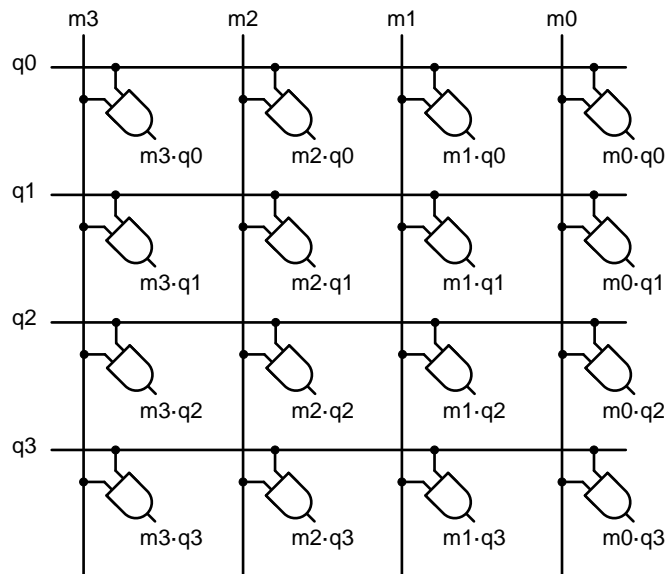


# EE 457 Unit 2c

## Fast Multipliers

# Multiplication Overview

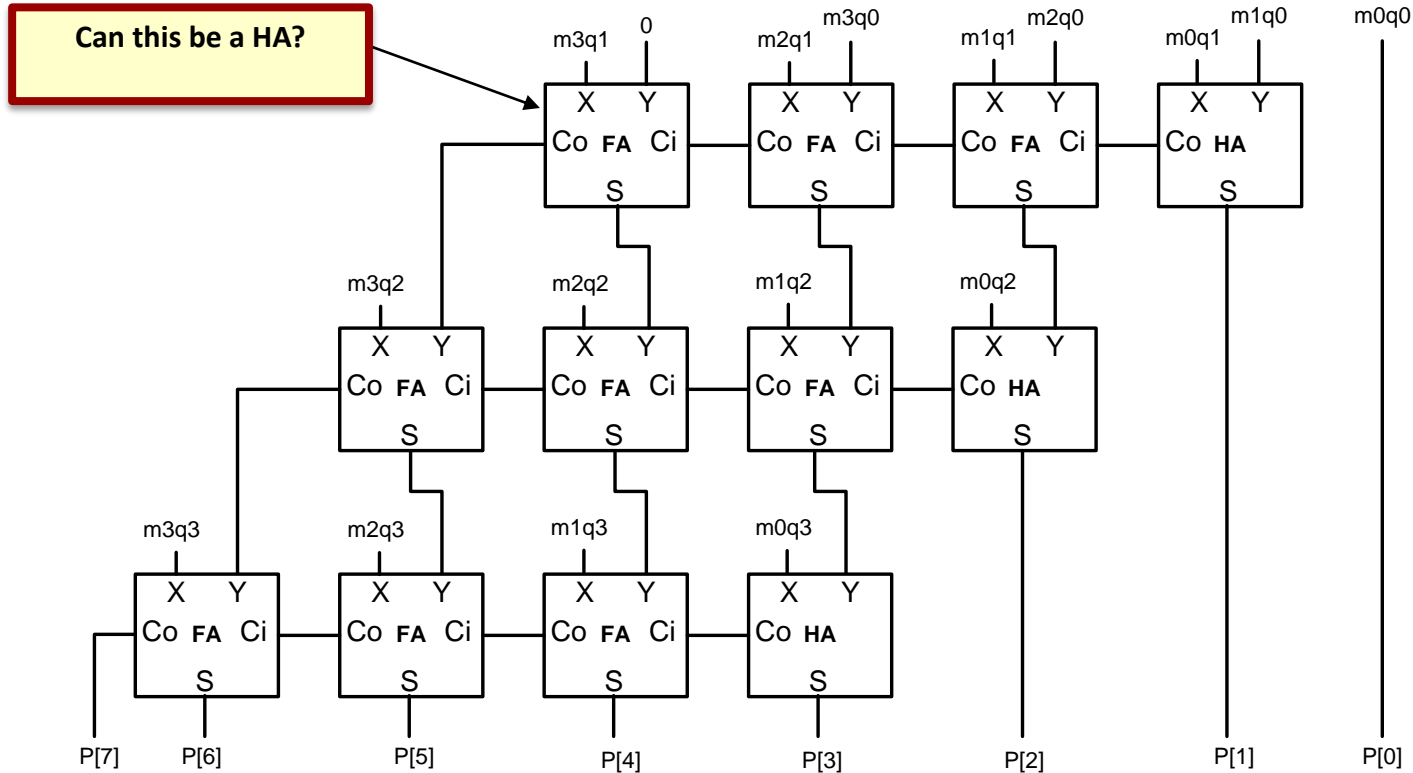
- Multiplication approaches:
  - Sequential: Shift-and-Add produces one product bit per clock cycle time (usually slow)
  - Combinational: Array multiplier uses an array of adders
    - Can be as simple as N-1 ripple-carry adders for an NxN multiplication



AND Gate Array produces partial product terms

	m3	m2	m1	m0				
<b>x</b>	q3	q2	q1	q0				
		m3q0	m2q0	m1q0	m0q0			
		m3q1	m2q1	m1q1	m0q1	+		
		m3q2	m2q2	m1q2	m0q2	↓	-	
+	m3q3	m2q3	m1q3	m0q3	↓	↓	↓	
	p7	p6	p5	p4	p3	p2	p1	p0

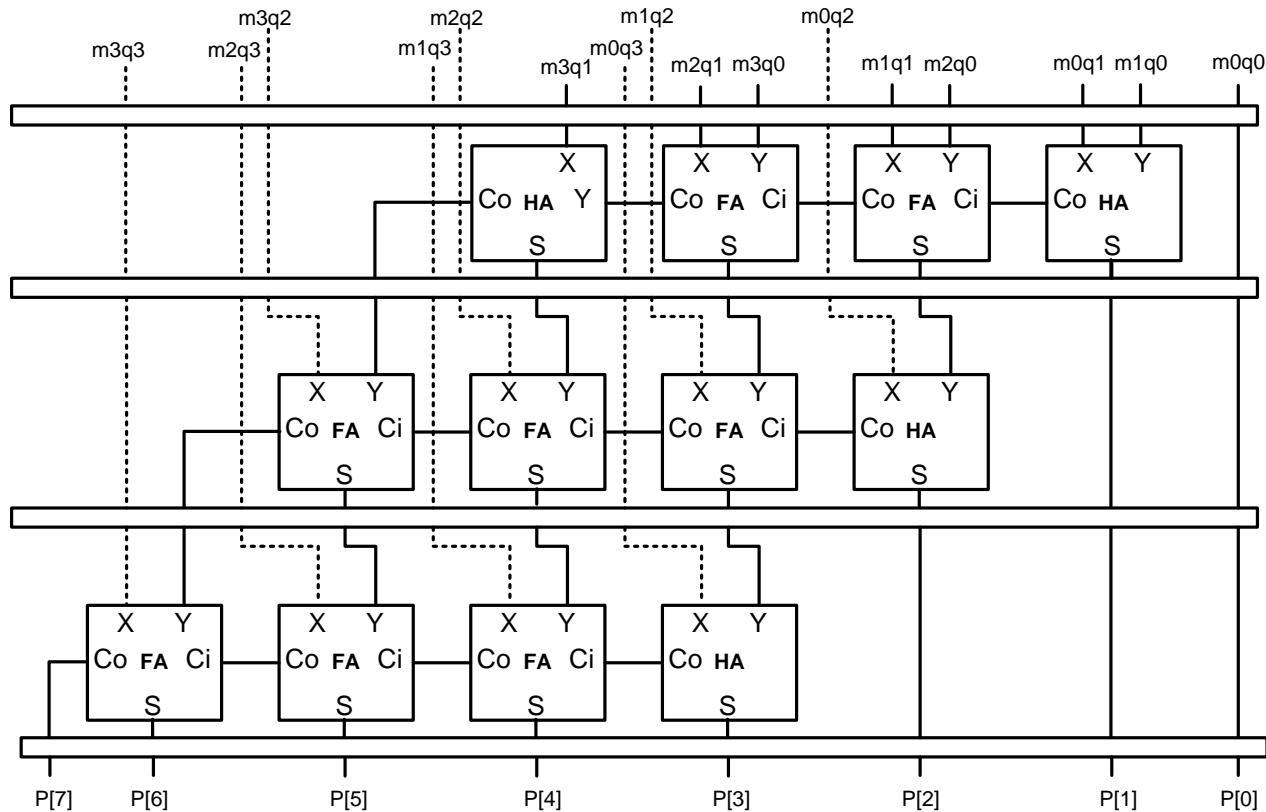
# Array Multiplier



- Maximum delay = ?
  - Do you look for the longest path or the shortest path between any input and output?
  - Compare with the delay of a shift-and-add method

# Pipelined Multiplier

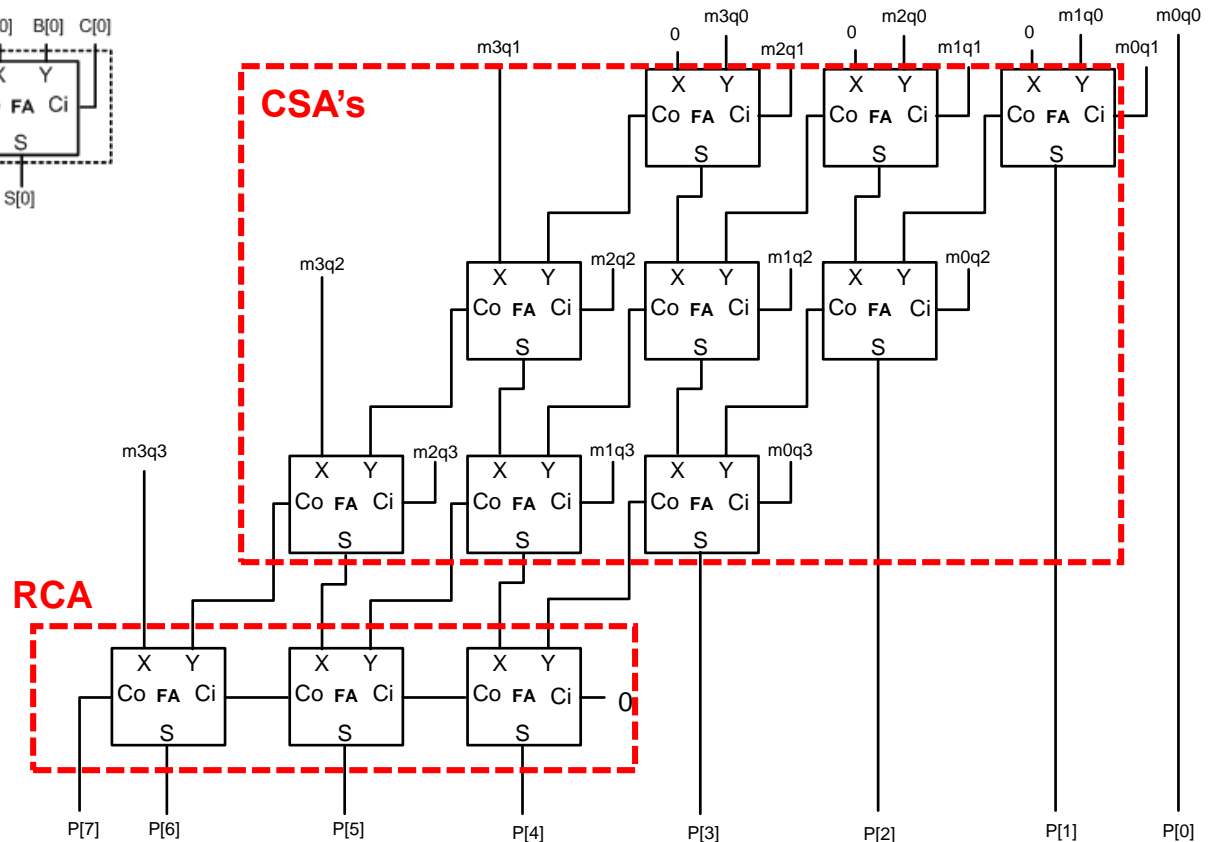
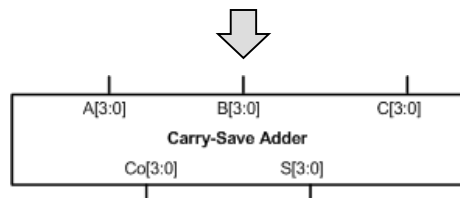
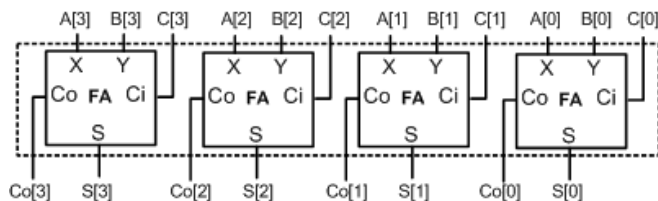
- Now try to pipeline the previous design



**Determine the maximum stage delay to decide the pipeline clock rate.**  
**Assume zero-delay for stage latches. How does the latency of the pipeline compare with the simple combinational array of the previous stage?**

# Carry-Save Multiplier

- Instead of propagating the carries to the left in the same row, carries are now sent down to the next stage to reduce stage delay and facilitate pipelining



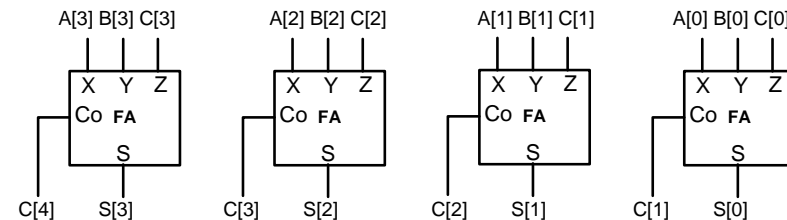
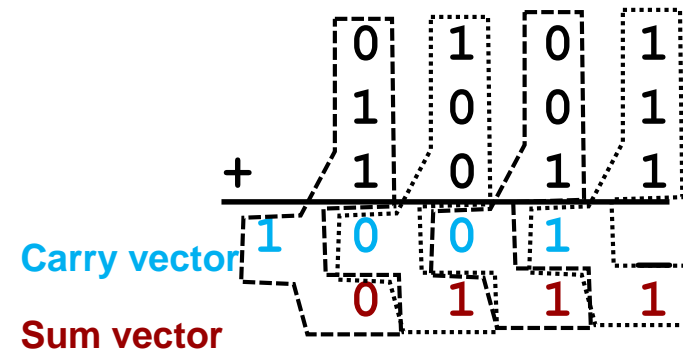
**The upper three stages are 3-bit Carry Save Adders (CSA's) each with 2-gate delays.**

**The last stage is a Ripple Carry Adder (RCA) which requires longer delay. It can be replaced by a CLA for larger multipliers.**



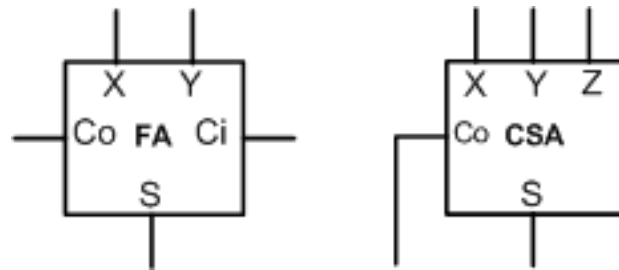
# Carry-Save (3,2) Adders

- A carry save adder is also called a (3,2) adder or a (3,2) counter (refer to Computer Arithmetic Algorithms by Israel Koren) as it takes three vectors, adds them up, and reduces them to two vectors, namely a **sum vector** and a **carry vector**
- CSA's are based on the principle that carries do not have to be added as soon as possible, but can be combined in a later step
- An n-bit CSA consist of n disjoint full adders



# 1-bit FA vs. 1-bit CSA

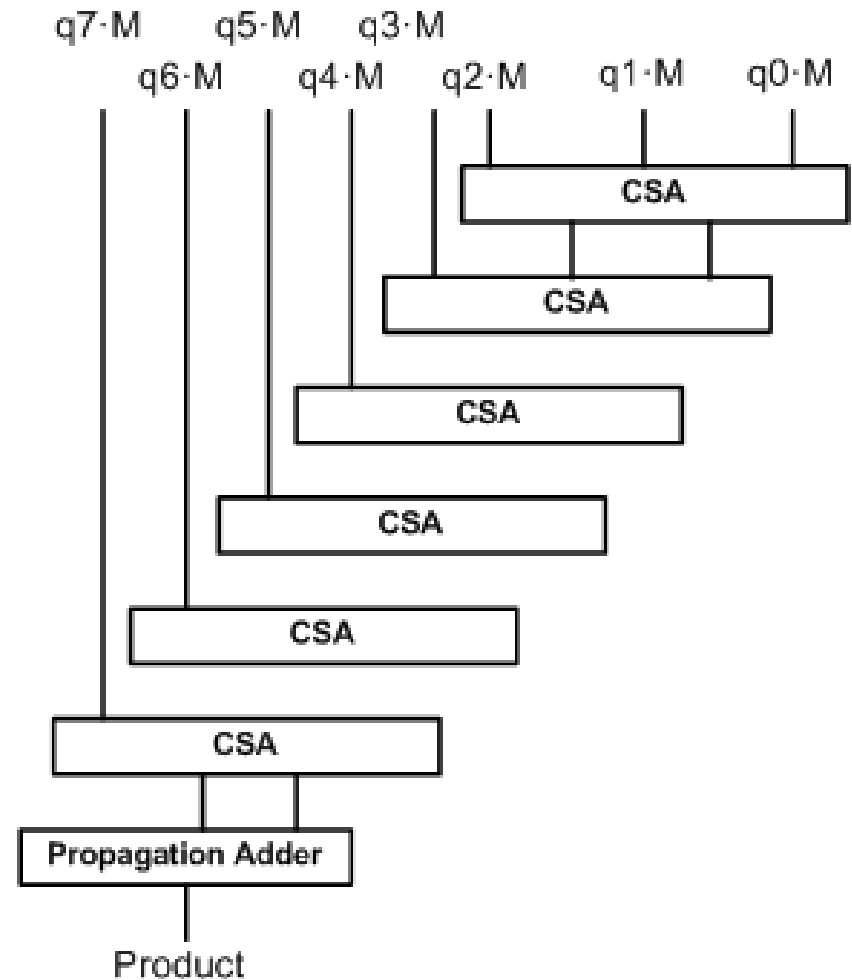
- Any difference between an ordinary full adder and 1-bit CSA? **NO!**



- 16-bit wide CSA takes (**more / equal / less**) time to produce its outputs compared to an 8-bit wide CSA
- Carry-save adder (**is / is not**) useful in adding only 2 numbers

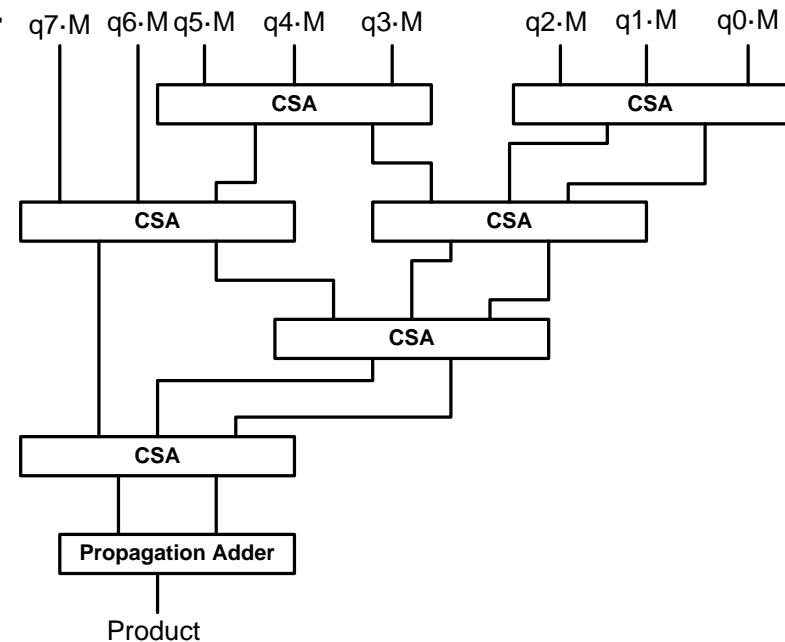
# CSA Organization

- We can arrange our CSA's in a linear manner where one partial product is added per CSA (after the first level)



# Wallace Tree Multiplier

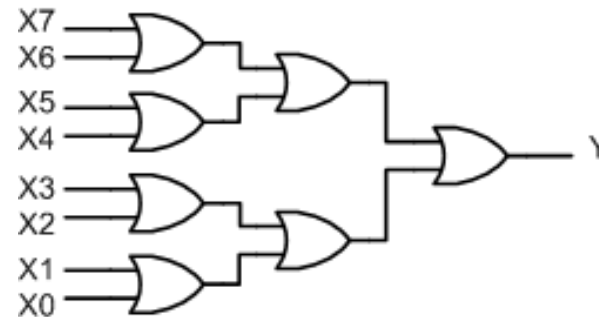
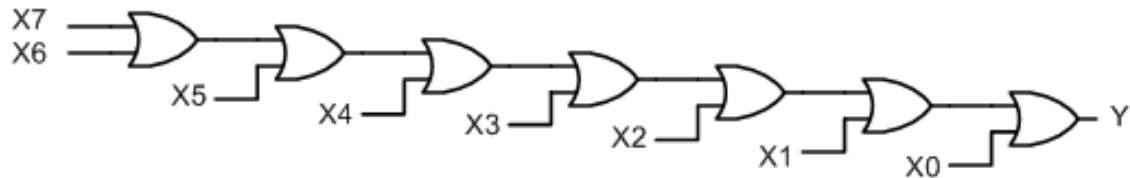
- Using the previous example as a template, to build an NxN multiplier you need (n-1) of (n-1) bit CSAs followed by a final (n-1)-bit RCA
- Delay = Delay of (n-1) CSA's  
 + Delay of (n-1) bit RCA  
 = 2 \* (n-1) \* Delay(FullAdder)
- We can reduce the CSA component of the delay by organizing the CSA's in a tree (i.e. logarithmic delay)



**Note:** The vectors (partial products) need to be aligned before summing. These details are not shown in the block diagram.

# Logic Delay

- Consider the gate arrangement for OR'ing 8 bits
- Linear:
  - Delay = 7 gates
- Tree
  - Depth of tree =  $\log_2 8 = 3$  levels
- Consider OR'ing 16-bits using 4-bit OR gates, how many levels would you need?



# Wallace Tree Discussion

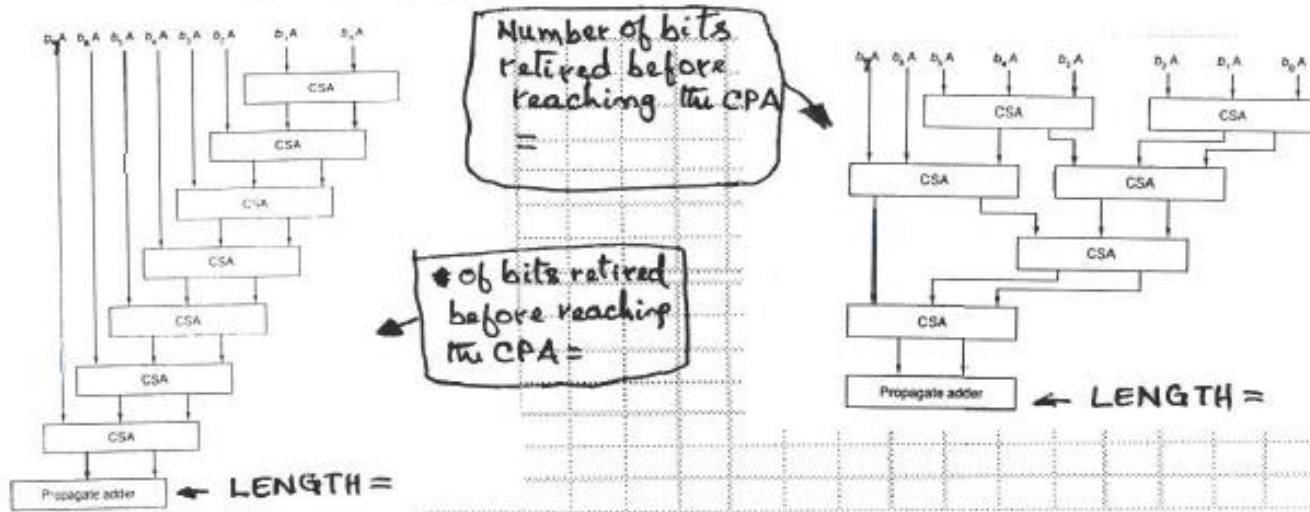
- A 4-input OR gate reduces 4 literals to 1 (i.e. a factor of 4 reduction)
- A CSA reduces 3 vectors to 2 vectors (i.e. a factor of 1.5)
  - This reduction factor may not be convenient to develop an efficient tree to sum 16 or 32 partial products
  - Wallace tree may not achieve a great reduction in delay due to wastage of an extra level
- Also note the Wallace tree shown earlier does not show...
  - Size of buses
  - What bits are “retired” progressively
  - Relative significance (alignment) of partial products
  - Size of the carry-propagate adder (e.g. RCA or CLA) needs to be figured out and overall delay estimated

2. (8+2+4+2 = 16 points) 25 min

Multiplication and CSAs:

2.1 Consider the following adder arrangements for 8 x 8 multiplication reproduced from your classnotes. Find the length of the "Propagate Adder" at the bottom of each. Consider the number of bits you will be "retiring" before reaching the propagate adder and also state the number.

8 pts



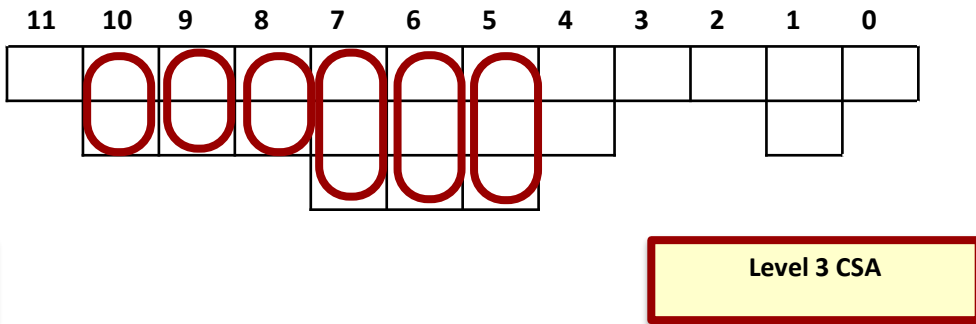
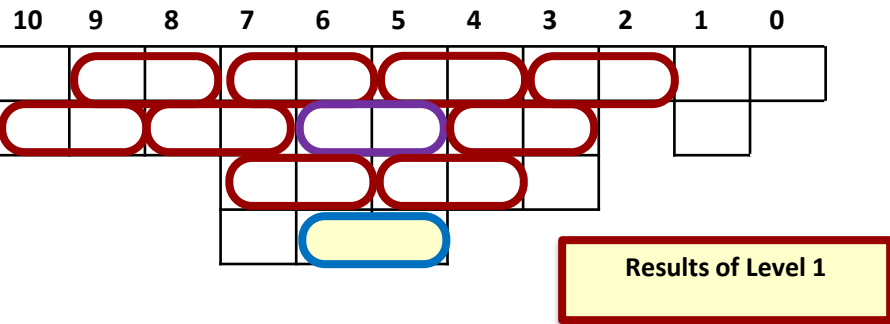
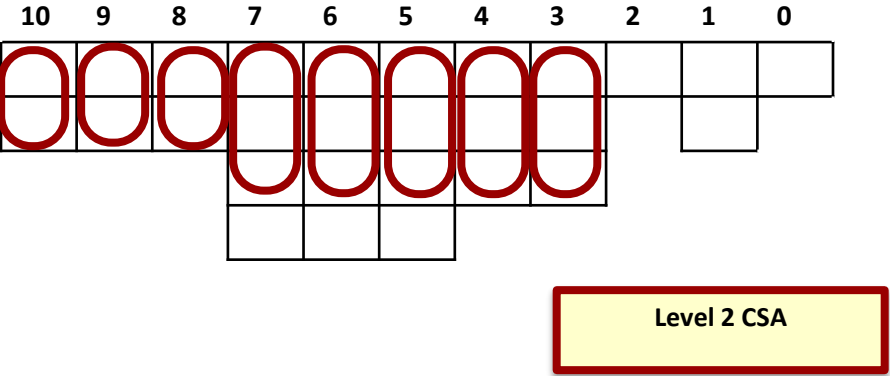
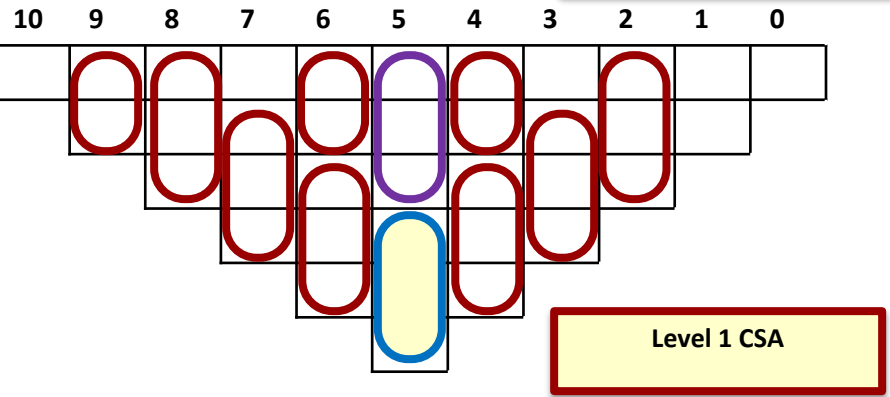
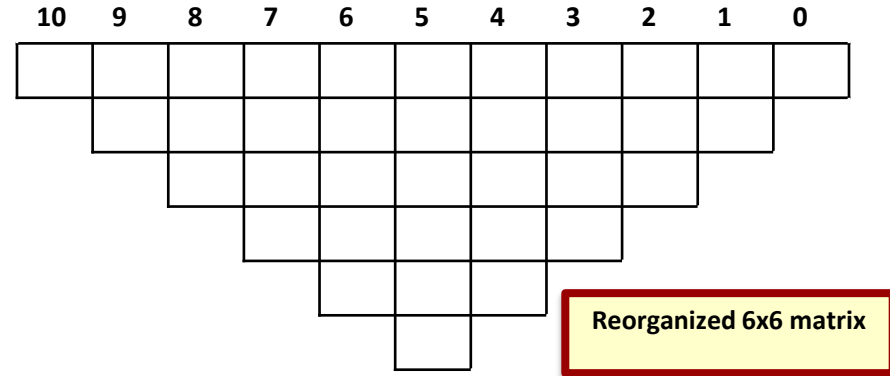
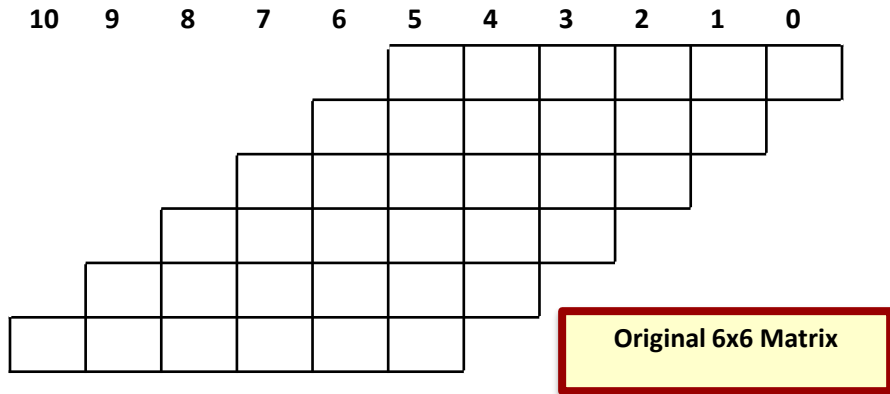
2.2 The lab #7 on Wallace Tree, based on the figure on the right above, produces the product of two \_\_\_\_\_ (signed/unsigned) 8-bit numbers.

2 pts

2.3.1 You are aware that the CSA (Carry Save Adder) accepts three vectors and produces two vectors. Hence it is sometimes called a 3-to-2 reducer. Find how many levels of CSAs you need to reduce 32 partial sums (in a 32x32 Wallace Tree multiplier) into two vectors.

4 pts

Level 1: 32 vectors - { [30] + 2 } ==> { [30 / (3/2)] + 2 } = { [20] + 2 } = 22



# Credits

- These slides were derived from Gandhi Puvvada's EE 457 Class Notes