16-Programmable Logic Devices

Text: Unit 9

ECEGR/ISSC 201
Digital Operations and Computations
Winter 2011

Overview

- Introduction
- Programmable Logic Device
- Programmable Logic Arrays
- Complex Programmable Logic Devices
- Field Programmable Gate Arrays

Programmable Logic Devices

- Programmable Logic Device (PLD): generic name for a digital IC that can be programmed to provide a variety of different logical functions
- Simple PLDs:
  - 2-10 functions
  - 4-16 variables
- Complex PLDs:
  - Thousands of gates and flip-flops

Programmable Logic Devices

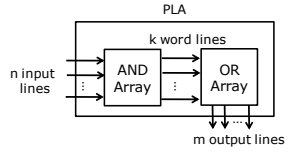
- Benefits of PLDs
  - Versatile
  - Single chip, which makes it inexpensive
  - Conducive to iterative design

Programmable Logic Array

- Programmable Logic Array (PLA)
- Same basic function as ROM
  - n inputs
  - m outputs
  - m functions of n variables
- Mask-programmable PLAs and Field-programmable PLAs are available
- For small numbers of variables, PROM may be more economical than PLA

Programmable Logic Array

- Recall that any function can be realized using AND and OR gates
- PLAs use SoP expressions

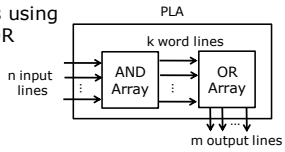


Programmable Logic Array

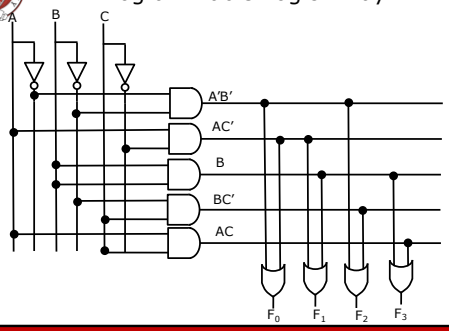
- Consider a PLA with
  - 3 inputs
  - 4 outputs
  - 5 unique product terms
- $F_0 = A'B' + AC'$
- $F_1 = AC' + B$
- $F_2 = A'B' + BC'$
- $F_3 = B + AC$

Programmable Logic Array

- We want to build this using AND gates fed into OR gates



Programmable Logic Array



Programmable Logic Array

- PLA Table for
  - $F_0 = A'B' + AC'$
  - $F_1 = AC' + B$
  - $F_2 = A'B' + BC'$
  - $F_3 = B + AC$

Product Term	Inputs A B C	Outputs F_0 F_1 F_2 F_3
A'B'	0 0 -	1 0 1 0
AC'	1 – 0	1 1 0 0
B	- 1 -	0 1 0 1
BC'	- 1 0	0 0 1 0
AC	1 - 1	0 0 0 1

Programmable Logic Array

- Design a PLA for
  - $F_0 = A'BD + ABD + AB'C' + B'C$
  - $F_1 = C + A'BD$
  - $F_2 = BC + AB'C' + ABD$
- Start by completing the PLA table

Product Term	Inputs A B C D	Outputs F_0 F_1 F_2

Programmable Logic Array

- $F_0 = A'BD + ABD + AB'C' + B'C$
- $F_1 = C + A'BD$
- $F_2 = BC + AB'C' + ABD$

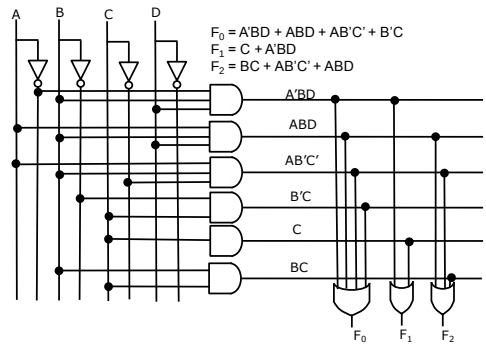
Product Term	Inputs A B C D	Outputs F_0 F_1 F_2
A'BD	0 1 – 1	1 1 0
ABD	1 1 - 1	1 0 1
AB'C'	1 0 0 -	1 0 1
B'C	- 0 1 -	1 0 0
C	- - 1 -	0 1 0
BC	- 1 1 -	0 0 1

Programmable Logic Array

- Next, draw the PLA
  - How many inputs?
  - How many outputs?
  - How many word lines?

Programmable Logic Array

- Next, draw the PLA
  - How many inputs? 4
  - How many outputs? 3
  - How many word lines? 6

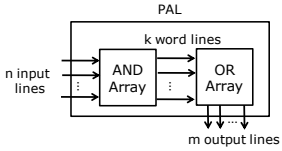


Programmable Array Logic

- Programmable Array Logic (PAL)
- Special case of PLA
  - AND is programmable
  - OR is fixed
- Less expensive than PLA
- Easier to program
- Less flexibility

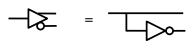
Programmable Array Logic

- Basic PAL configuration is the same as a PLA
- The number of AND gates fed to each OR gate is fixed
- AND terms are not shared by OR gates

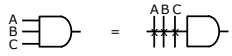


Programmable Array Logic

- Each PAL input must drive many AND gates
- Buffers must be used
- An unprogrammed segment

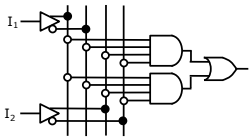


- Notation:



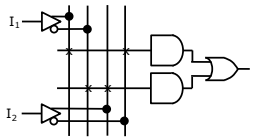
Programmable Array Logic

- An unprogrammed PAL segment:



Programmable Array Logic

- To realize: $I_1I_2' + I_1'I_2$

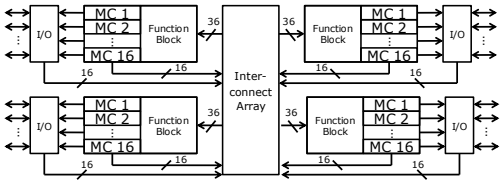


Complex Programmable Logic Devices

- Complex Programmable Logic Devices (CPLDs)
  - Multiple PLAs or PALs on the same chip
  - May contain storage (flip-flops)

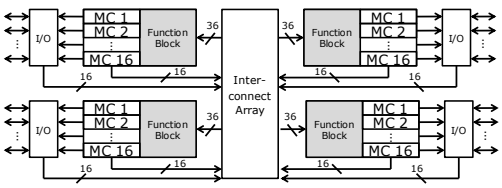
Complex Programmable Logic Devices

- Consider the Xilinx XCR3064XL CPLD



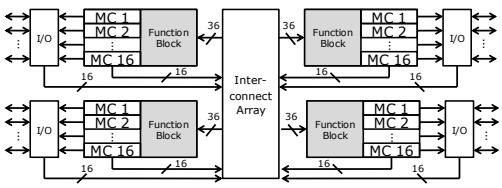
Complex Programmable Logic Devices

- 4 function blocks: each a programmable AND-OR array as in an PLA



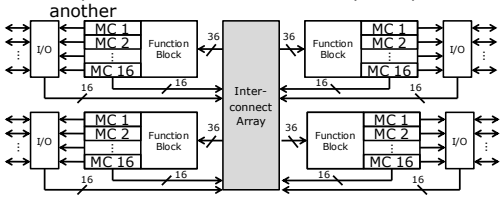
Complex Programmable Logic Devices

- Each has 16 associated macrocells (MC)
  - Route signals from I/O or to the Interconnect Array (IA)



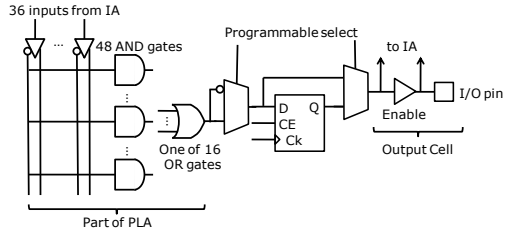
Complex Programmable Logic Devices

- IA: select signals from the MCs or the I/O blocks and connects them back to the function blocks
  - Output from one function block may be input to another



Complex Programmable Logic Devices

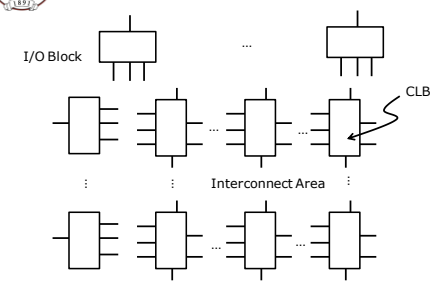
- MC internal



Field Programmable Gate Arrays

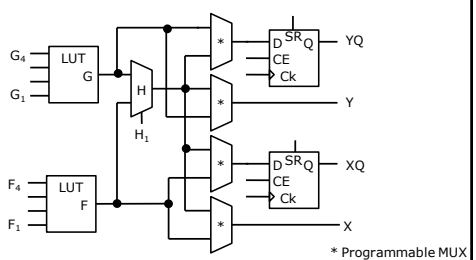
- Field Programmable Gate Arrays (FPGAs)
- FPGA: IC that contains an array of identical logic cells whose interconnections are programmable
- Programmed using a Hardware Description Language (HDL)
- Logic cells: also called configurable logic block (CLB)

Field Programmable Gate Arrays

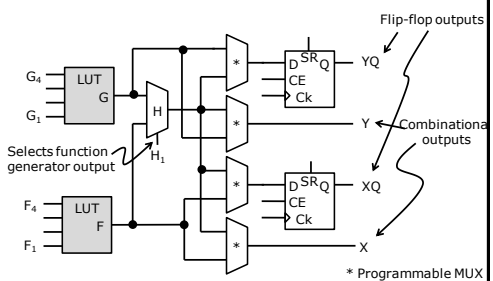


Field Programmable Gate Arrays

- Simplified version of a CLB



Field Programmable Gate Arrays

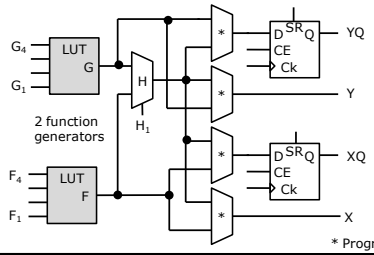


Field Programmable Gate Arrays

- Programmable MUX: inputs are programmed when the FPGA is configured
- For example:
  - X could come from F function generator
  - Y could come from the H MUX

Field Programmable Gate Arrays

- Simplified version of a CLB



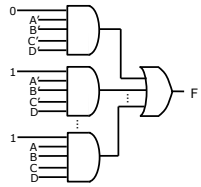
Field Programmable Gate Arrays

- Function generators: realized as a Look Up Table (LUT)
- 4 input LUT is essentially a reprogrammable ROM with 16 1-bit words
  - Stores the truth table of the function being generated

Field Programmable Gate Arrays

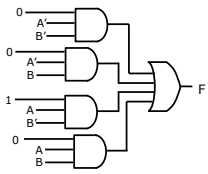
- Implementation of a 4-input look-up table
  - F = A + B + C + D

A B C D	F
0 0 0 0	0
0 0 0 1	1
	. . .
1 1 1 1	1



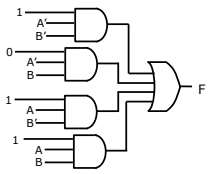
Field Programmable Gate Arrays

- Implementation in a 2-input look-up table
  - F = AB'



Field Programmable Gate Arrays

- Implementation in a 2-input look-up table
  - F = A + B'



Field Programmable Gate Arrays

- To implement an expression of more than four variables using a four input LUT, the expression must be decomposed
- Shannon's expansion theorem
  - F(A, B, C, D) = A'F(0, B, C, D) + AF(1, B, C, D)
  - F_0 = A'F(0, B, C, D)
  - F_1 = AF(1, B, C, D)
  - F = A'F_0 + AF_1

Field Programmable Gate Arrays

- Example:
  - F(A, B, C, D) = C'D' + A'B'C + BCD + AC'
  - F = A'(C'D' + 1B'C + BCD + 0C') + A(C'D' + 0B'C + BCD + 1C')
  - F = A'(C'D' + B'C + BCD) + A(C'D' + BCD +C')
  - F = A'(C'D' + B'C + BCD) + A(BD +C')
  - F_0 = C'D' + B'C + BCD
  - F_1 = BD +C'

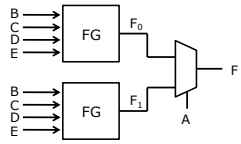
Field Programmable Gate Arrays

- Using a K-map

AB \ CD	A = 0: 00	01	A = 1: 11	10
00	1	1	1	1
01	0	0	1	1
11	1	1	1	0
10	1	0	0	0
	F_0		F_1	

Field Programmable Gate Arrays

- Implementation of a 5-variable function using 4 input function generators



Field Programmable Gate Arrays

- What about functions of more than 5 variables?
- Simply apply Shannon's expansion again

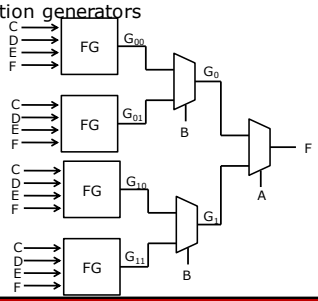
G(A,B,C,D,E,F) = A'G(0,B,C,D,E,F) + AG(1,B,C,D,E,F) = A'G_0 + AG_1

G0=B'(0,0,C,D,E,F) + B(0,1,C,D,E,F) = B'G_{00} + BG_{01}

G1=B'(1,0,C,D,E,F) + B(1,1,C,D,E,F) = B'G_{10} + BG_{11}

Field Programmable Gate Arrays

implementation of a 6-variable function using 4 input function generators



Field Programmable Gate Arrays

- In general any n variable function (where n > 4) can be realized using 2^{n-4} function generators and one 2^{n-4} to-1 MUX