EE671: VLSI Design
Spring 2024/25

Laxmeesha Somappa

Department of Electrical Engineering

IIT Bombay

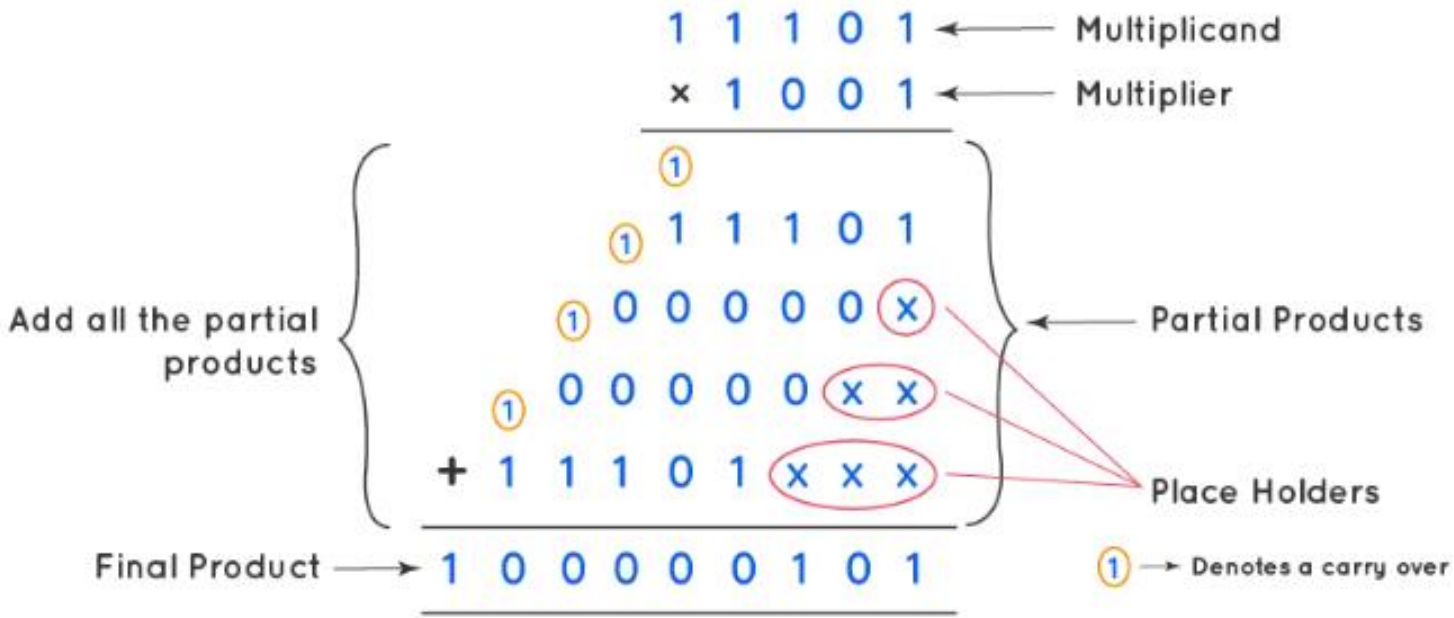
laxmeesha@ee.iitb.ac.in

Lecture – 18
Arithmetic IP: Multipliers

Acknowledgement: Prof. D.K.S

laxmeesha@ee.iitb.ac.in

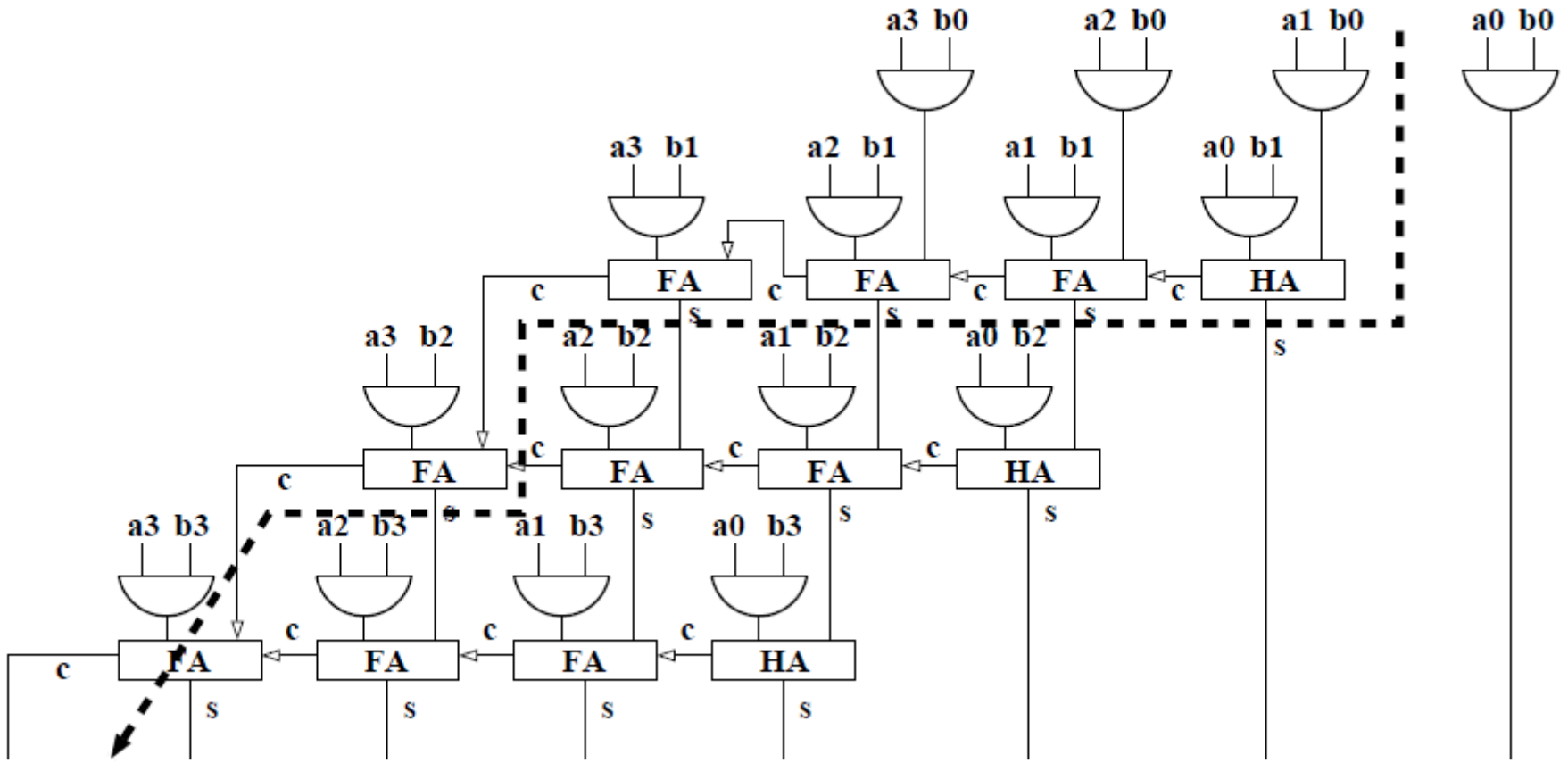
BINARY MULTIPLIER: BASICS



- Rules: a) bit-wise multiplication (AND gate)
  b) Repeat bit-wise multiplication → generates partial product
  c) Shift & add all partial products → final product

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 ?

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

- Same example: multiply A = 229 and B = 222
- A = 11100101, B = 11011110
- Booth Encoding Multiplier: Recall booth encoding table

B_{i+1}	B_i	B_{i-1}	PP
0	0	0	0
0	0	1	A
0	1	0	A
0	1	1	2A
1	0	0	-2A
1	0	1	-A
1	1	0	-A
1	1	1	0

Modified Booth Encoding

A = 0000000011100101

2A = 0000000111001010

-A = 1111111100011011

-2A = 1111111000110110

B = 11011110

B = 11|01|11|10

B = 00|11|01|11|100

PP = A|-A|2A|0|-2A

B_{i+1}	B_i	B_{i-1}	PP
0	0	0	0
0	0	1	A
0	1	0	A
0	1	1	2A
1	0	0	-2A
1	0	1	-A
1	1	0	-A
1	1	1	0

Modified Booth Encoding

A = 0000000011100101 -A = 1111111100011011

2A = 0000000111001010 -2A = 1111111000110110

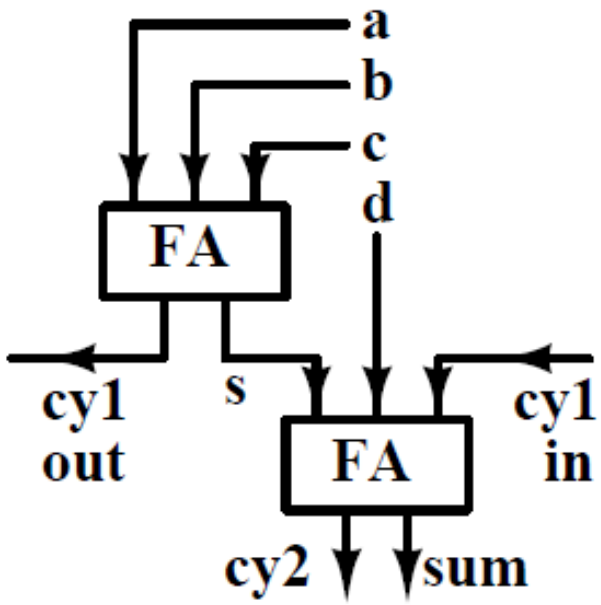
PP = A|-A|2A|0|-2A

```
  1111111000110110
  0000000000000xx
----------------
  1111111000110110
  0001110010xxxx
----------------
 10001101011010110
1111111100011011xxxxxx
----------------
10000001110000110010110
00000000111001010xxxxxxx
----------------
01000001100011010010110
```

A X B = 1100011010010110 with only 5 partial products instead of 8 in an array multiplier

Modified Booth Encoding

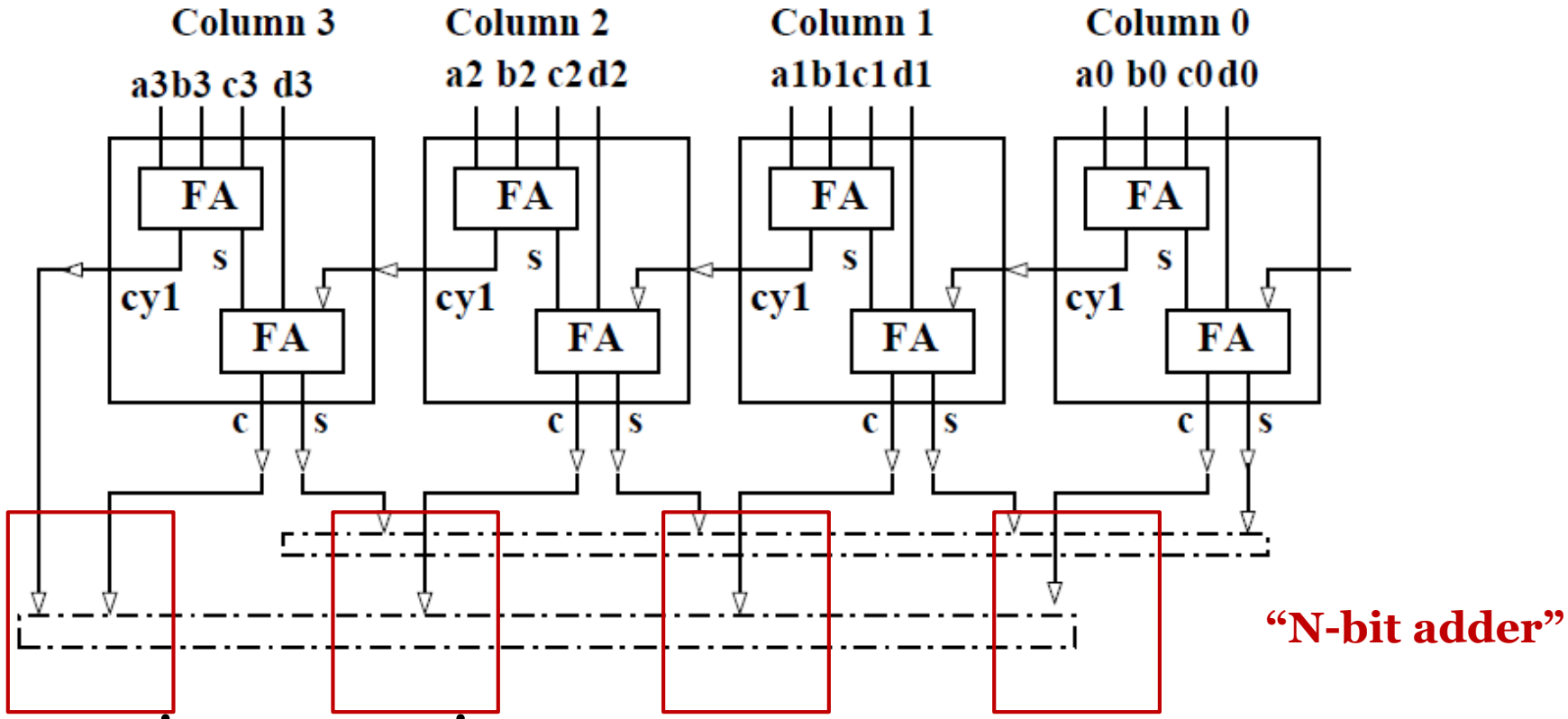
- We looked at 1-bit adders to implement 'N' bit adders
- Can we also have a 2-bit adder to add partial products?



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

MODIFIED BOOTH ENCODING

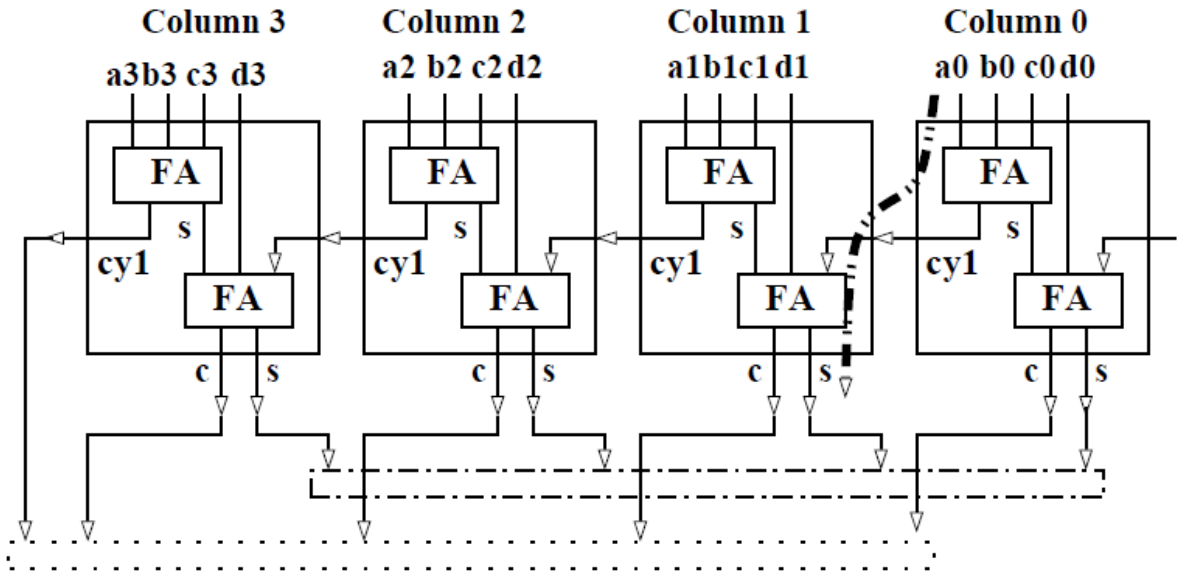
- 2-bit adders to implement 'N' bit adders: for PP addition

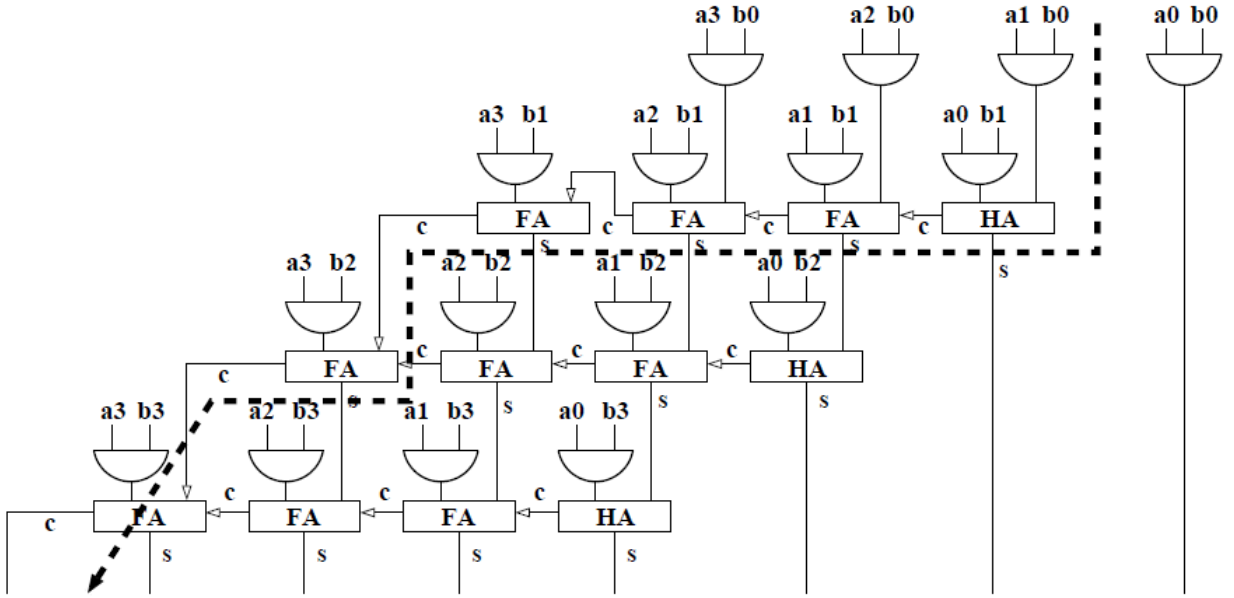


- Save output in two registers
- Add these two registers using any adder that we have seen

Modified Booth Encoding

- PP addition: critical path (no carry ripple except for the last 2-register addition) – compare with normal ripple adder!





laxmeesha@ee.iitb.ac.in

WALLACE MULTIPLIERS

- Consider 4x4 multiplication

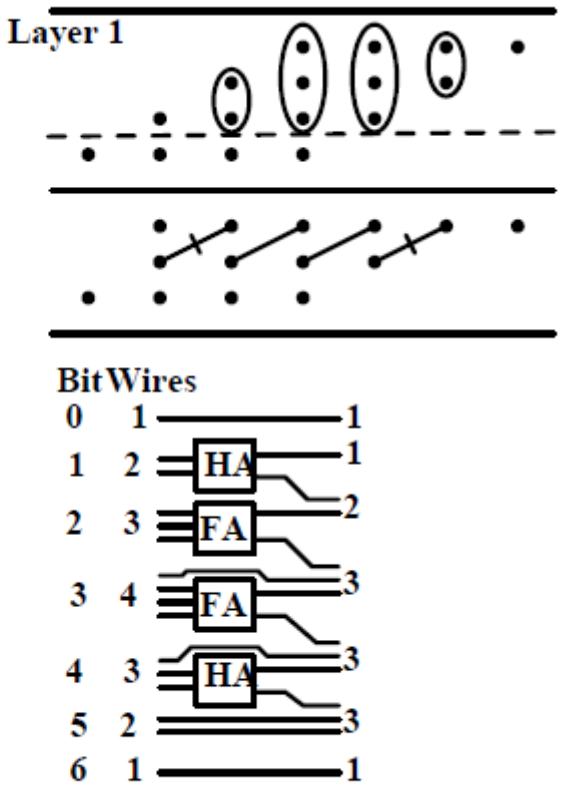
						a3	a2	a1	a0
					x	b3	b2	b1	b0
				a3.b0	a2.b0	a1.b0	a0.b0
			a3.b1	a2.b1	a1.b1	a0.b1	
		a3.b2	a2.b2	a1.b2	a0.b2		
	a3.b3	a2.b3	a1.b3	a0.b3			

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: after first reduction

After first reduction

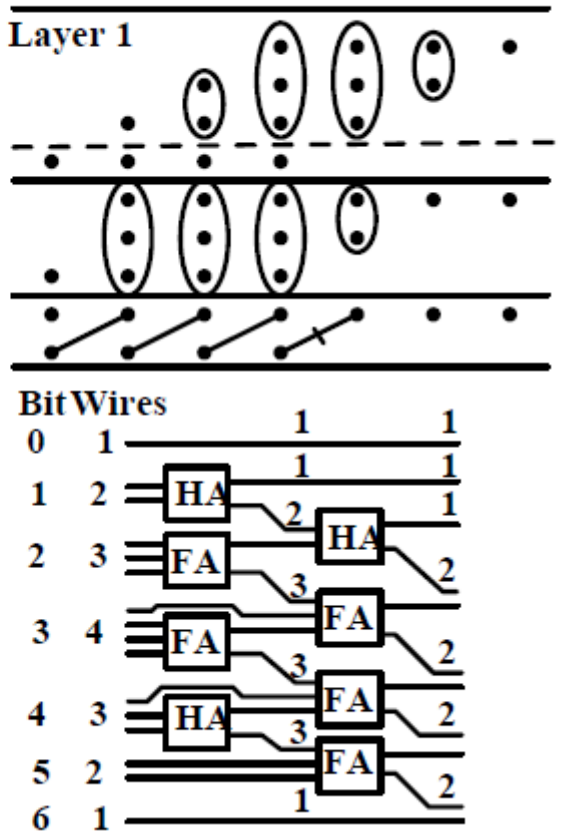
- Bits 0, 1 and 6 have a single wire.
- Bit 2 has 2 wires: carry of the half adder at bit 1 and the sum of full adder at bit 2.
- Bits 3 and 4 have 3 wires: carry of the full adder at lower weight, the sum wire from their full/half adder and a passed through wire.
- Bit 5 has 3 wires: carry of bit 4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



A 4x4 Wallace Multiplier: After Second Reduction

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

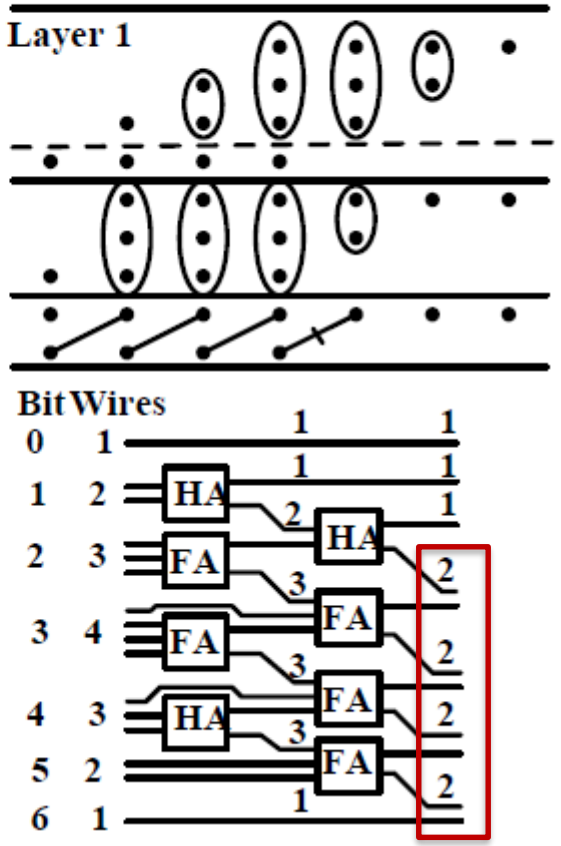
- Bit 3 has 2 wires: carry of the half adder at bit 2 and sum of the full adder at bit 3.
- Bit 4 has 2 wires: carry of the full adder at bit 3, and sum of the full adder at bit 4.
- Bit 5 has 2 wires, carry of bit 4 and sum of bit 5.
- Bit 6 has 2 wires, carry of bit 5 and 1 fed through wire.



A 4x4 Wallace Multiplier: Final addition

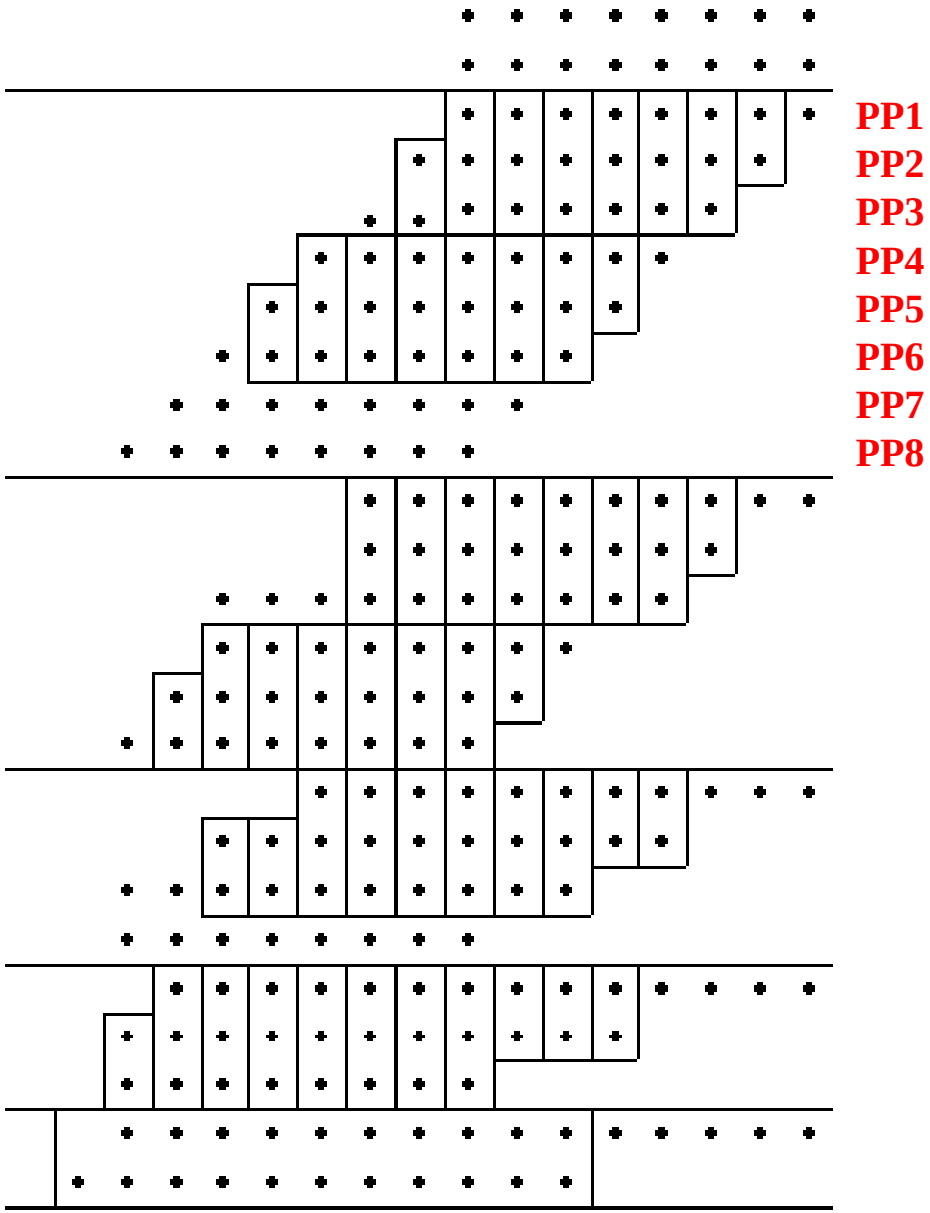
After the second layer, no bit has more than 2 wires. Single wires at bits 0, 1 and 2 are fed through to the output.

- 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



First reduction:
HAs: 4, FAs: 12

Second reduction:
HAs: 3, FAs: 13

Third reduction:
HAs: 4, FAs: 6

Fourth reduction:
HAs: 4, FAs: 7

Final:11-bit fast adder

DADDA MULTIPLIER

- Similar to Wallace: but,
  - Wallace: reduce the PP bits as soon as possible (i.e., moment you see more than or equal to 2 bits, put HA or FA
  - Dadda: reduce the PP bits as late as possible, based on

$$d_{k+1} = \text{floor}\,(3/2\ d_k) \text{ and } d_1 = 2$$

Possible d = 2,3,4,6,9,13,19 .....

- For an NxM multiplier,

$$\mathbf{d < min(N,M)}$$

- Example: for 4x4, d = 3, for 8x8, d = 6
- At every stage, if there are **more than** *d* bits, we reduce using FA/HA
  - Reduce d after every reduction

DADDA MULTIPLIERS

- Consider 4x4 multiplication

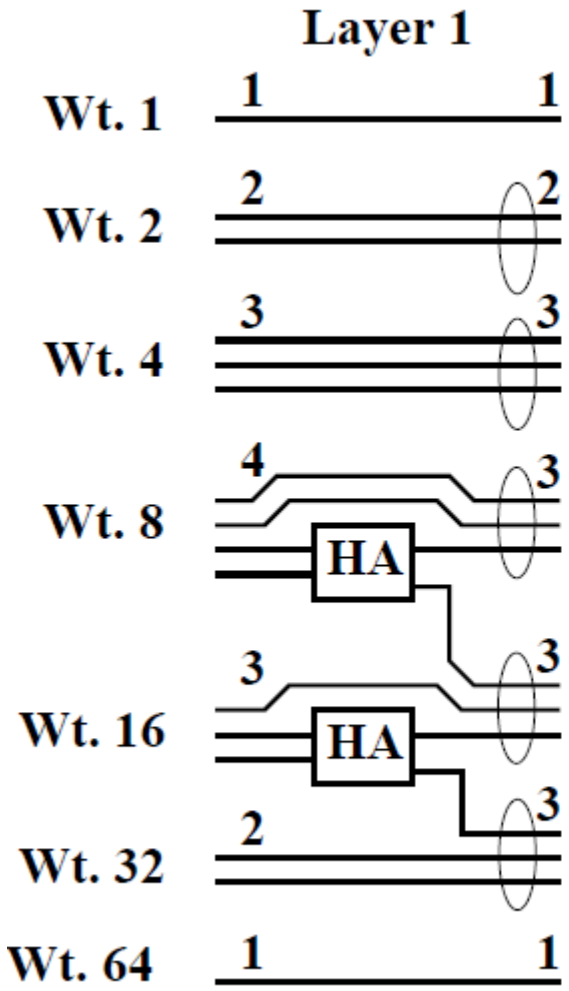
						a3	a2	a1	a0
					x	b3	b2	b1	b0
				a3.b0	a2.b0	a1.b0	a0.b0
			a3.b1	a2.b1	a1.b1	a0.b1	
		a3.b2	a2.b2	a1.b2	a0.b2		
	a3.b3	a2.b3	a1.b3	a0.b3			

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

- d = 3 for 4x4 multiplier

DADDA MULTIPLIERS: FIRST REDUCTION

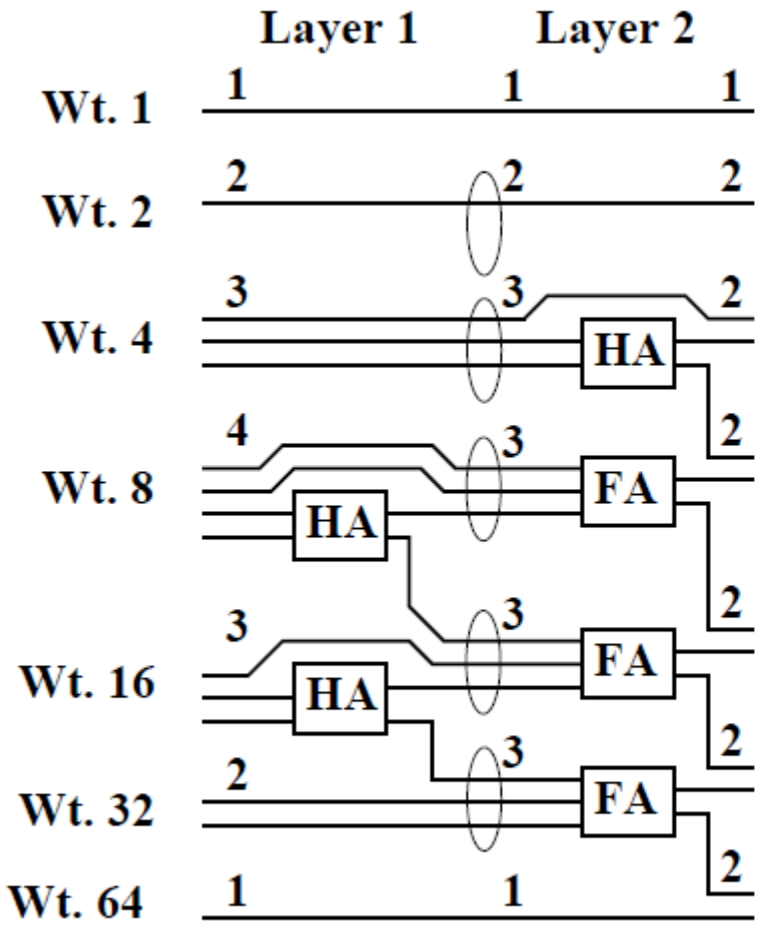
- First reduction, d = 3



- Wt.1 has the single wire which was fed through.
- Wt.2 has 2 fed through wires.
- Wt.4 has 3 wires: all passed through.
- Wt.8 has 3 wires: sum of the half adder at wt.4, and 2 passed through.
- Wt.16 has 3 wires: carry of wt. 8, sum of half adder at 16 and 1 passed through.
- Wt.32 has 3 wires: carry of wt. 16 and 2 passed through.
- Wt.64 has 1 fed through wire.

DADDA MULTIPLIERS: FIRST REDUCTION

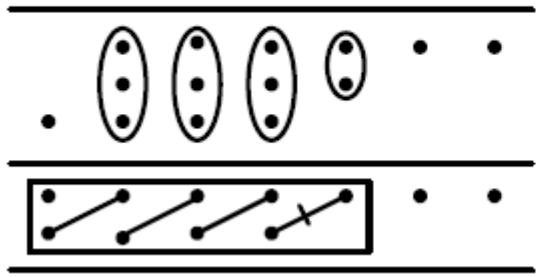
- Second reduction, d = 2

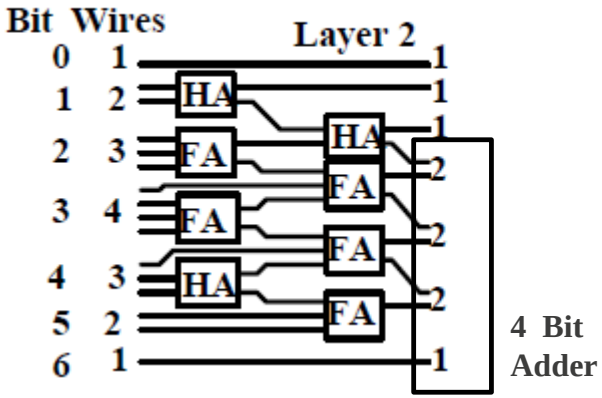


- 3 wires at Wt. 4 are reduced to 2 by a half adder: No carry in expected.
- Wt. 8 has 3 input wires. Carry will arrive from Wt. 4. Reduced using a Full adder.
- Wt. 16 has 3 input wires. Carry will arrive from Wt. 8. Reduced using a full adder.
- Wt. 32 has 3 input wires. Carry will arrive from Wt. 16. Reduced using a full adder.
- Wt. 64 has 1 input wire. Carry will arrive from Wt. 32, making it 2 output wires, which will be fed through.

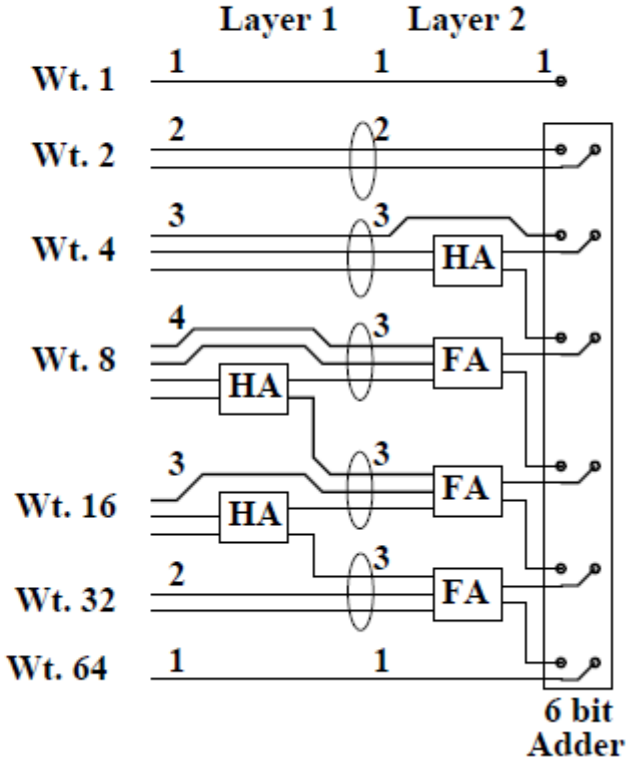
4X4 COMPARISON

Wallace Multiplier



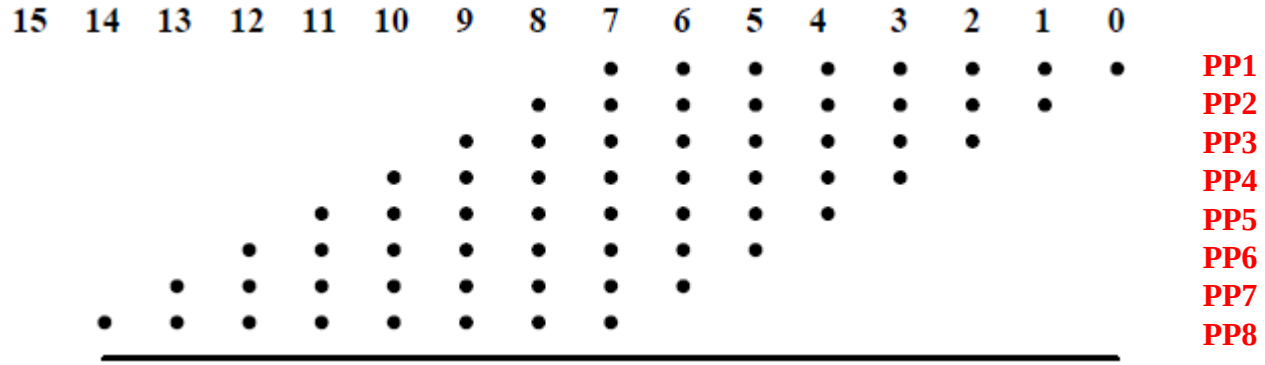


Dadda Multiplier

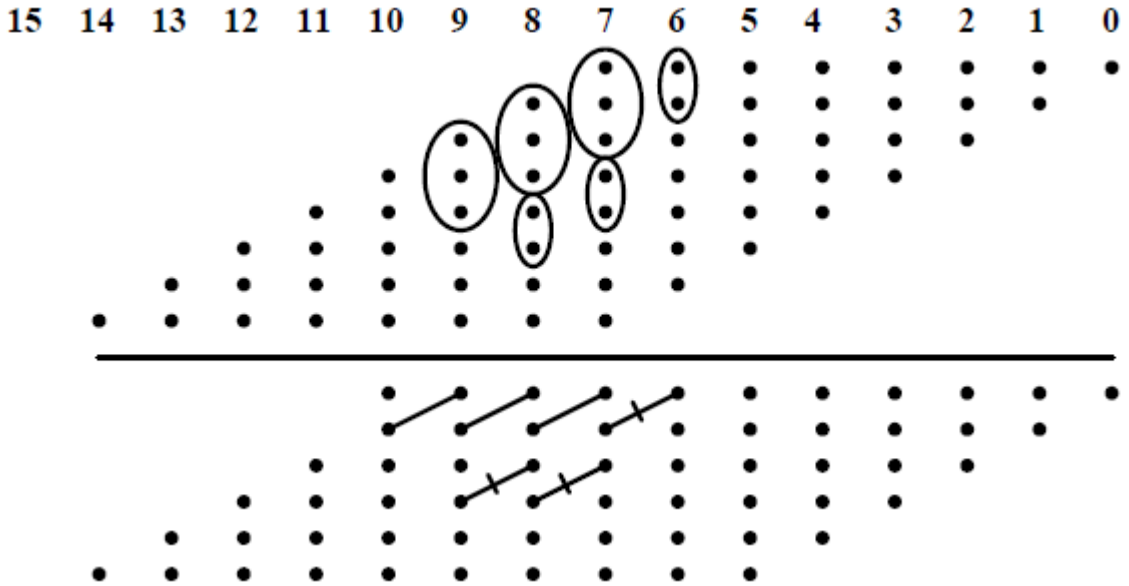


- Both use 2 layers of reduction
- Wallace: 3 HA + 5 FA, Dadda: 3 HA + 3 FA
- Wallace: 4-bit fast adder, Dadda: 6-bit fast adder

DADDA: 8X8

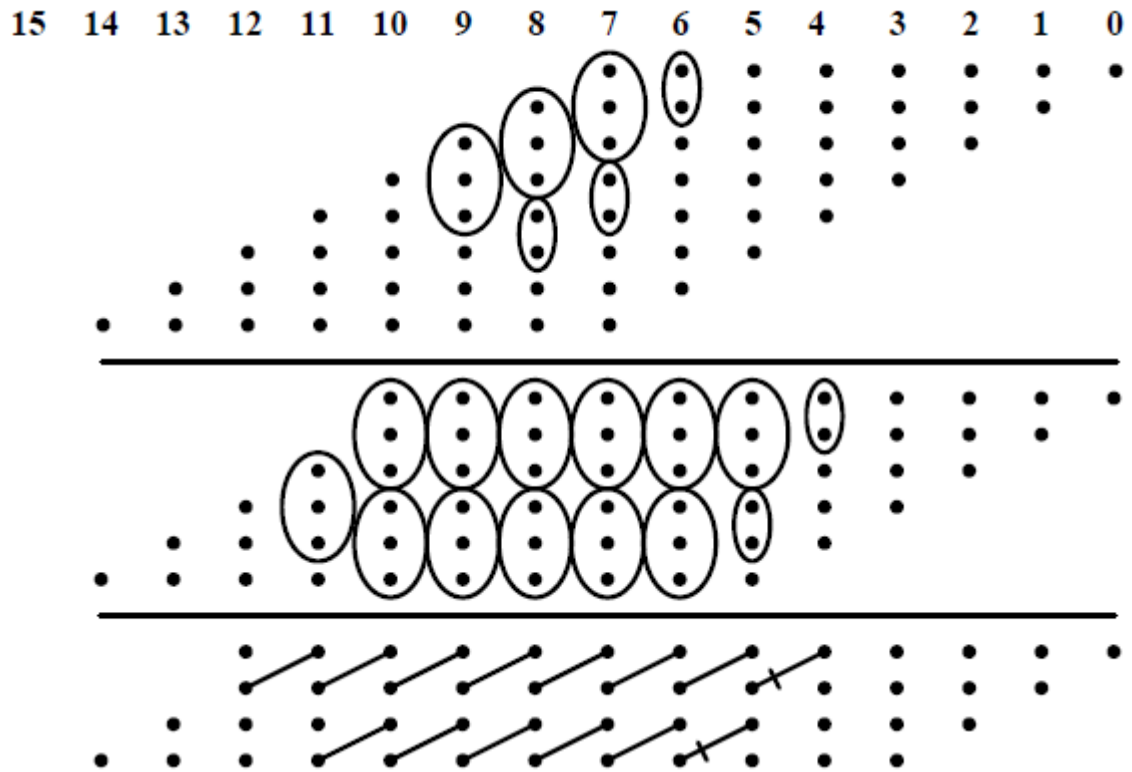


- First reduction, d = 6



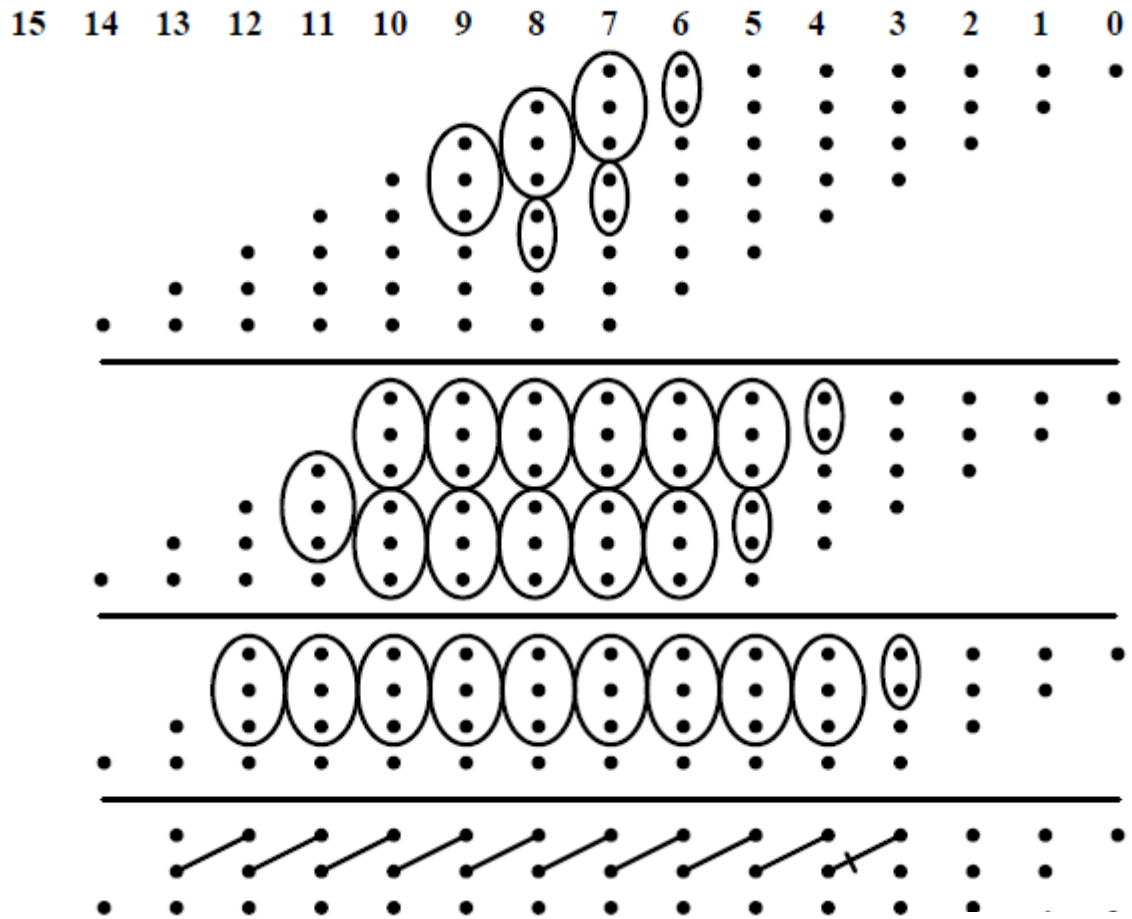
DADDA: 8X8

- Second reduction, d = 4



DADDA: 8X8

- Third reduction, d = 3



DADDA: 8X8

- Final reduction, d = 2 (14-bit fast –adder)