Lecture 11-1 FPGA

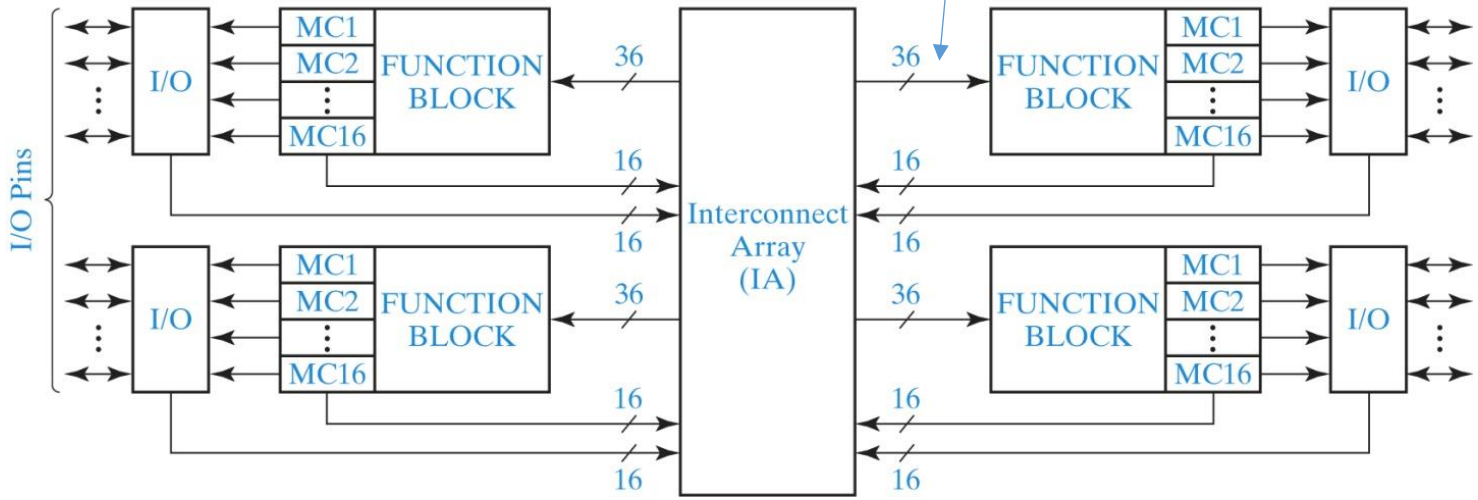
- We have finished combinational circuits, and learned registers. Now are ready to see the inside of an FPGA.

Complex Programming Logic Devices

- Xilinx XCR3064XL CPLD
  - Function block (16 macrocells)= PLA
  - Macrocell = a flip flop + multiplexers
  - IA routes signals

Input of function block

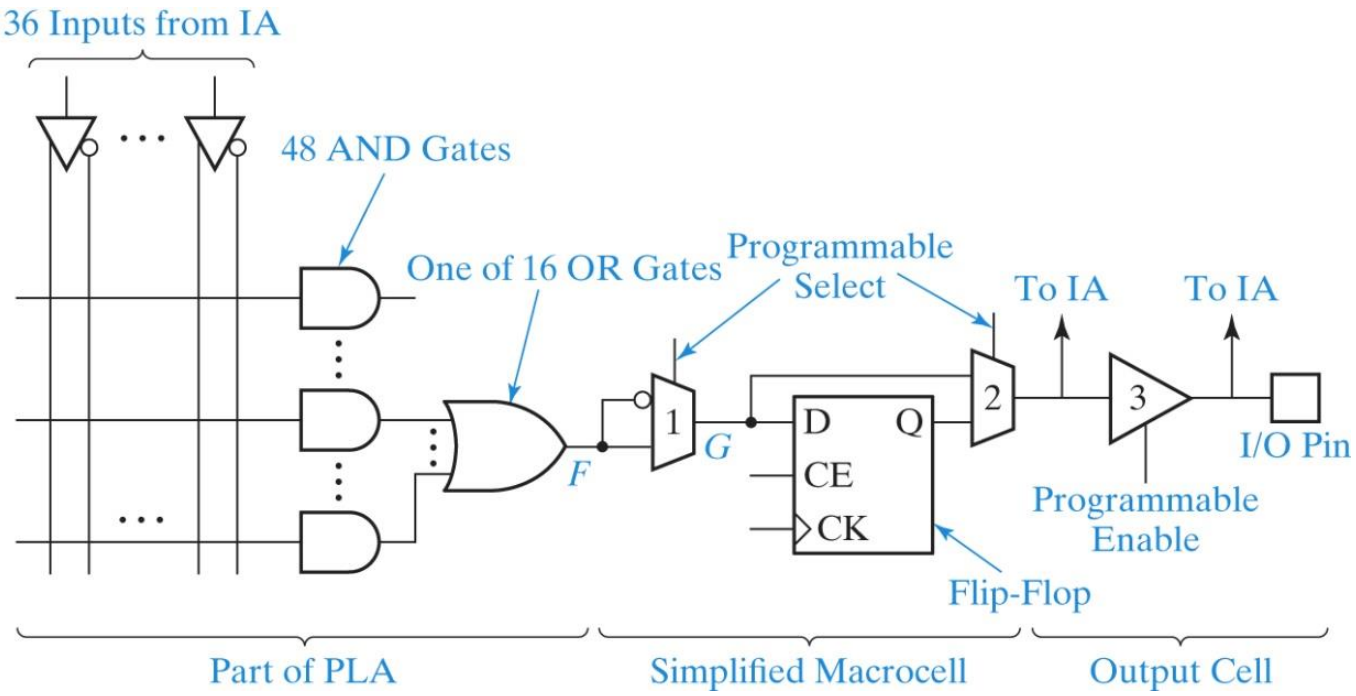
FIGURE 9-34 Architecture of Xilinx XCR3064XL CPLD (Figure based on figures and text owned by Xilinx, Inc., Courtesy of Xilinx, Inc. © Xilinx, Inc. 1999–2003. All rights reserved.)



Function Block and MC

- Signal from PLA -> marcocell -> I/O pin
- Use CAD tool to fit the design into the PLD.

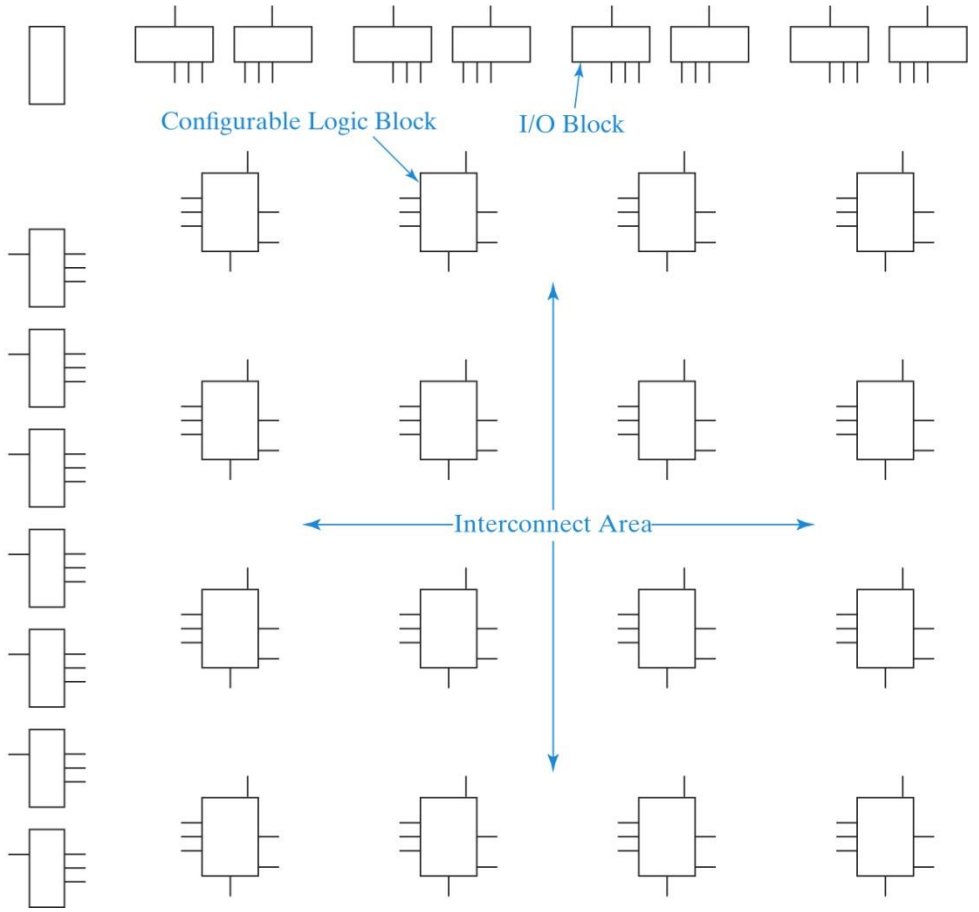
FIGURE 9-35
CPLD Function Block and Macrocell (A Simplified Version of XCR3064XL)



Field Programmable Gate Arrays (FPGA)

- Logic cell: configurable logic blocks (CLBs)
- Input/Output blocks (I/O blocks)

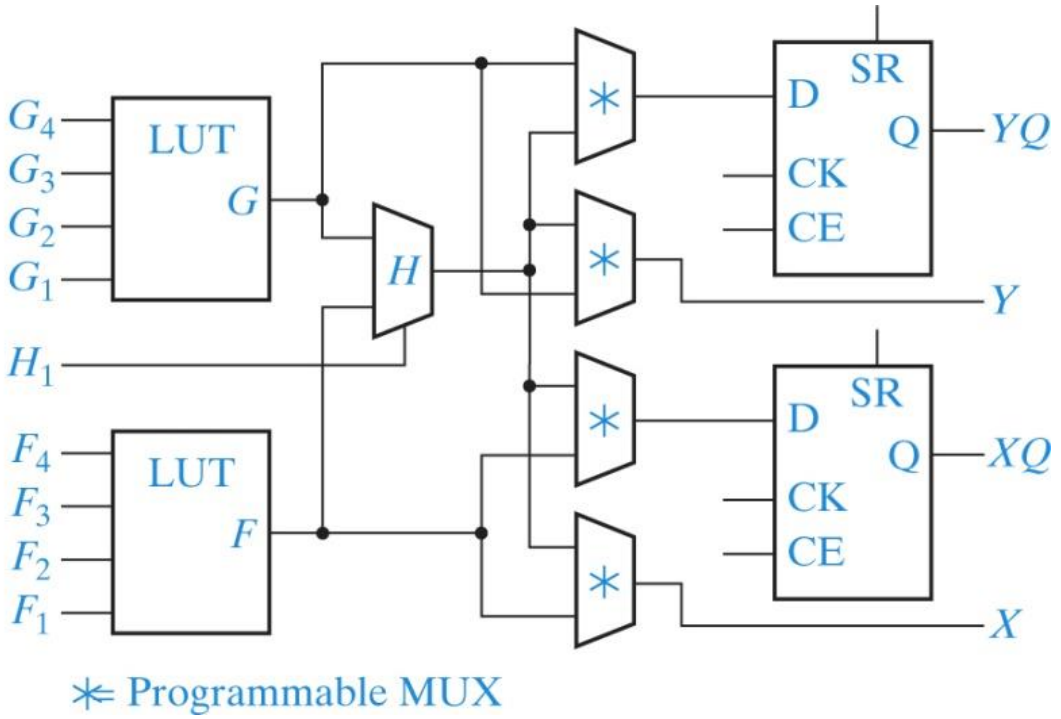
FIGURE 9-36
Layout of a Typical FPGA



Configurable Logic Block

- Inside a CLB: function generators (LUT), FFs, and MUXs
- LUT: lookup table (truth table) is a reprogrammable ROM (16 1-bit words)

FIGURE 9-37
Simplified Configurable Logic Block (CLB)

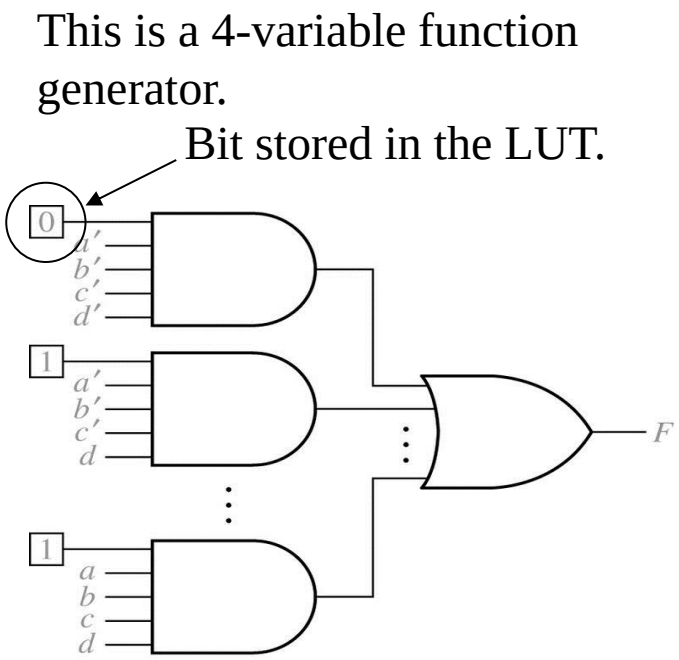


A Lookup Table (LUT)

- If we want F = abc (one minterm)
  - 1110 (and F=1) + 1111 (and F=1)
- Or if we want F = a'b'c'd' + a'b'cd' + …abcd. (15 minterms)
- Require a single function generator. Program the LUT table to get what we want.

To be implemented

a b c d	F
0 0 0 0	0
0 0 0 1	1
0 0 1 0	1
…..	
1 1 1 1	1



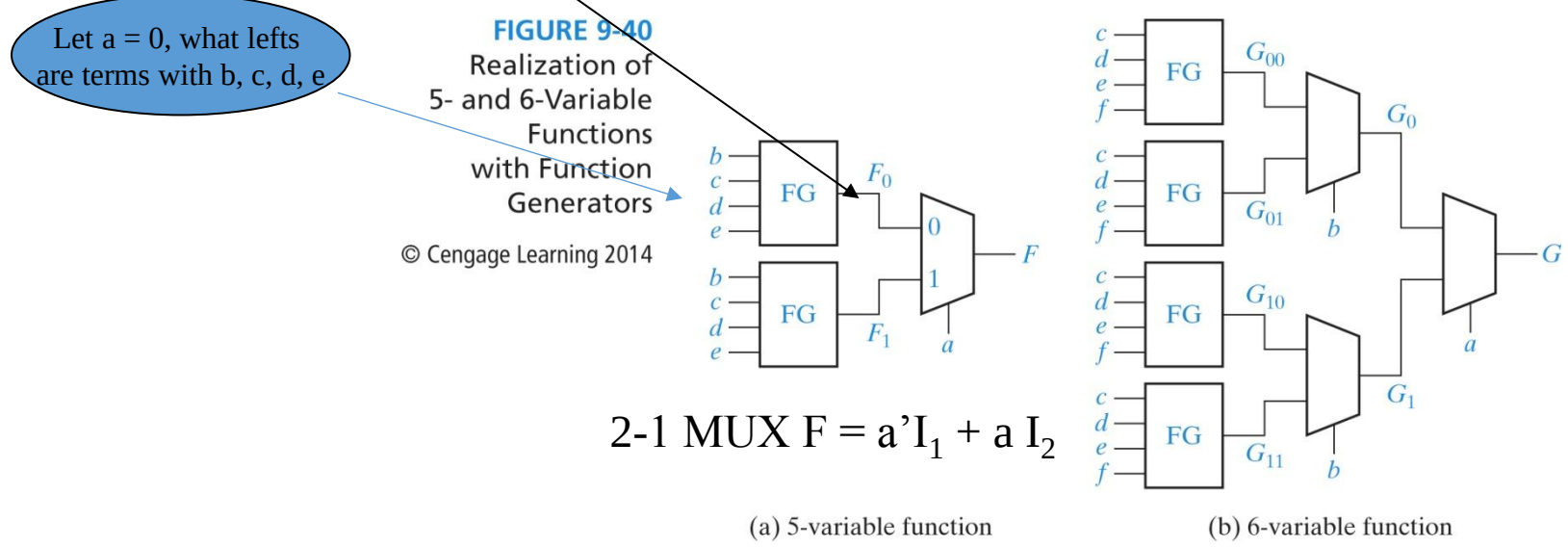
Application of Shannon's Expansion Theorem

- What if # of variables > 4 variables

$$f(x_1, x_2, \ldots x_n)$$
$$= x_i' f(x_1, x_2, ..x_{i-1}, 0, x_{i+1}, \ldots x_n) +$$
$$x_i f(x_1, x_2, ..x_{i-1}, 1, x_{i+1}, \ldots x_n)$$
$$= x_i' f_0 + x_i f_1$$

$$f(a, b, c, d, e) = a' f(0, b, c, d, e) + a f(1, b, c, d, e)$$

Let a = 0, what lefts are terms with b, c, d, e



2-1 MUX $F = a'I_1 + a\,I_2$

(a) 5-variable function (b) 6-variable function

FIGURE 9-40 Realization of 5- and 6-Variable Functions with Function Generators

Shannon's Expansion Theorem: a majority function

$f = x_1x_2 + x_1x_3 + x_2x_3$

$= \sim x_1(x_2x_3) + x_1(x_2+x_3+x_2x_3)$; //let $x_1 = 0$ for f, and let $x_1 = 1$ for f

$= \sim x_1(x_2x_3) + x_1(x_2+x_3)$

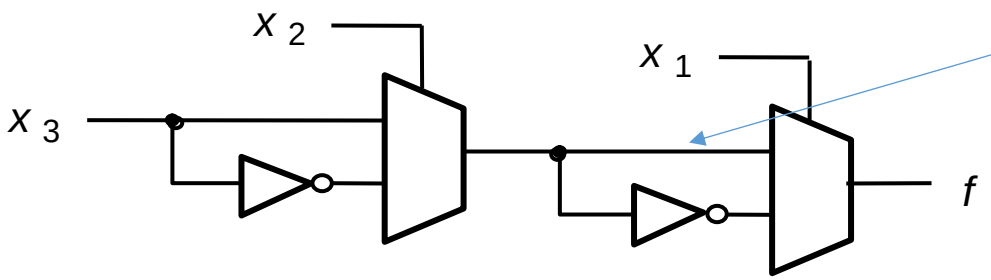
What is the circuit using a 2-1 MUX for x_1 ?

Shannon's Expansion Theorem: XOR

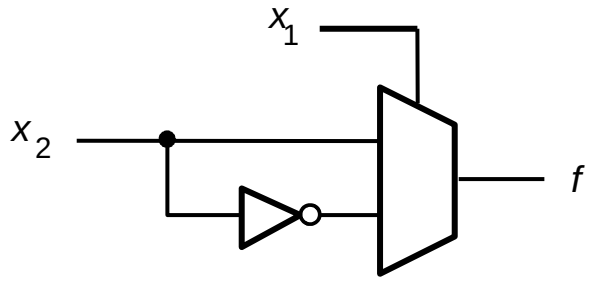
- Ex: $\mathbf{f = x_1 \oplus x_2 \oplus x_3}$

$\mathbf{= \sim x_1(0 \oplus x_2 \oplus x_3) + x_1(1 \oplus x_2 \oplus x_3)}$

$\mathbf{= \sim x_1(x_2 \oplus x_3) + x_1 \sim (x_2 \oplus x_3)}$



x_1	x_2	x_3	f	
0	0	0	0	
0	0	1	1	$x_2 \oplus x_3$
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	$\overline{x_2 \oplus x_3}$
1	1	0	0	
1	1	1	1	



f = x1 ⊕ **x2**

Shannon's expansion for more than two variables

$f(x_1, x_2, \ldots x_n)$

$= x_i' f(x_1, x_2, ..x_{i-1}, 0, x_{i+1}, \ldots x_n) + x_i f(x_1, x_2, ..x_{i-1}, 1, x_{i+1}, \ldots x_n)$

$= x_i' f_0 + x_i f_1$

Say for x_1 and x_2;

$f(x_1, x_2, \ldots x_n)$

$= x_1' x_2' f(0, 0, x_3 \ldots x_n) + x_1' x_2 f(0, 1, x_3 \ldots x_n) + x_1 x_2' f(1, 0, x_3 \ldots x_n) + x_1 x_2 f(1, 1, x_3 \ldots x_n)$

This expansion can be implemented by a 4-to-1 multiplexer where x_1 and x_2 are the selection signals.