

Binary Multiplication

Binary Multiplication

$$\begin{array}{rcccc} 0 & 1 & 0 & 1 & = & 5 & \leftarrow \text{Multiplicand} \\ 0 & 0 & 1 & 1 & = & 3 & \leftarrow \text{Multiplier} \\ \hline \end{array}$$

Binary Multiplication

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0101 \end{array}$$

Binary Multiplication

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0101 \\ 0101 \end{array}$$

Binary Multiplication

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0101 \\ 0101 \\ 0000 \\ \hline \end{array}$$

Binary Multiplication

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0101 \\ 0101 \\ 0000 \\ 0000 \\ 0000 \end{array}$$

Binary Multiplication

[illegible]

This is identical to grade-school multiplication except everything is binary...

The problem is that you must add n numbers in a column. Can we do them 2 at a time?

Binary Shift/Add Multiplication

Add

0	1	0	1
0	0	1	1
0	1	0	1

Binary Shift/Add Multiplication

Shift

0	1	0	1	
0	0	1	1	
0	0	1	0	1

Binary Shift/Add Multiplication

Add

$$\begin{array}{rcccc} 0 & 1 & 0 & 1 & \\ 0 & 0 & 1 & 1 & \\ \hline 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & \\ \hline 0 & 1 & 1 & 1 & 1 \end{array}$$

Binary Shift/Add Multiplication

Shift

0	1	0	1		
0	0	1	1		
0	0	1	0	1	
0	1	0	1		
0	0	1	1	1	1

Binary Shift/Add Multiplication

Add

$$\begin{array}{rcccccc} & 0 & 1 & 0 & 1 & & \\ & 0 & 0 & 1 & 1 & & \\ \hline & 0 & 0 & 1 & 0 & 1 & \\ & 0 & 1 & 0 & 1 & & \\ \hline & 0 & 0 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 0 & 0 & & \\ \hline & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

Binary Shift/Add Multiplication

Shift

0	1	0	1				
0	0	1	1				
0	0	1	0	1			
0	1	0	1				
0	0	1	1	1	1		
0	0	0	0				
0	0	0	1	1	1	1	1

Binary Shift/Add Multiplication

Add

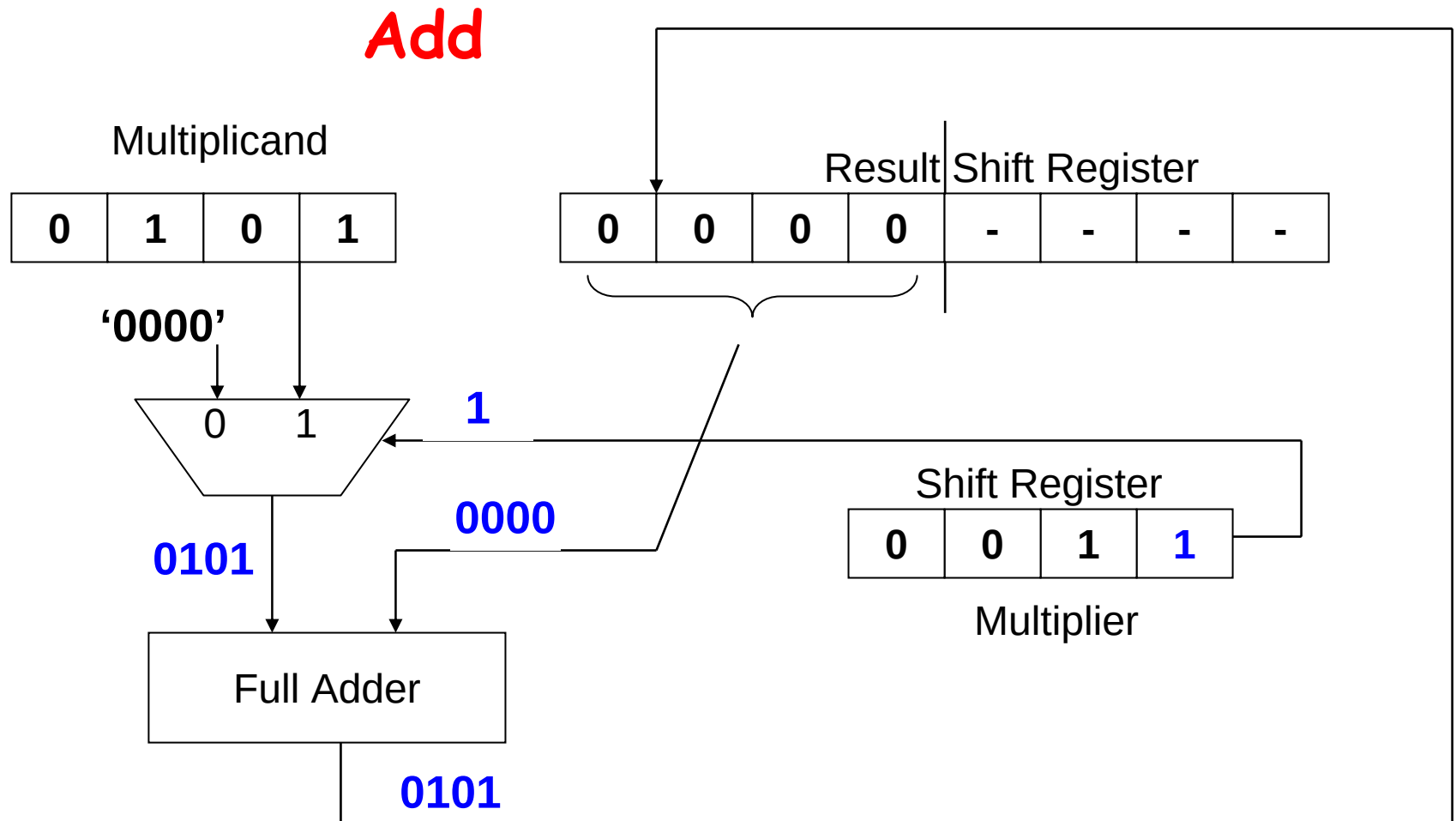
0	1	0	1			
0	0	1	1			
0	0	1	0	1		
0	1	0	1			
0	0	1	1	1	1	
0	0	0	0			
0	0	0	1	1	1	1
0	0	0	0			
0	0	0	1	1	1	1

Binary Shift/Add Multiplication

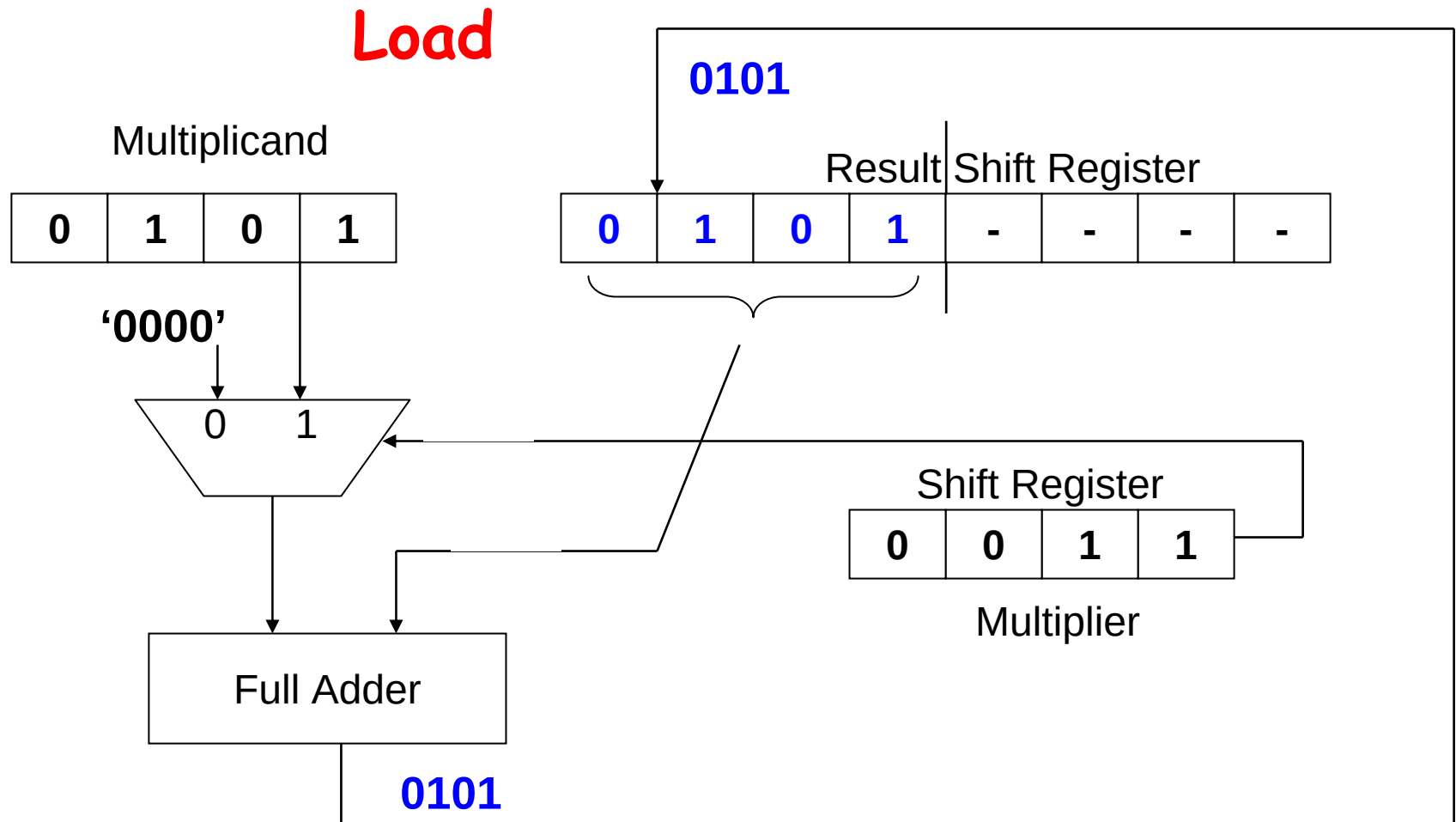
Shift

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 00101 \\ 0101 \\ \hline 001111 \\ 0000 \\ \hline 0001111 \\ 0000 \\ \hline 00001111 = 15_{10} \end{array}$$

Binary Shift/Add Multiplication

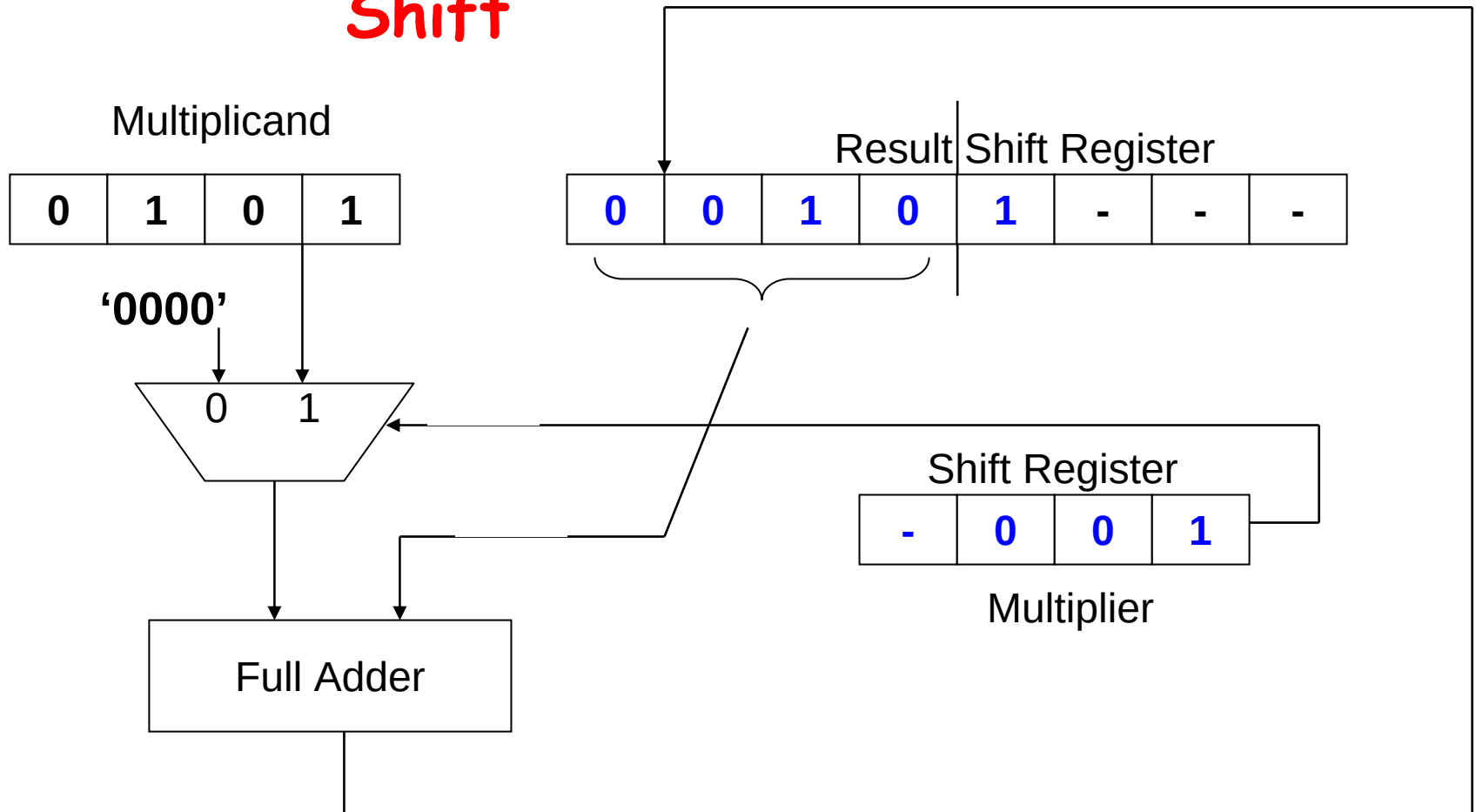


Binary Shift/Add Multiplication

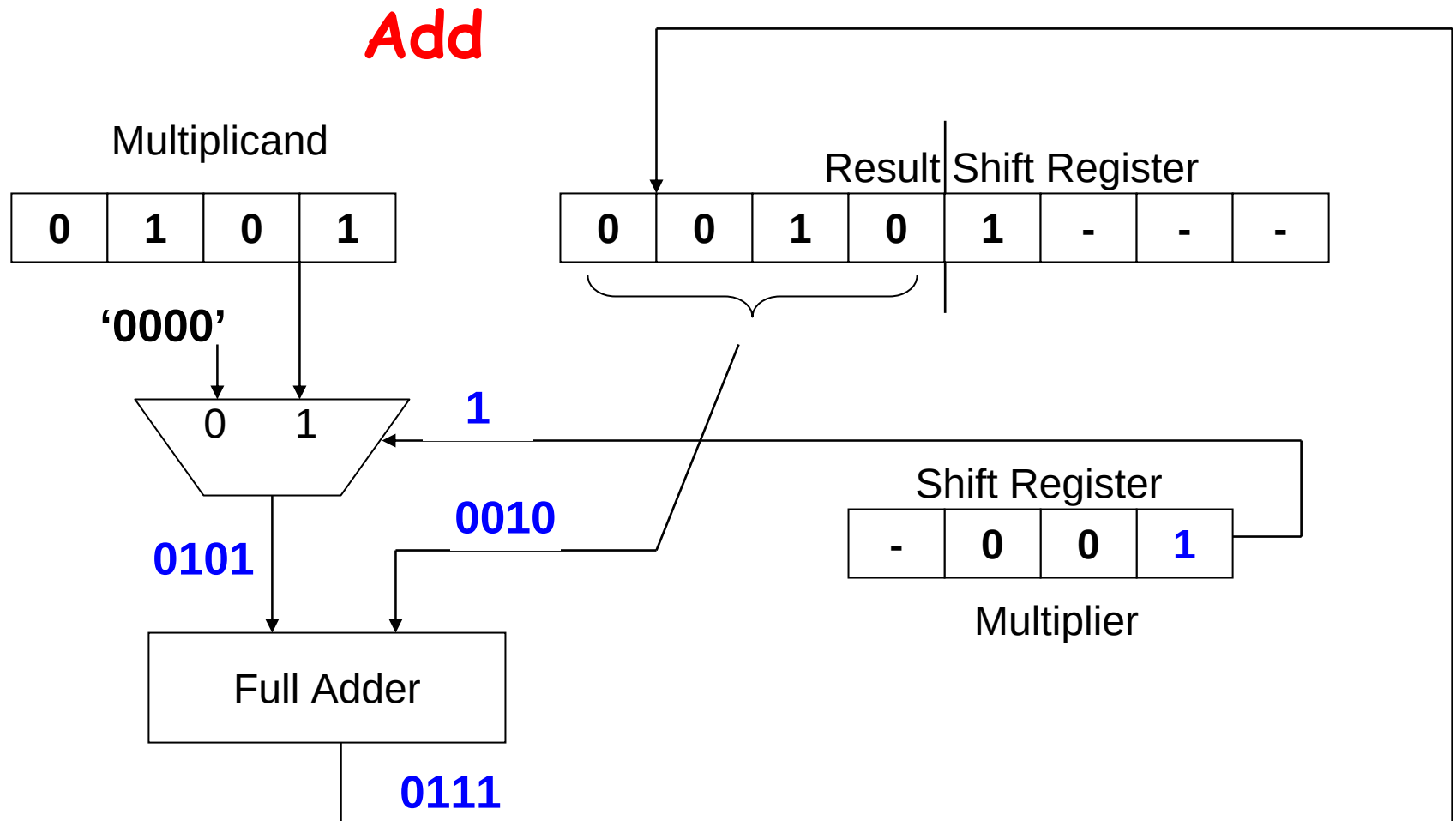


Binary Shift/Add Multiplication

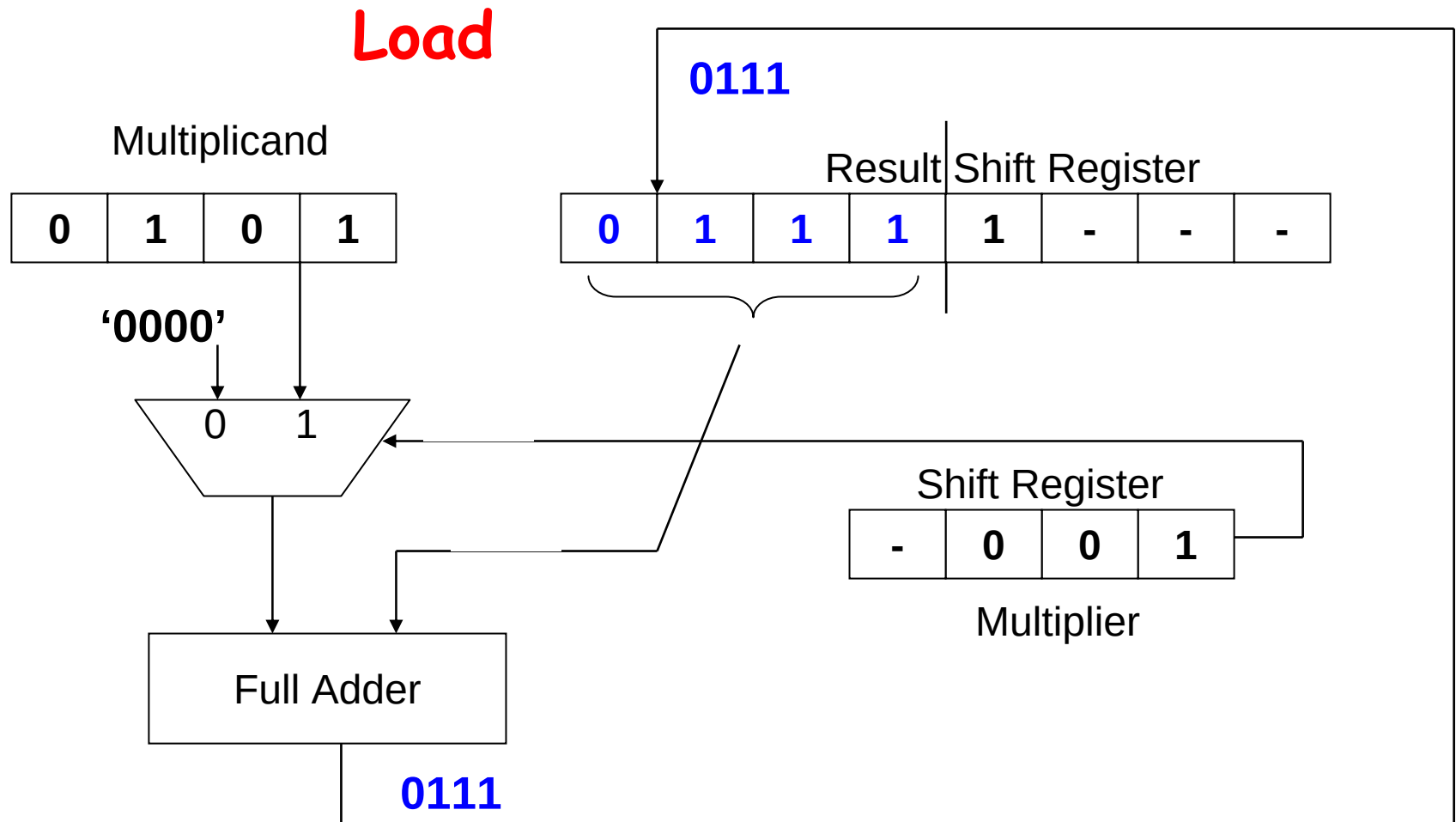
Shift



Binary Shift/Add Multiplication

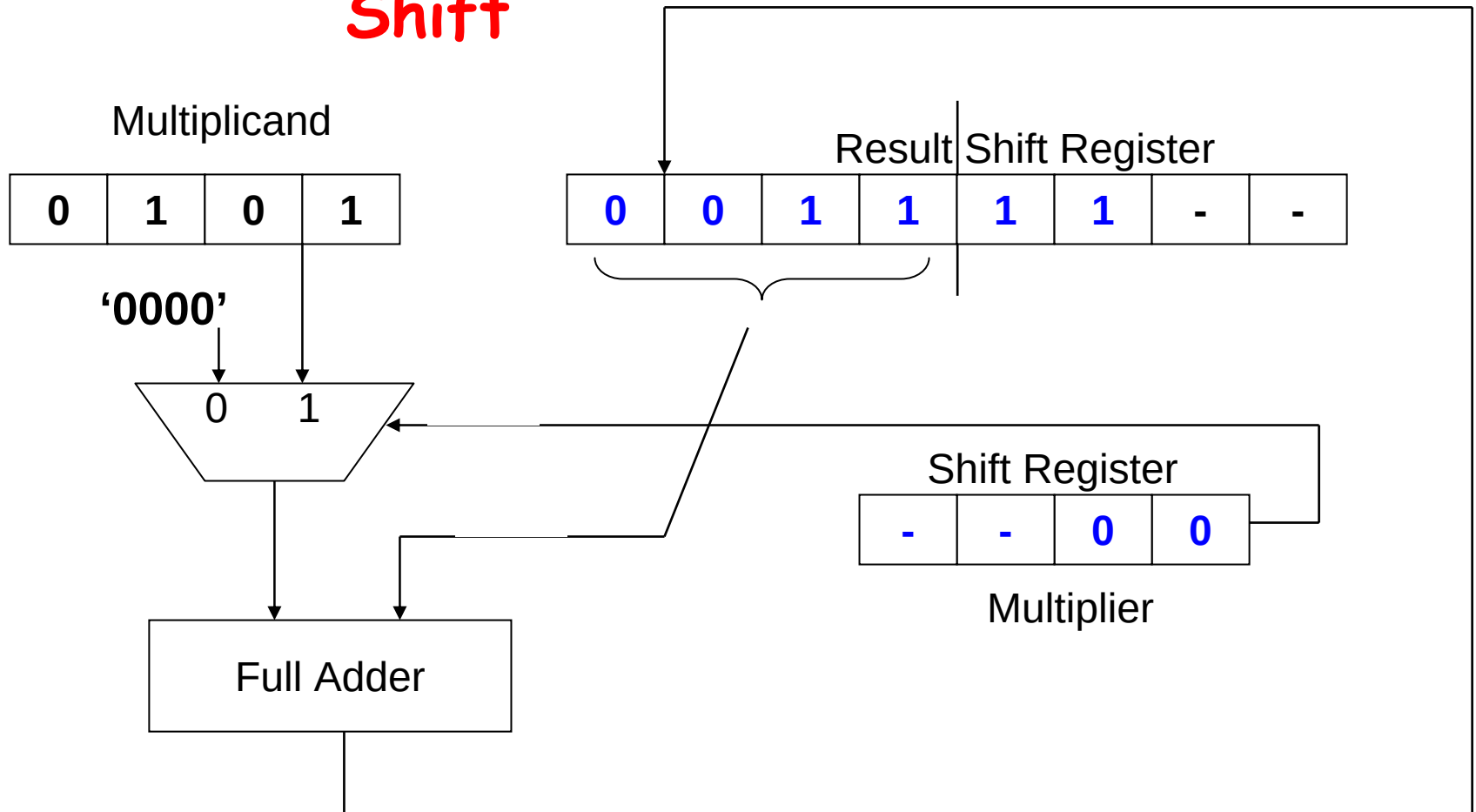


Binary Shift/Add Multiplication

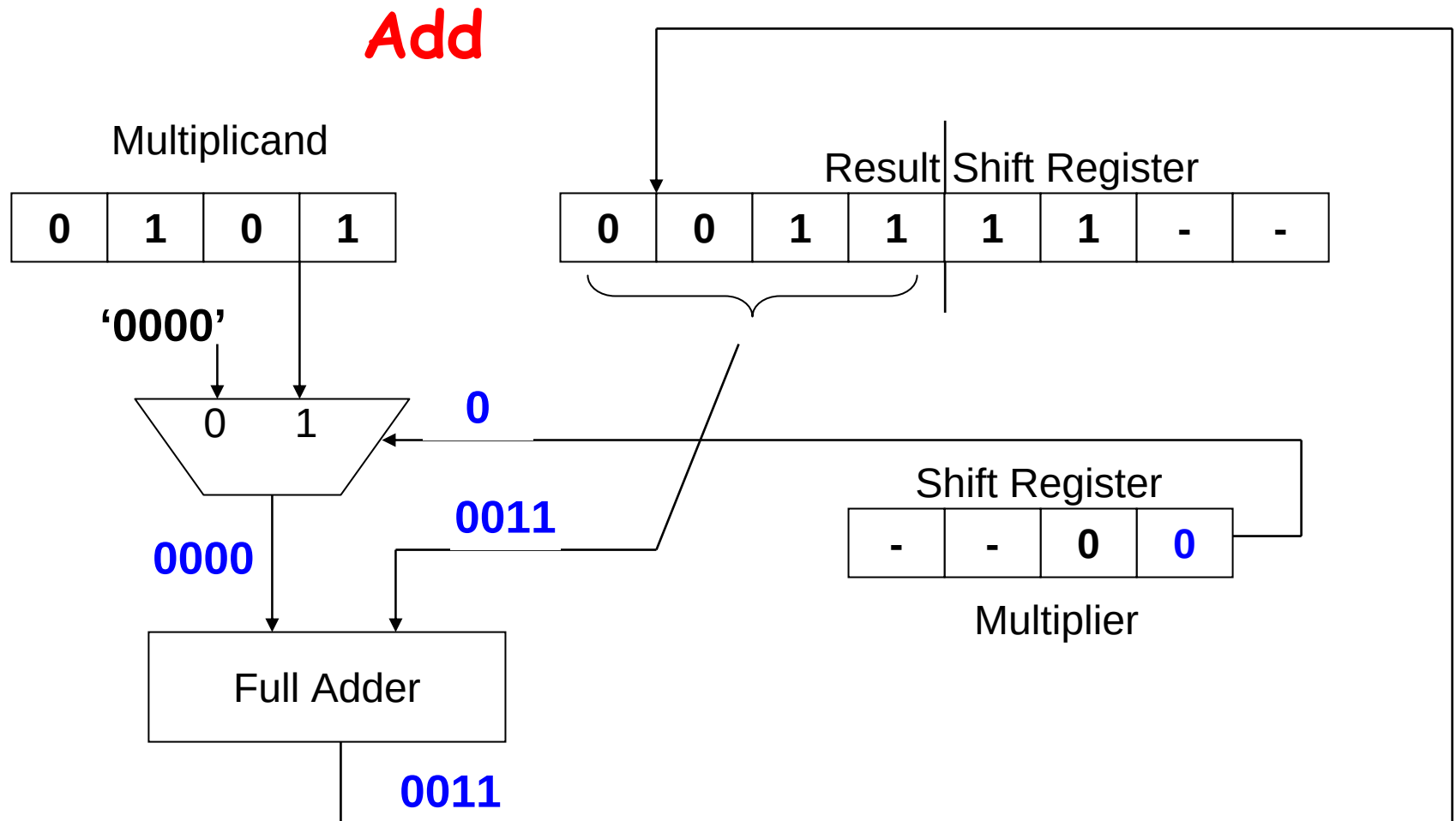


Binary Shift/Add Multiplication

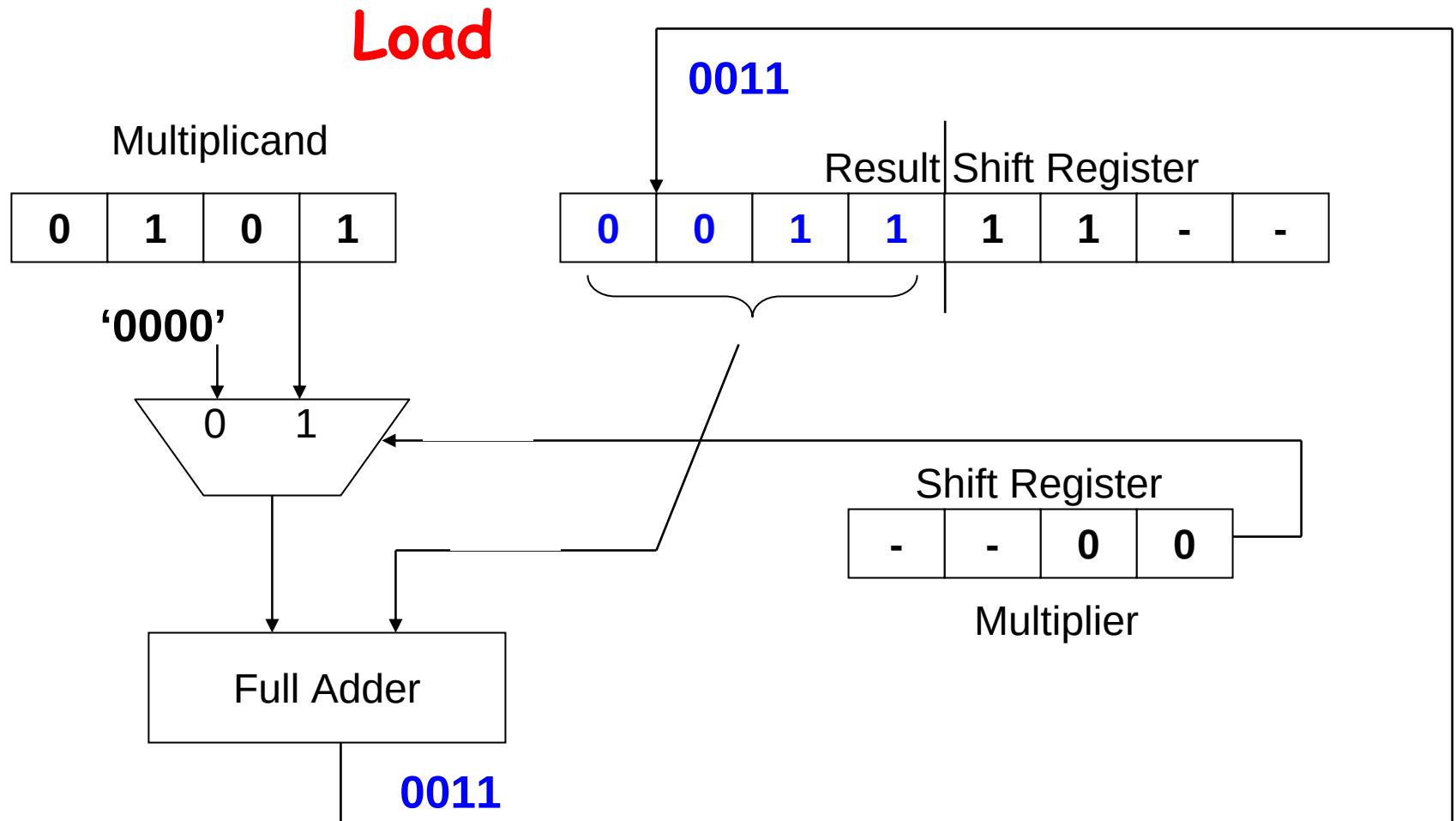
Shift



Binary Shift/Add Multiplication

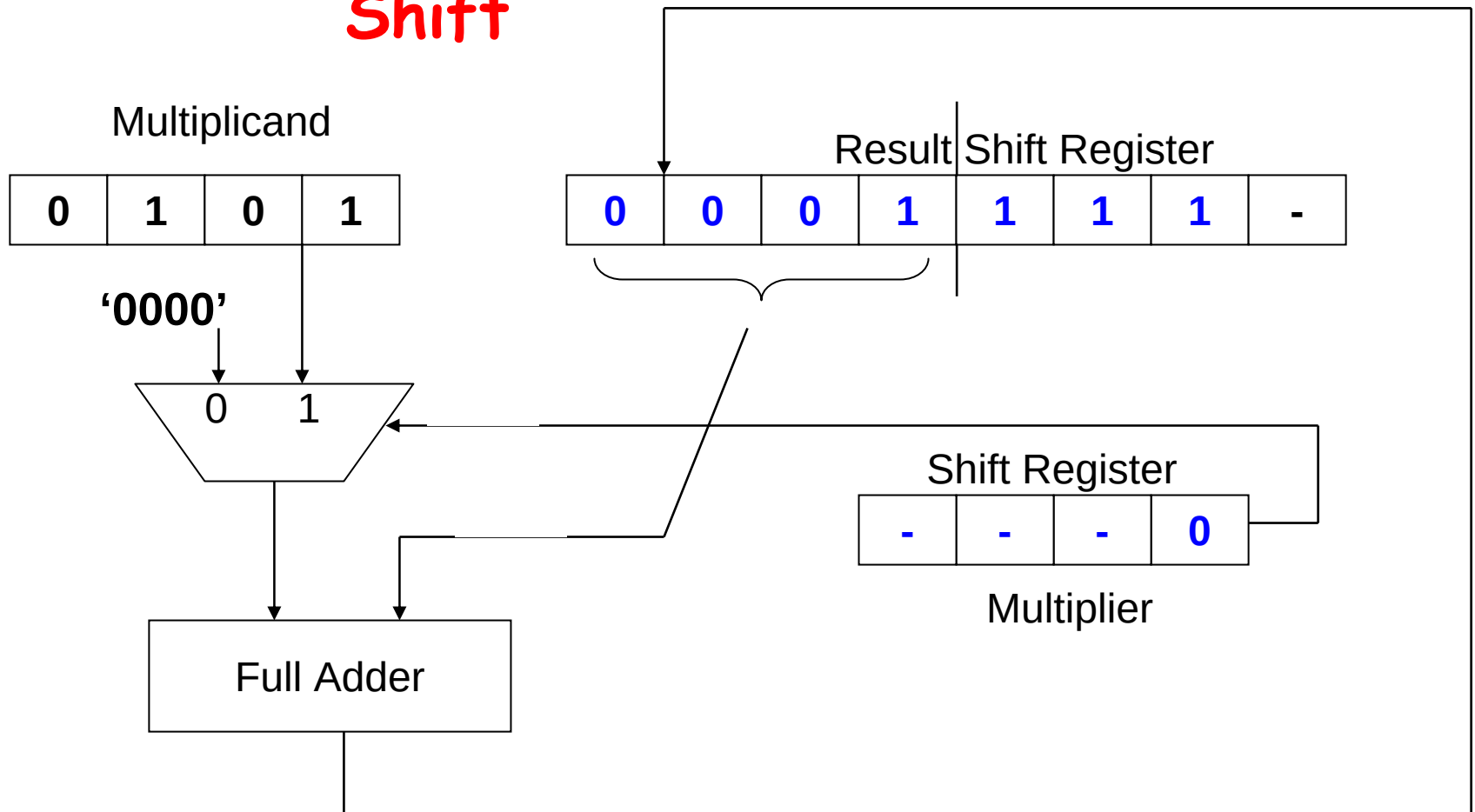


Binary Shift/Add Multiplication

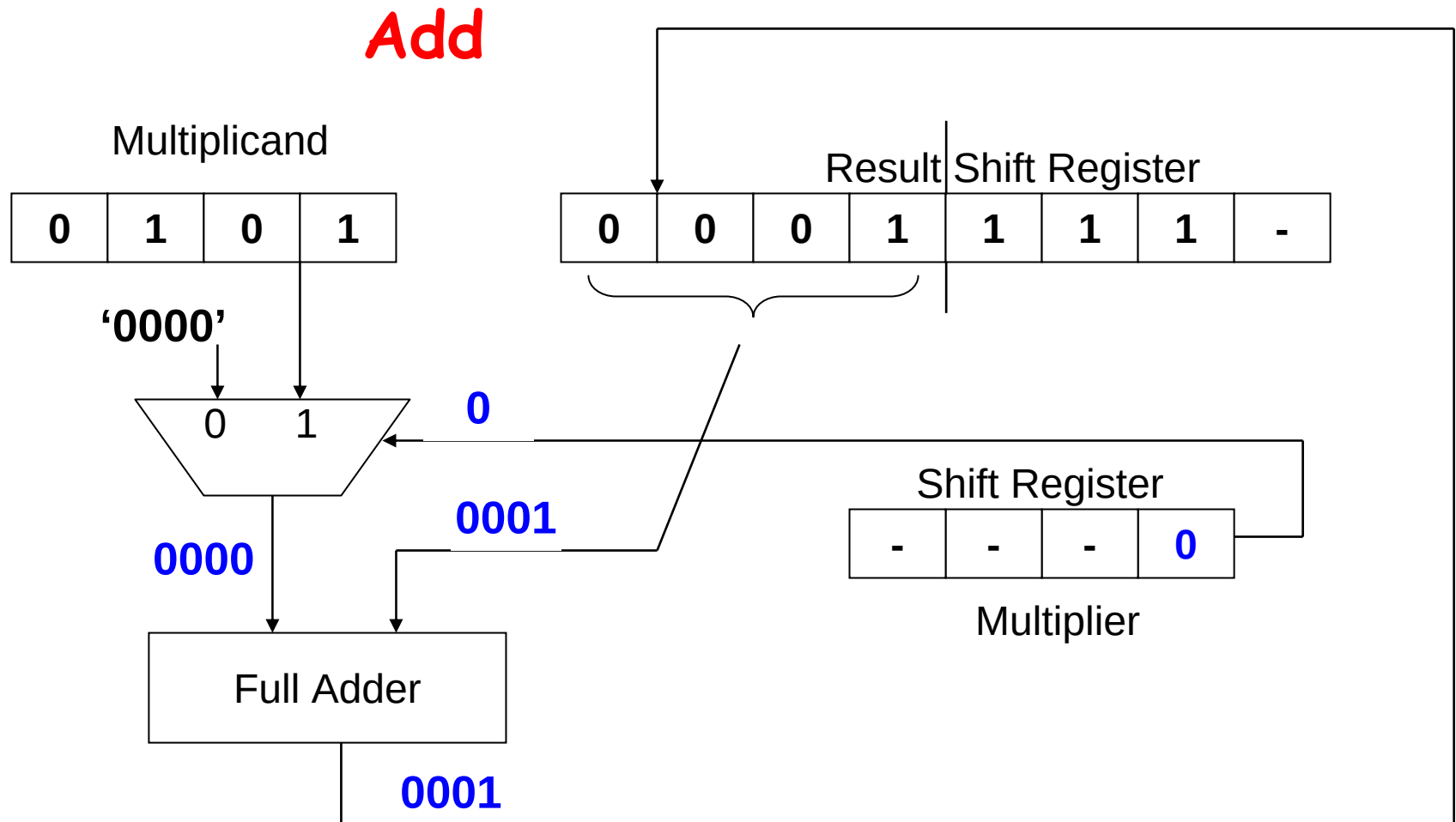


Binary Shift/Add Multiplication

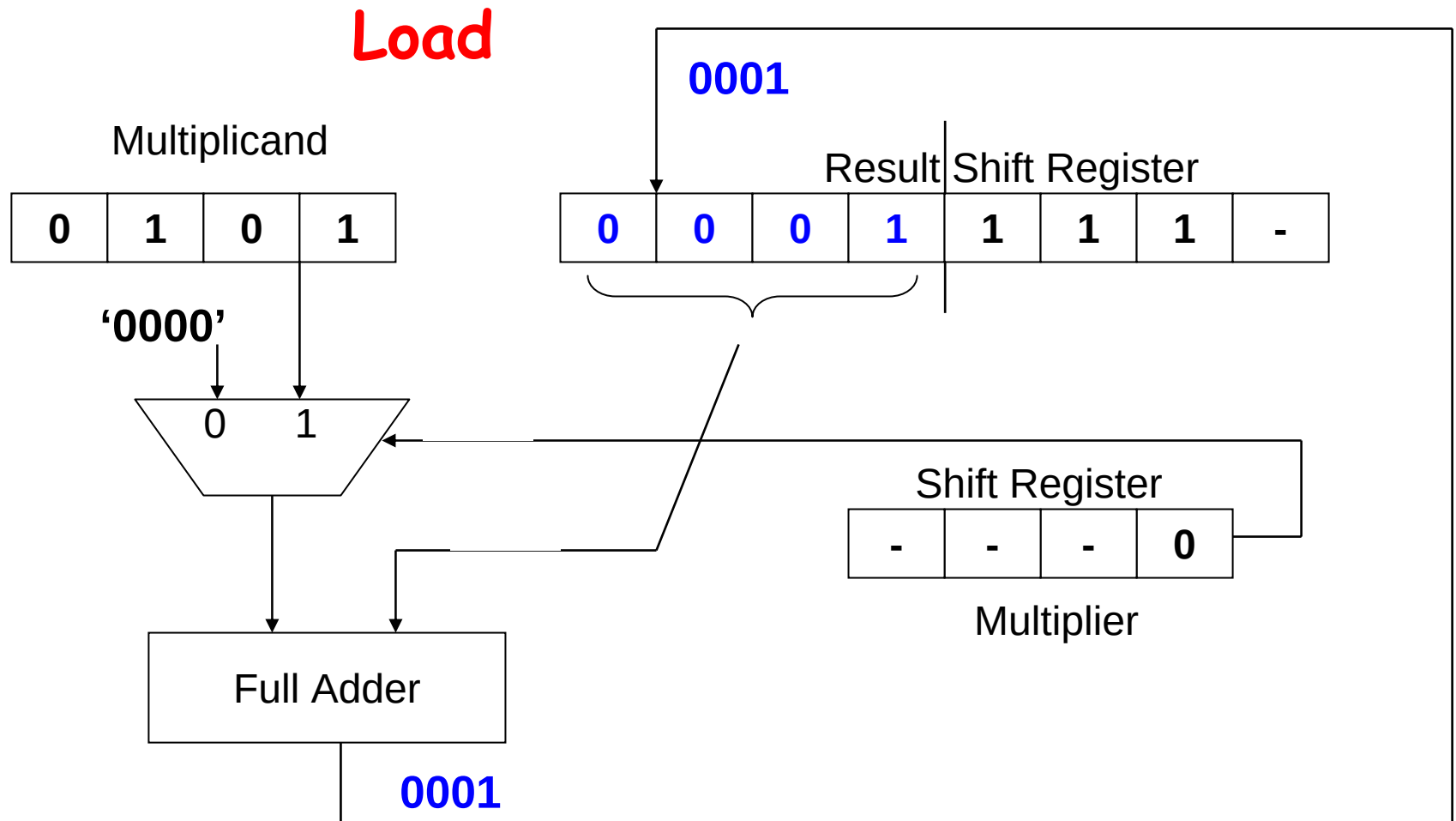
Shift



Binary Shift/Add Multiplication

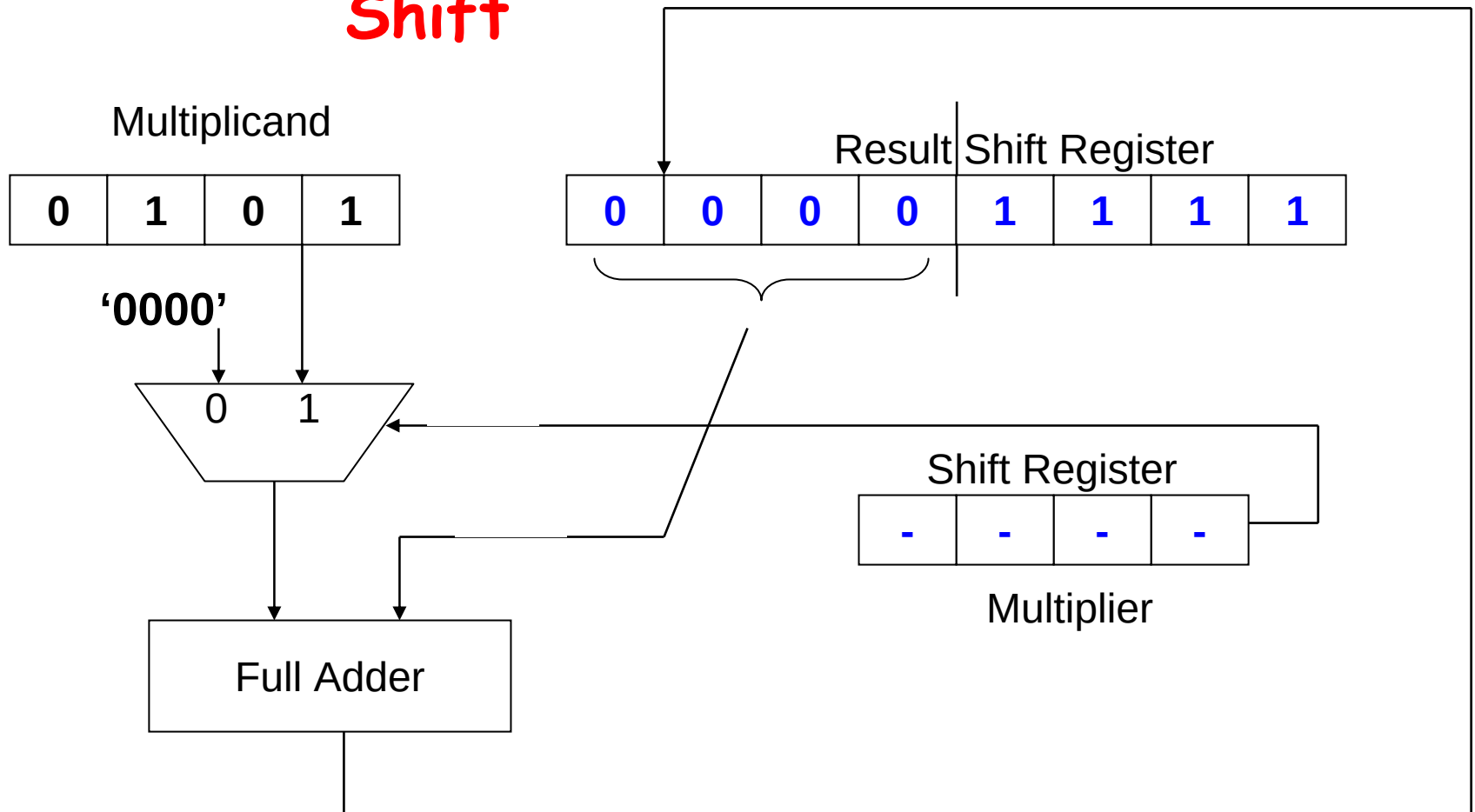


Binary Shift/Add Multiplication



Binary Shift/Add Multiplication

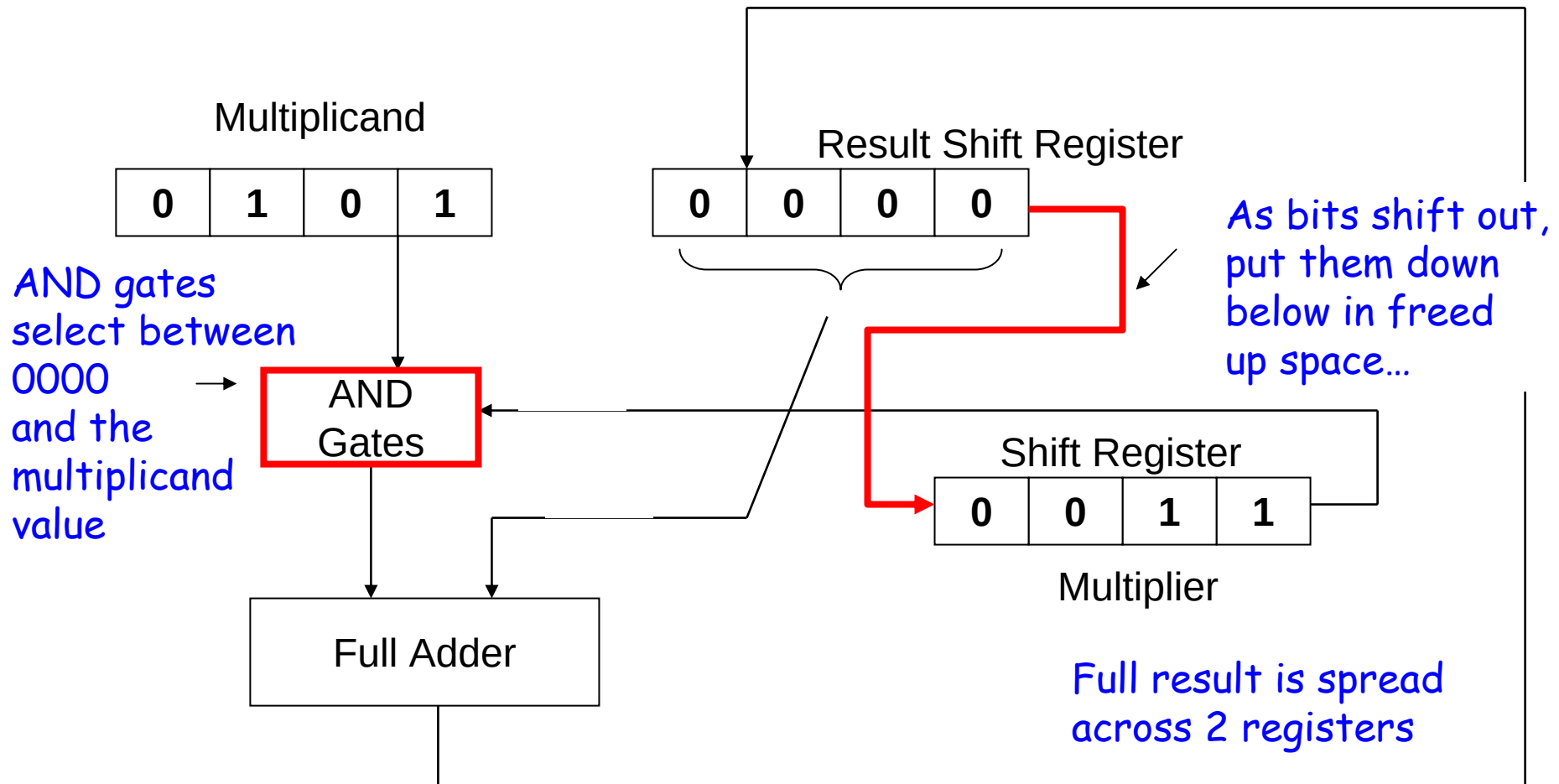
Shift



Performance/Area Analysis

- 2 cycles per bit
 - Add and Load happen in one cycle
 - Shift happens in the other cycle
 - 4x4 multiply => 8 cycles
 - ? *How can it be reduced to 1 cycle/bit?*
- Area
 - 16 register bits
 - 4-bit adder
 - 4-bit MUX
 - ? *How to simplify MUX?*
 - ? *How to reduce register bits needed?*

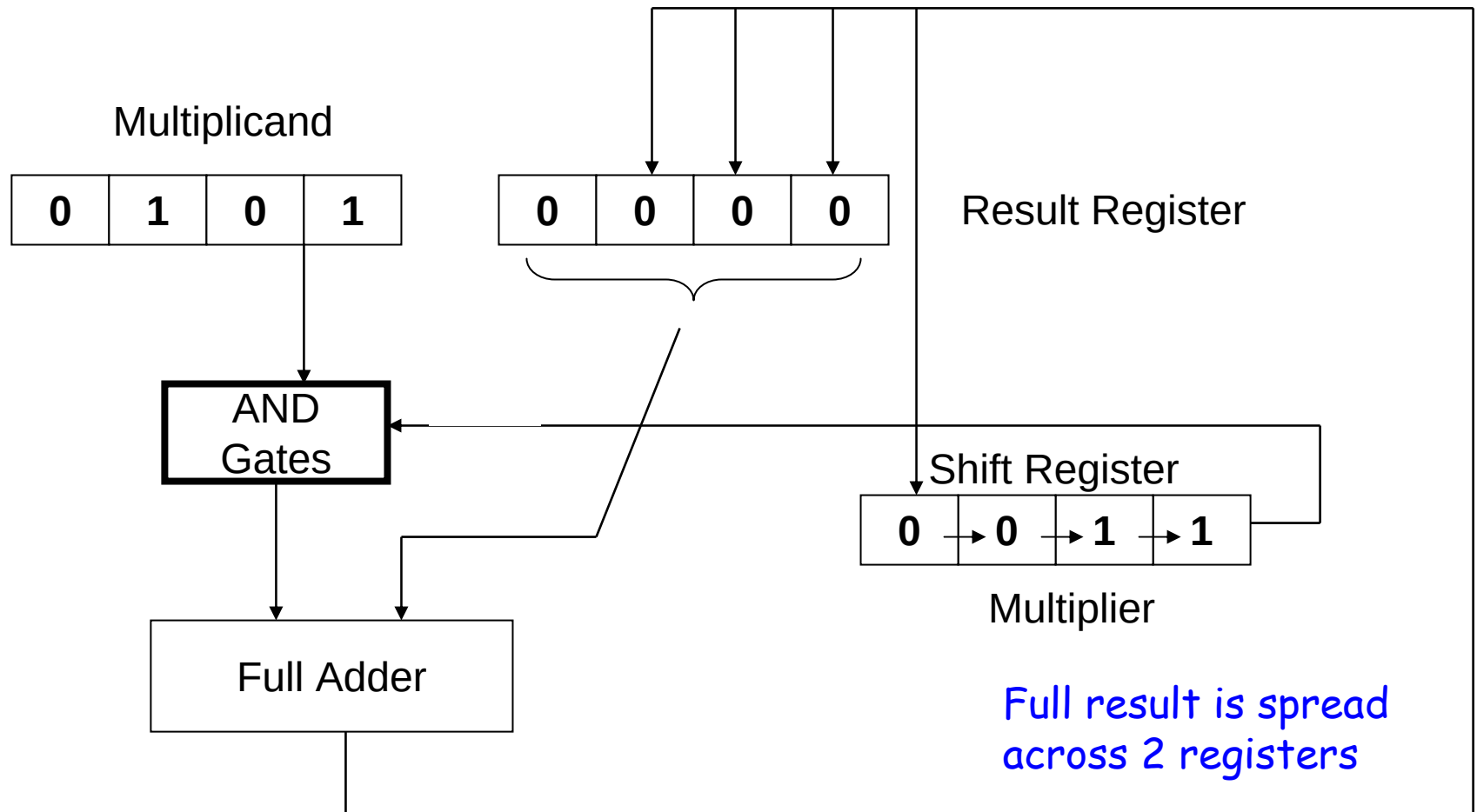
Simplified MUX and Reduced Register Bits



One Cycle Per Bit

- Load result and shift at same time
 - Load result into register already shifted...
 - Shift bottom half of result register and multiplier register at same time
- Can combine with "Simplified MUX and Reduced Register Bits" design

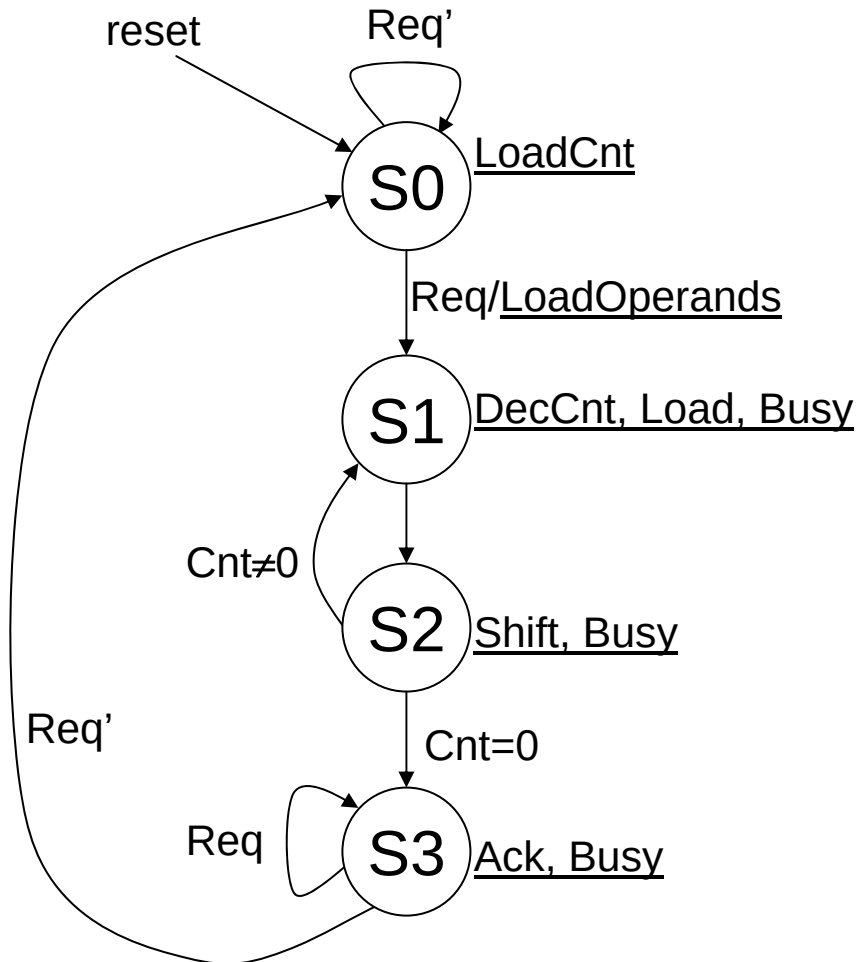
Simplified MUX and Reduced Register Bits



System Level Issues

- How to get data into and out of registers?
 - Multiplicand and multiplier registers should be loadable
- Handshaking signals with outside
 - REQ/ACK
 - BUSY
- FSM to control entire multiplier?

FSM Controller



S0: wait for Req=1, load operands when Req=1

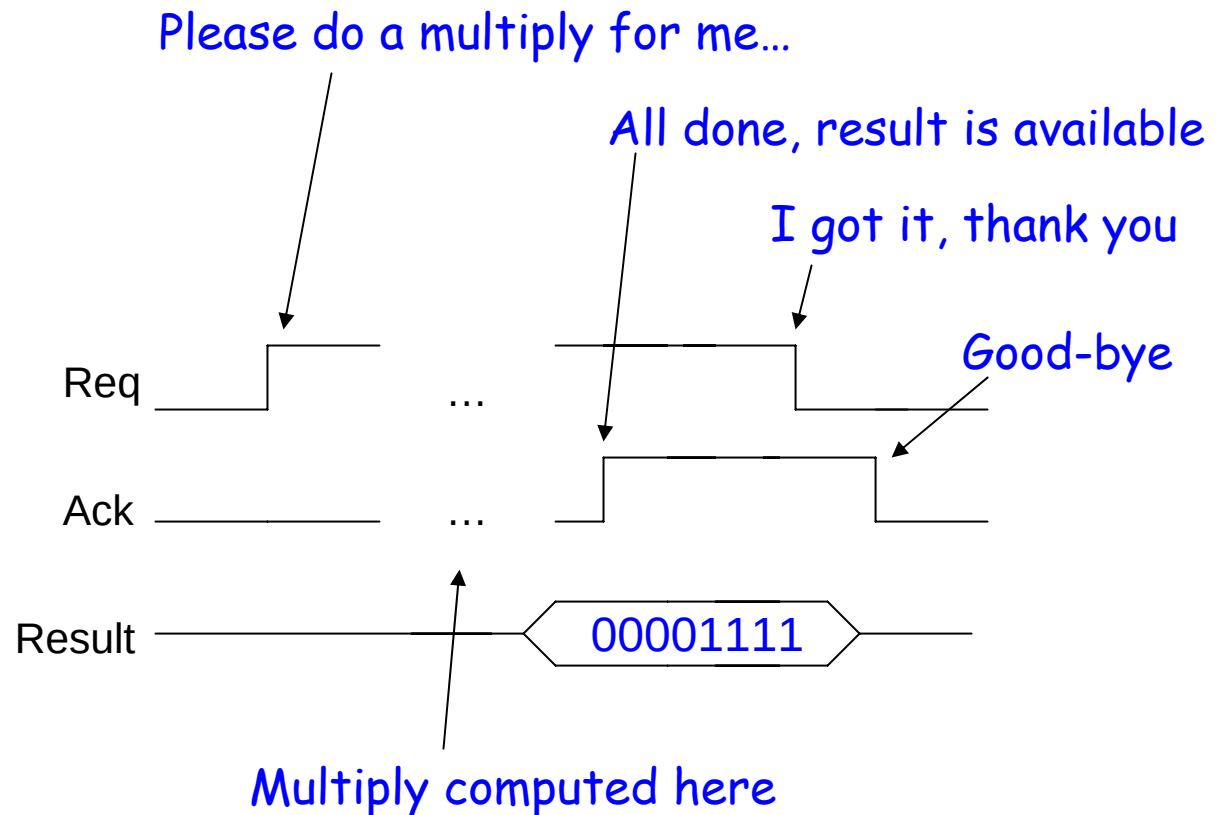
S1: Increment counter, do Load op

S2: Do Shift op

S3: All done, raise Ack, wait for Req=0

Request/Ack Handshake

- Called a 4-cycle handshake



Binary Multiplication

[illegible]

This is identical to grade-school multiplication except everything is binary...

The problem is that
you must add n numbers
in a column 2 at a time?

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1
	b_3	b_2	b_1	b_0
b	0	0	1	1

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1

	b_3	b_2	b_1	b_0
b	0	0	1	1

0	1	0	1
0	0	1	1
0	1	0	1

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1

	b_3	b_2	b_1	b_0
b	0	0	1	1

0	1	0	1
0	0	1	1
0	1	0	1
$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1
	b_3	b_2	b_1	b_0
b	0	0	1	1

	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1
	b_3	b_2	b_1	b_0
b	0	0	1	1

	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
0	0	0	0	
$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$	

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1
	b_3	b_2	b_1	b_0
b	0	0	1	1

	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
0	0	0	0	
$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$	
0	0	0	0	
$a_3 \cdot b_3$	$a_2 \cdot b_3$	$a_1 \cdot b_3$	$a_0 \cdot b_3$	

Binary Multiplication

	a_3	a_2	a_1	a_0
a	0	1	0	1
	b_3	b_2	b_1	b_0
b	0	0	1	1

0	1	0	1
0	0	1	1
0	1	0	1
$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$

0	1	0	1
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$

0	0	0	0
$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$

0	0	0	0
$a_3 \cdot b_3$	$a_2 \cdot b_3$	$a_1 \cdot b_3$	$a_0 \cdot b_3$

0
0
0
1
1
1
1

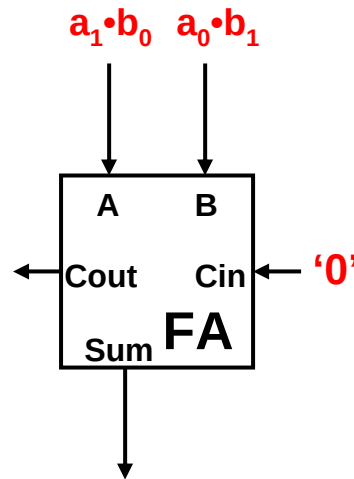
Binary Multiplication

Using partial sums

$$\begin{array}{r}
 \begin{array}{cccc}
 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1 \\
 \hline
 0 & 1 & 0 & 1 \\
 a_3 \cdot b_0 & a_2 \cdot b_0 & a_1 \cdot b_0 & a_0 \cdot b_0
 \end{array} \\
 \begin{array}{cccc}
 0 & 1 & 0 & 1 \\
 a_3 \cdot b_1 & a_2 \cdot b_1 & a_1 \cdot b_1 & a_0 \cdot b_1
 \end{array} \\
 \hline
 0 & 1 & 1 & 1 & 1
 \end{array}$$

Binary Multiplication

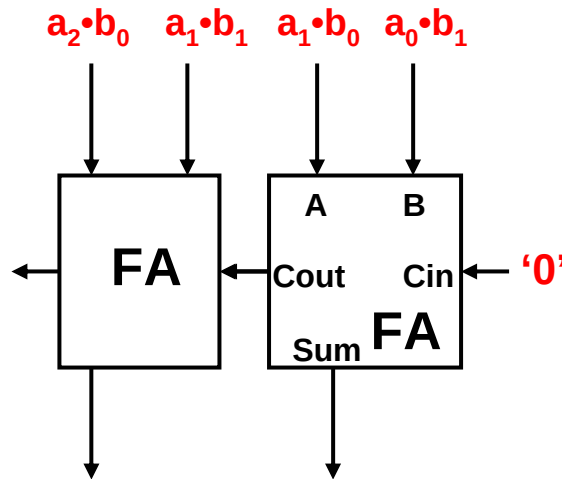
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
0	1	1	1	1

Binary Multiplication

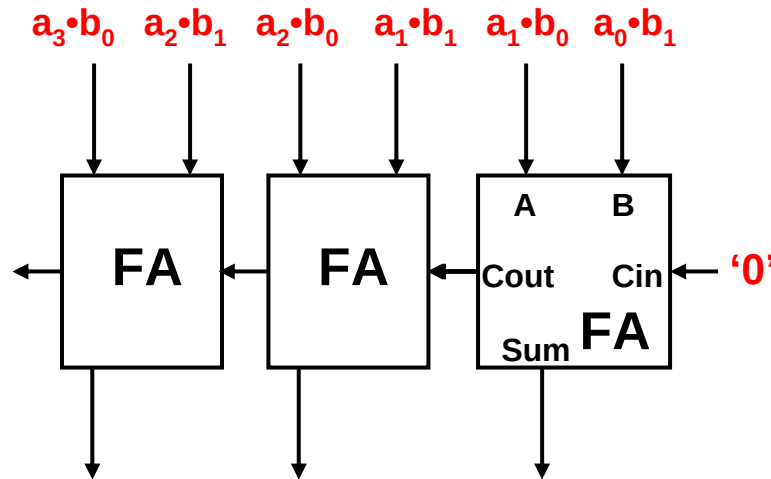
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
0	1	1	1	1

Binary Multiplication

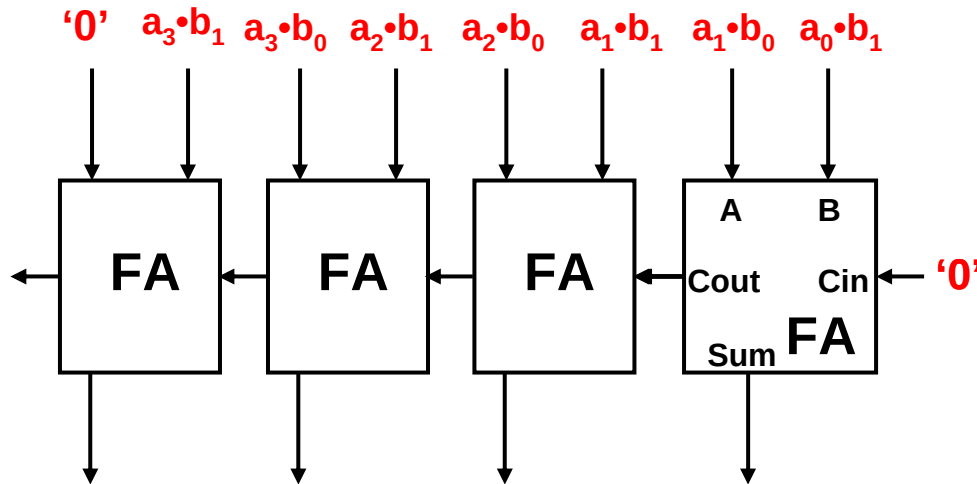
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
0	1	0	1	
$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
	0	1	1	1

Binary Multiplication

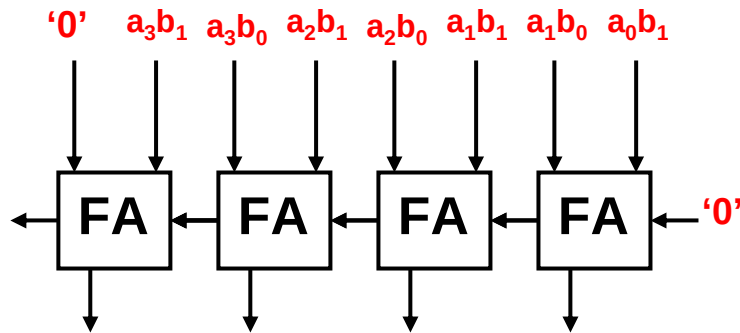
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
	1	0	1	
	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$	
	0	1	1	1
	0	1	1	1

Binary Multiplication

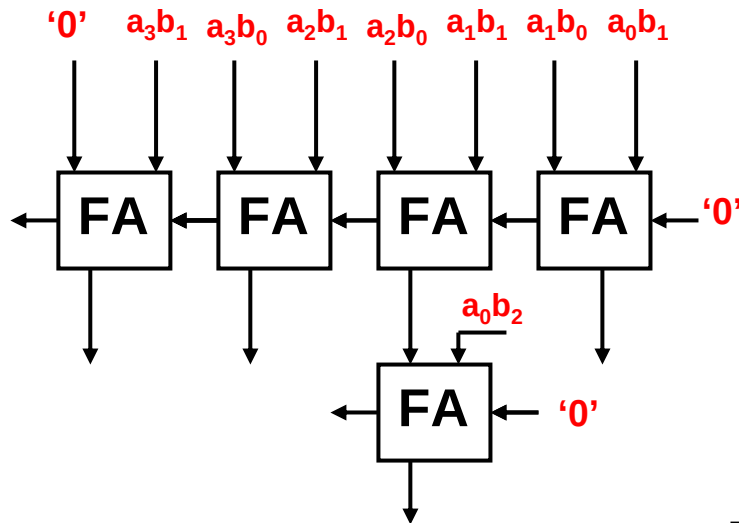
Using partial sums



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Binary Multiplication

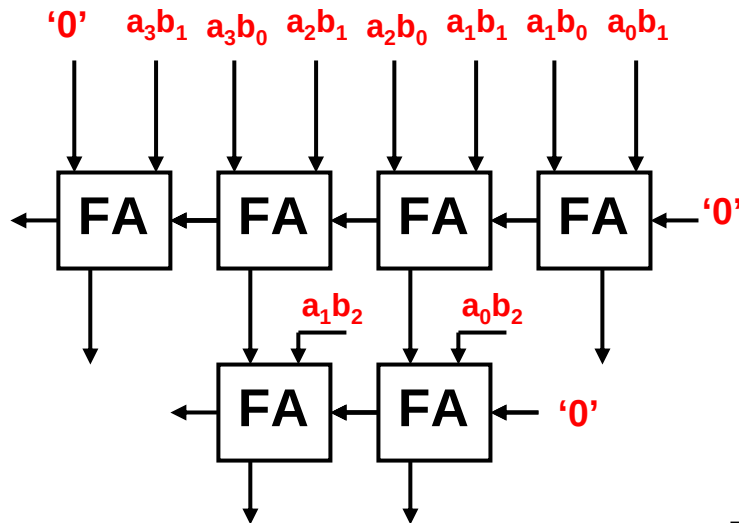
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
	0	1	0	1
	$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$
	0	1	1	1
	0	0	0	0
	$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$
	0	0	1	1

Binary Multiplication

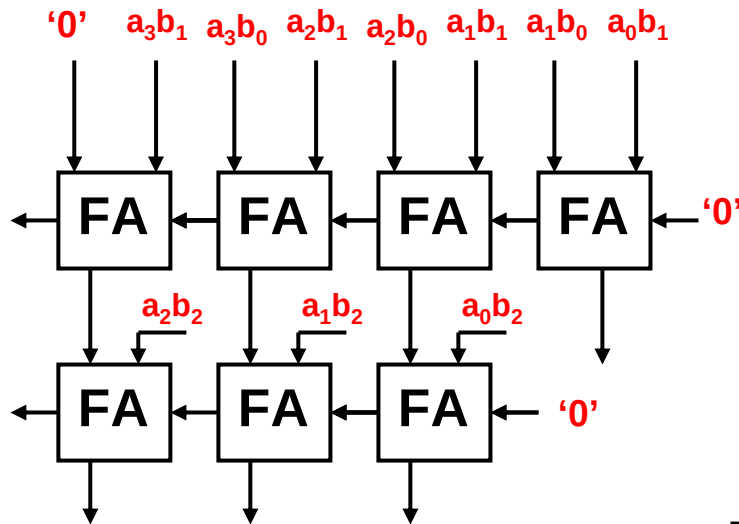
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
	0	1	0	1
	$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$
	0	1	1	1
	0	0	0	0
	$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$
	0	0	1	1

Binary Multiplication

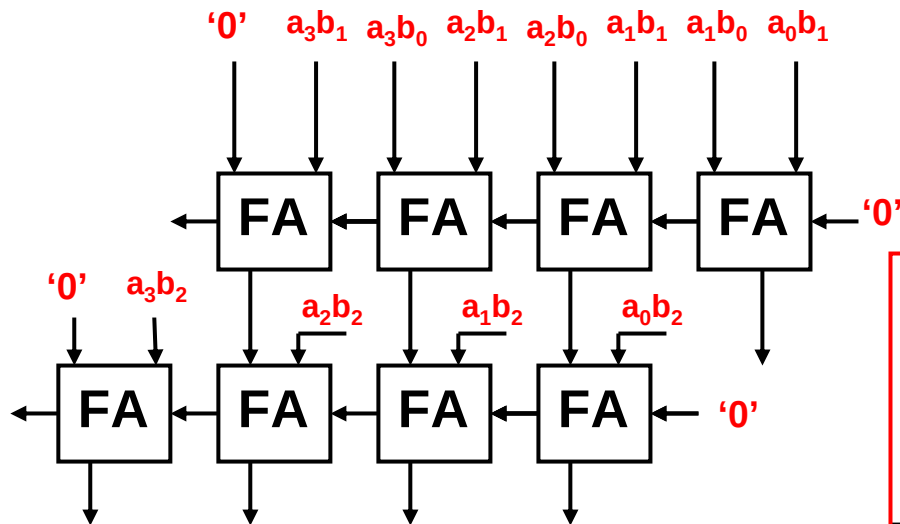
Using partial sums



	0	1	0	1
	0	0	1	1
	0	1	0	1
	$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$
	0	1	0	1
	$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$
	0	1	1	1
	0	0	0	0
	$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$
	0	0	1	1

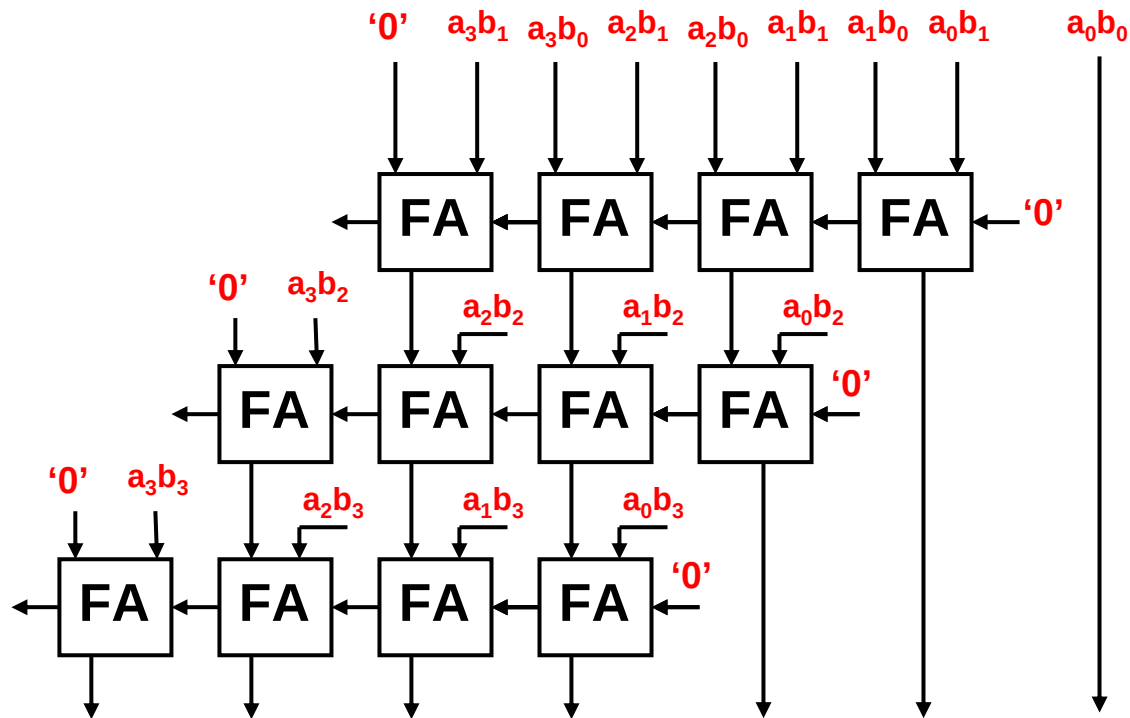
Binary Multiplication

Using partial sums

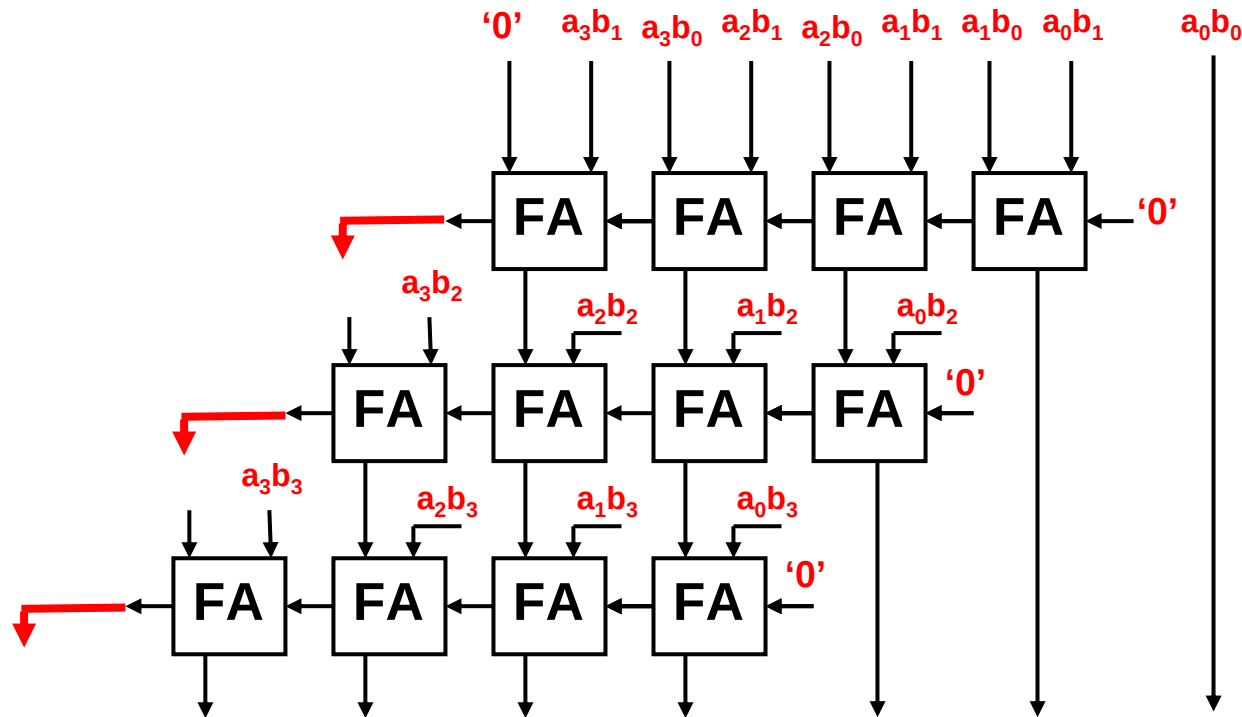


		0	1	0	1	
		0	0	1	1	
		0	1	0	1	
		$a_3 \cdot b_0$	$a_2 \cdot b_0$	$a_1 \cdot b_0$	$a_0 \cdot b_0$	
	0	0	1	0	1	
	$a_3 \cdot b_1$	$a_2 \cdot b_1$	$a_1 \cdot b_1$	$a_0 \cdot b_1$		
0,		0	1	1	1	1
	0	0	0	0		
	$a_3 \cdot b_2$	$a_2 \cdot b_2$	$a_1 \cdot b_2$	$a_0 \cdot b_2$		
	0	0	1	1		

Binary Multiplication using an adder array



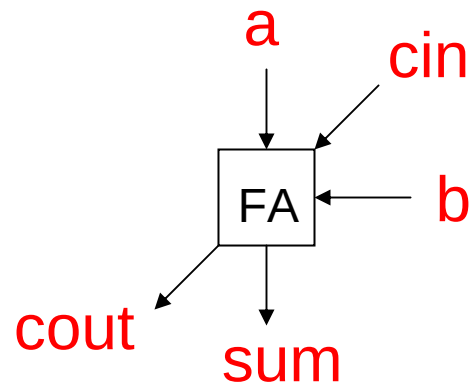
Binary Multiplication using an adder array



Look at the carry chains: Where do the carry outs from the MSB go?
Can we avoid the carry propagation delay ?

Binary Multiplication using an adder array

Propagate the carries to the next column but down into the next partial sum



Binary Multiplication using an adder array

