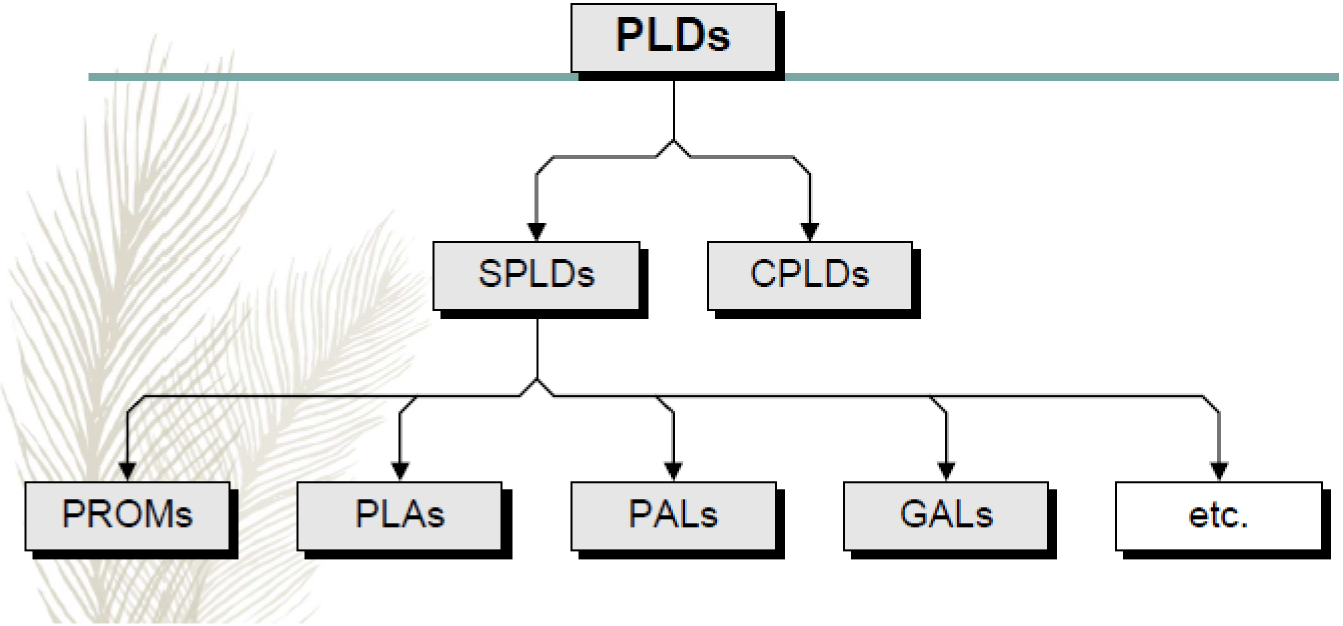
MODULE - 7

Programmable Logic Devices

- General purpose chip for implementing circuits
- Can be customized using programmable switches
- Main types of PLDs
- PLA
- PAL
- CPLD
- FPGA

Programmable Logic Devices

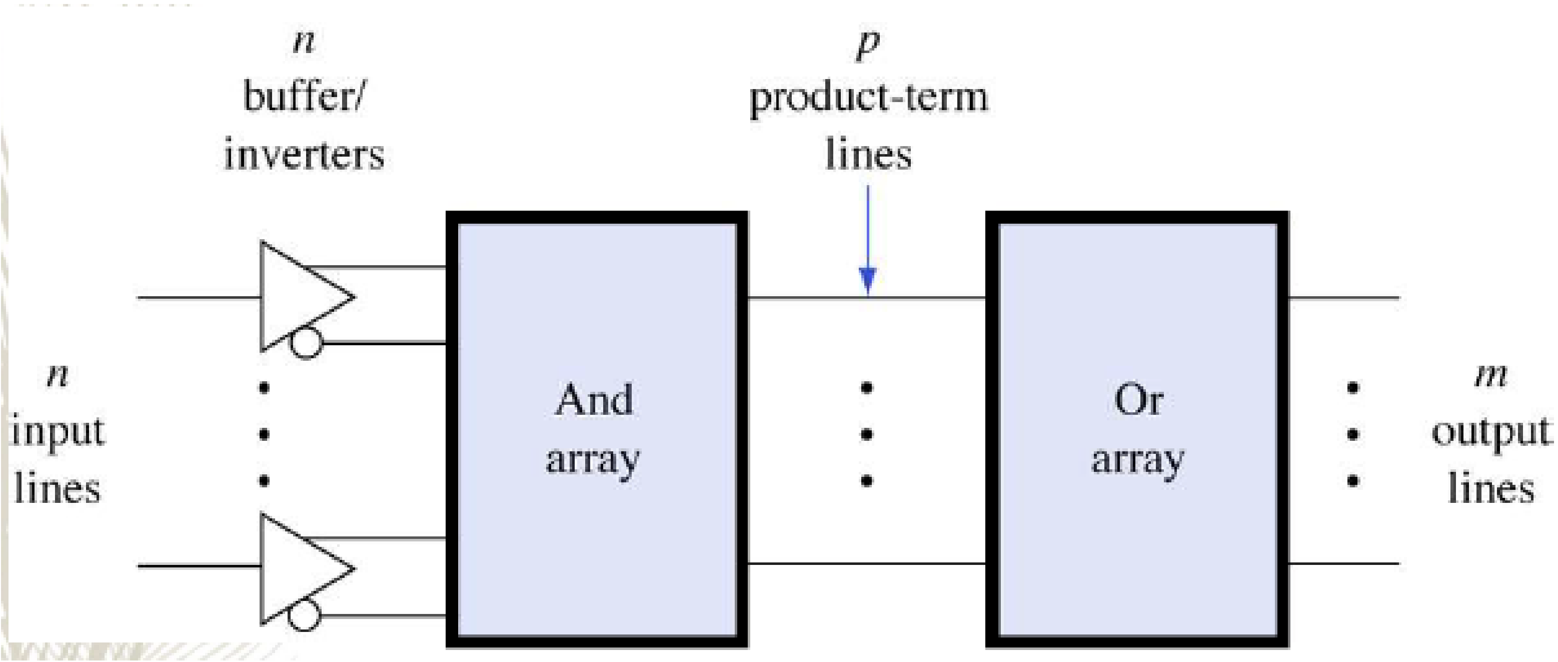


Programmable Logic Devices

- The purpose of a PLD device is to permit elaborate digital logic designs to be implemented by the user in a single device.
- Can be erased electrically and reprogrammed with a new design, making them very well suited for academic and prototyping

- Types of Programmable Logic Devices
- SPLDs (Simple Programmable Logic Devices)
    - ROM (Read-Only Memory)
    - PLA (Programmable Logic Array)
    - PAL (Programmable Array Logic)
    - GAL (Generic Array Logic)
- CPLD (Complex Programmable Logic Device)
- FPGA (Field-Programmable Gate Array)

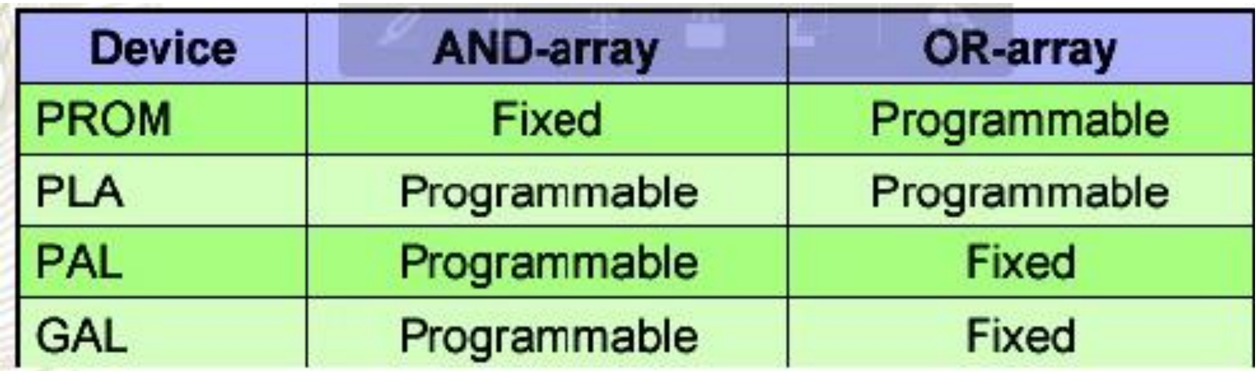
General structure of PLDs



General structure of PLDs

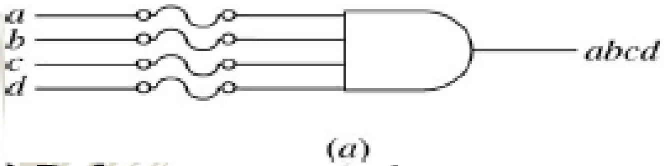
The differences between the first three categories are these:

- In a ROM, the input connection matrix is hardwired. The user can modify the output connection matrix.
- In a PAL/GAL the output connection matrix is hardwired. The user can modify the input connection matrix.
- In a PLA the user can modify both the input connection matrix and the output connection matrix.

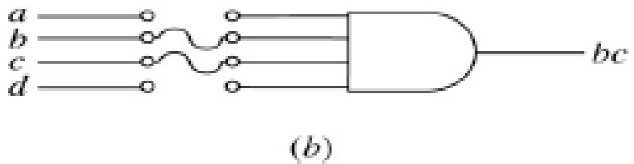


Device	AND-array	OR-array
PROM	Fixed	Programmable
PLA	Programmable	Programmable
PAL	Programmable	Fixed
GAL	Programmable	Fixed

Programming by blowing fuses

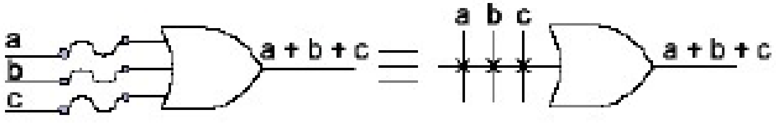


(*a*) Before programming.

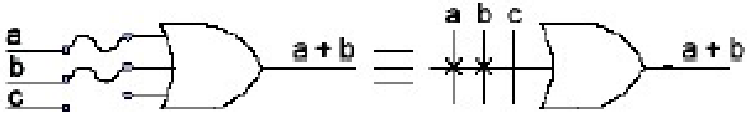


(*b*) After programming.

OR - PLD Notation

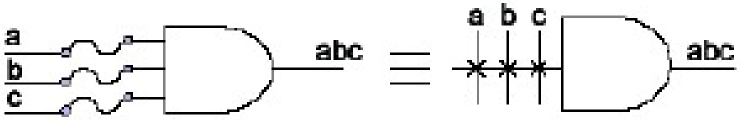


OR gate before programming

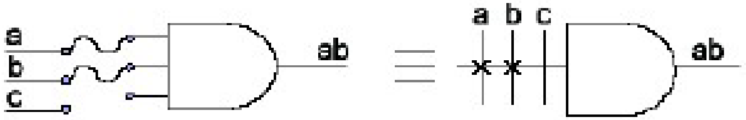


OR gate after programming

AND - PLD Notation

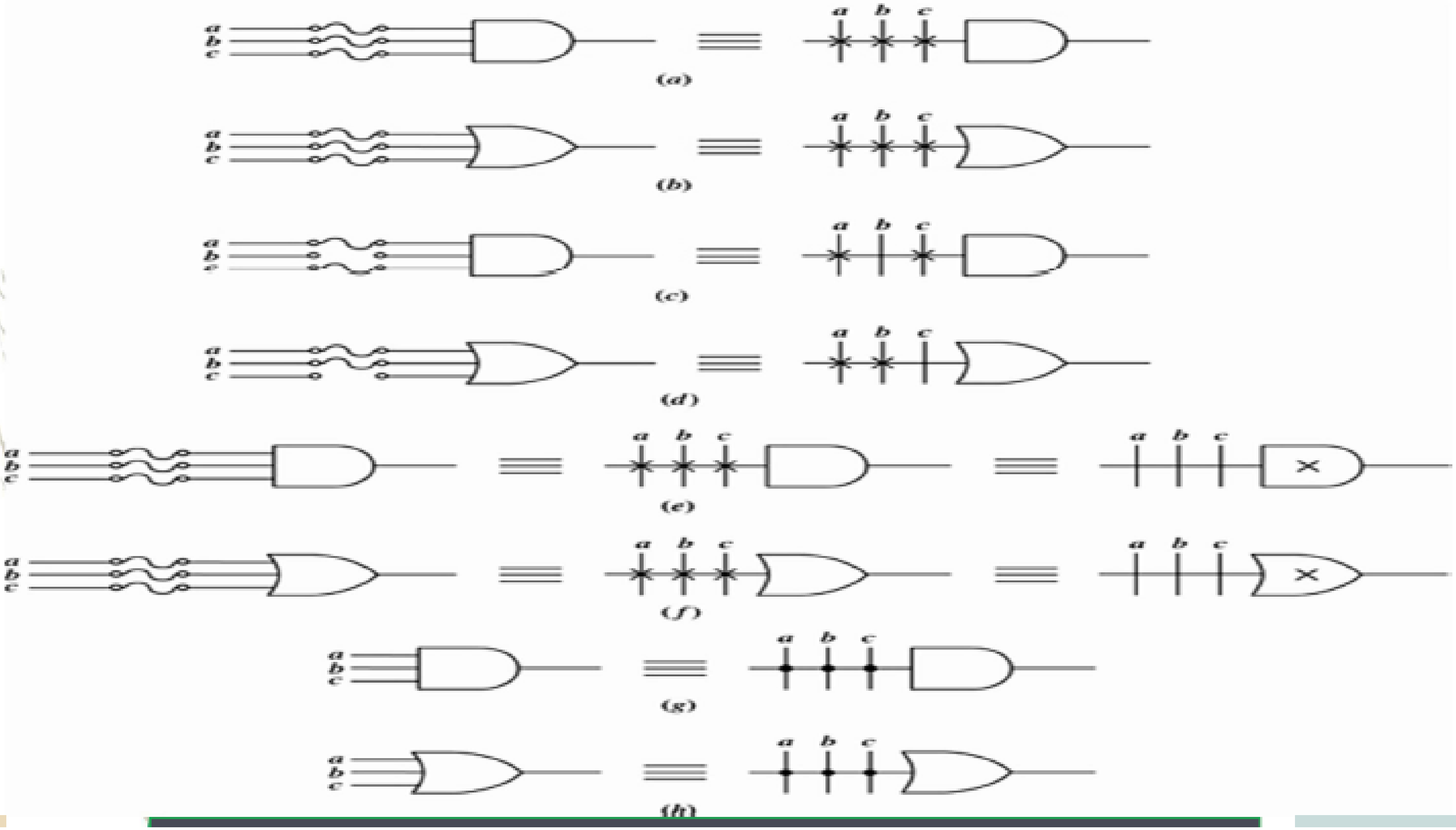


AND gate before programming



AND gate after programming

Programming by blowing fuses



(a)

(b)

(c)

(d)

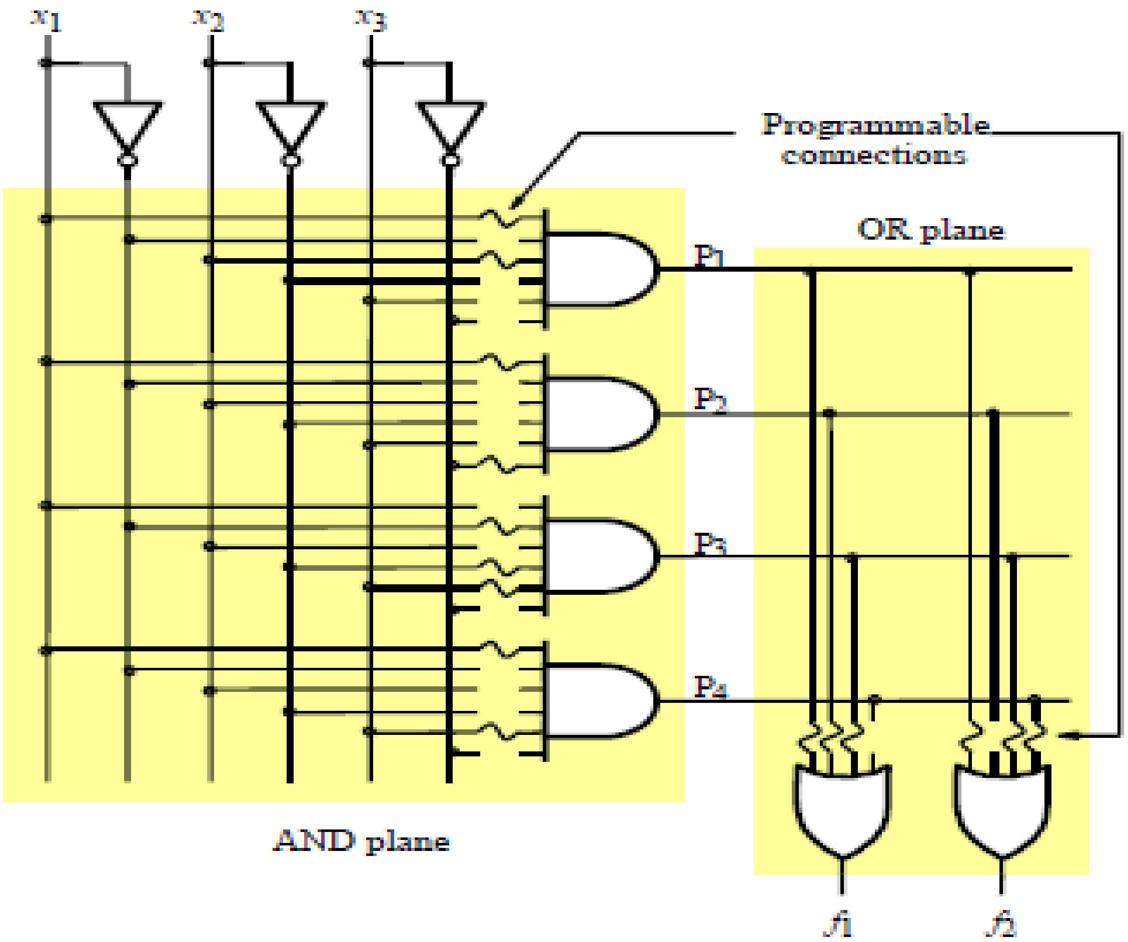
(e)

(f)

(g)

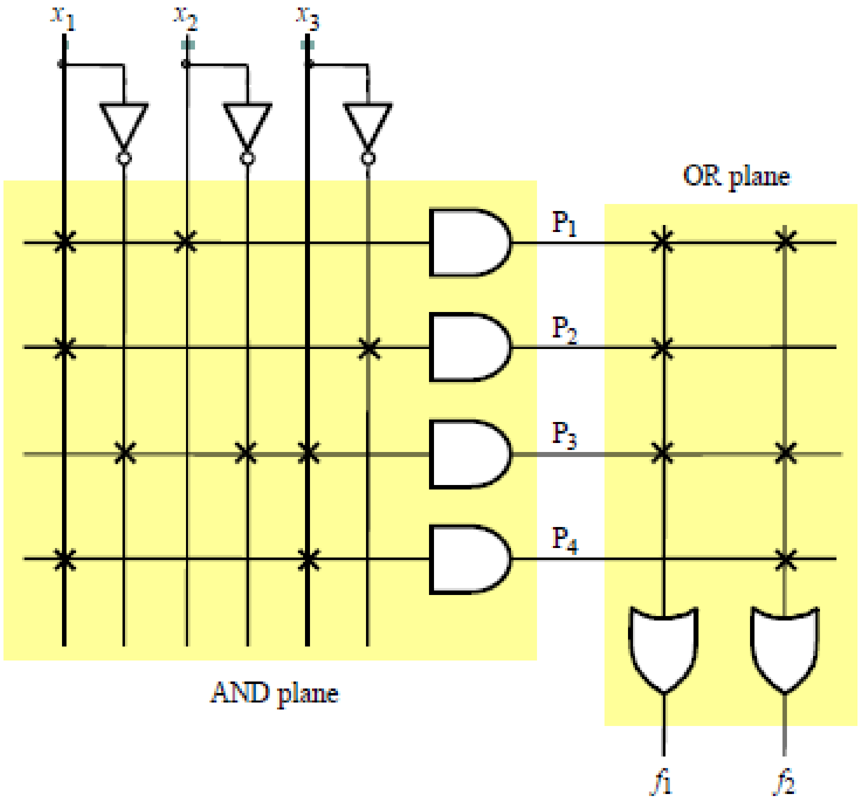
(h)

Gate Level Version of PLA



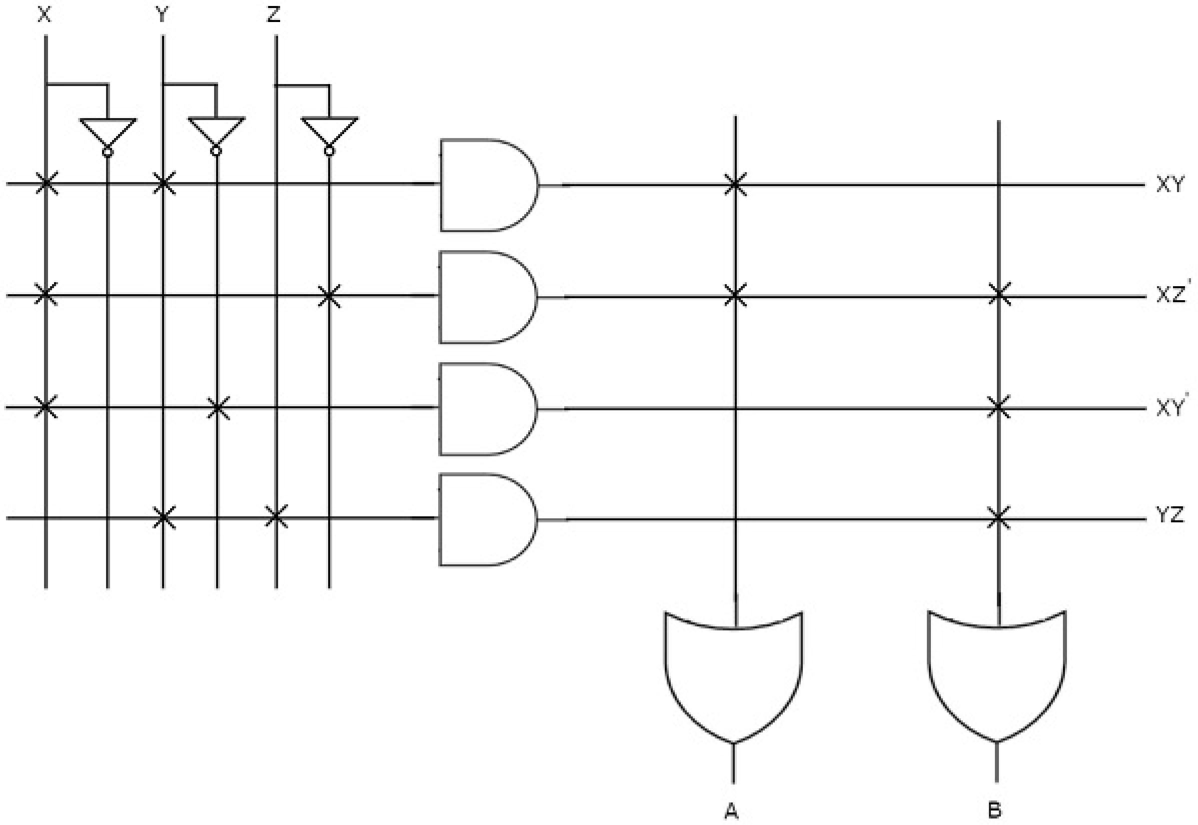
Function Implementation in PLA

f1 = x1x2+x1x3'+x1'x2'x3 & f2 = x1x2+x1'x2'x3+x1x3



Example

- A=XY+XZ'
- B=XY'+YZ+XZ'



Ex2

- F1= AB’+AC+A’BC’
- F2=(AC+BC)’

Ex3

- $F1(A,B,C)=\sum(0,1,2,4)$
- $F2(A,B,C)=\sum(0,5,6,7)$

Ex4

- F=$\sum(1,2,4,5,7)$ USING pla

- Ex5
- F=$\sum(0,1,3,5,7)$

Ex6

- $F=\sum m(0,2,6,7,8,9,12,13,14)$

Design for PLA

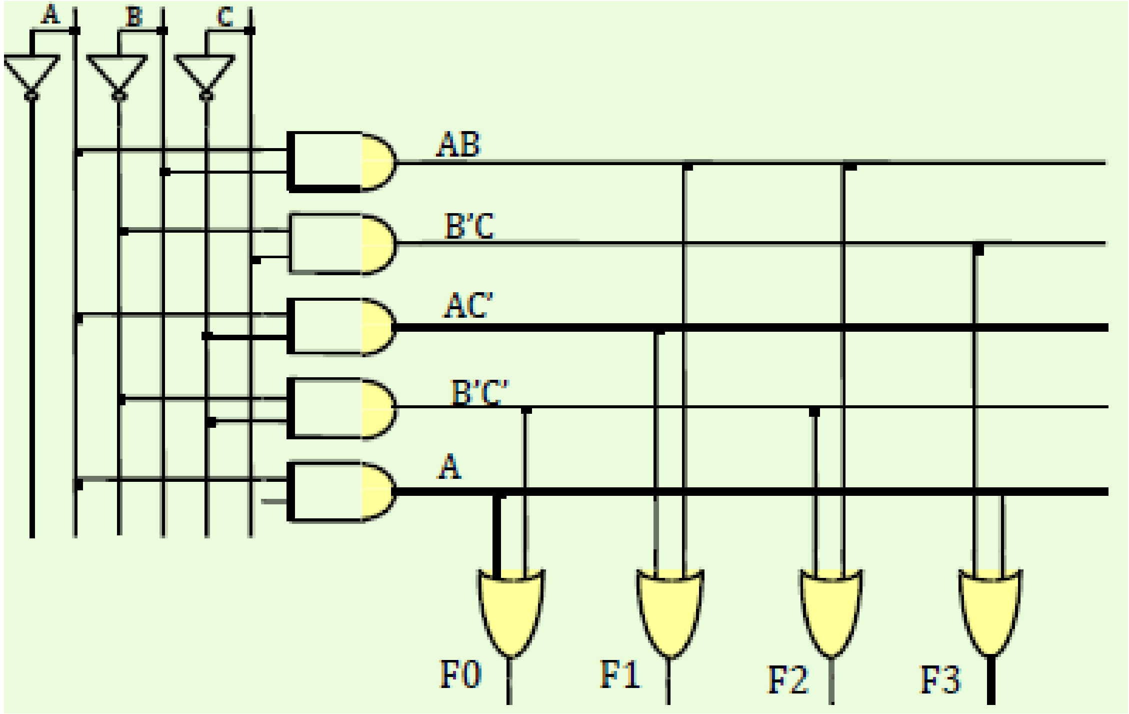
Implement the following functions using PLA

F0 = A + B' C'

F1 = A C' + A B

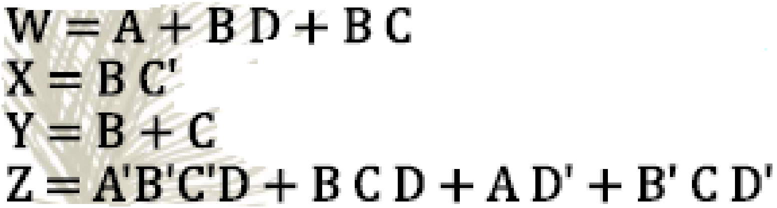
F2 = B' C' + A B

F3 = B' C + A


A
B
C
AB
B'C
AC'
B'C'
A
F0
F1
F2
F3

Design for PLA

Implement the following functions using PLA

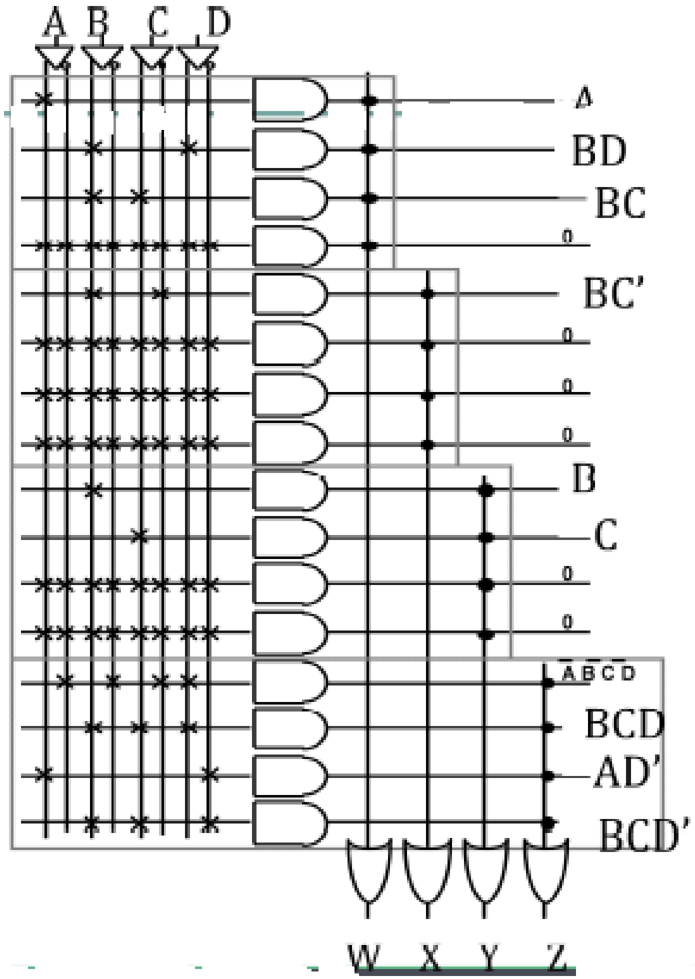


$W = A + B D + B C$

$X = B C'$

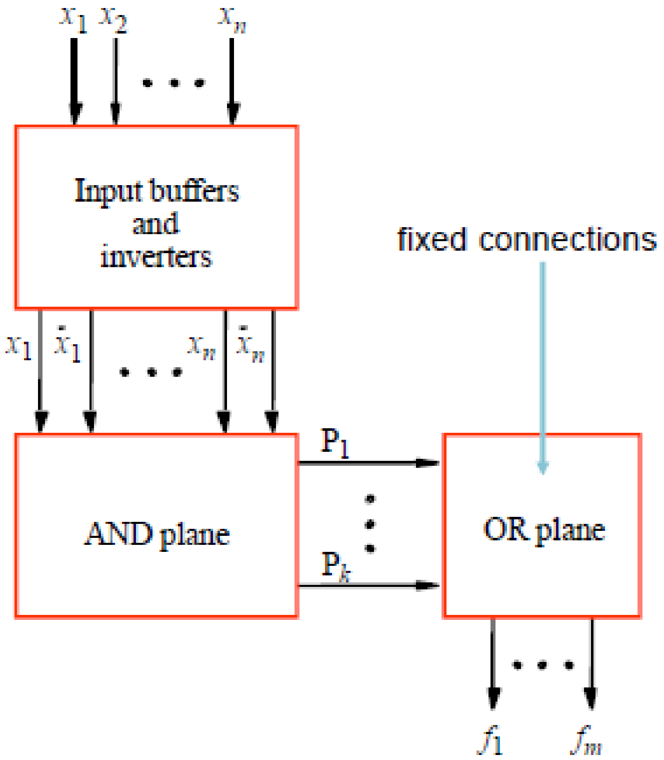
$Y = B + C$

$Z = A'B'C'D + B C D + A D' + B' C D'$


A
B
C
D
A
BD
BC
0
BC’
0
0
0
B
C
0
0
$\bar{A}\bar{B}\bar{C}D$
BCD
AD’
BCD’
W
X
Y
Z

Programmable Array Logic (PAL)

- Also used to implement circuits in SOP form
- The connections in the AND plane are programmable
- The connections in the OR plane are NOT programmable

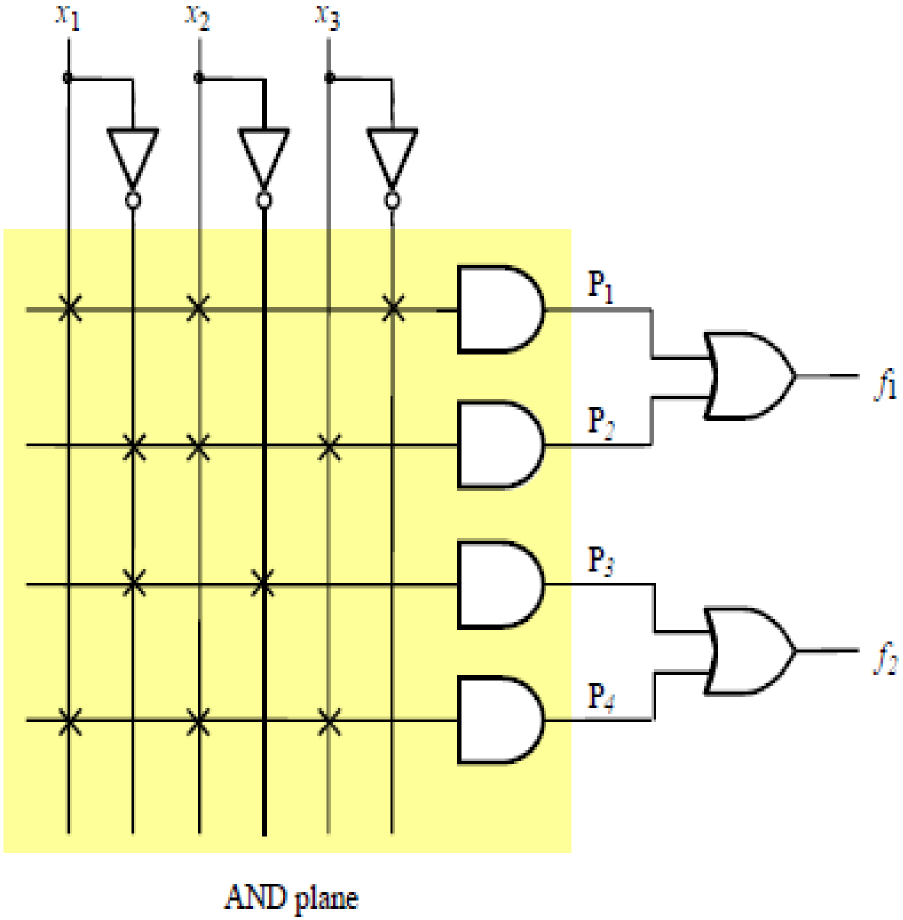


Programmable Array Logic (PAL)

Implement the following functions in PAL

f1 = x1x2x3'+x1'x2x3

f2 = x1'x2'+x1x2x3



Programmable Array Logic (PAL)

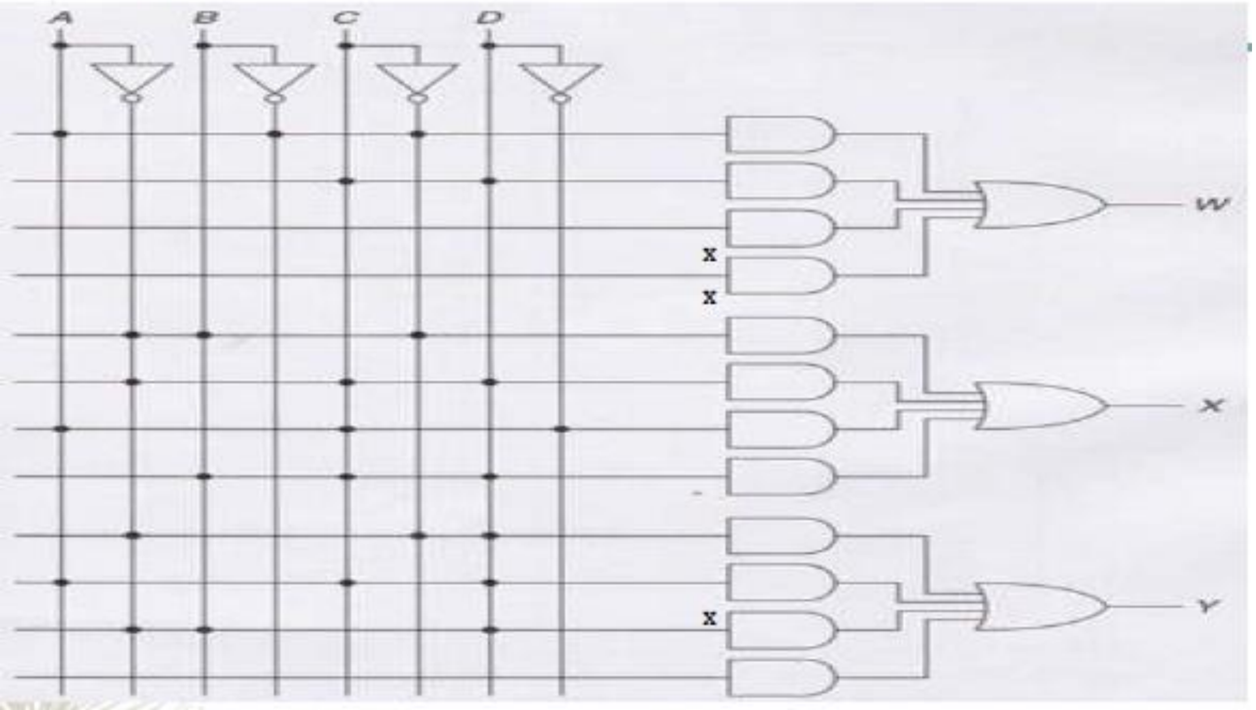
Implement the following functions in PAL

$W = AB'C' + CD$

$X = A'BC' + A'CD + ACD' + BCD$

$Y = A'C D + ACD + A'BD:$

PAL



Comparison of PALs and PLAs

- PALs have the same limitations as PLAs (small number of allowed AND terms) plus they have a fixed OR plane less flexibility than PLAs
- PALs are simpler to manufacture, cheaper, and faster (better performance)
- PALs also often have extra circuitry connected to the output of each OR gate

Ex

- $W(A,B,C,D)=\sum m(2,12,13)$
- $X(A,B,C,D)=\sum m(7,8,9,10,11,12,13,14,15)$
- $Y(A,B,C,D)=\sum m(0,2,3,4,5,6,7,8,10,11,15)$
- $Z(A,B,C,D)=\sum (1,2,8,12,13)$

Ex4

- $F=\sum(1,2,4,5,7)$
- $F=\sum(0,1,3,5,7)$

Ex5

- $F=\sum m(0,2,6,7,8,9,12,13,14)$

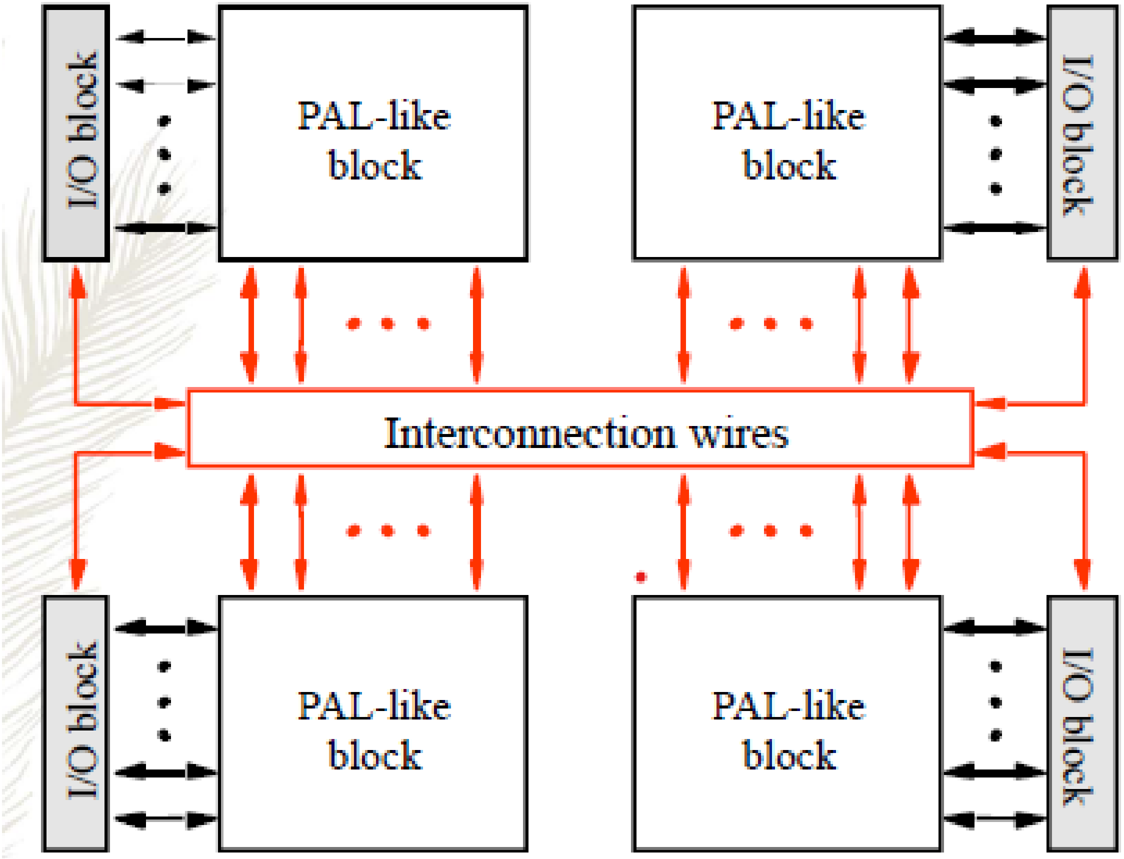
FPGA AND CPLD

- FPGA - Field-Programmable Gate Array.
- CPLD - Complex Programmable Logic Device
- FPGA and CPLD is an advance PLD.
- Support thousands of gate where as PLD only support hundreds of gates.

CPLD

- CPLDs contain multiple circuit blocks on a single chip
- Each block is like a PAL: PAL-like block
- Connections are provided between PAL-like blocks via an interconnection network that is programmable
- Each block is connected to an I/O block as well

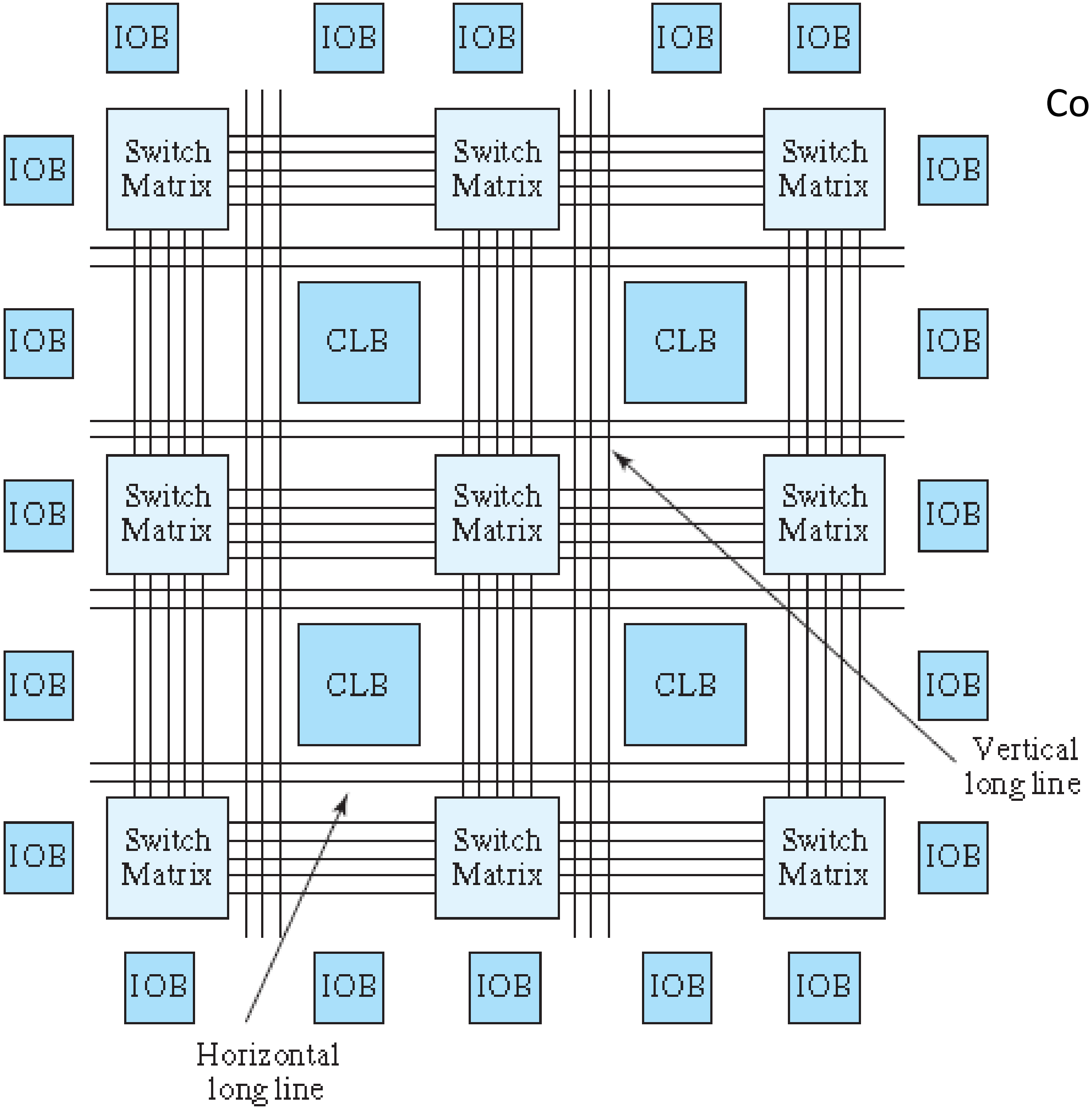
Structure of a CPLD



FPGA

- SPLDs and CPLDs are relatively small and useful for simple logic devices Up to about 20000 gates
- Field Programmable Gate Arrays (FPGA) can handle larger circuits
- No AND/OR planes
- Provide logic blocks, I/O blocks, and interconnection wires and switches
- Logic blocks provide functionality
- Interconnection switches allow logic blocks to be connected to each other and to the I/O pins

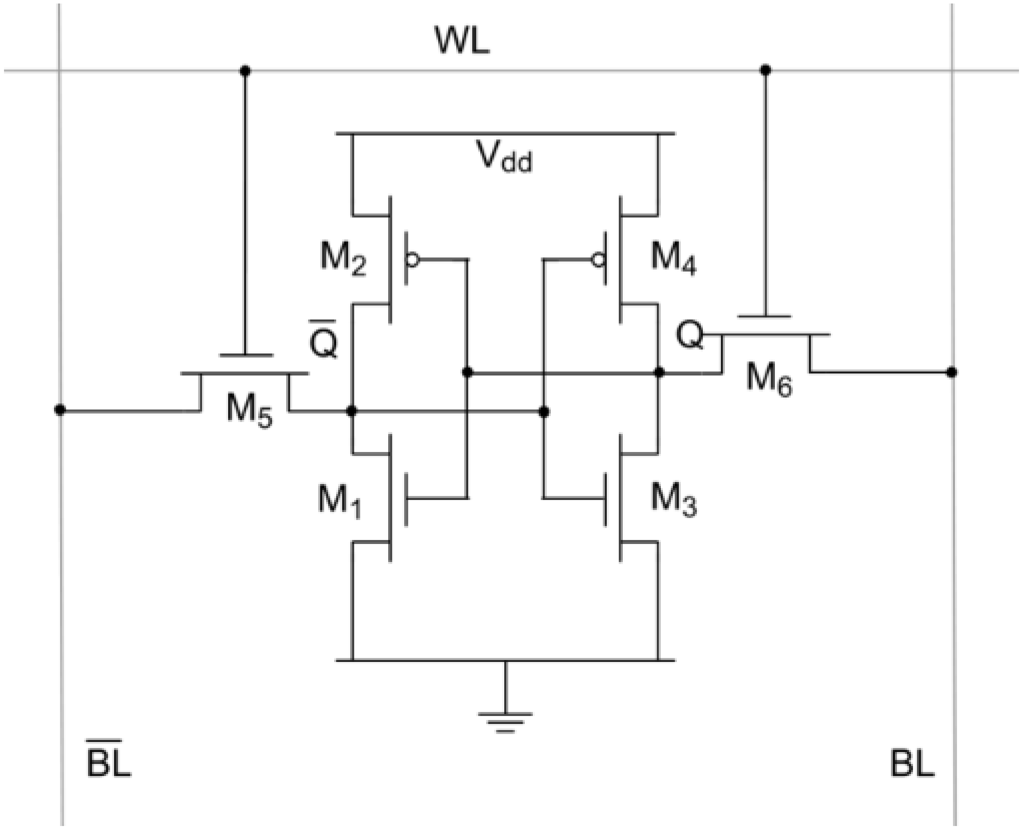
Configurable Logic Block- CLB



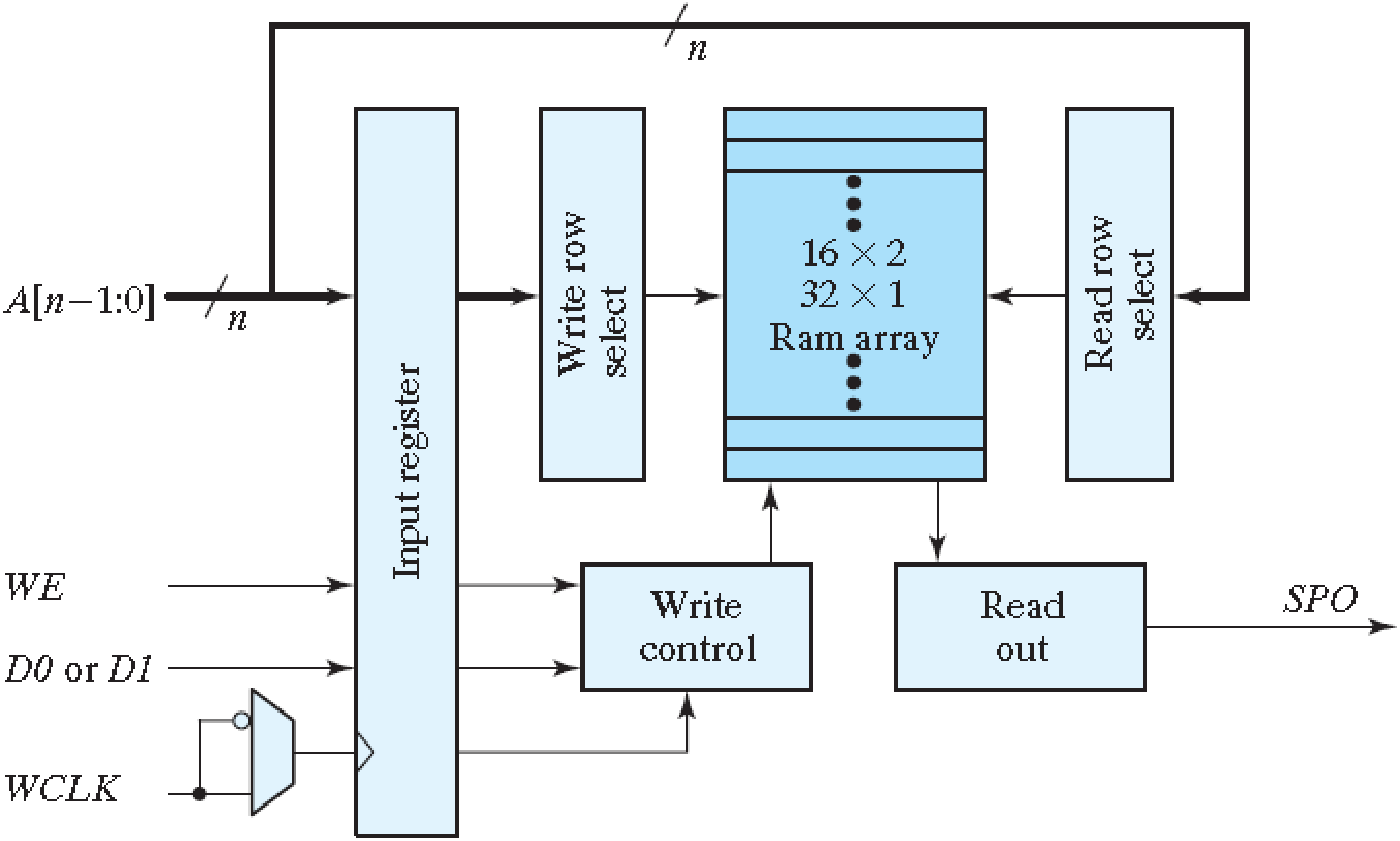
Spartan board configuration

- CLB consists of a
  - programmable lookup table
  - multiplexers
  - registers
  - paths for control signals

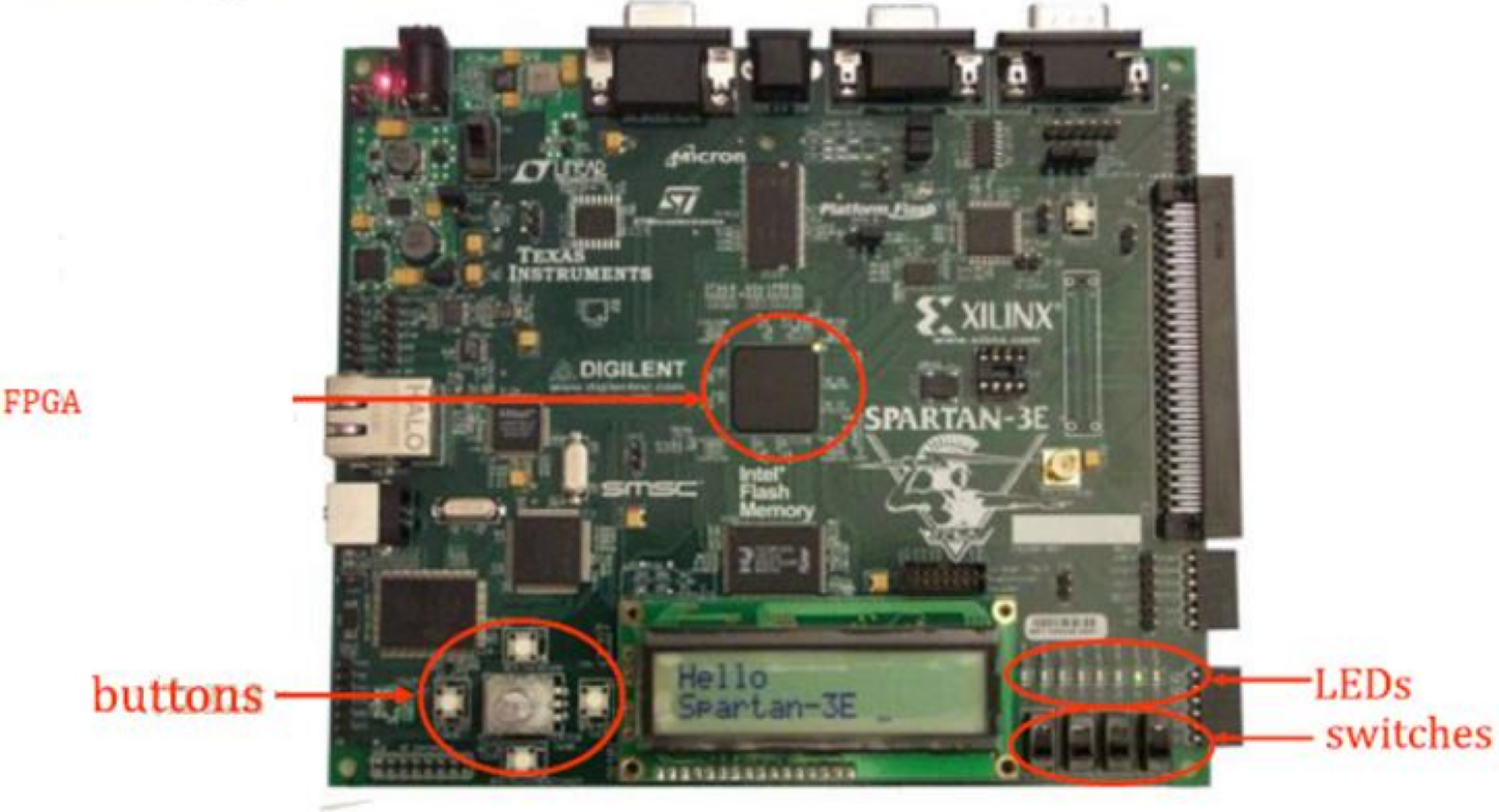
6T SRAM cell

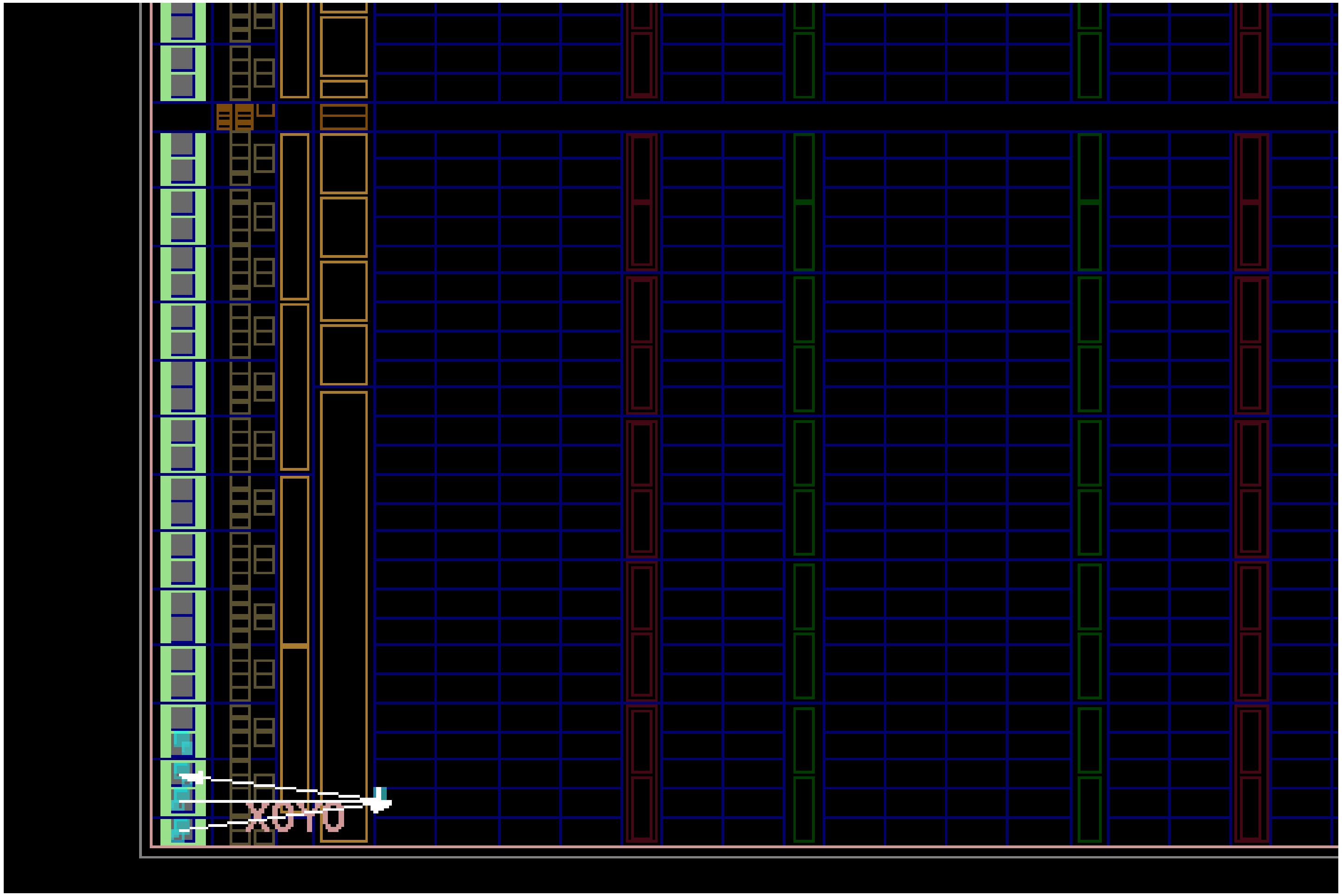


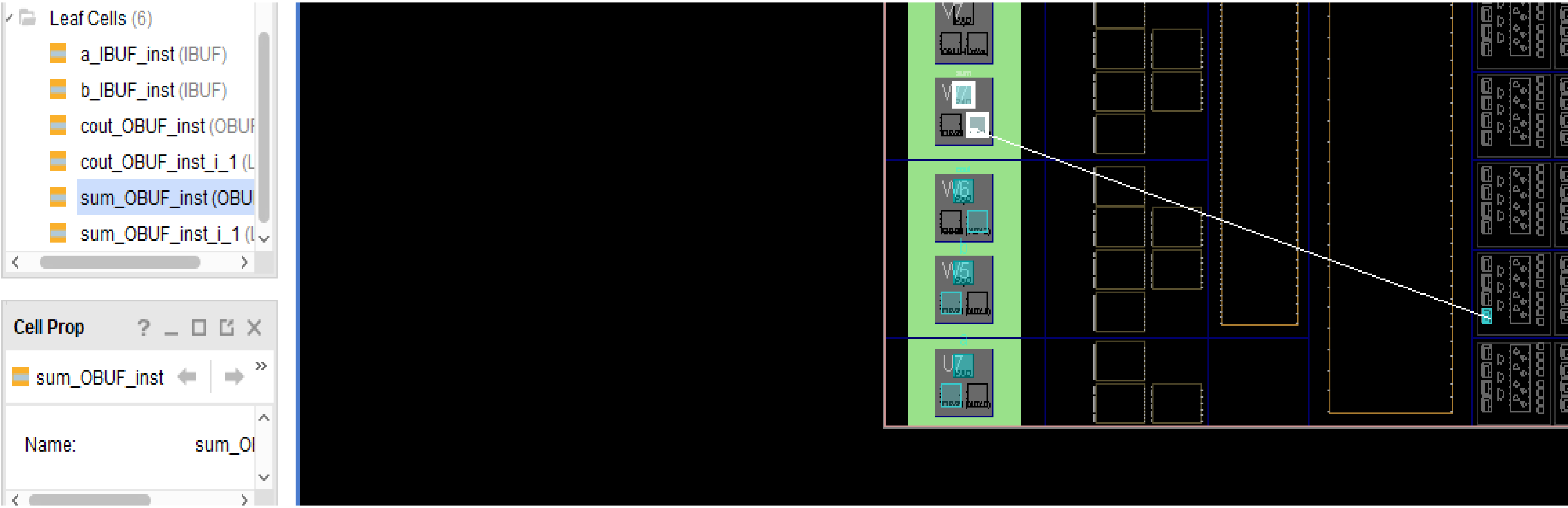
LUTs


n
A[n−1:0]
n
Input register
Write row select
16 × 2
32 × 1
Ram array
Read row select
WE
D0 or D1
WCLK
Write control
Read out
SPO

Xilinx Spartan-3E Starter Kit




X0Y0



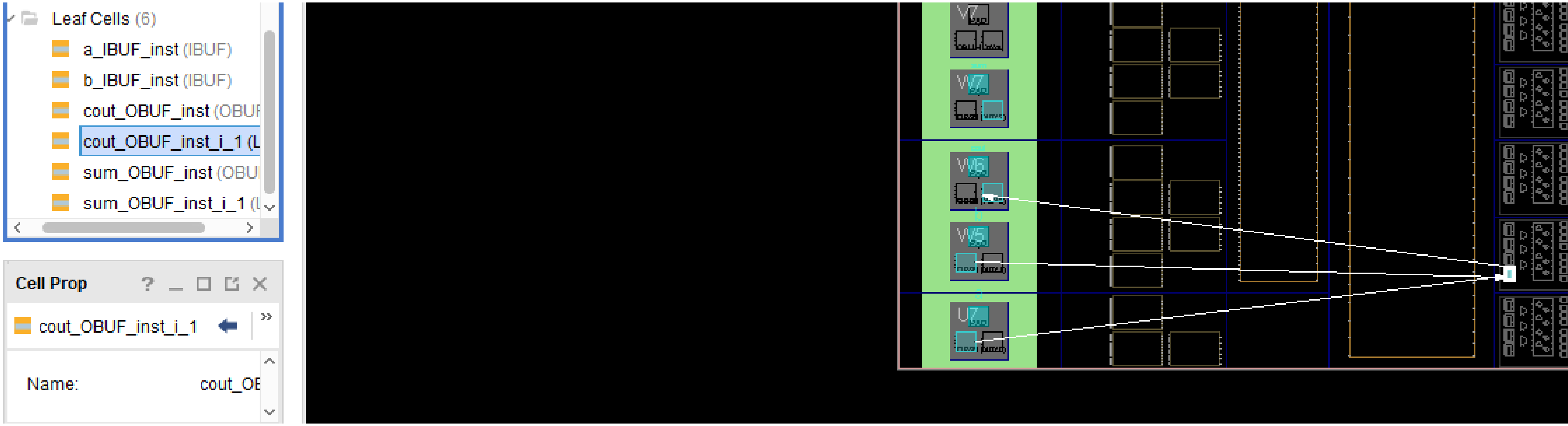
Leaf Cells (6)

- a_IBUF_inst (IBUF)
- b_IBUF_inst (IBUF)
- cout_OBUF_inst (OBUF
- cout_OBUF_inst_i_1 (L
- sum_OBUF_inst (OBU
- sum_OBUF_inst_i_1 (L

Cell Prop

sum_OBUF_inst

Name: sum_OI



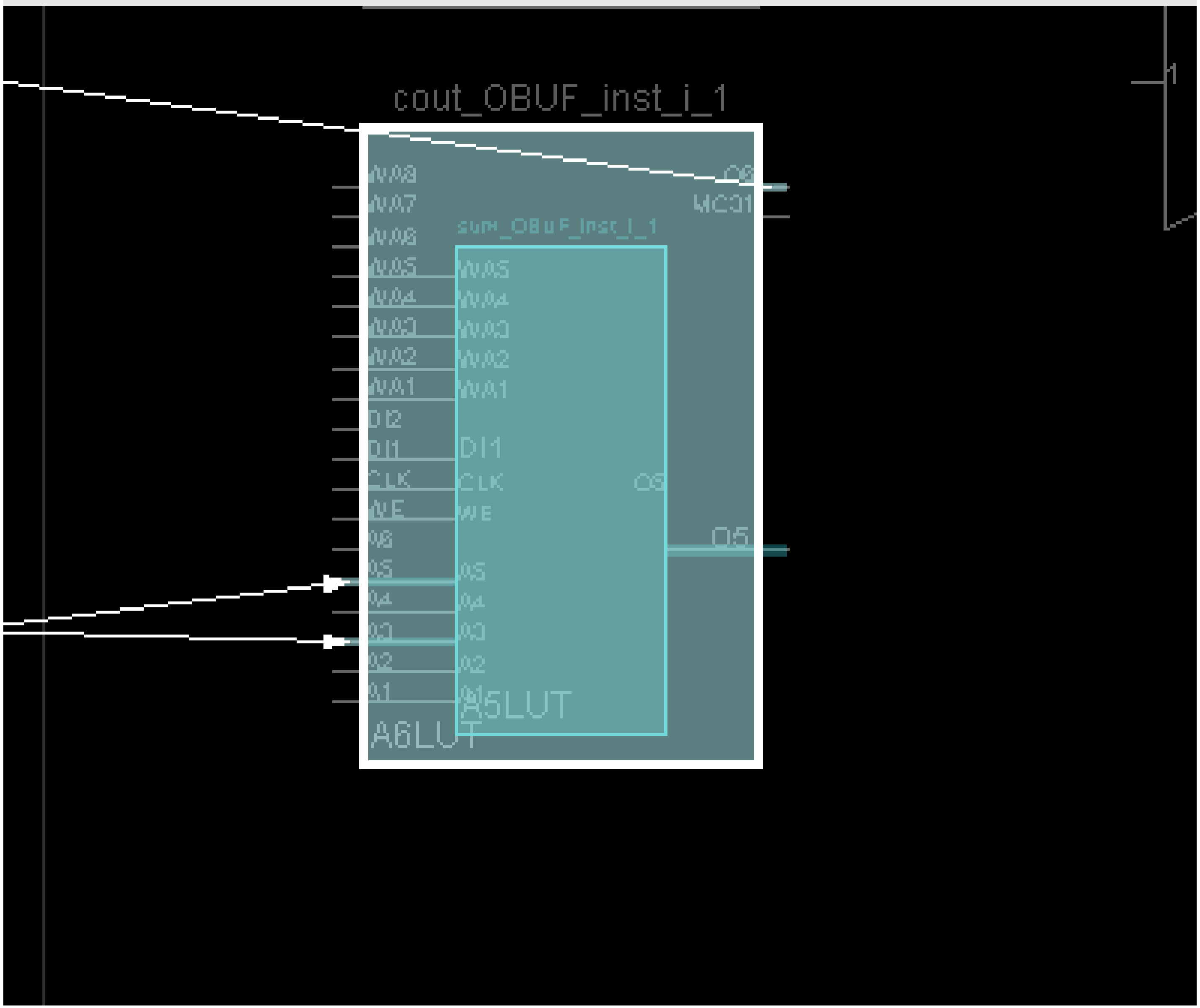
Leaf Cells (6)

- a_IBUF_inst (IBUF)
- b_IBUF_inst (IBUF)
- cout_OBUF_inst (OBUF
- cout_OBUF_inst_i_1 (L
- sum_OBUF_inst (OBU
- sum_OBUF_inst_i_1 (L

Cell Prop

cout_OBUF_inst_i_1

Name: cout_OE


cout_OBUF_inst_i_1
WA8
WA7
WA6
WA5
WA4
WA3
WA2
WA1
DI2
DI1
CLK
WE
A6
A5
A4
A3
A2
A1
A6LUT
sum_OBUF_inst_i_1
WA5
WA4
WA3
WA2
WA1
DI1
CLK
WE
A5
A4
A3
A2
A1
A5LUT
O6
MC31
O5
1

FPGA Advantages

- Long time availability
- Can be updated and upgraded at your customer's site
- Extremely short time to market
- Fast and efficient systems
- Performance gain for software applications
- Real time applications
- Massively parallel data processing