EE-421: Digital System Design

Multiplier on FPGA and It's Performance

Dr. Rehan Ahmed [rehan.ahmed@seecs.edu.pk]

NUST
SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

Multiplication

Multiplication

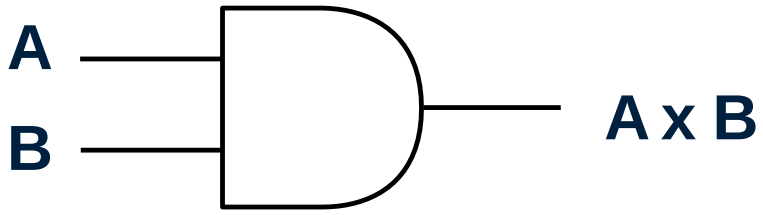
Multiplying 1-bit numbers is an AND operation

0 x 0 = 0

0 x 1 = 0

1 x 0 = 0

1 x 1 = 1



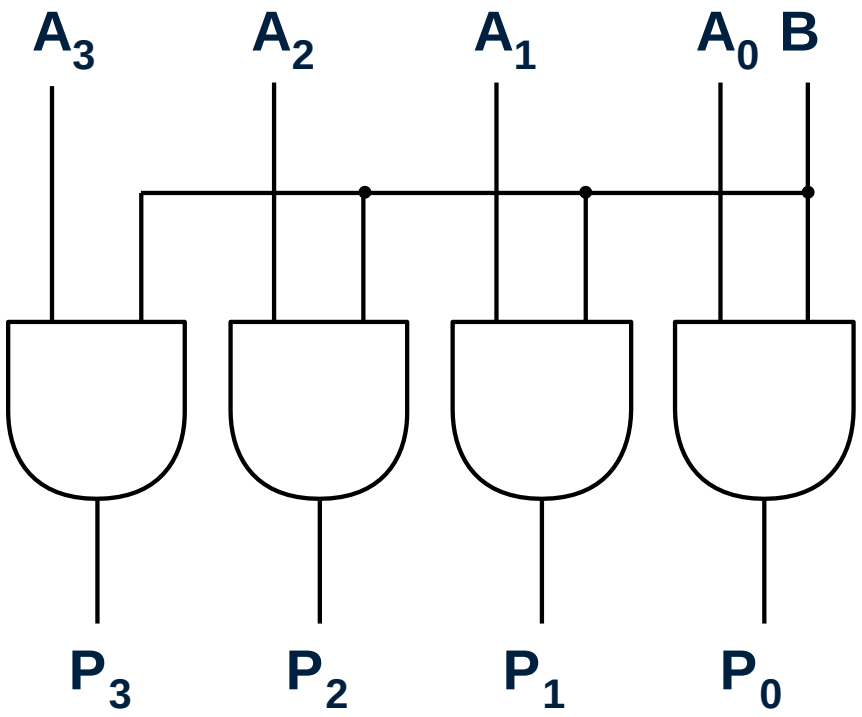
Multiplication

Multiplying 1-bit x N-bit is AND operation

A x B = P

1011 x 0 = 0000

1011 x 1 = 1011



N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

$$
\begin{array}{cccccc}
 & & A_3 & A_2 & A_1 & A_0 \\
X & & B_3 & B_2 & B_1 & B_0 \\
\hline
\end{array}
$$

N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

$$
\begin{array}{cccccc}
 & A_3 & A_2 & A_1 & A_0 \\
X & B_3 & B_2 & B_1 & B_0 \\
\hline
 & A_3 B_0 & A_2 B_0 & A_1 B_0 & A_0 B_0
\end{array}
$$

N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

	A_3	A_2	A_1	A_0
X	B_3	B_2	B_1	B_0
	$A_3 \cdot B_0$	$A_2 \cdot B_0$	$A_1 \cdot B_0$	$A_0 \cdot B_0$
$A_3 \cdot B_1$	$A_2 \cdot B_1$	$A_1 \cdot B_1$	$A_0 \cdot B_1$	

N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

		A_3	A_2	A_1	A_0
	X	B_3	B_2	B_1	B_0
		$A_3 \cdot B_0$	$A_2 \cdot B_0$	$A_1 \cdot B_0$	$A_0 \cdot B_0$
	$A_3 \cdot B_1$	$A_2 \cdot B_1$	$A_1 \cdot B_1$	$A_0 \cdot B_1$	
$A_3 \cdot B_2$	$A_2 \cdot B_2$	$A_1 \cdot B_2$	$A_0 \cdot B_2$		

N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

				A_3	A_2	A_1	A_0
			X	B_3	B_2	B_1	B_0
				$A_3 B_0$	$A_2 B_0$	$A_1 B_0$	$A_0 B_0$
			$A_3 B_1$	$A_2 B_1$	$A_1 B_1$	$A_0 B_1$	
		$A_3 B_2$	$A_2 B_2$	$A_1 B_2$	$A_0 B_2$		
	$A_3 B_3$	$A_2 B_3$	$A_1 B_3$	$A_0 B_3$			

N-Bit x N-Bit (Unsigned) Multiplication

Consider 4-bit x 4-bit Multiplication

				A_3	A_2	A_1	A_0
			X	B_3	B_2	B_1	B_0
				$A_3 \cdot B_0$	$A_2 \cdot B_0$	$A_1 \cdot B_0$	$A_0 \cdot B_0$
			$A_3 \cdot B_1$	$A_2 \cdot B_1$	$A_1 \cdot B_1$	$A_0 \cdot B_1$	
		$A_3 \cdot B_2$	$A_2 \cdot B_2$	$A_1 \cdot B_2$	$A_0 \cdot B_2$		
+	$A_3 \cdot B_3$	$A_2 \cdot B_3$	$A_1 \cdot B_3$	$A_0 \cdot B_3$			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

N-Bit x N-Bit (Unsigned) Multiplication

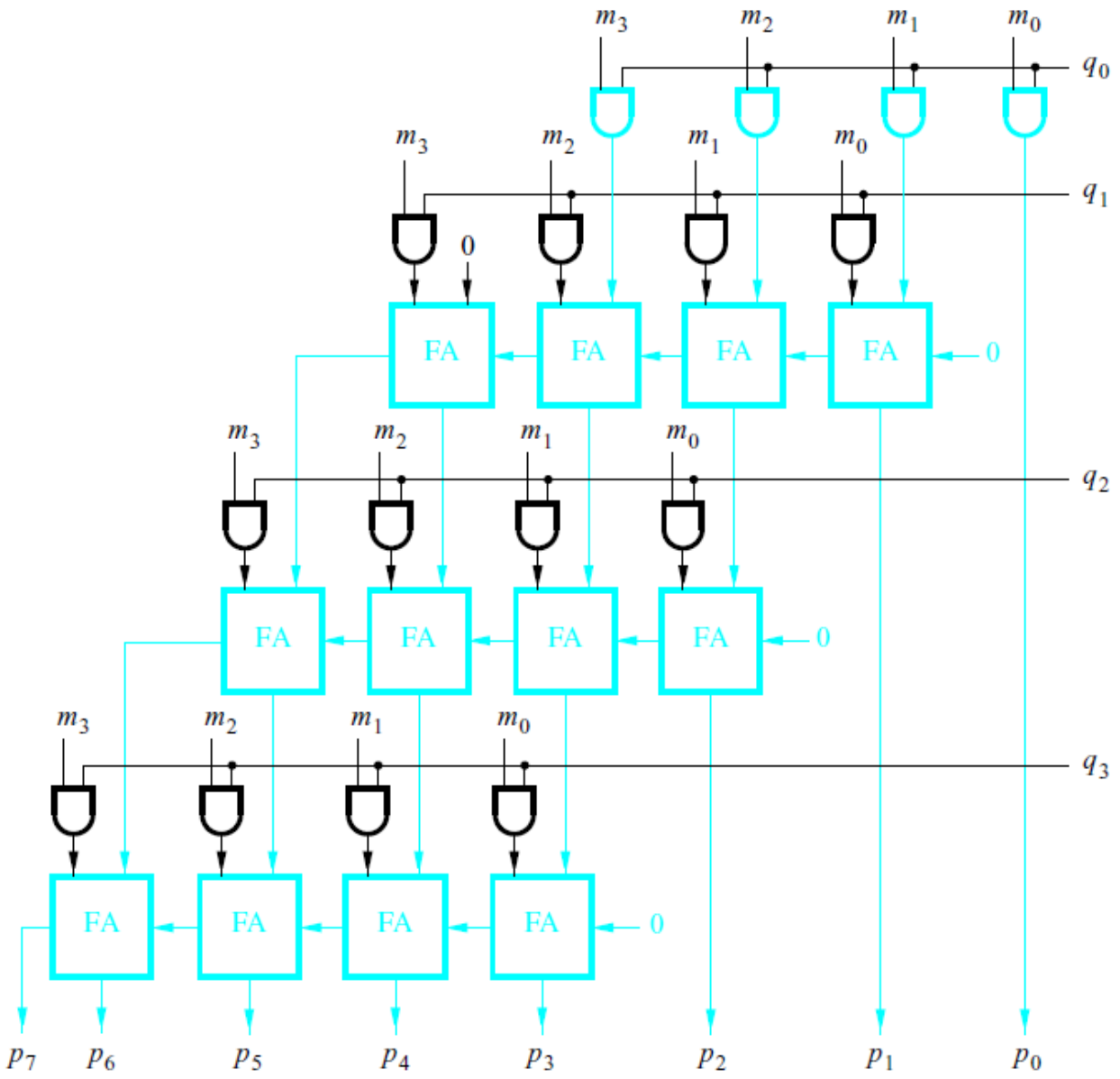
Consider 4-bit x 4-bit Multiplication

				A_3	A_2	A_1	A_0
			X	B_3	B_2	B_1	B_0
AB_i called a "partial product"			→	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			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Sounds familiar, Right! : Multiplication by Hand

How to express this multiplication in Hardware?

Hardware Implementation: N-Bit x N-Bit (Unsigned) Multiplication



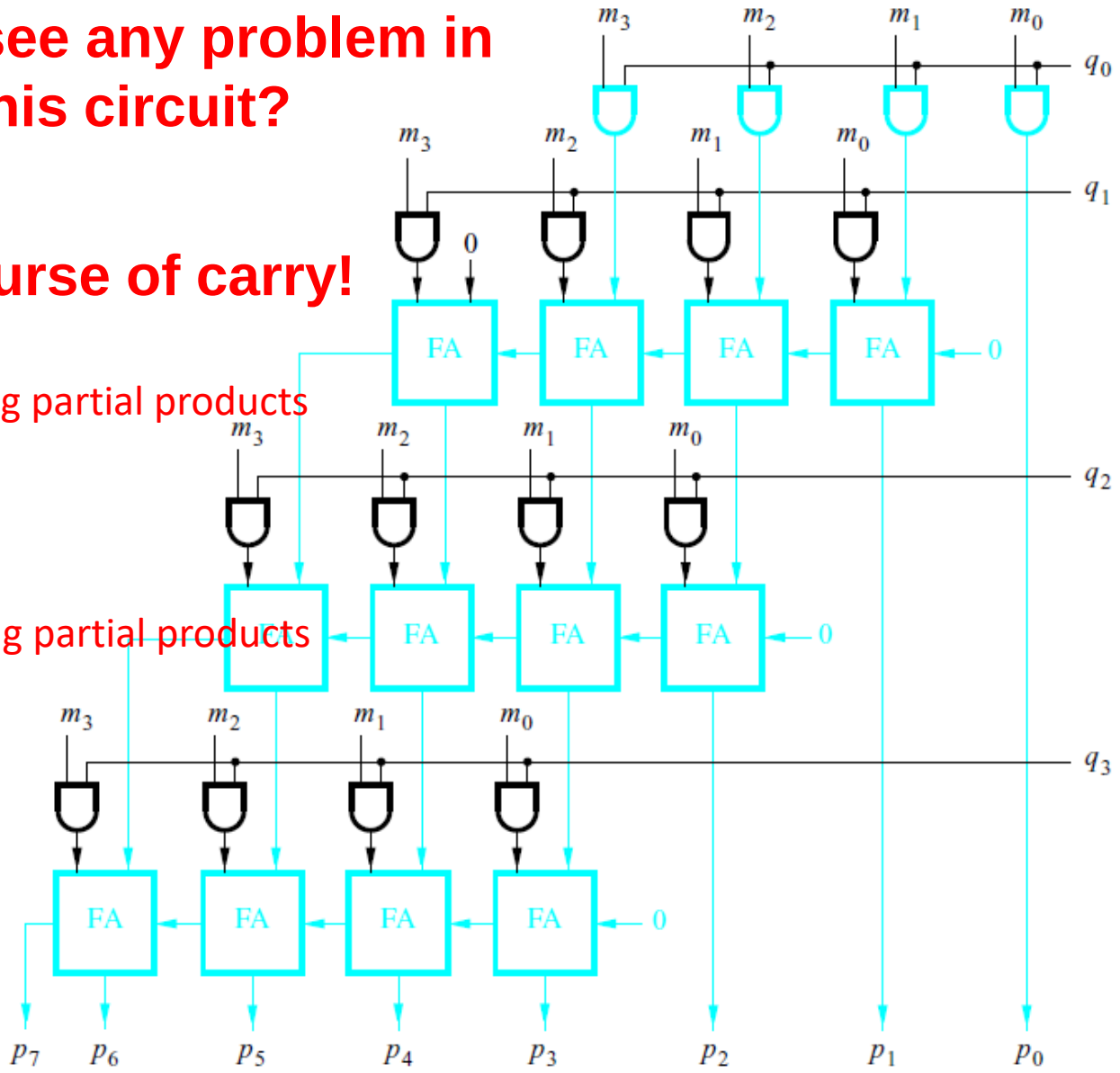
Hardware Implementation: N-Bit x N-Bit (Unsigned) Multiplication

Do you see any problem in this circuit?

Yes, the curse of carry!

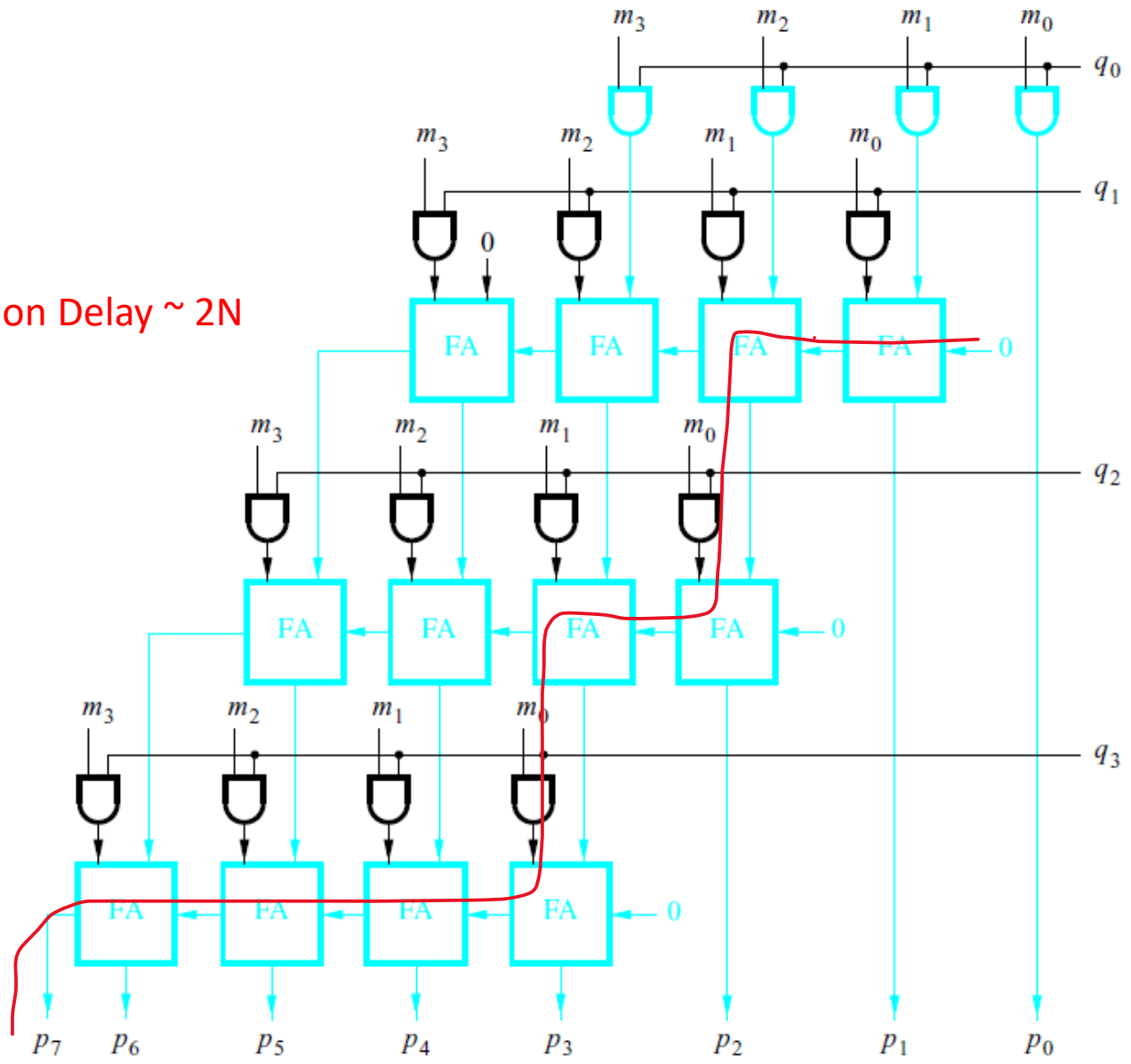
Easy Part: forming partial products

Hard Part: Adding partial products

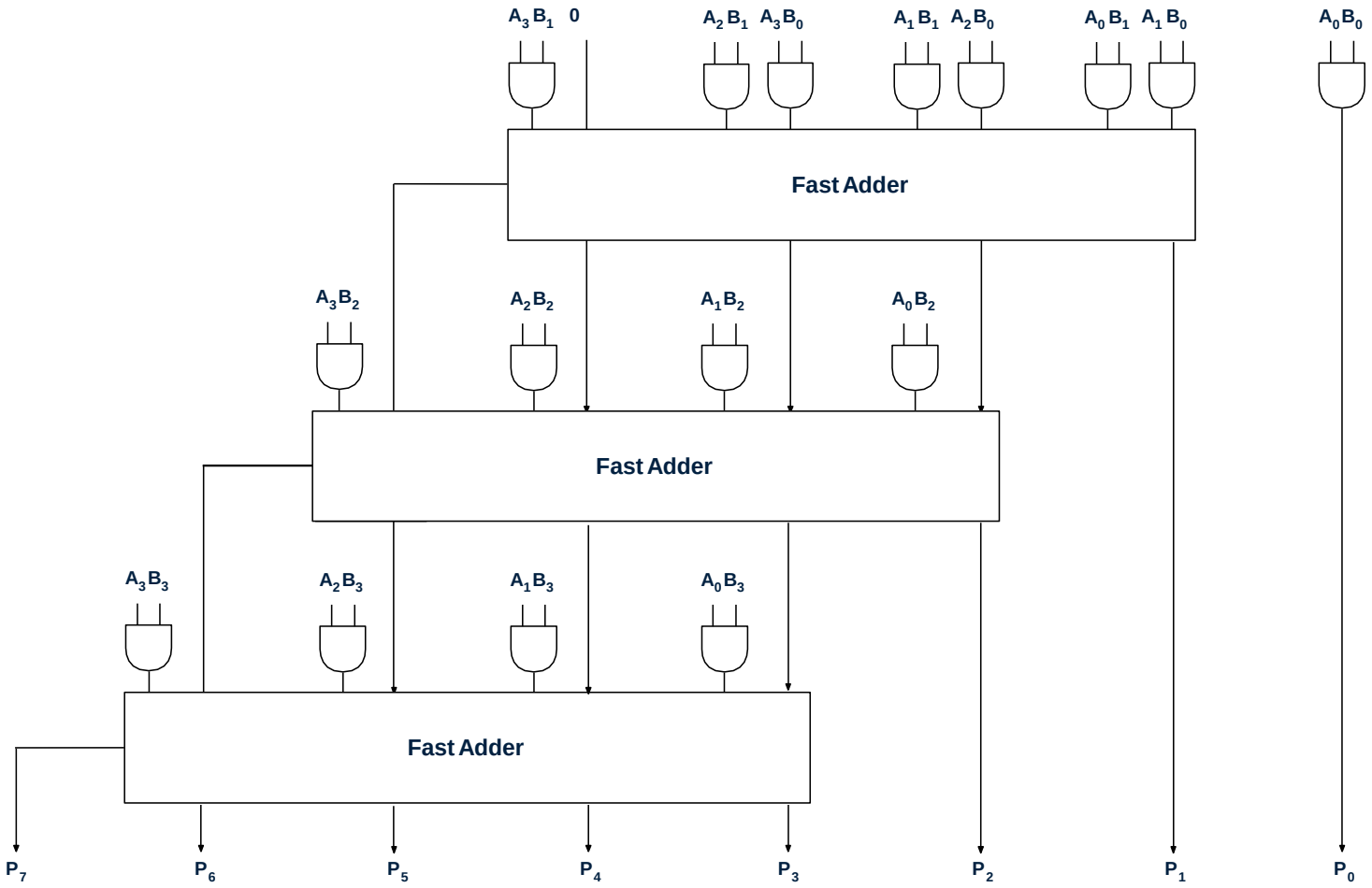


Propagation Delay

Propagation Delay ~ 2N



Using Fast Adders as Discussed Previously



FPGA Implementation of a Combinational Multiplier

- In Verilog:

```
P = A * B;
```

(make sure **P** has twice the number of bits of **A** and **B**)

- Synthesis tool will create a combinational multiplier if DSP Block inference is turned off.

F_{max} of a Combinational Multiplier

- Measured on our Cyclone FPGA:

Multiplier Width (in bits)	F_{max} (MHz)
8 x 8	464
16 x 16	396
32 x 32	104
64 x 64	66

THANK YOU