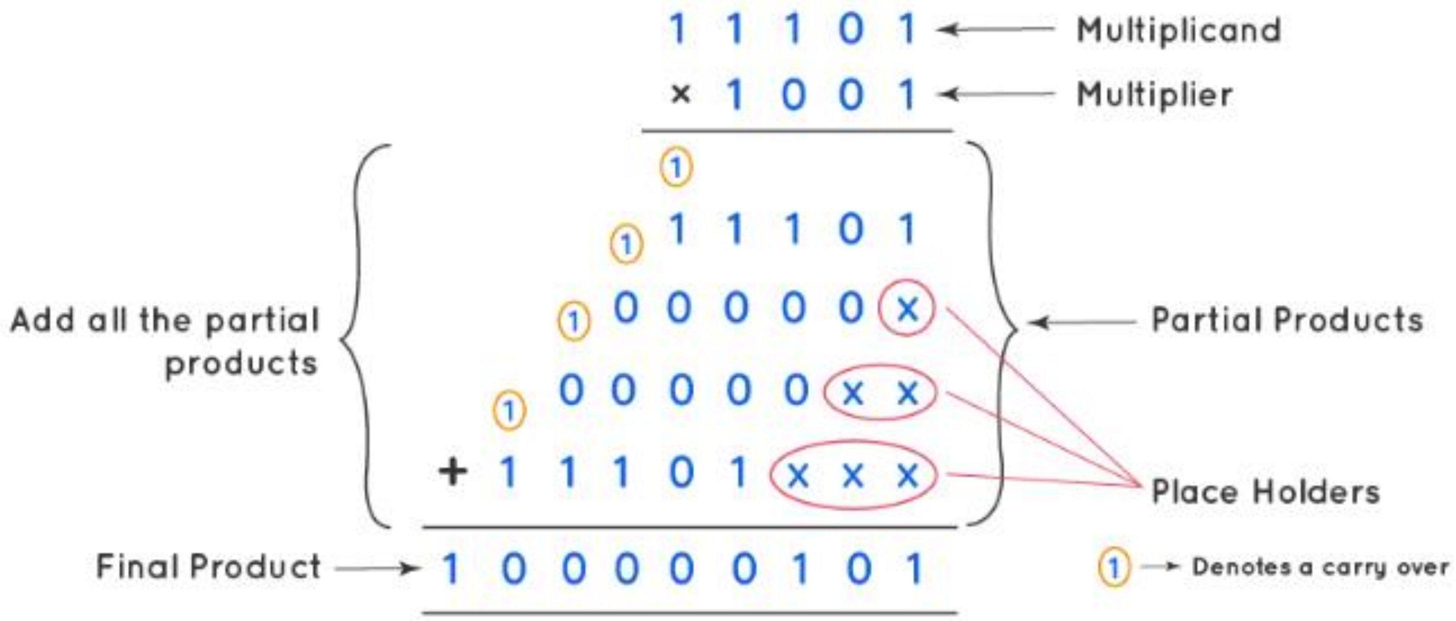
laxmeesha@ee.iitb.ac.in

LECTURE – 17
ARITHMETIC IP: MULTIPLIERS

Acknowledgement: Prof. D.K.S

laxmeesha@ee.iitb.ac.in

BINARY MULTIPLIER: BASICS



- ❑ Rules: a) bit-wise multiplication (AND gate)

  b) Repeat bit-wise multiplication → generates partial product

  c) Shift & add all partial products → final product

Binary Multiplier: Basics

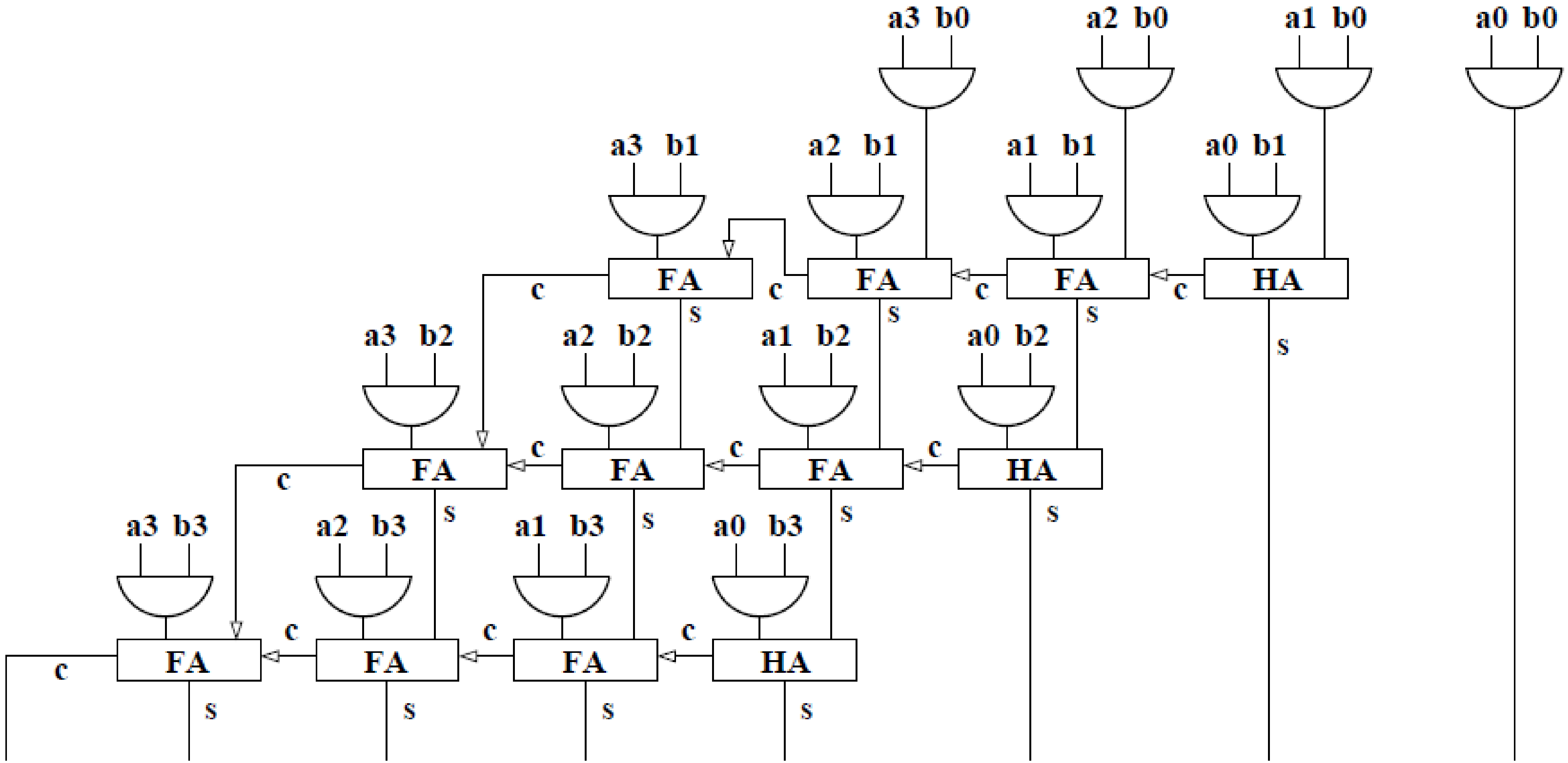
- ❑ Say we want to perform A X B (A and B are binary numbers)

$$A = \sum_{i=0}^{n-1} 2^i a_i \qquad B = \sum_{j=0}^{n-1} 2^j b_j$$

- ❑ We can generate any bit of a partial product using AND of (a_i,b_j)
- ❑ All bits of partial products can be computed using AND gates
- ❑ Use adders to add the partial products
- ❑ This architecture is called an Array Multiplier

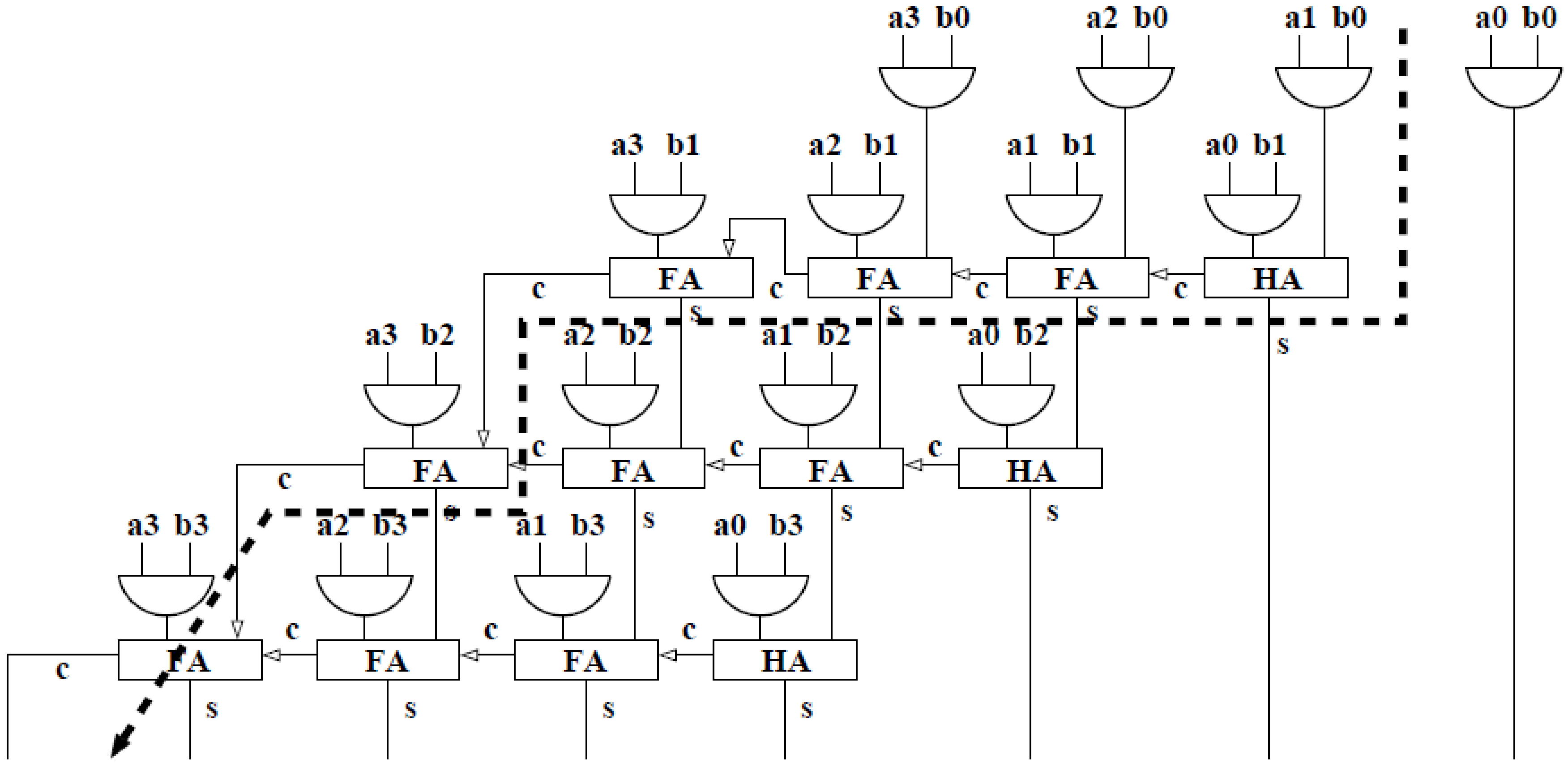
Array Multiplier

- A 4-bit array multiplier structure



Array Multiplier

- A 4-bit array multiplier: critical path → adders in the path !



- Can we speed-up the multiplier further?
  - Can we reduce number of partial products using a 2-bit multiplication instead ?

Booth Encoding

- Consider A X B, and we want to multiply two bits (of multiplier B) at a time
- If B = 00 → partial product will be 0
- If B = 01 → partial product will be A
- If B = 10 → partial product will be 2A
    - Simple left shift operation
- If B = 11 → partial product will be 3A

- How to perform 3A with a simple operation?
    - 3A = 4A – A
    - What if 4A operation is shifted to next pair of bits?
    - The next pair already has higher binary weight (higher by factor 4)
        - 4A is as simple as adding 1 to the next pair of bits in B
    - -A can be generated from a subtractor (2's complement adder)

Modified Booth Encoding

- ❑ To simplify the logic of “when to” generate 4A
  - ❑ Generating 2A is done similar to 3A
  - ❑ 3A = 4A - A (subtract A and tell next group to add 4)
  - ❑ 2A = 4A -2A (subtract 2A and tell next group to add 4)

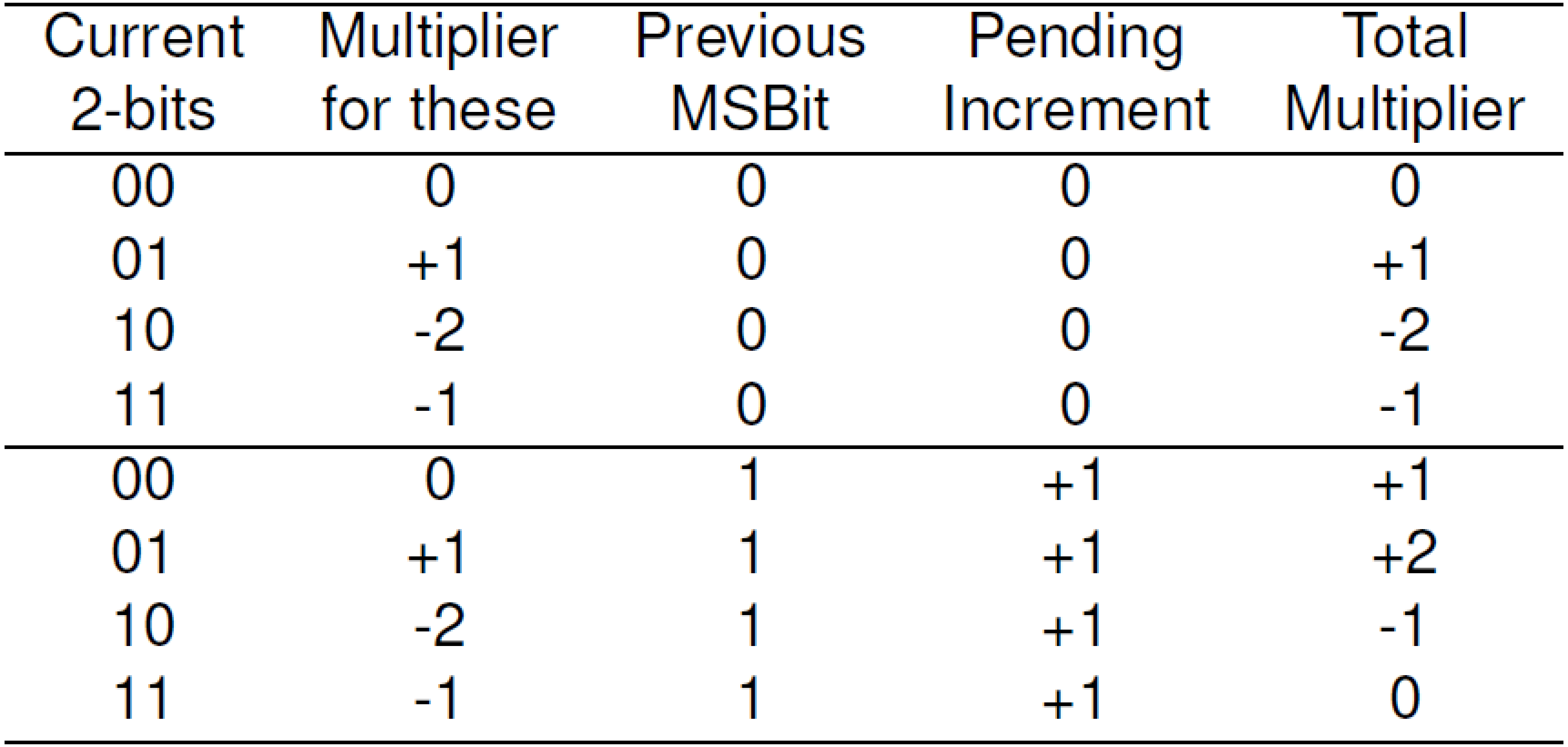
- ❑ In summary (look at group of 2 bits in B):
  - ❑ 00 → No operation
  - ❑ 01 -> add A
  - ❑ 10 → subtract 2A and ask next group to add 4A
  - ❑ 11 → subtract A and ask next group to add 4A
  - ❑ Next group will see the MSB of previous group → if “1”, add 1 to the multiplier bit group (current two bits in B)

Modified Booth Encoding

- Modified Algorithm:
  - Look at 3 bits (with one bit overlap) in B (multiplier)
  - Current 2 bits and MSB of the previous group of 2 bits (in B)
  - For the first group of 2 bits, assume a 0 to the right
  - Move from right to left
  - After handling the previous group, A is shifted left by two positions → inherent multiplication by 4
    - This means adding 4A on behalf of previous group is same as adding +1 to the multiplier (bits of B) to the current group

Modified Booth Encoding

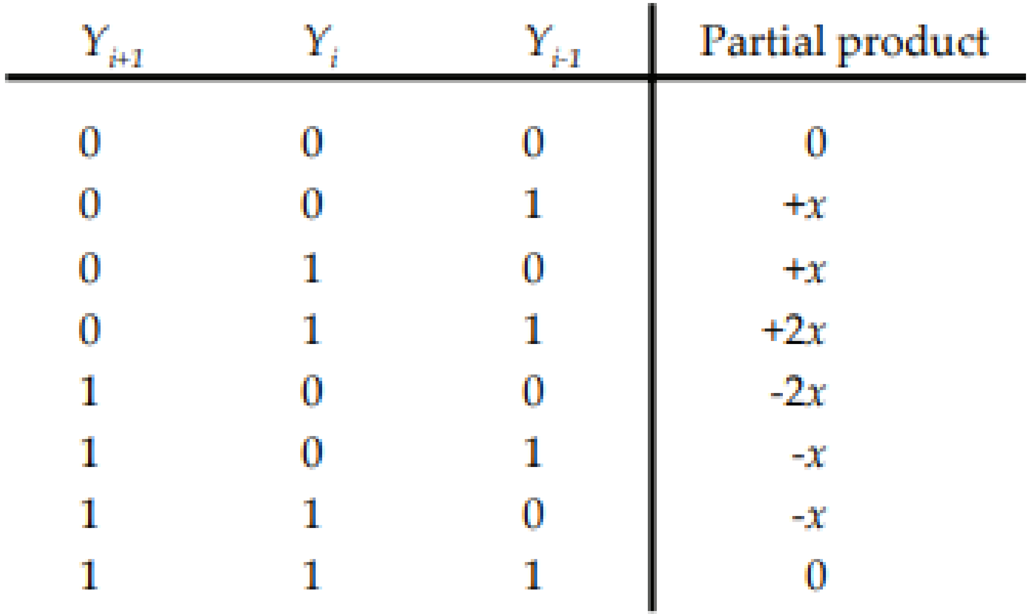
- Algorithm: to generate partial products



Current 2-bits	Multiplier for these	Previous MSBit	Pending Increment	Total Multiplier
00	0	0	0	0
01	+1	0	0	+1
10	-2	0	0	-2
11	-1	0	0	-1
00	0	1	+1	+1
01	+1	1	+1	+2
10	-2	1	+1	-1
11	-1	1	+1	0

Modified Booth Encoding

- ❑ Algorithm: say multiplying Y and X
  - ❑ Generate partial products with group of 2 bits



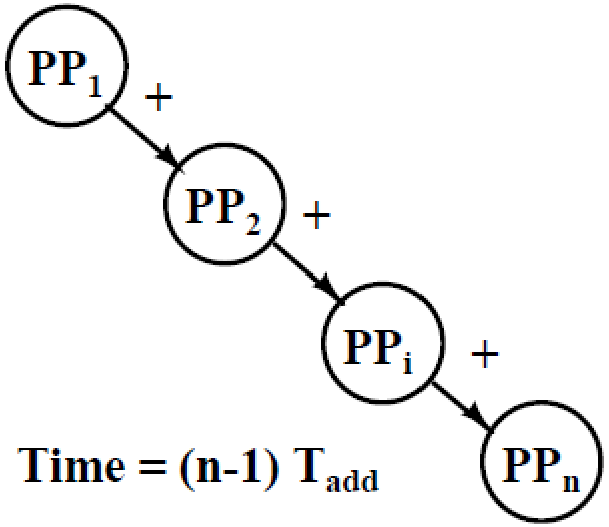
Y_{i+1}	Y_i	Y_{i-1}	Partial product
0	0	0	0
0	0	1	$+x$
0	1	0	$+x$
0	1	1	$+2x$
1	0	0	$-2x$
1	0	1	$-x$
1	1	0	$-x$
1	1	1	0

- ❑ Add partial products using any of the adders you have learnt before to get the final product – how many adders?

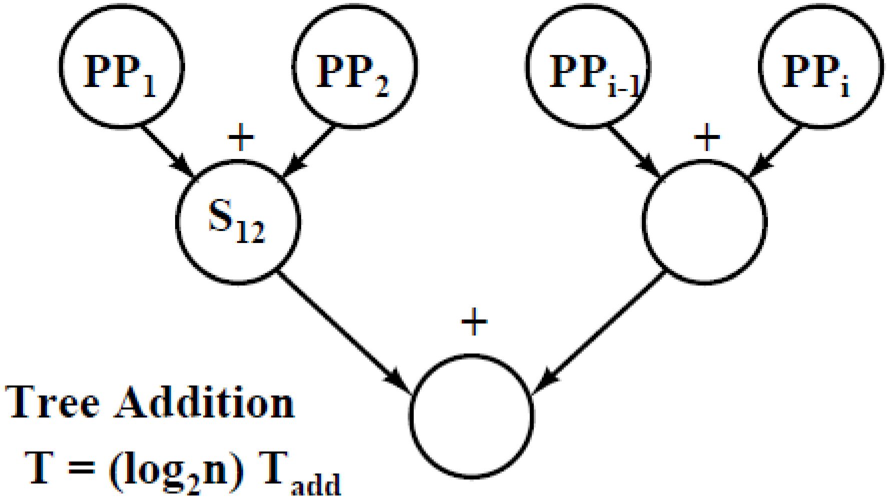
Modified Booth Encoding

- Adding partial products:
  - Adders in critical path (array multiplier)
  - Speed up the partial production addition?

  - We have already looked at tree adders for faster additions! – Similar idea



Time = (n-1) T_{add}

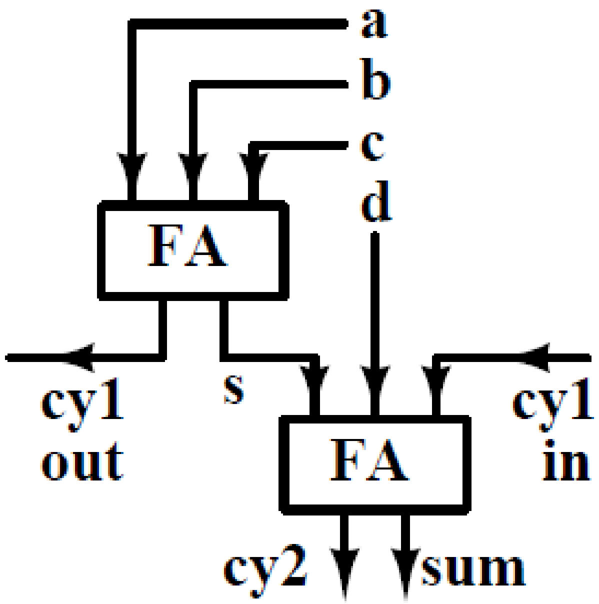


Tree Addition

T = ($\log_2 n$) T_{add}

MODIFIED BOOTH ENCODING

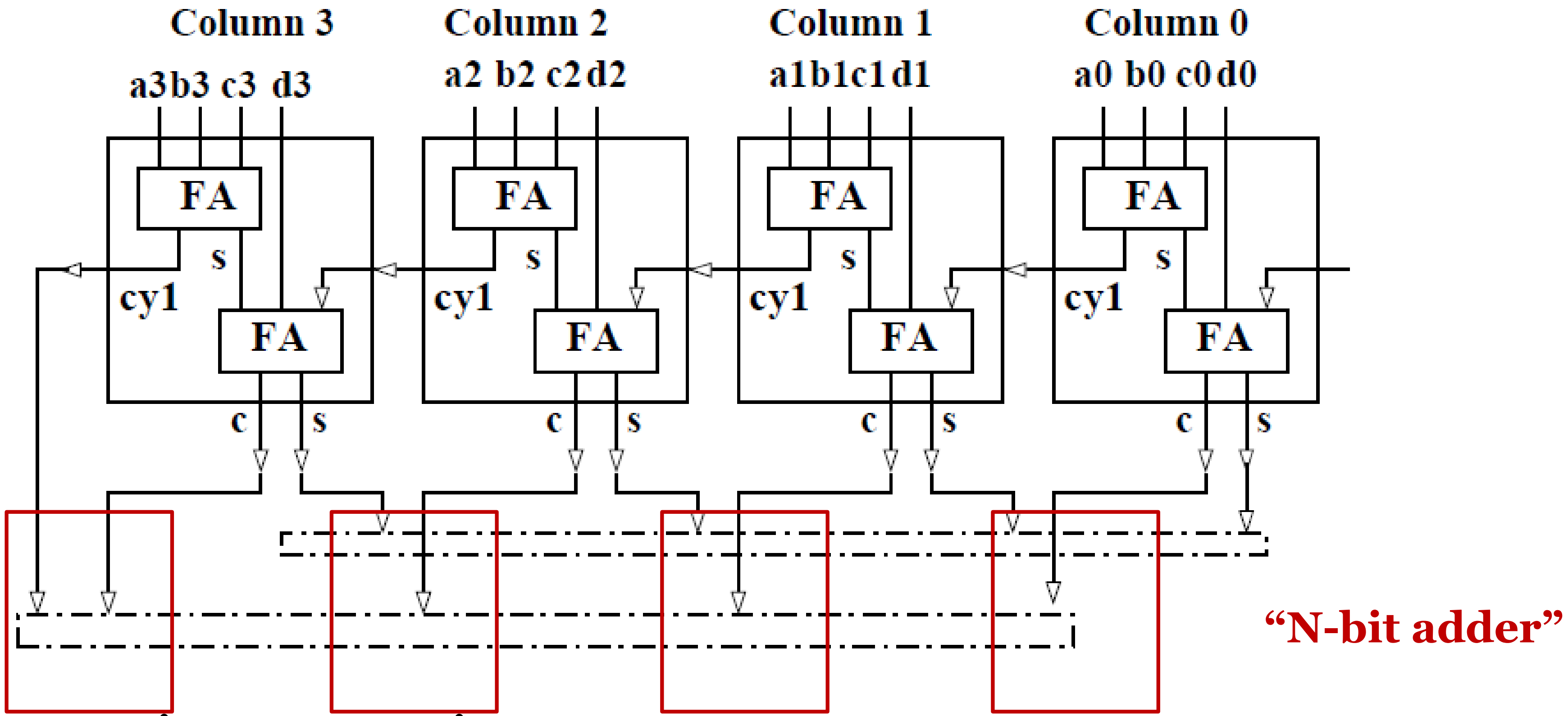
- ❑ We looked at 1-bit adders to implement ‘N’ bit adders
- ❑ Can we also have a 2-bit adder to add partial products?



- ❑ cy1 is the carry from the previous 2-bit group partial product addition
- ❑ No rippling of carry from right to left!!!

MODIFIED BOOTH ENCODING

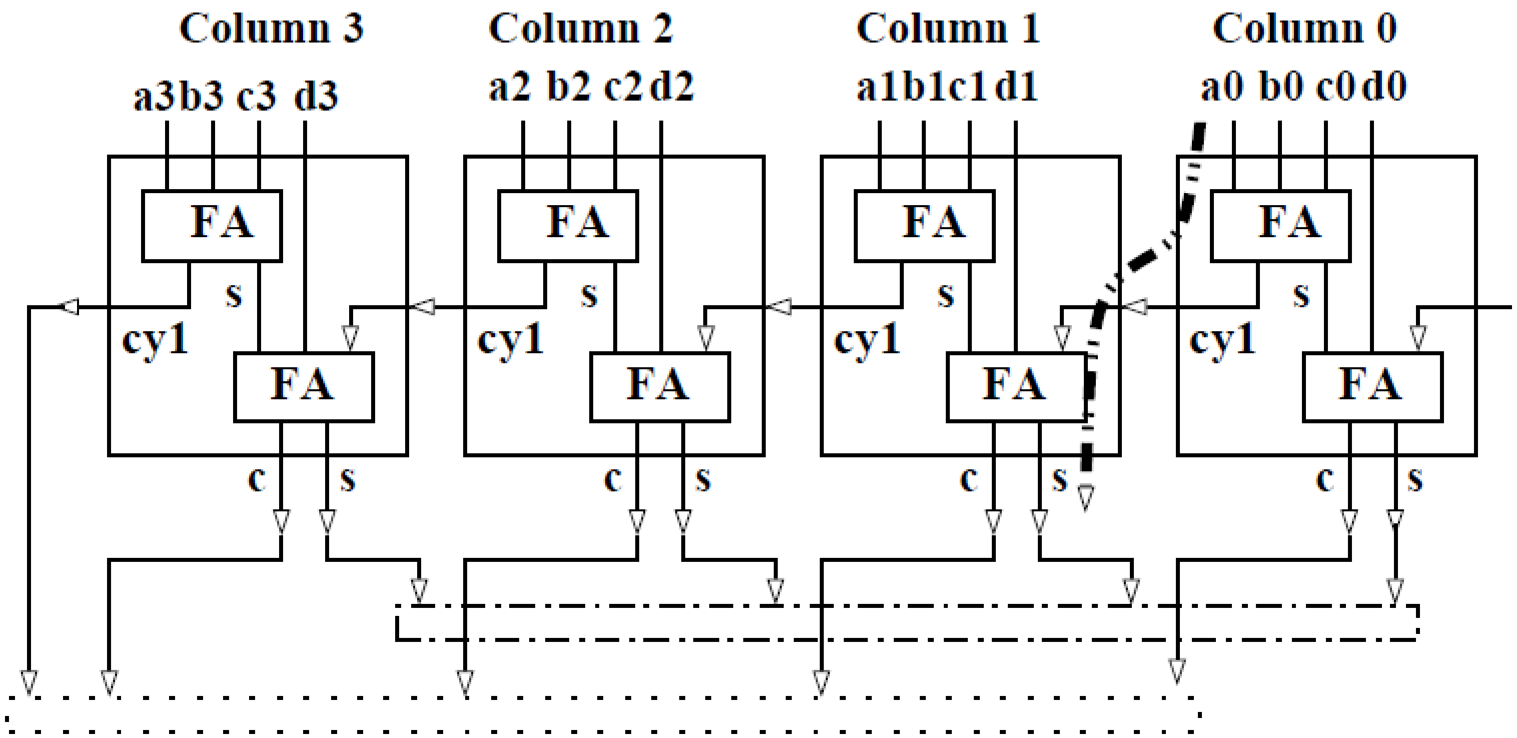
- 2-bit adders to implement 'N' bit adders: for PP addition



- Save output in two registers
- Add these two registers using any adder that we have seen

Modified Booth Encoding

- PP addition: critical path (no carry ripple except for the last 2-register addition)



Wallace Multipliers

- Consider 4x4 multiplication

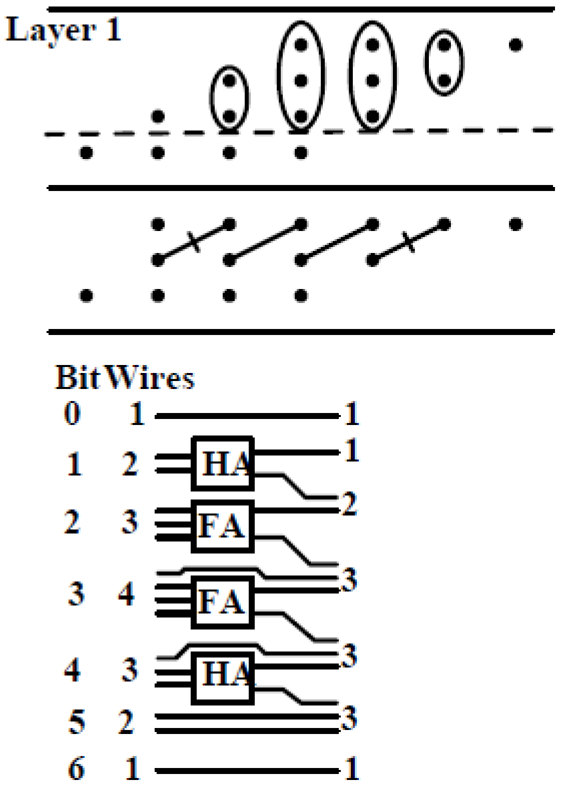
			a3	a2	a1	a0
		x	b3	b2	b1	b0
			a3.b0	a2.b0	a1.b0	a0.b0
		a3.b1	a2.b1	a1.b1	a0.b1	
	a3.b2	a2.b2	a1.b2	a0.b2		
a3.b3	a2.b3	a1.b3	a0.b3			

Bit	Terms	Wires
0	a0b0	1
1	a0b1, a1b0	2
2	a0b2, a1b1, a2b0	3
3	a0b3, a1b2, a2b1, a3b0	4
4	a1b3, a2b2, a3b1	3
5	a2b3, a3b2	2
6	a3b3	1

A 4X4 WALLACE MULTIPLIER: FIRST REDUCTION

For wires within the top 3 rows:

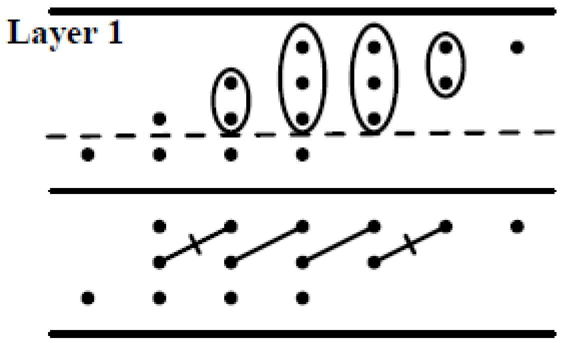
- Bit 0 has a single wire: passed through.
- Bit 1 has 2 wires: fed to a half adder.
- Bits 2 and 3 have 3 wires: fed to full adders.
- Bit 4 has 2 wires (in the group of 3): fed to a half adder.
- Bit 5 has 1 wire: passed through.

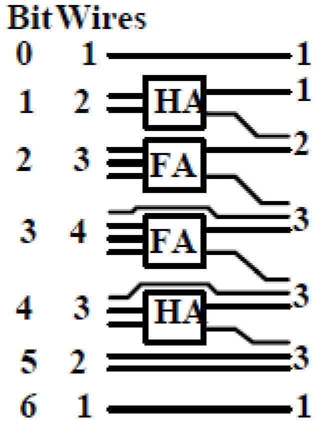


A 4x4 Wallace Multiplier: after first reduction

After first reduction

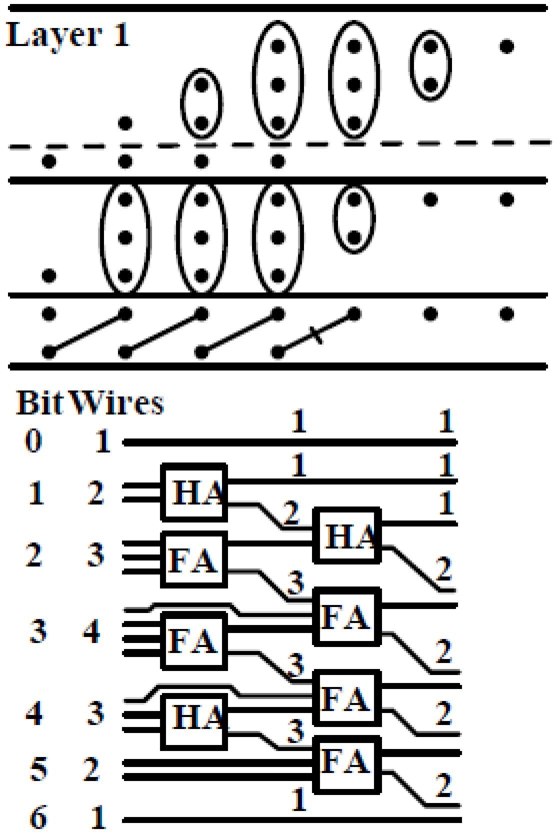
- Bits 0, 1 and 6 have a single wire.
- Bit 2 has 2 wires: carry of the half adder at bit 1 and the sum of full adder at bit 2.
- Bits 3 and 4 have 3 wires: carry of the full adder at lower weight, the sum wire from their full/half adder and a passed through wire.
- Bit 5 has 3 wires: carry of bit 4 plus 2 fed through wires.





A 4X4 WALLACE MULTIPLIER: SECOND REDUCTION

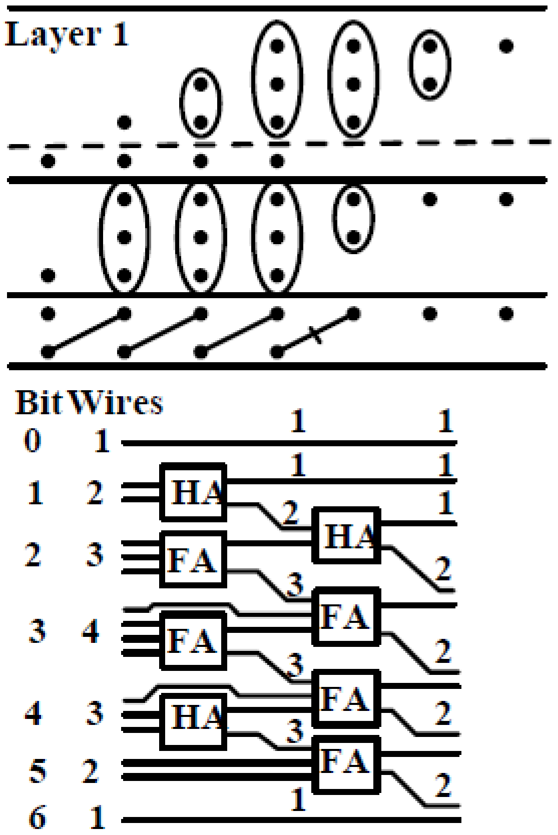
Since Bits 3, 4 and 5 have 3 wires each, we need another reduction layer. This will be the last reduction layer.



- Bits 0 and 1 have single wires. These are fed through.
- Bit 2 has 2 wires: these are fed to a half adder.
- Bits 3, 4 and 5 have 3 wires: These are fed to full adders.

A 4x4 Wallace Multiplier: After Second Reduction

Bits 0, 1, and 2 have single wires which carry the final result.

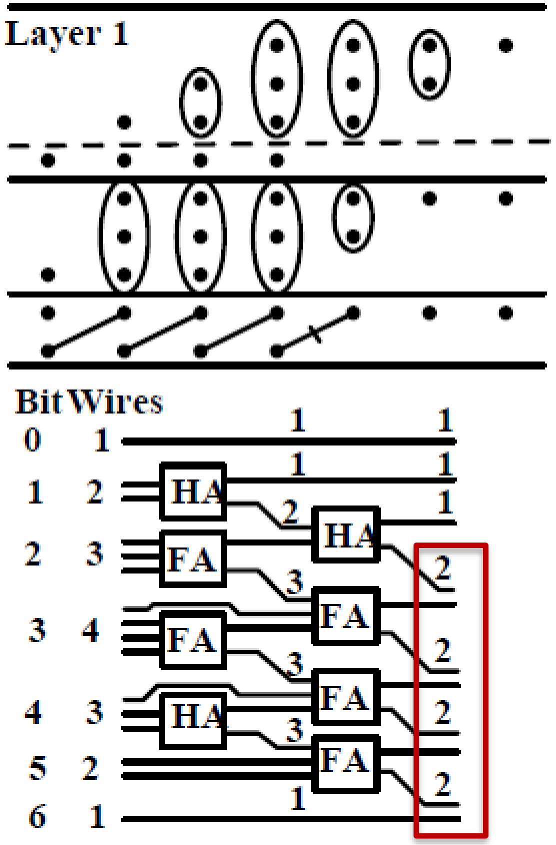


- Bit 3 has 2 wires: carry of the half adder at bit 2 and sum of the full adder at bit 3.
- Bit 4 has 2 wires: carry of the full adder at bit 3, and sum of the full adder at bit 4.
- Bit 5 has 2 wires, carry of bit 4 and sum of bit 5.
- Bit 6 has 2 wires, carry of bit 5 and 1 fed through wire.

A 4X4 WALLACE MULTIPLIER: FINAL ADDITION

After the second layer, no bit has more than 2 wires. Single wires at bits 0, 1 and 2 are fed through to the output.

- A fast conventional adder is used to add the 2 bits each at Bits 3, 4, 5 and 6.
- Notice that we do not need a full width fast adder.
- This is because the half adders at low weights have already rippled the carry while the rest of weights were being reduced.
- This makes the final adder smaller and faster.



A 4-bit adder

8x8 Wallace Multiplier