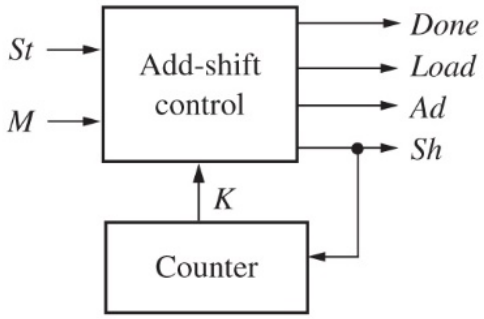
Lecture 11 (Chapter 4)

Design Examples: Array Multipliers

Khaza Anuarul Hoque
ECE 4250/7250

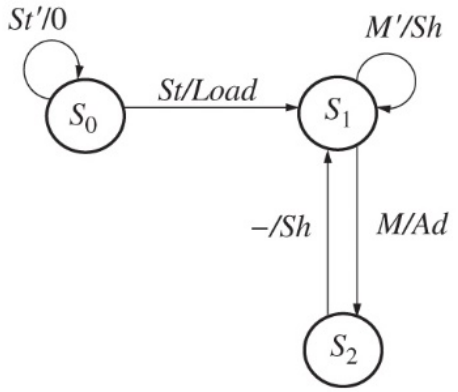
Add-and-Shift Multiplier (continued)

- As the state graph for the multiplier indicates, the control performs 2 functions—generating add or shift signals as needed and counting the number of shifts. If the number of bits is large, it is convenient to divide the control circuit into a counter and an add-shift control:



Add-and-Shift Multiplier (continued)

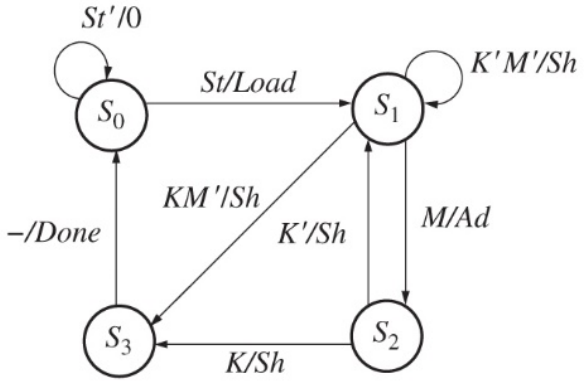
- First, derive a state graph for the add-shift control that tests *St* and *M* and outputs the proper sequence of add and shift signals.



- Then we will add a completion signal (*K*) from the counter that stops the multiplier after the proper number of shifts have been completed.

Add-and-Shift Multiplier (continued)

- The counter is incremented each time a shift signal is generated. If the multiplier is *n* bits, *n* shifts are required.
- Design the counter so that a completion signal (*K*) is generated after *n* - 1 shifts have occurred.
- When K = 1, the circuit should perform one more addition, if necessary, and then do the final shift.
- Final state graph for add-shift control:



Array Multiplier

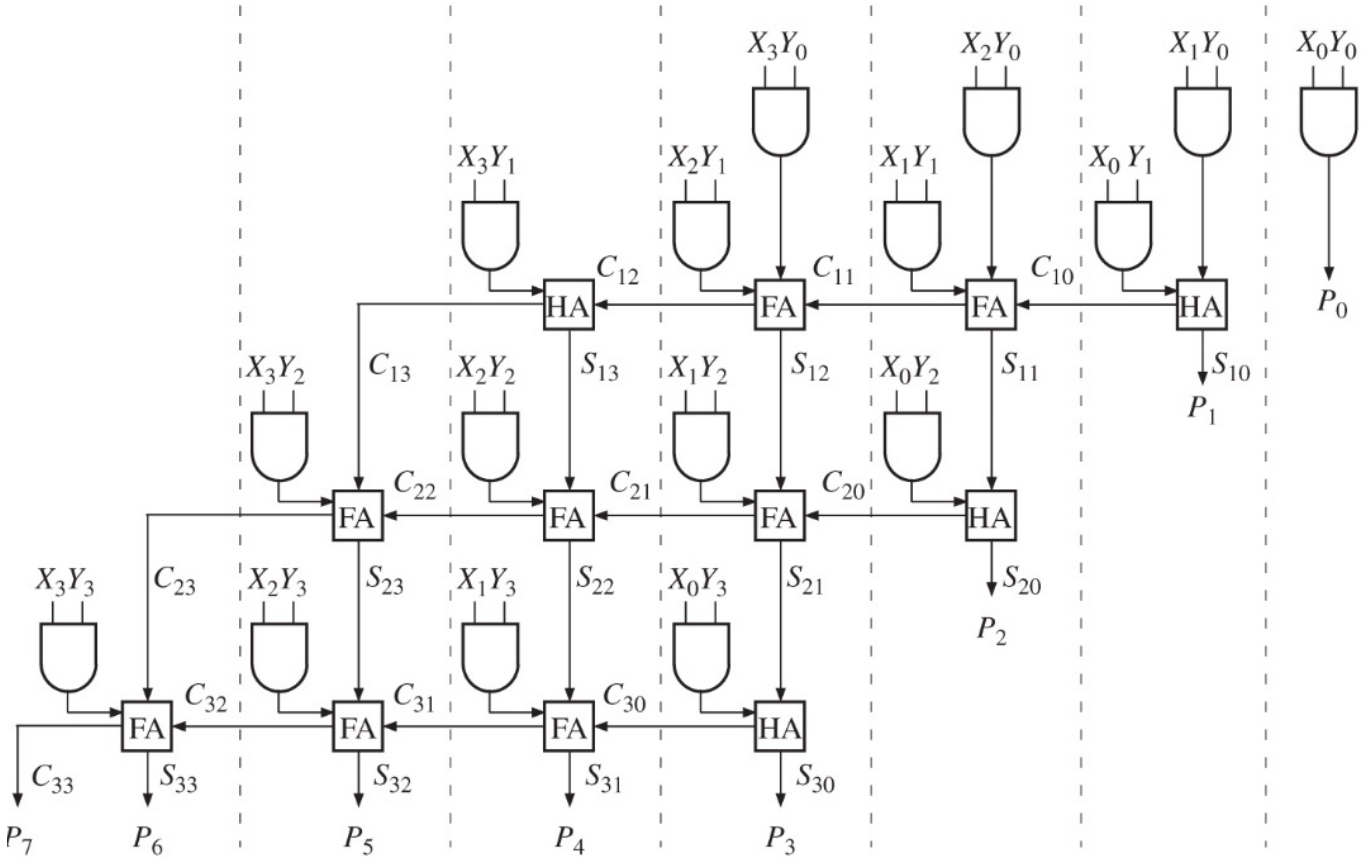
- Array multiplier: a parallel multiplier that generates the partial products in a parallel fashion. Partial products are added as soon as they are available.

Array Multiplier (continued)

- 4-bit multiplier partial products:

			X_3	X_2	X_1	X_0	Multiplicand
			Y_3	Y_2	Y_1	Y_0	Multiplier
			X_3Y_0	X_2Y_0	X_1Y_0	X_0Y_0	Partial product 0
		X_3Y_1	X_2Y_1	X_1Y_1	X_0Y_1		Partial product 1
		C_{12}	C_{11}	C_{10}			First row carries
	C_{13}	S_{13}	S_{12}	S_{11}	S_{10}		First row sums
	X_3Y_2	X_2Y_2	X_1Y_2	X_0Y_2			Partial product 2
	C_{22}	C_{21}	C_{20}				Second row carries
C_{23}	S_{23}	S_{22}	S_{21}	S_{20}			Second row sums
X_3Y_3	X_2Y_3	X_1Y_3	X_0Y_3				Partial product 3
C_{32}	C_{31}	C_{30}					Third row carries
C_{33}	S_{33}	S_{32}	S_{31}	S_{30}			Third row sums
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0 Final product

Array Multiplier (continued)



Array Multiplier (continued)

- The longest path (from input to output): 8 adders. If t_{ad} is the worst-case delay through an adder, and t_g is the longest AND gate delay, then the worst-case time to complete the multiplication is $8t_{ad} + t_g$.
- An n-bit-by-n-bit array multiplier requires n^2 AND gates, $n(n - 2)$ full adders, and n half-adders.

Array Multiplier (continued)

- For an $n \times n$ array multiplier, the longest path from input to output goes through n adders in the top row, n-1 adders in the bottom row and n-3 adders in the middle rows.
  - Worst-case multiply time: $(3n - 4)t_{ad} + t_g$.
  - The longest delay in a circuit: **critical path**.
  - Worse-case can be improved to $2nt_{ad} + t_g$ by forwarding carry from each adder to the diagonally lower adder.