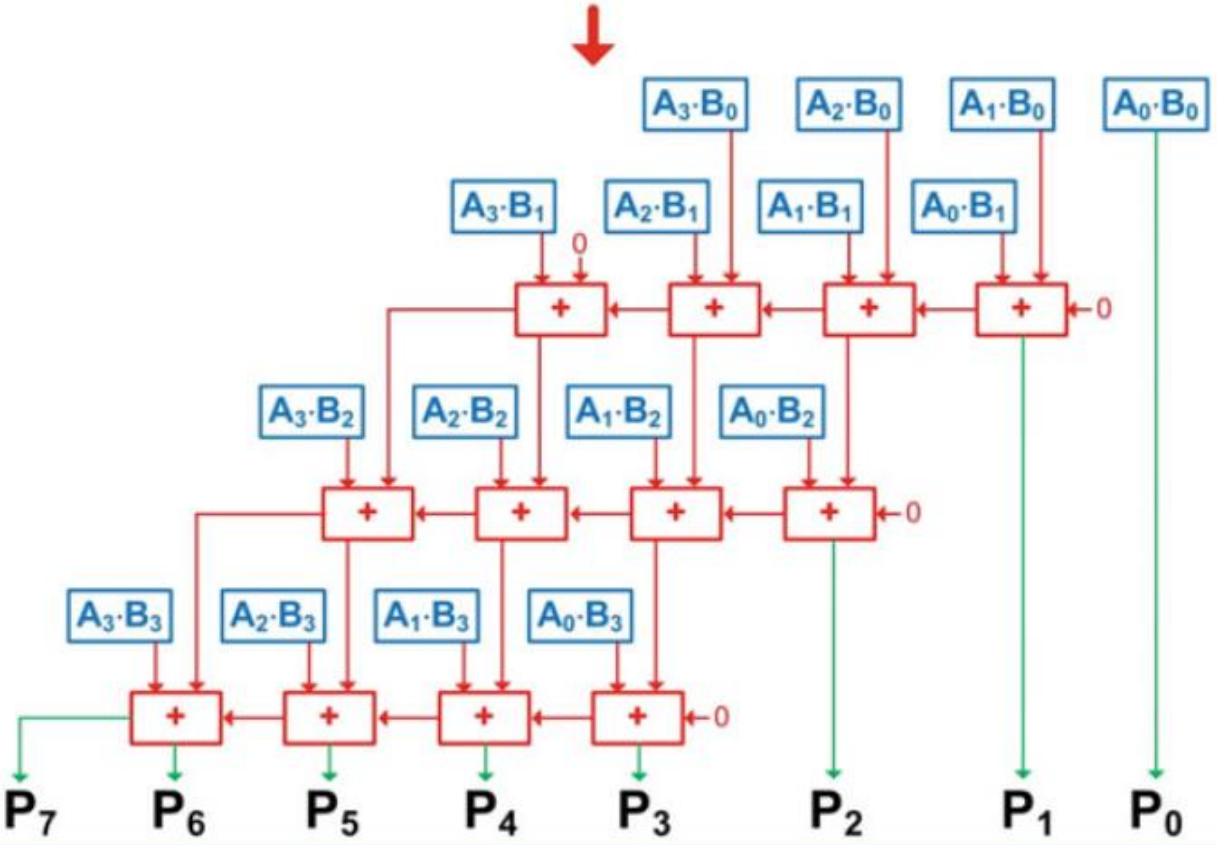
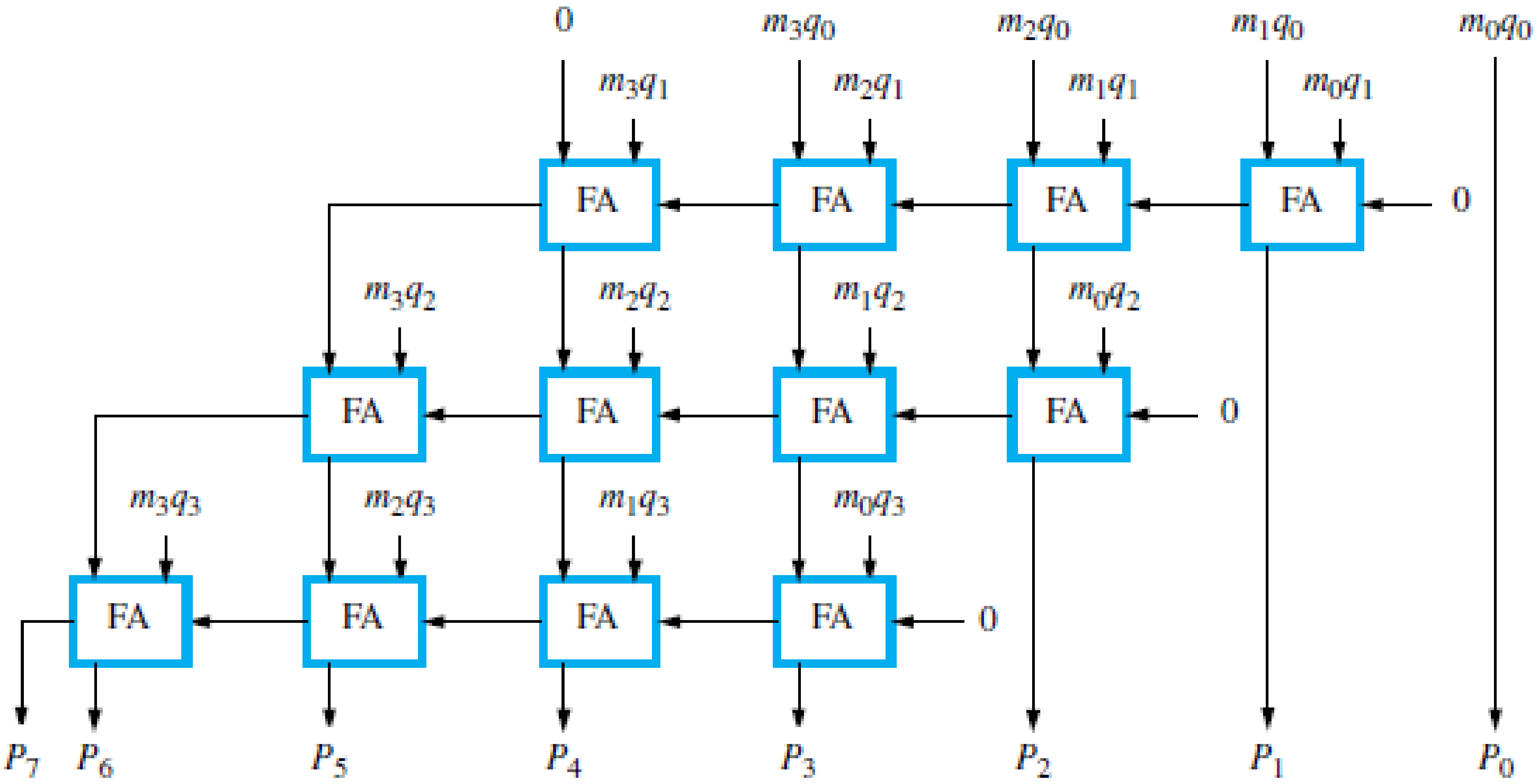
Example: Design of a 4-Bit Unsigned Multiplier

If we break the sum of the partial product columns into incremental addition steps, we can then use full adders.

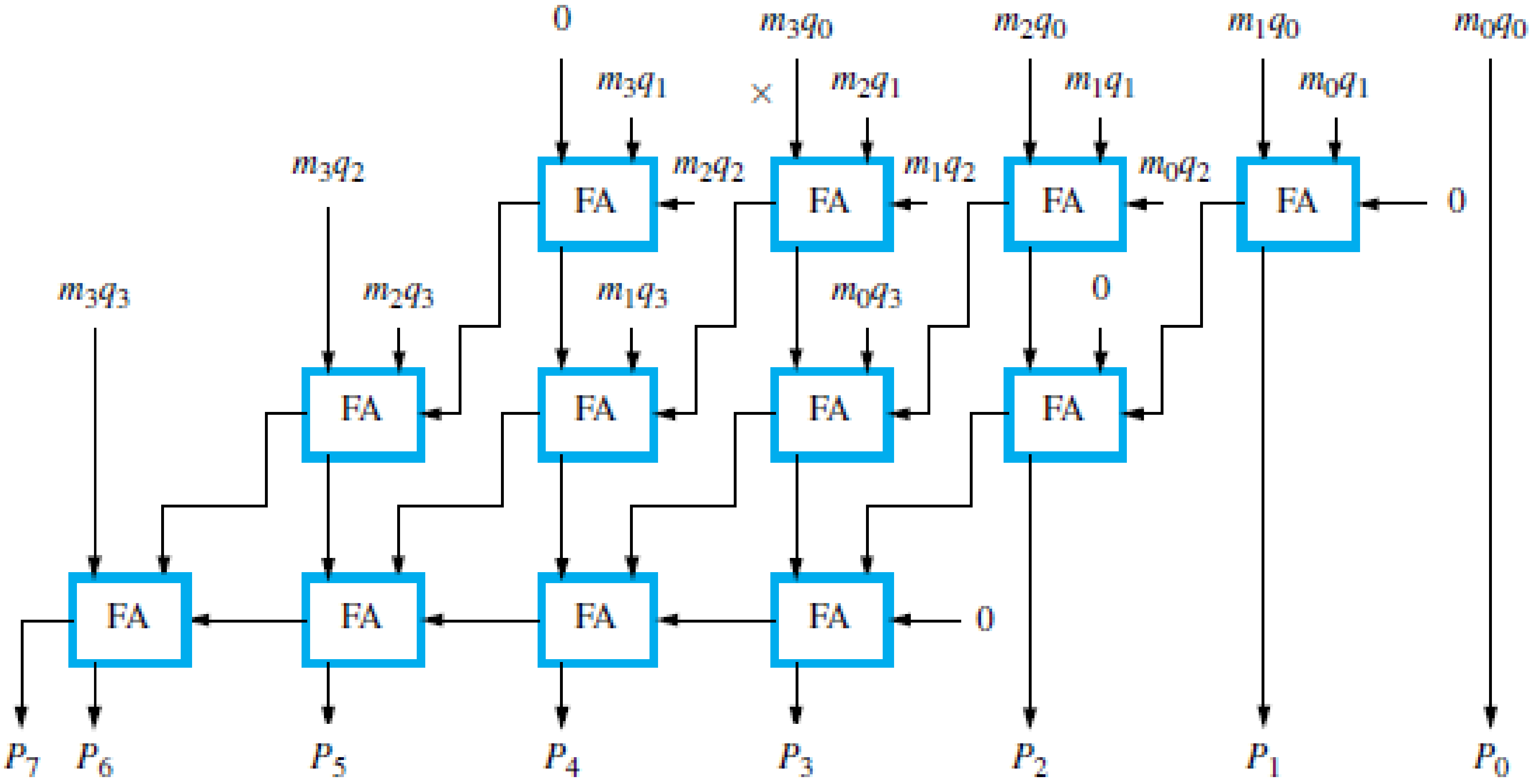
$$\begin{array}{r} A_3A_2A_1A_0 \\ \times \quad\quad B_3B_2B_1B_0 \\ \hline P_7P_6P_5P_4P_3P_2P_1P_0 \end{array}$$



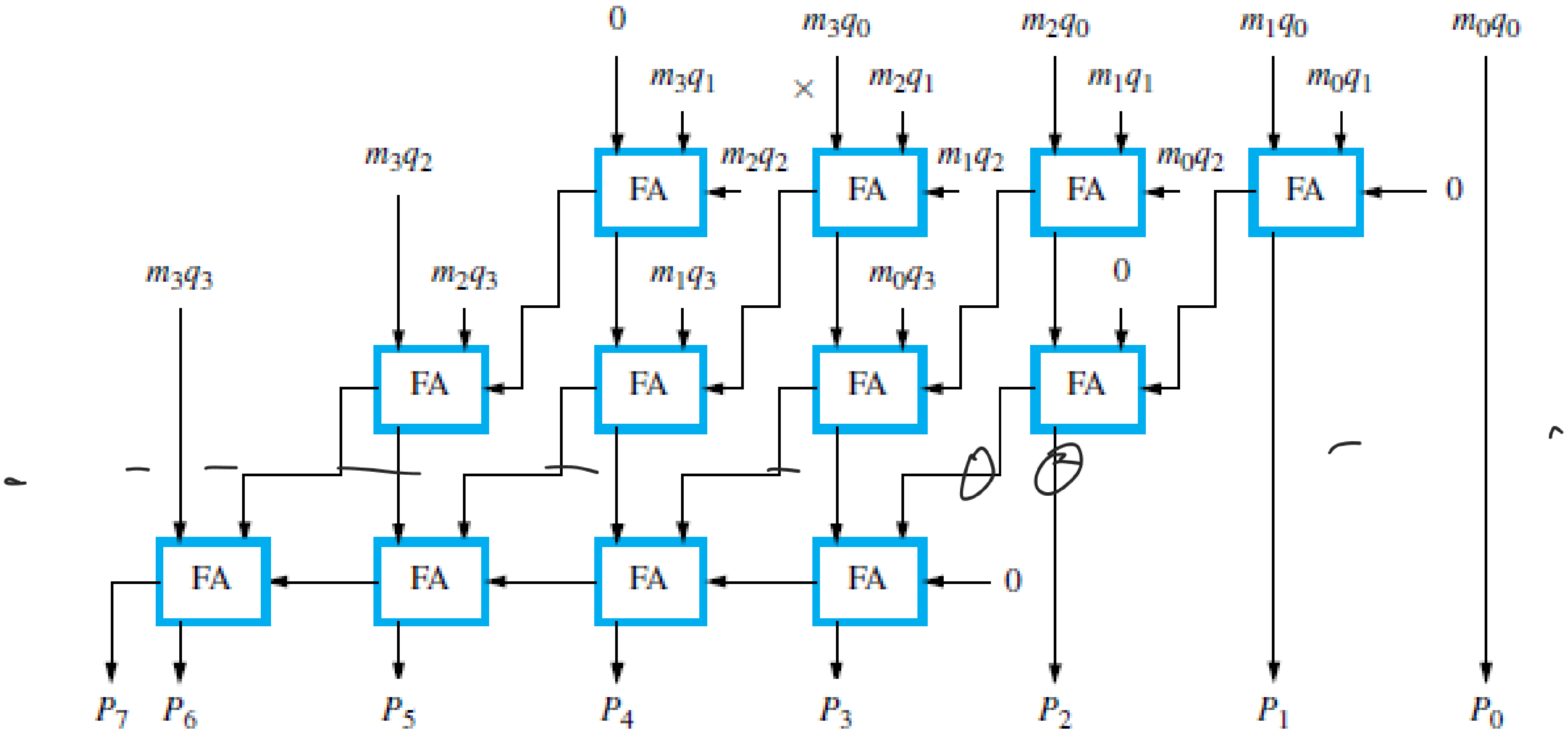


0
m_3q_0
m_2q_0
m_1q_0
m_0q_0
m_3q_1
m_2q_1
m_1q_1
m_0q_1
FA
FA
FA
FA
0
m_3q_2
m_2q_2
m_1q_2
m_0q_2
FA
FA
FA
FA
0
m_3q_3
m_2q_3
m_1q_3
m_0q_3
FA
FA
FA
FA
0
P_7
P_6
P_5
P_4
P_3
P_2
P_1
P_0

Carry Save Adder

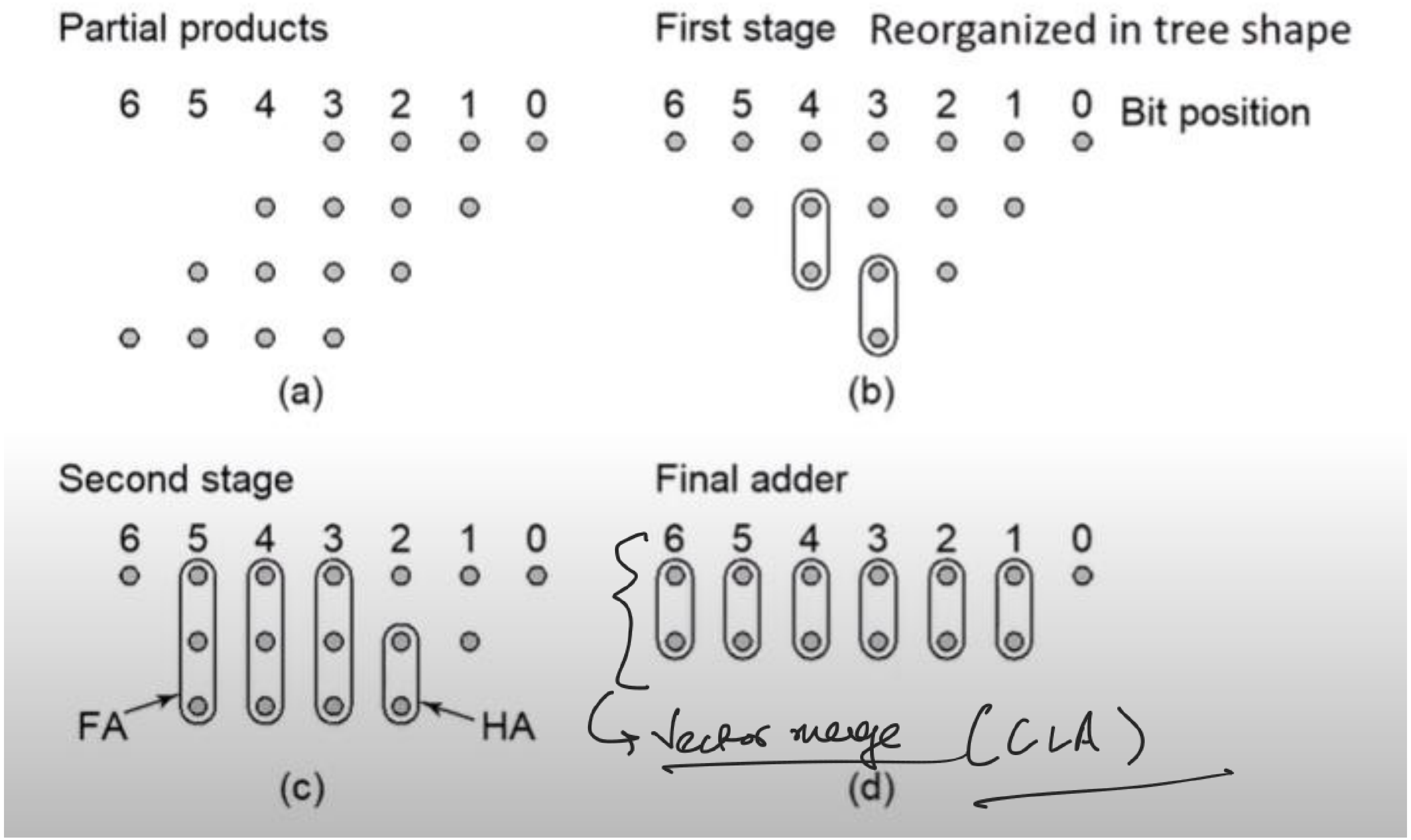


Timing analysis



Overflow

Wallace Tree Multiplier



x
y

$x_3 \quad x_2 \quad x_1 \quad x_0$

$y_3 \quad y_2 \quad y_1 \quad y_0$

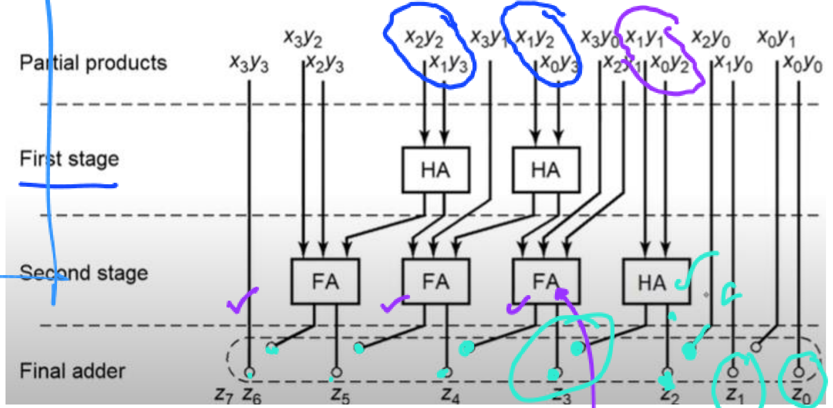
Sum of Subscripts is a way to check

1τ

2τ

2τ

			x_3y_0	x_2y_0	x_1y_0	x_0y_0	PP_0
		x_3y_1	x_2y_1	x_1y_1	x_0y_1		PP_1
	x_3y_2	x_2y_2	x_1y_2	x_0y_2			PP_2
x_3y_3	x_2y_3	x_1y_3	x_0y_3				PP_3



CLA

$\boxed{6\tau}$ 5τ $\boxed{11\tau}$

1011
0111

n-bit addition

$\boxed{n = 4}$

$2\left\lceil \frac{n}{4} \right\rceil + 2$

$2\left\lceil \frac{2n}{4} \right\rceil + 2$

$2 + \frac{8}{4} \to 2$
$= 6\tau$.

x_3y_3	x_3y_2	x_3y_1	x_3y_0	x_1y_n	x_1y_0	x_0y_0
	x_2y_3	$\boxed{x_2y_2}$	x_2y_1	$\boxed{x_1y_1}$	x_0y_1	
		$\boxed{x_1y_3}$	$\boxed{x_1y_2}$	$\boxed{x_0y_2}$		
			$\boxed{x_0y_3}$			

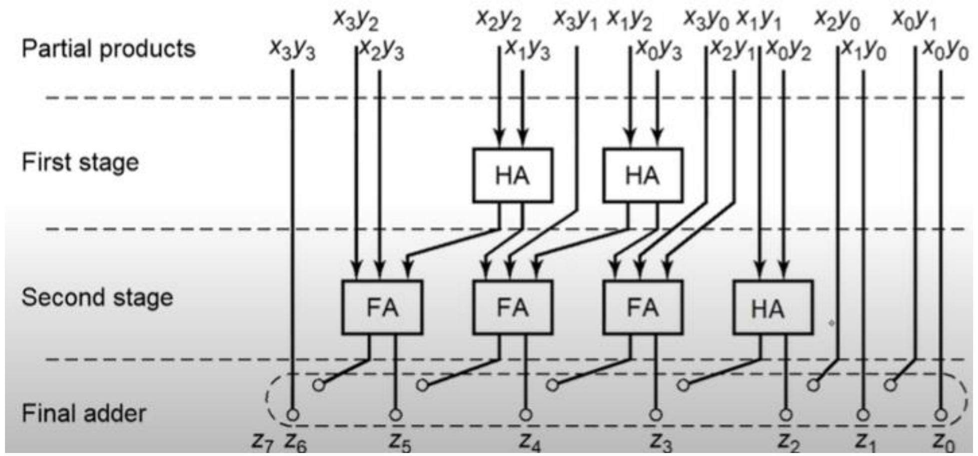
CLA

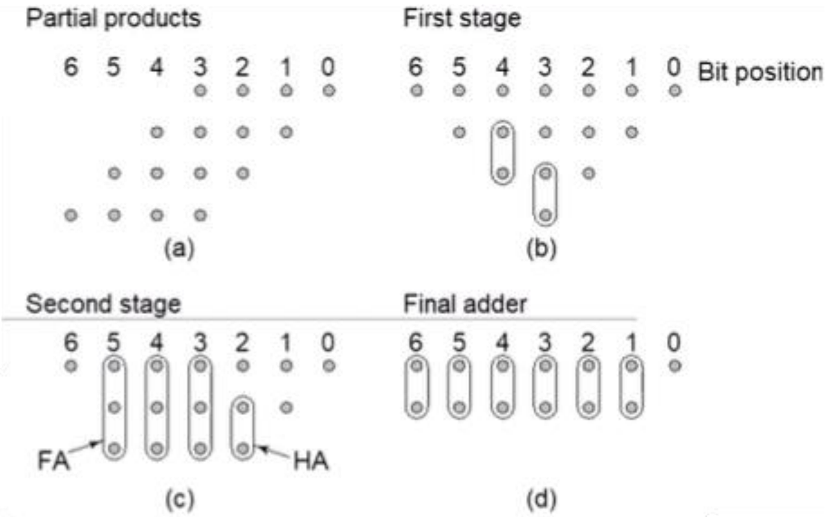
S S z_0

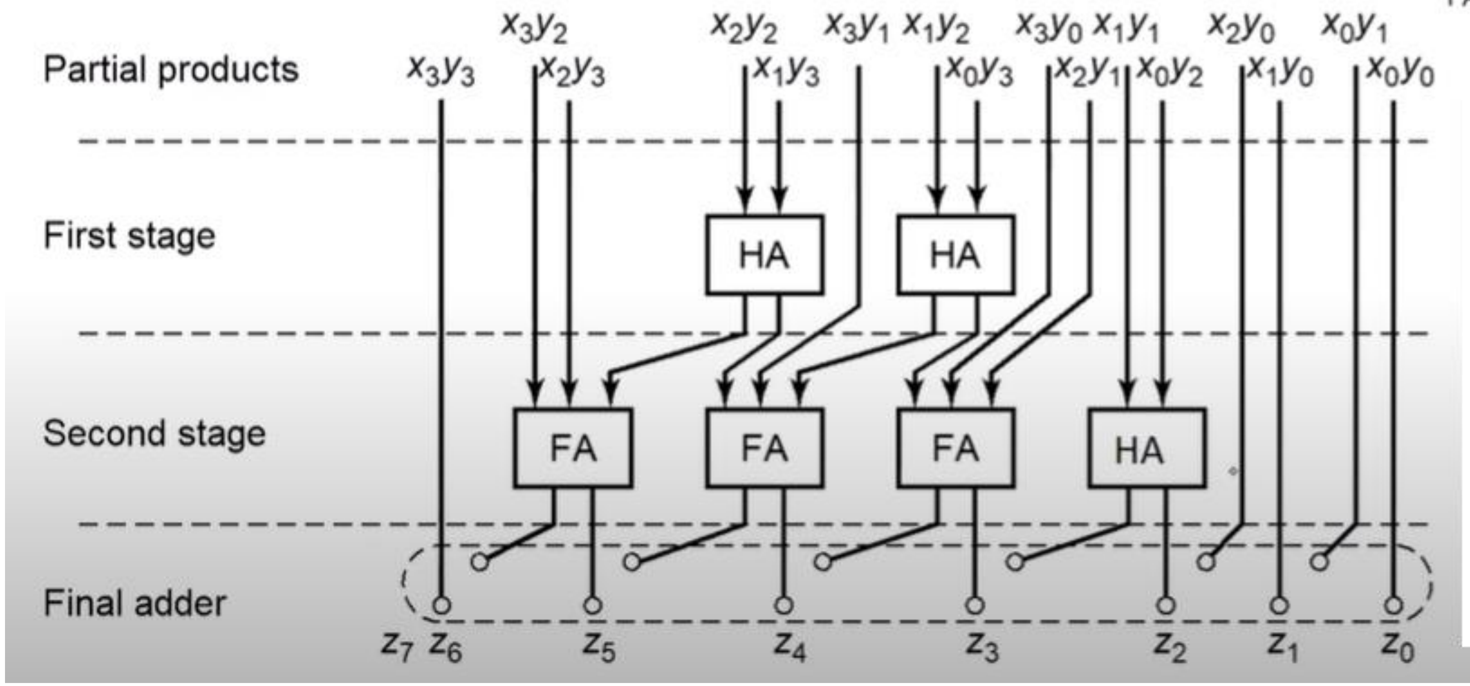
S

(TO BE COMPLETED)

[Figs: To be acknowledged (Textbook).]




Partial products
First stage
Bit position
6 5 4 3 2 1 0
(a)
(b)
Second stage
Final adder
FA
HA
(c)
(d)


Partial products
First stage
Second stage
Final adder
x_3y_3
x_3y_2
x_2y_3
x_2y_2
x_1y_3
x_3y_1
x_1y_2
x_0y_3
x_3y_0
x_2y_1
x_1y_1
x_0y_2
x_2y_0
x_1y_0
x_0y_1
x_0y_0
HA
HA
FA
FA
FA
HA
z_7 z_6 z_5 z_4 z_3 z_2 z_1 z_0

Summand Addition Tree using 3-2 Reducers

Hamacher

						1	0	1	1	0	1	(45)	M	
					×	1	1	1	1	1	1	(63)	Q	
						1	0	1	1	0	1		A	PP_0
					1	0	1	1	0	1			B	PP_1
				1	0	1	1	0	1				C	
			1	0	1	1	0	1					D	
		1	0	1	1	0	1						E	
	1	0	1	1	0	1							F	PP_i
1	0	1	1	0	0	0	1	0	0	1	1	(2,835)	Product	

Summands →

3-to-2 reduction

$(n-2)$

Summand Addition Tree using 3-2 Reducers.

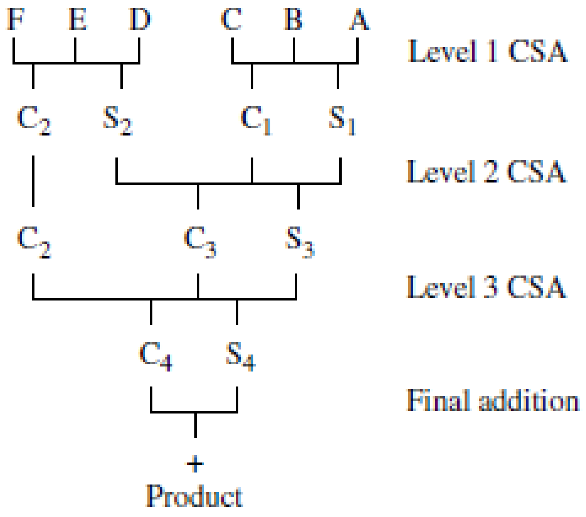
						1	0	1	1	0	1	(45)	M
					×	1	1	1	1	1	1	(63)	Q
						1	0	1	1	0	1		A
					1	0	1	1	0	1			B
				1	0	1	1	0	1				C
			1	0	1	1	0	1					D
		1	0	1	1	0	1						E
	1	0	1	1	0	1							F
1	0	1	1	0	0	0	1	0	0	1	1	(2,835)	Product

							1	0	1	1	0	1	M
						×	1	1	1	1	1	1	Q
							1	0	1	1	0	1	A
						1	0	1	1	0	1		B
					1	0	1	1	0	1			C
					1	1	0	0	0	0	1	1	S_1
				0	0	1	1	1	1	0	0		C_1
				1	0	1	1	0	1				D
			1	0	1	1	0	1					E
		1	0	1	1	0	1						F
		1	1	0	0	0	0	1	1				S_2
	0	0	1	1	1	1	0	0					C_2
					1	1	0	0	0	0	1	1	S_1
				0	0	1	1	1	1	0	0		C_1
		1	1	0	0	0	0	1	1				S_2
		1	1	0	1	0	1	0	0	0	1	1	S_3
	0	0	0	0	1	0	1	1	0	0	0		C_3
	0	0	1	1	1	1	0	0					C_2
	0	1	0	1	1	1	0	1	0	0	1	1	S_4
+	0	1	0	1	0	1	0	0	0	0	0		C_4
	1	0	1	1	0	0	0	1	0	0	1	1	Product

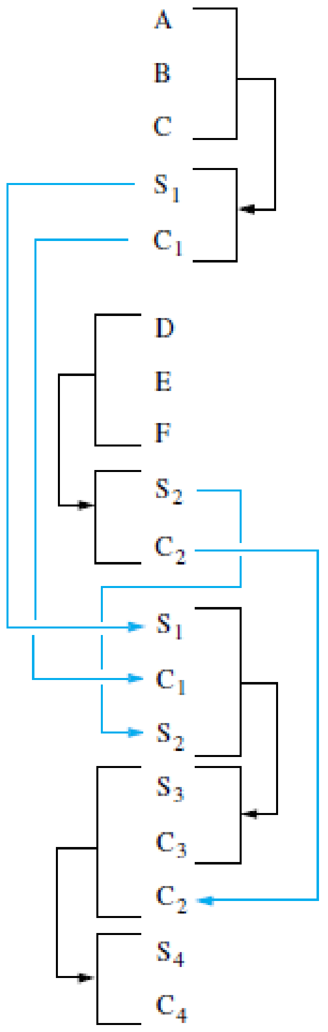
FA

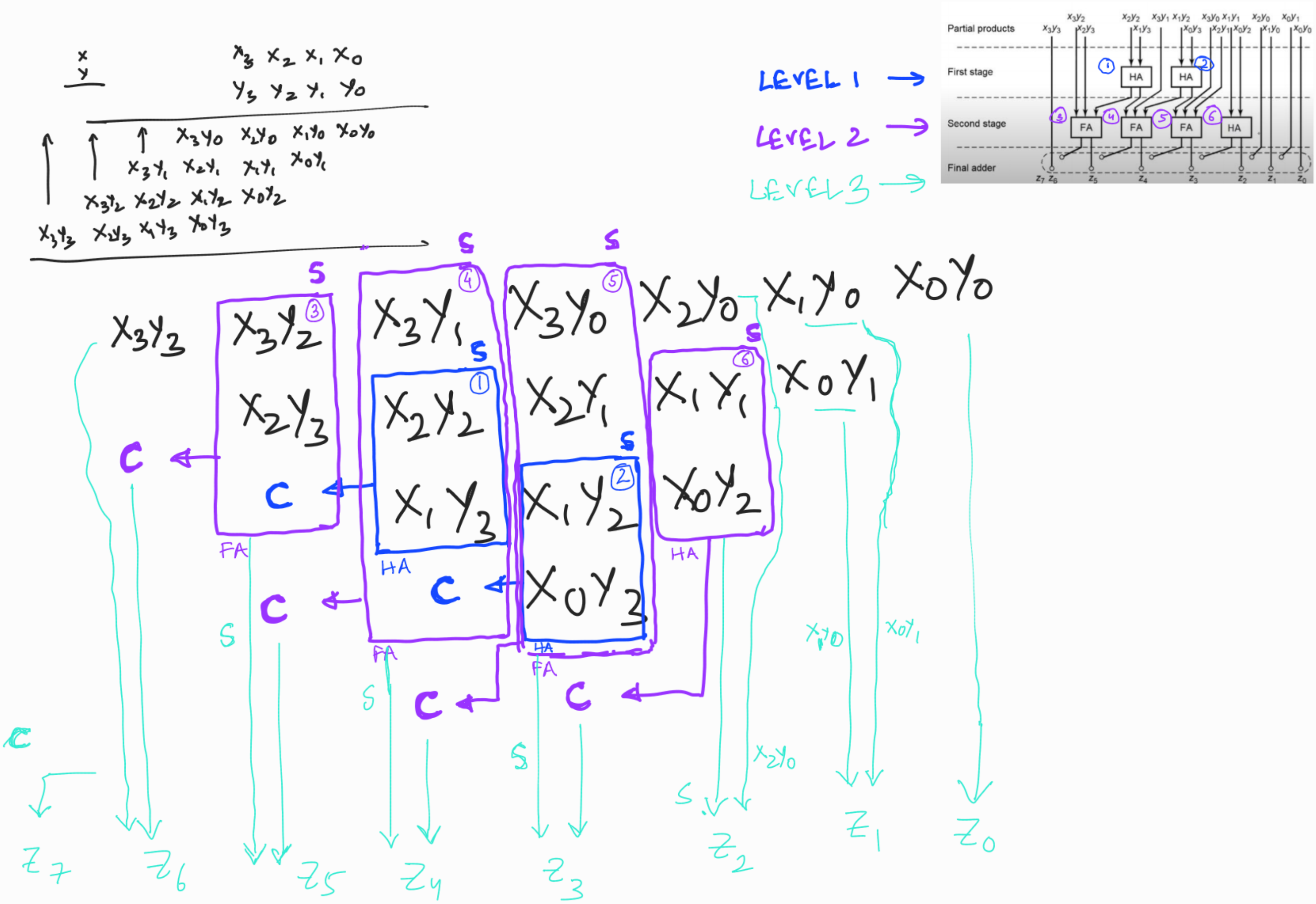
CLA

						1	0	1	1	0	1	(45)	M
					×	1	1	1	1	1	1	(63)	Q
						1	0	1	1	0	1		A
					1	0	1	1	0	1			B
				1	0	1	1	0	1				C
			1	0	1	1	0	1					D
		1	0	1	1	0	1						E
	1	0	1	1	0	1							F
1	0	1	1	0	0	0	1	0	0	1	1	(2,835)	Product



						1	0	1	1	0	1	M
					×	1	1	1	1	1	1	Q
						1	0	1	1	0	1	
					1	0	1	1	0	1		
				1	0	1	1	0	1			
				1	1	0	0	0	0	1	1	
			0	0	1	1	1	1	0	0		
			1	0	1	1	0	1				
		1	0	1	1	0	1					
	1	0	1	1	0	1						
	1	1	0	0	0	0	1	1				
0	0	1	1	1	1	0	0					
				1	1	0	0	0	0	1	1	
			0	0	1	1	1	1	0	0		
	1	1	0	0	0	0	1	1				
	1	1	0	1	0	1	0	0	0	1	1	
0	0	0	0	1	0	1	1	0	0	0		
0	0	1	1	1	1	0	0					
0	1	0	1	1	1	0	1	0	0	1	1	
+ 0	1	0	1	0	1	0	0	0	0	0		
1	0	1	1	0	0	0	1	0	0	1	1	Product





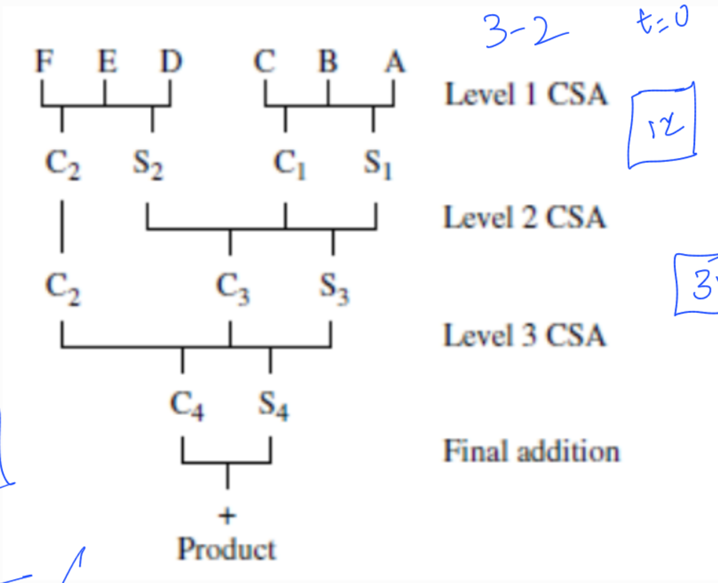
							1	0	1	1	0	1	(45)	M
						×	1	1	1	1	1	1	(63)	Q
PP_0							1	0	1	1	0	1		A
PP_1						1	0	1	1	0	1			B
PP_2					1	0	1	1	0	1				C
PP_3				1	0	1	1	0	1					D
PP_4			1	0	1	1	0	1						E
PP_5		1	0	1	1	0	1							F
	1	0	1	1	0	0	0	1	0	0	1	1	(2,835)	Product

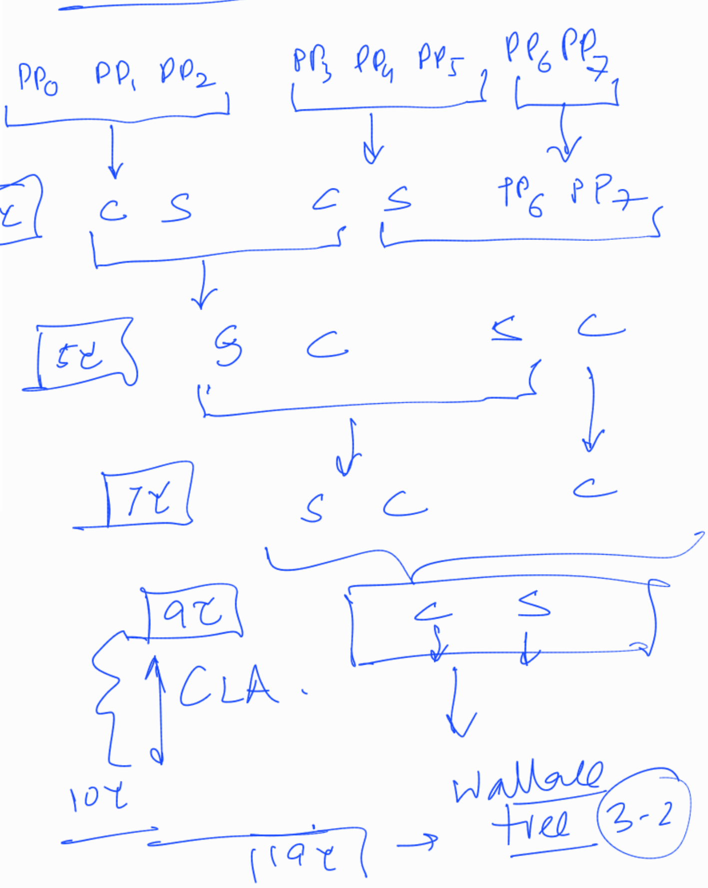
S C PP_3 S C PP_4 S C PP_5 S C CLA

3-to-2 Summand addition ~~tree~~

3-2 $t=0$

8-bit multiplier





CAM

$6n-7$

41τ

$$\left(2\times\left\lceil\frac{n}{4}\right\rceil+2\right)\tau$$

CSA

23τ

$$2\times\frac{16}{4}+2 = 10$$

Analysis for n-bits (k-summands) 3-to-2 Wallace Tree

$$\boxed{k\left(\frac{2}{3}\right)^{L}=2}$$

$$\log_2 k + L\log_2\left(\frac{2}{3}\right) = \log_2 2$$

$$\log_2 k + L(1-\log_2 3) = 1.$$

$$\log_2 k + L(1-1.59) = 1$$

$$L = (1-\log_2 k)/(-.59)$$

$$\rightarrow \; = \boxed{1.7\log_2 k - 1.7}$$

4-2

$$\boxed{k\left(\frac{1}{2}\right)^{L}=2.}$$

$$\boxed{L=\log_2 k - 1}$$

-7×8

-2^5 $\quad$ -2^4

multiplier

$\boxed{\text{multiplicant}}$

$\boxed{10110}$

00110

$\rightarrow$ negative

$\boxed{01001}$

+1

1010

00000

1111 $\boxed{1011}$ 0

1110110

00000

$\boxed{\text{Booth's Algorithm}}$

++

+−

−+

−−

$\ominus \times \oplus$

$\oplus \times \ominus$

−