Integer Multiplication

Mohammad-Reza Movahedin

Computer Architecture

1393/11/27

Decimal Multiplication

$$\begin{array}{ccccc} & 1 & 3 & 2 & 1 \\ \times & 2 & 1 & 3 & 2 \\ \hline \\ \\ \\ \hline \end{array}$$

Decimal Multiplication

$$\begin{array}{rrrrr} & 1 & 3 & 2 & 1 \\ \times & 2 & 1 & 3 & 2 \\ \hline & & & & 2 \\ & & & & \\ & & & & \\ \hline \end{array}$$

Decimal Multiplication

$$\begin{array}{rrrrr}
 & 1 & 3 & 2 & 1 \\
\times & 2 & 1 & 3 & 2 \\
\hline
 & & & 4 & 2 \\
 & & & & \\
 & & & & \\
\hline
\end{array}$$

Decimal Multiplication

$$\begin{array}{rrrrr} & 1 & 3 & 2 & 1 \\ \times & 2 & 1 & 3 & 2 \\ \hline & & 6 & 4 & 2 \\ \\ \\ \hline \end{array}$$

Decimal Multiplication

$$
\begin{array}{ccccc}
 & 1 & 3 & 2 & 1 \\
\times & 2 & 1 & 3 & 2 \\
\hline
 & 2 & 6 & 4 & 2 \\
 & & & & \\
 & & & & \\
\hline
\end{array}
$$

We do need a Multiplication Table to generate Partial Products

Decimal Multiplication

$$\begin{array}{ccccc}
 & 1 & 3 & 2 & 1 \\
\times & 2 & 1 & 3 & 2 \\
\hline
 & 2 & 6 & 4 & 2 \\
3 & 9 & 6 & 3 & \\
 & & & & \\
\hline
\end{array}$$

Decimal Multiplication

$$\begin{array}{cccccc} & & 1 & 3 & 2 & 1 \\ & \times & 2 & 1 & 3 & 2 \\ \hline & & 2 & 6 & 4 & 2 \\ & 3 & 9 & 6 & 3 & \\ 1 & 3 & 2 & 1 & & \\ & & & & & \\ \hline \end{array}$$

Decimal Multiplication

$$
\begin{array}{ccccccc}
 & & & 1 & 3 & 2 & 1 \\
 & & \times & 2 & 1 & 3 & 2 \\
\hline
 & & & 2 & 6 & 4 & 2 \\
 & & 3 & 9 & 6 & 3 & \\
 & 1 & 3 & 2 & 1 & & \\
2 & 6 & 4 & 2 & & & \\
\hline
\end{array}
$$

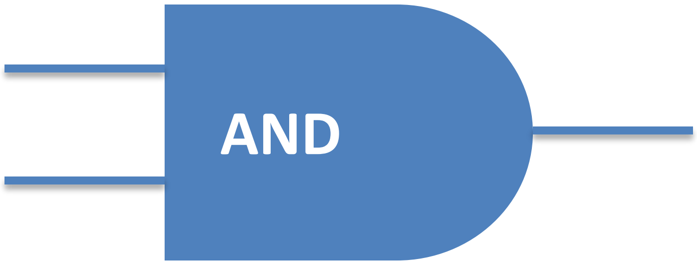
Decimal Multiplication

$$\begin{array}{ccccccc}
 & & & 1 & 3 & 2 & 1 \\
 & & \times & 2 & 1 & 3 & 2 \\
\hline
 & & & 2 & 6 & 4 & 2 \\
 & & 3 & 9 & 6 & 3 & \\
 & 1 & 3 & 2 & 1 & & \\
2 & 6 & 4 & 2 & & & \\
\hline
2 & 8 & 1 & 6 & 3 & 7 & 2
\end{array}$$

Binary Multiplication Table

×	0	1
0	0	0
1	0	1

Note: despite decimal table, each cell is only one digit



Unsigned Integer Multiplication

A:		**1**	**0**	**1**	**1**
B:	×	**1**	**1**	**0**	**1**

Unsigned Integer Multiplication

A:		1	0	1	1
B:	×	1	1	0	1
		1	0	1	1

Unsigned Integer Multiplication

A:		**1**	**0**	**1**	**1**
B:	×	**1**	**1**	**0**	**1**
		1	**0**	**1**	**1**
	0	**0**	**0**	**0**	

Unsigned Integer Multiplication

$$
\begin{array}{lcccccc}
\textbf{A:} & & & 1 & 0 & 1 & 1 \\
\textbf{B:} & & \times & 1 & 1 & 0 & 1 \\
\hline
 & & & 1 & 0 & 1 & 1 \\
 & & 0 & 0 & 0 & 0 & \\
 & 1 & 0 & 1 & 1 & & \\
 & & & & & & \\
\hline
\end{array}
$$

Unsigned Integer Multiplication

$$
\begin{array}{lccccccc}
\mathbf{A:} & & & & 1 & 0 & 1 & 1 \\
\mathbf{B:} & & & \times & 1 & 1 & 0 & 1 \\
\hline
 & & & & 1 & 0 & 1 & 1 \\
 & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 0 & 1 & 1 & & \\
 & 1 & 0 & 1 & 1 & & & \\
\hline
\end{array}
$$

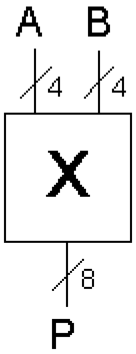
Unsigned Integer Multiplication

$$
\begin{array}{cccccccc}
\textbf{A:} & & & & 1 & 0 & 1 & 1 \\
\textbf{B:} & & & \times & 1 & 1 & 0 & 1 \\
\hline
 & & & & 1 & 0 & 1 & 1 \\
 & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 0 & 1 & 1 & & \\
 & 1 & 0 & 1 & 1 & & & \\
\hline
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}
$$

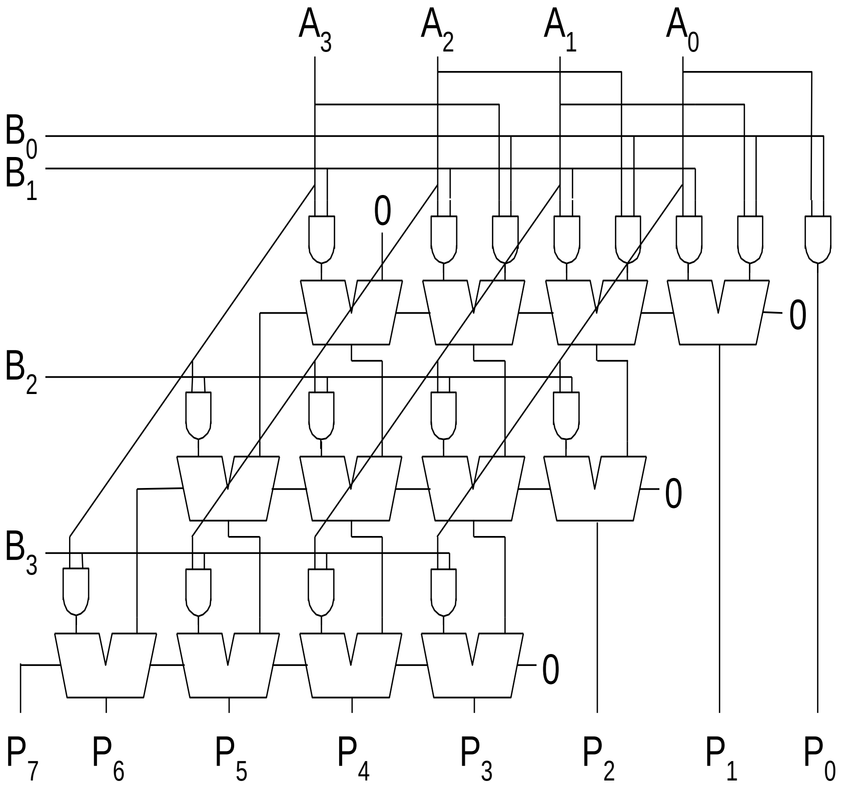
Integer Multiplication Procedure

- Generate Partial Products
  - $PP_i = B_i$? A : 0
- Position each PP_i at 2^i location
  - Put *i* 0's in front of PP_i
- Sum Partial Products
  - Be aware of addition overflow
- The summation can be done in two ways:
  - Combinational aka Parallel
  - Sequential aka Serial

Parallel Array Multiplier

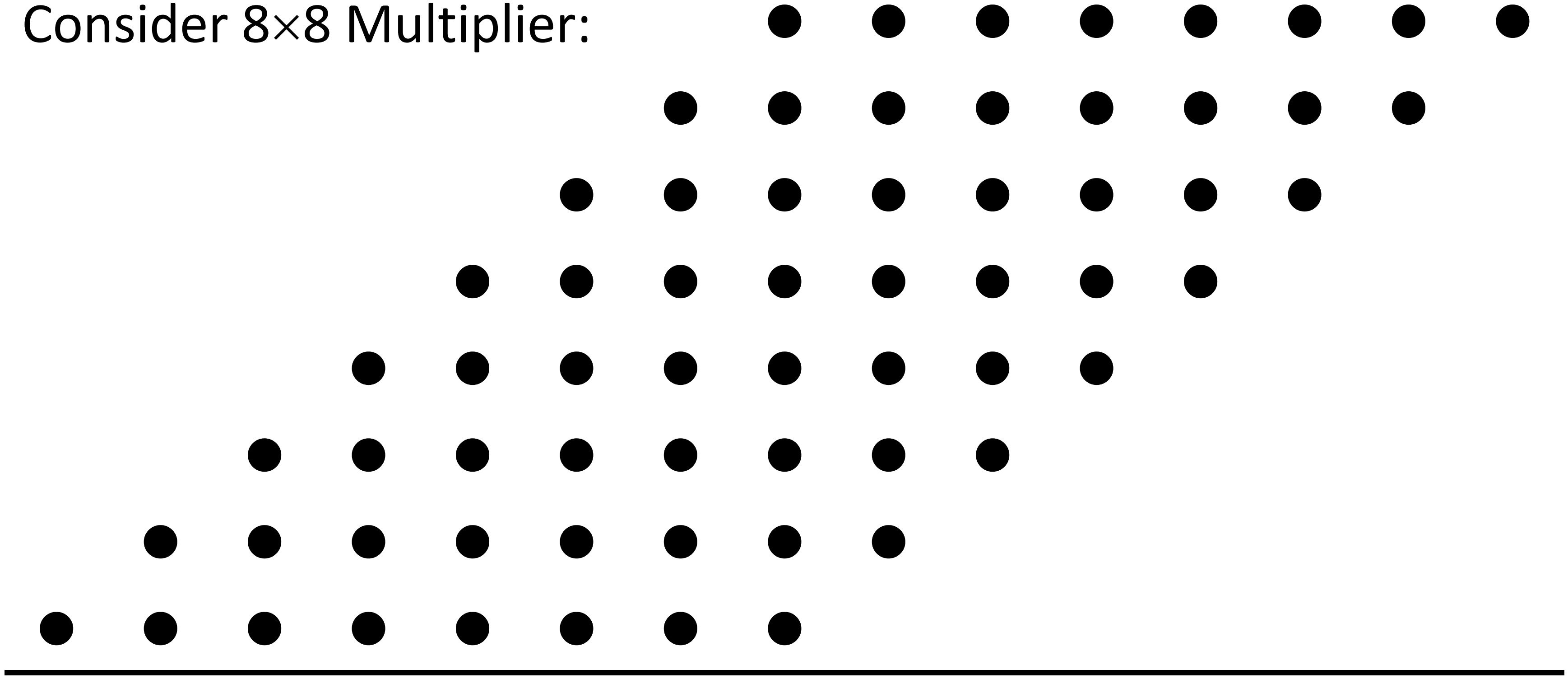


$$
\begin{array}{ccccccccc}
 & & & & & A_3 & A_2 & A_1 & A_0 \\
 & & & x & & B_3 & B_2 & B_1 & B_0 \\
\hline
 & & & & & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\
 & & & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & \\
 & & & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & & \\
+ & & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 & & & \\
\hline
 & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$



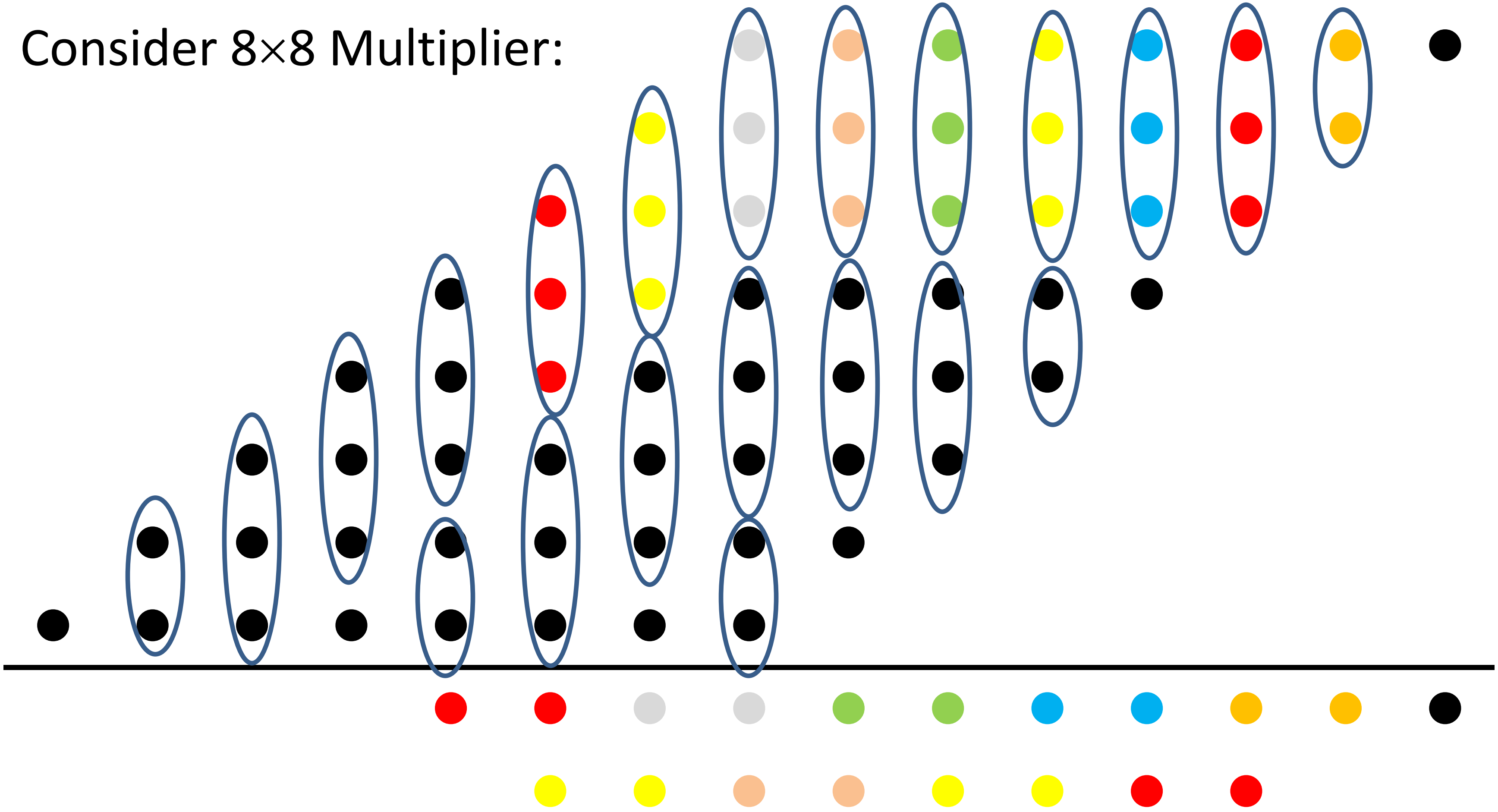
Wallace Tree Multiplier: Compression by Half and Full Adders

Consider 8×8 Multiplier:



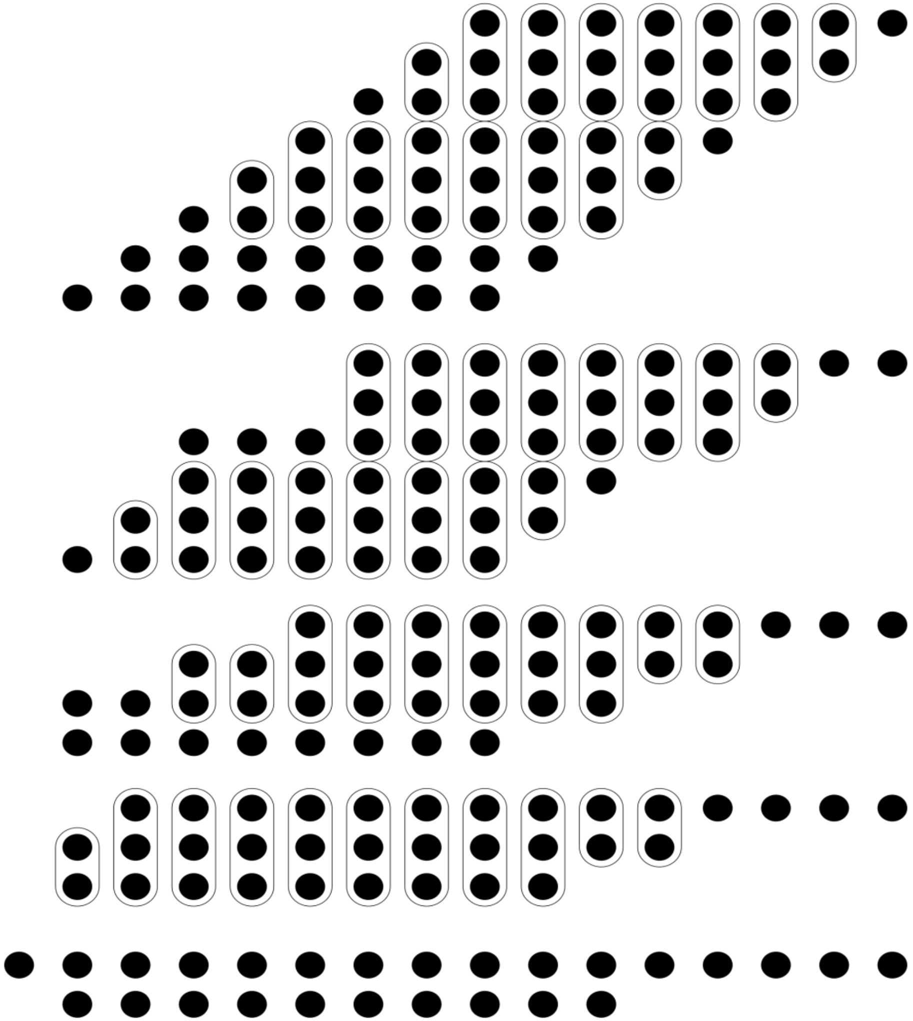
Wallace Tree Multiplier: Compression by Half and Full Adders

Consider 8×8 Multiplier:



. . . to be continued

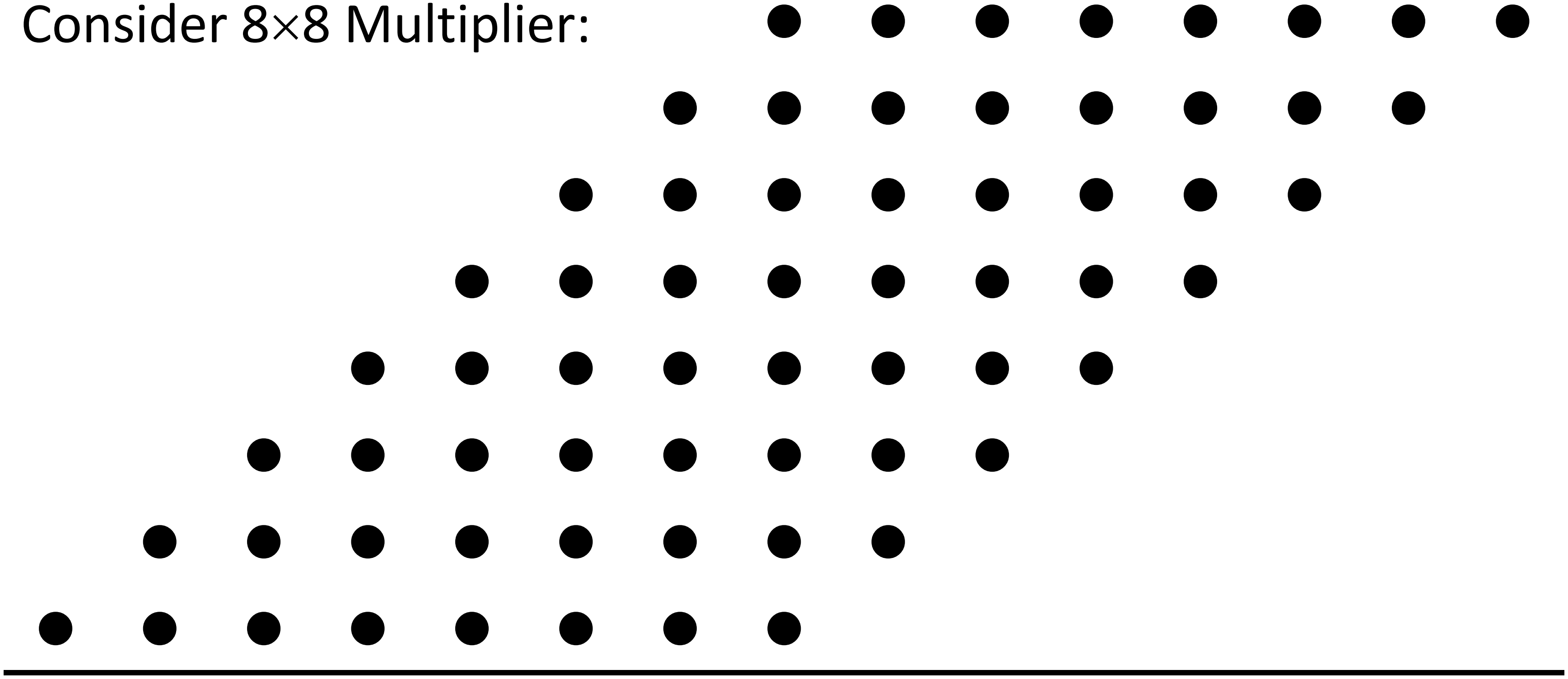
Wallace Tree Multiplier



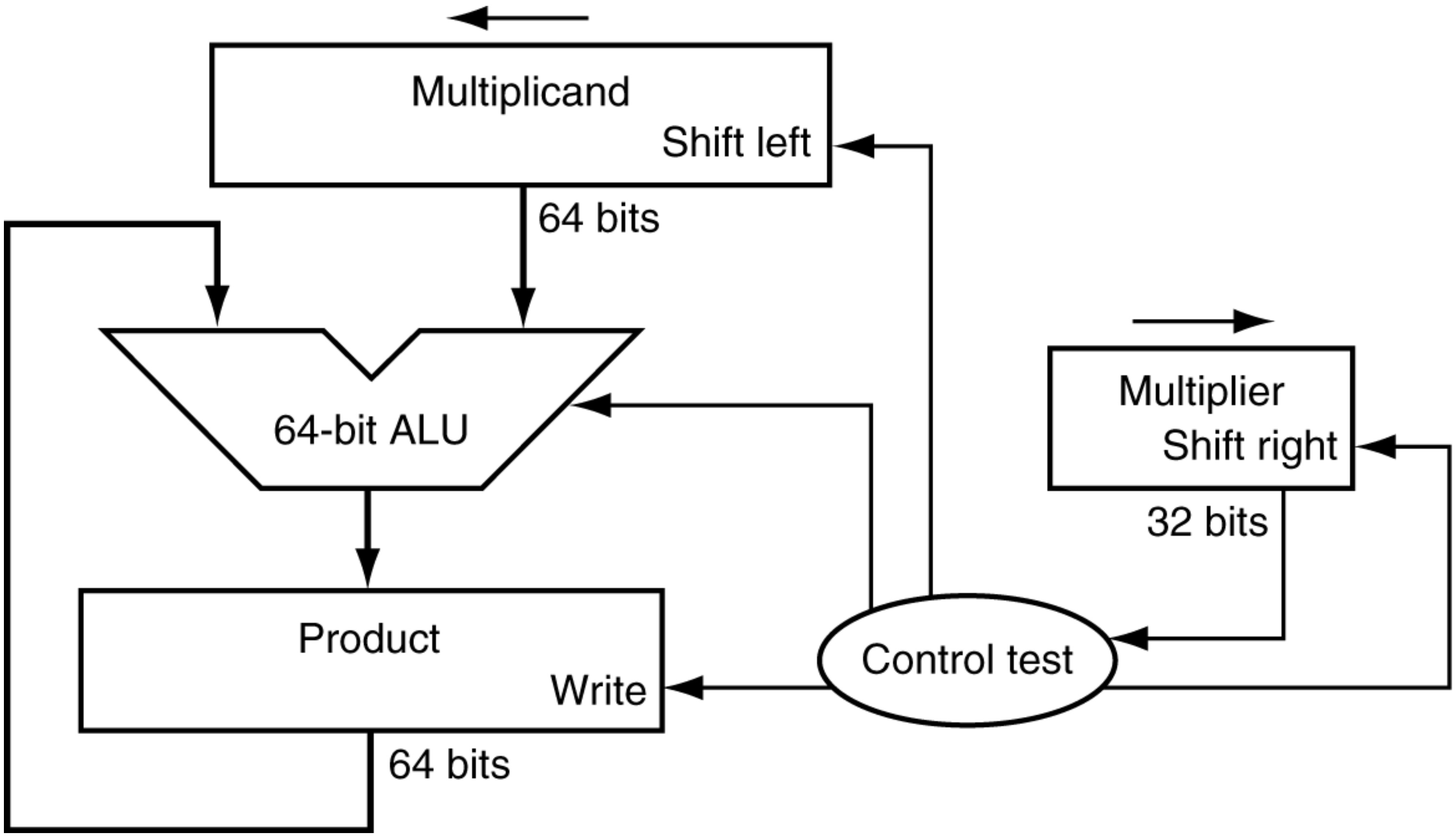
Source: Wikipedia

Serial Multipliers: Summation of partial products in several clocks

Consider 8×8 Multiplier:

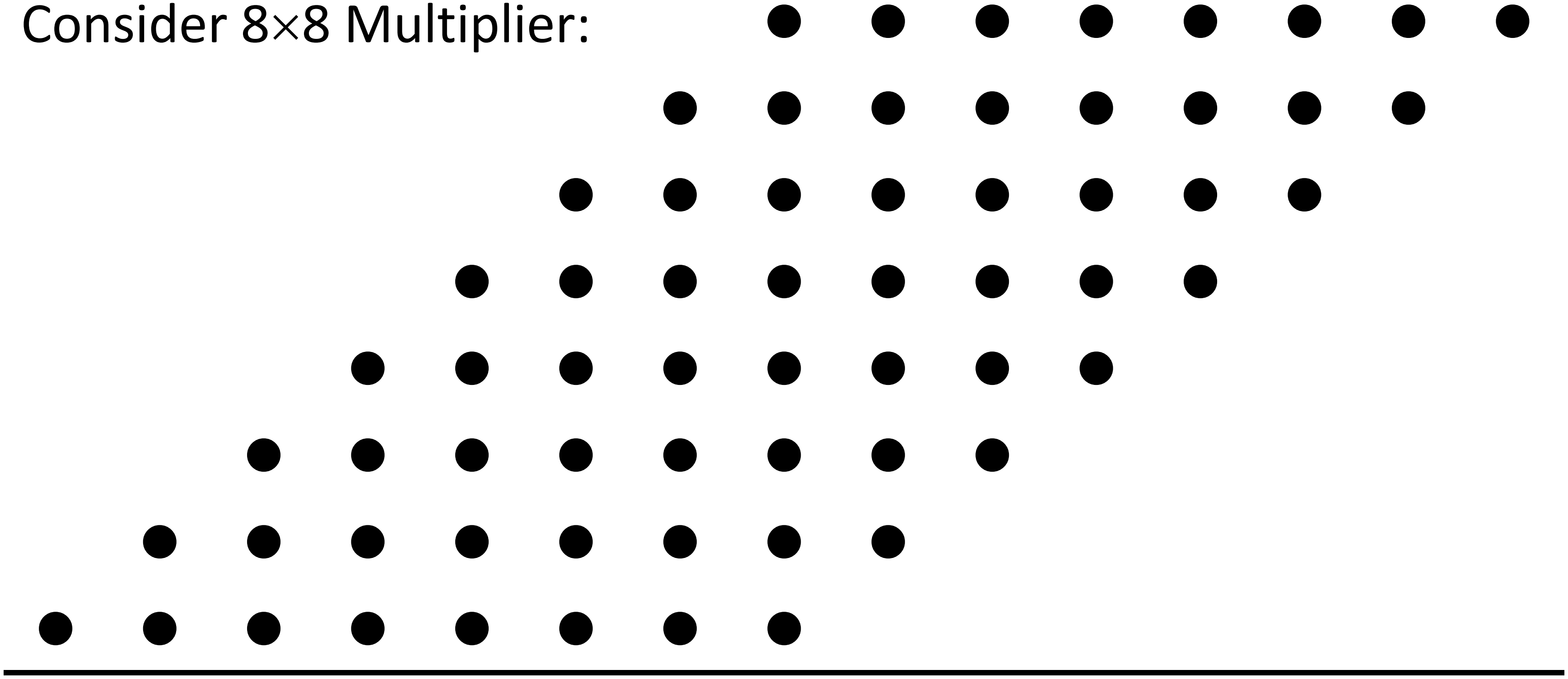


Serial Multiplier



Serial Multipliers: summation of partial products in several clocks

Consider 8×8 Multiplier:

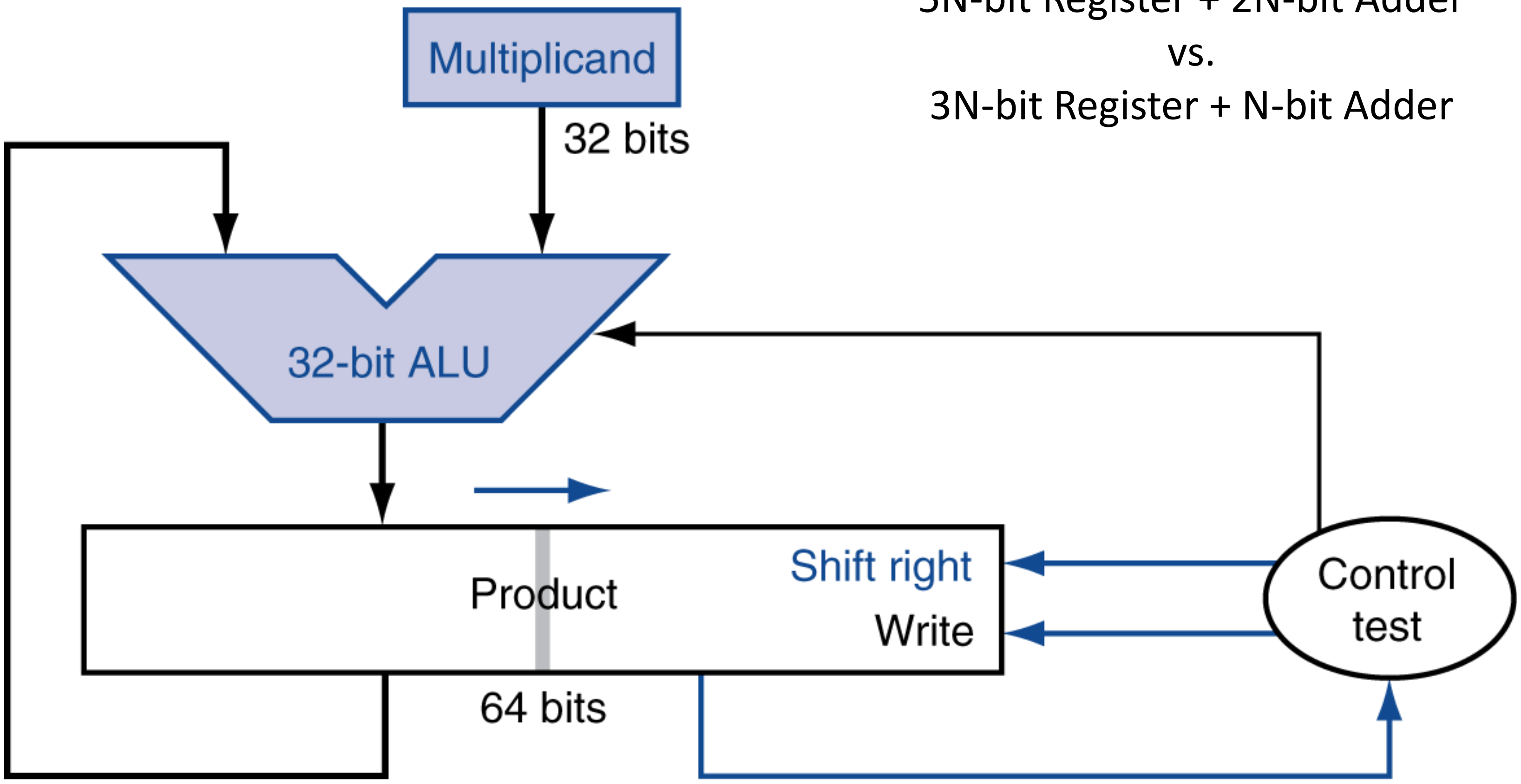


Optimized Serial Multiplier

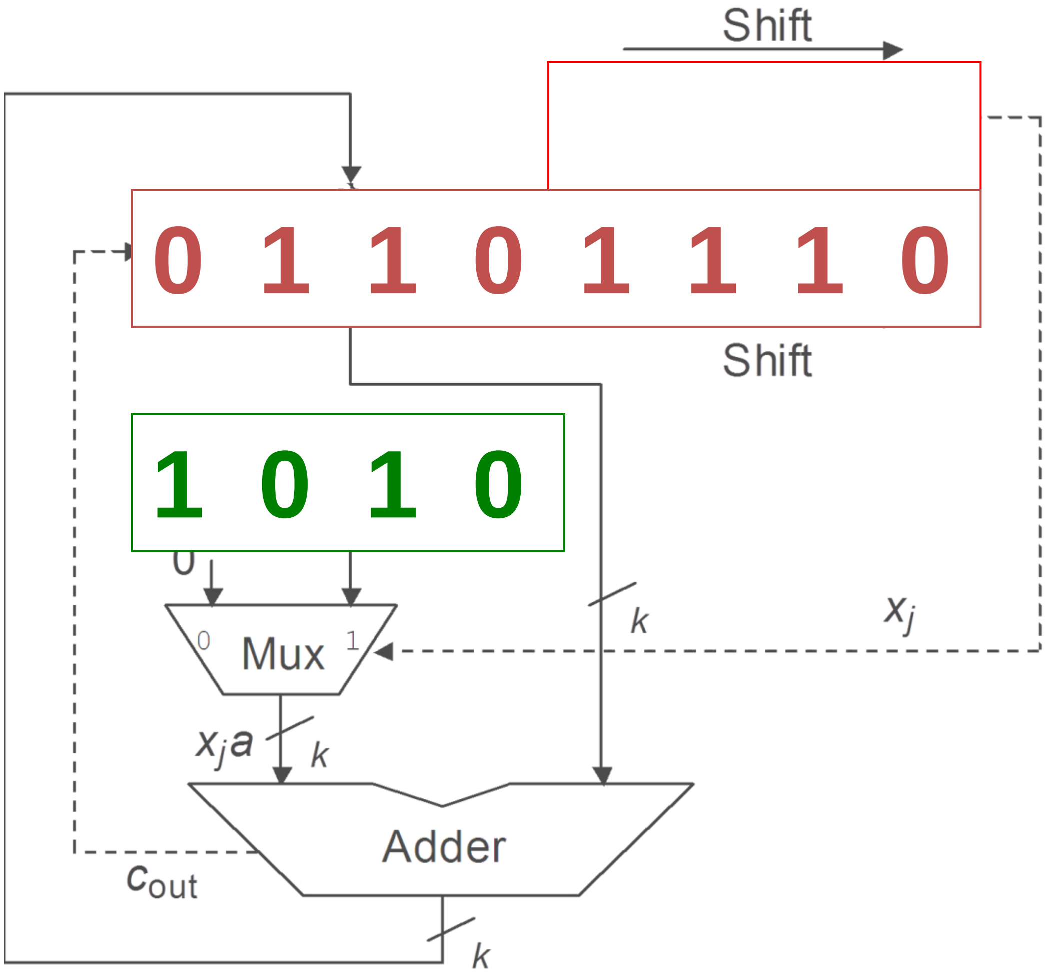
5N-bit Register + 2N-bit Adder

vs.

3N-bit Register + N-bit Adder



Sequential Multiplier: Example



$(11)_{ten}$

$(110)_{ten}$

$(10)_{ten}$

Signed Multiplication: When only A is signed

$$\begin{array}{lccccccc}
\textbf{A:} & & & & 1 & 0 & 1 & 1 \\
\textbf{B:} & & & \times & 1 & 1 & 0 & 1 \\
\hline
 & & & & 1 & 0 & 1 & 1 \\
 & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 0 & 1 & 1 & & \\
 & 1 & 0 & 1 & 1 & & & \\
\hline
\end{array}$$

Signed Multiplication: When only A is signed

A:				1	0	1	1
B:			×	1	1	0	1
	1	1	1	1	0	1	1
	0	0	0	0	0	0	
	1	1	0	1	1		
	1	0	1	1			

Signed Multiplication: When only A is signed

A:				1	0	1	1
B:			×	1	1	0	1
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	
1	1	1	0	1	1		
1	1	0	1	1			

Signed Multiplication: When only A is signed

A:				1	0	1	1
B:			×	1	1	0	1
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	
1	1	1	0	1	1		
1	1	0	1	1			
1	0	1	1	1	1	1	1

Signed Multiplication: When A and B both are signed

A:				**1**	**0**	**1**	**1**
B:			×	**1**	**1**	**0**	**1**
				1	**0**	**1**	**1**
			0	**0**	**0**	**0**	
		1	**0**	**1**	**1**		
	1	**0**	**1**	**1**			

Signed Multiplication: When A and B both are signed

A:					**1**	**0**	**1**	**1**
B:				×	**1**	**1**	**0**	**1**
	1	**1**	**1**	**1**	**1**	**0**	**1**	**1**
	0	**0**	**0**	**0**	**0**	**0**	**0**	
	1	**1**	**1**	**0**	**1**	**1**		
	1	**1**	**0**	**1**	**1**			

Signed Multiplication: When A and B both are signed

A:					1	0	1	1
B:				×	1	1	0	1
+	1	1	1	1	1	0	1	1
+	0	0	0	0	0	0	0	
+	1	1	1	0	1	1		
-	1	1	0	1	1			

Signed Multiplication:
When A and B both are signed

A:					1	0	1	1
B:				×	1	1	0	1
+	1	1	1	1	1	0	1	1
+	0	0	0	0	0	0	0	
+	1	1	1	0	1	1		
+	0	0	1	0	1			

Signed Multiplication:
When A and B both are signed

A:					1	0	1	1
B:				×	1	1	0	1
+	1	1	1	1	1	0	1	1
+	0	0	0	0	0	0	0	
+	1	1	1	0	1	1		
+	0	0	1	0	1			
	0	0	0	0	1	1	1	1

What to do with Carry Out?
Be aware of Signed Addition Overflow!

Signed Serial Multiplier

Radix-4 Multiplication

- Let’s look at two bits of Multiplier

$B_{2i+1}B_{2i}$	$PP_{(2i+1,\ 2i)}$
00	0
01	A
10	2A
11	3A

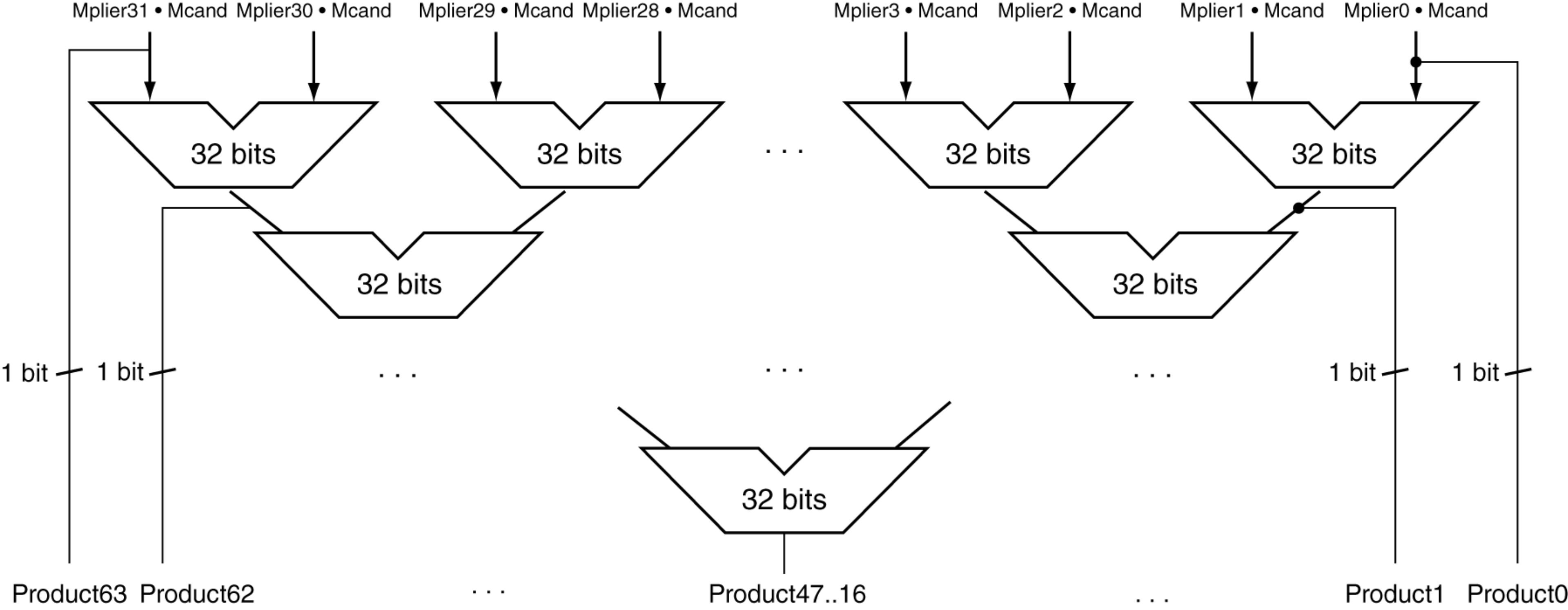
- It requires 3A calculation, not easy ☹

Radix-4 Multiplier: Booth Encoded

- Keep radix = 4, but digit set as: $\{-2, -1, 0, 1, 2\}$

$B_{2i+1}B_{2i}B_{2i-1}$	$PP_{(2i+1, 2i)}$
00_0	0
00_1	A
01_0	A
01_1	2A
10_0	-2A
10_1	-A
11_0	-A
11_1	0

Faster Parallel Multiplier, WRONG?


Mplier31 • Mcand
Mplier30 • Mcand
Mplier29 • Mcand
Mplier28 • Mcand
Mplier3 • Mcand
Mplier2 • Mcand
Mplier1 • Mcand
Mplier0 • Mcand
32 bits
32 bits
. . .
32 bits
32 bits
32 bits
32 bits
1 bit
1 bit
. . .
. . .
. . .
1 bit
1 bit
32 bits
Product63
Product62
. . .
Product47..16
. . .
Product1
Product0