

14. Programmable Logic Devices



Overview

- Introduction
- Programming Technologies
- Basic Programmable Logic Device (PLD) Concepts
- Complex PLD
- Field Programmable Gate Array (FPGA)
- CAD (Computer Aided Design) for FPGAs
- Design flow for Xilinx FPGAs
- Economical Considerations
- Logic design Alternatives

Introduction

- A **Programmable Logic Device** is an integrated circuit with internal logic gates and interconnects. These gates can be connected to obtain the required logic configuration.
- The term “programmable” means changing either hardware or software configuration of an internal logic and interconnects.
- The configuration of the internal logic is done by the user.
- PROM, EPROM, PAL, GAL etc. are examples of Programmable Logic Devices.

Programming Technologies

Programmable Logic Device can be programmed in two ways:

1. Mask programming (in some few cases)
2. Field programming (typical)

1.) Mask programming: programming of device is done in the mask level.

- + good timing performance due to internal connections hardwired during manufacture
- + cheap at high volume production
- programmed by manufacturer
- development cycle = weeks or months
- not re-programmable

Programming Technologies (II)

2.) **Field programming:** Programming of device is done by the user. The programming technologies are of two types

— **Permanent type (Non-volatile):**

- Fuse (normal on) - 'CLOSE (intact)' 'OPEN (blown)'
- Anti-fuse (normal off) - just the opposite of a FUSE
- EPROM
- EEPROM

— **Nonpermanent type (Volatile):**

- driving n-MOS pass transistor by SRAM
- NOTE:
 - When power of device is switched off then the content of SRAM is lost.

Basic PLD Concepts

1.) **PLA** (Programmable Logic Array):

- array of AND and OR gates are programmable
- product term sharing: every product term of the AND array can be connected to the input of any OR gate
- unidirectional input/output pins

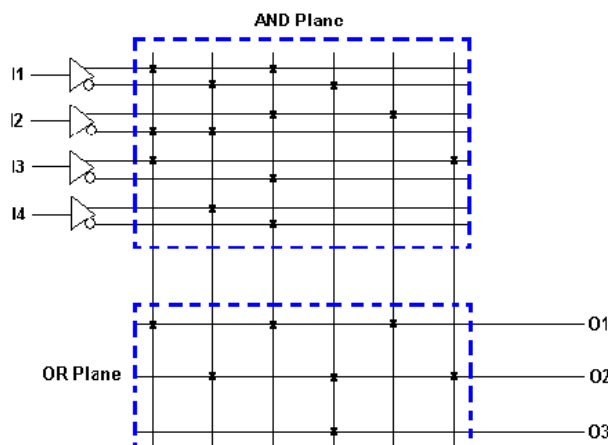


Figure 1: PLA device

Basic PLD Concepts (II)

2.) **Memory based**: Device with fixed AND array and programmable OR array

- output of OR gate has fixed connection with input of AND gates
- PROM, EPROM and EEPROM are memory based PLD device

3.) **PAL/GAL**(Programmable Array Logic/ Gate Array Logic):

AND array is programmable and OR array has fix connection with outputs of AND gates. PAL/GAL devices may have bi-directional I/O pins.

There are three different types of PAL/GAL devices

- **combinational** PAL devices are used for the implementation of logic function
- **sequential** PAL devices are used for the implementation of sequential logic (finite state machines)
- **arithmetic** PAL devices sum of product terms may be combined by **XOR** gates at the input of the macrocell D flip-flop__

Basic PLD Concepts (IV)

Additional features of PAL/GAL devices

- **PAL**:
 - **EPROM** - based programming Technology
- **GAL**:
 - has array of programmable AND gates and OLMC (Output Logic Macro Cell)
 - **EEPROM** - based programming Technology
 - programmable output polarity
 - device can be configured as dedicated input and output mode

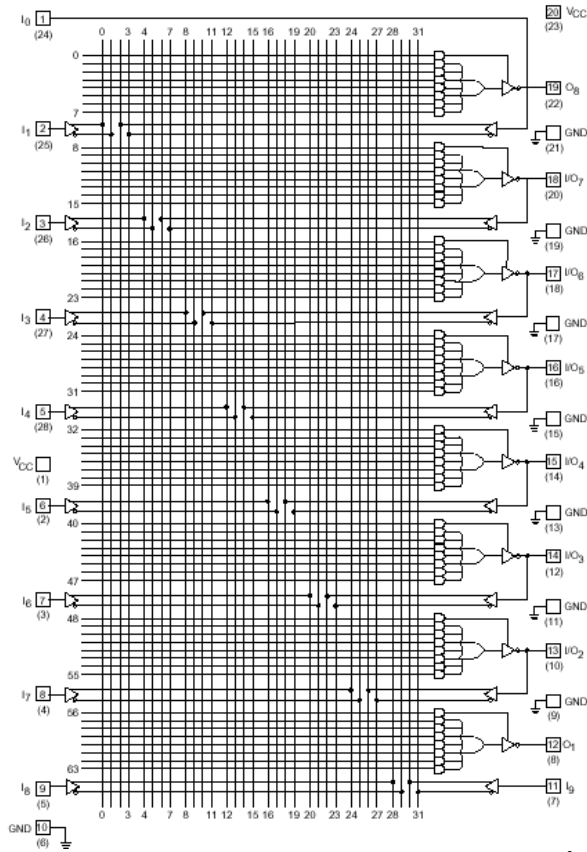


Figure 2:
Combinational PAL
device, AMD PAL16L8

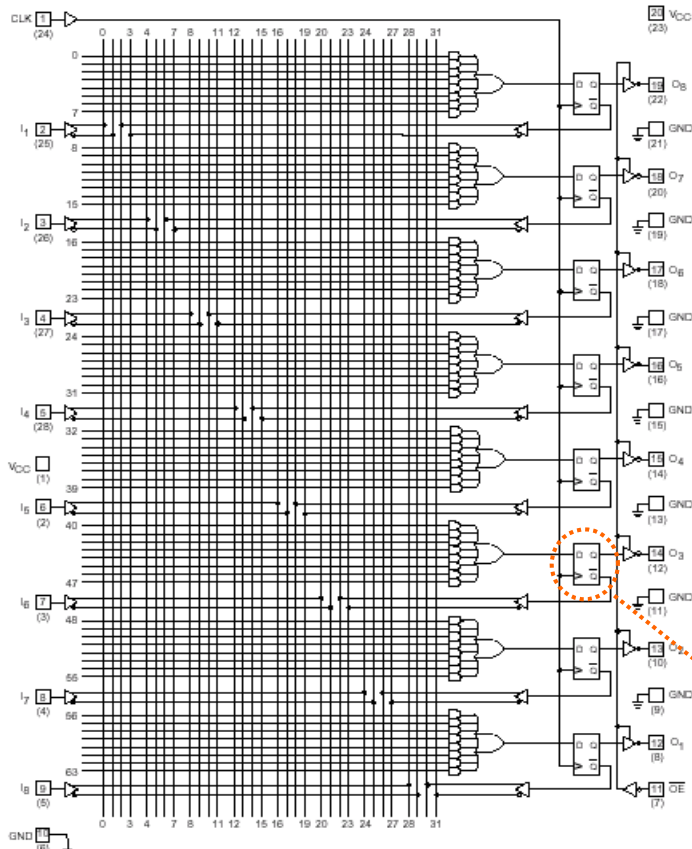


Figure 3:
Sequential PAL devices,
AMD PAL16R8

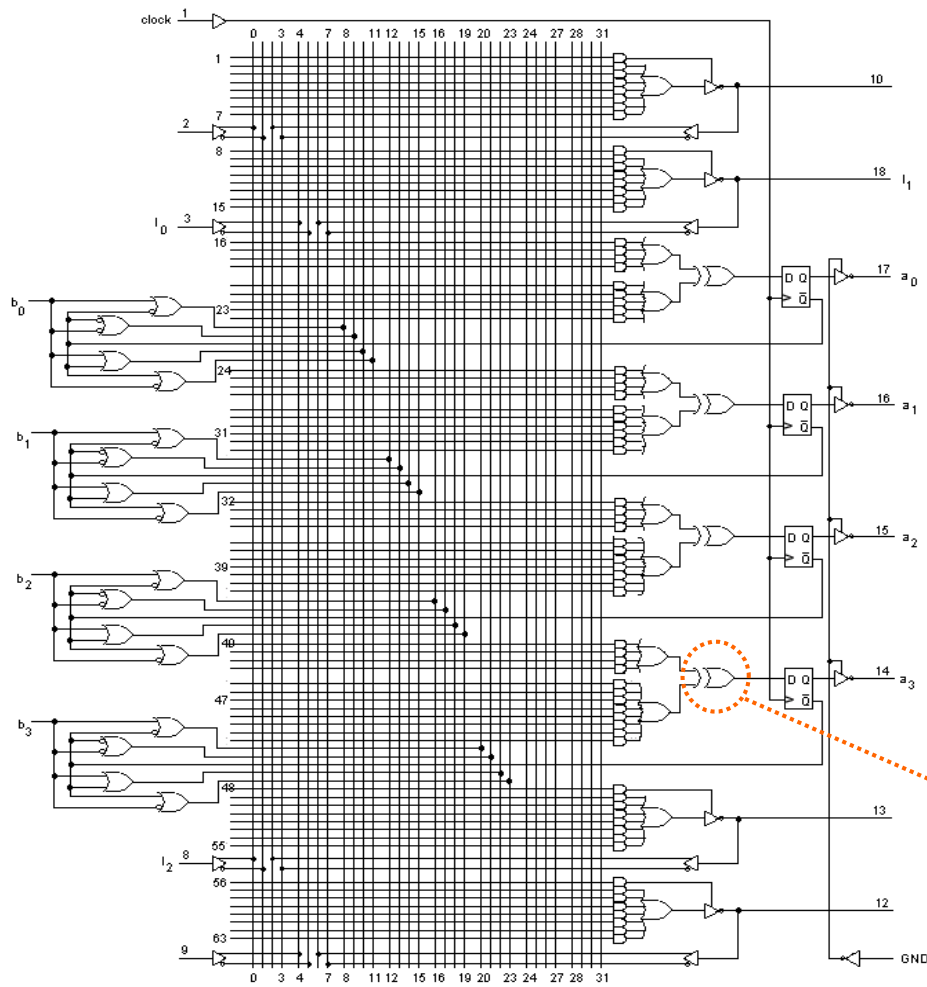
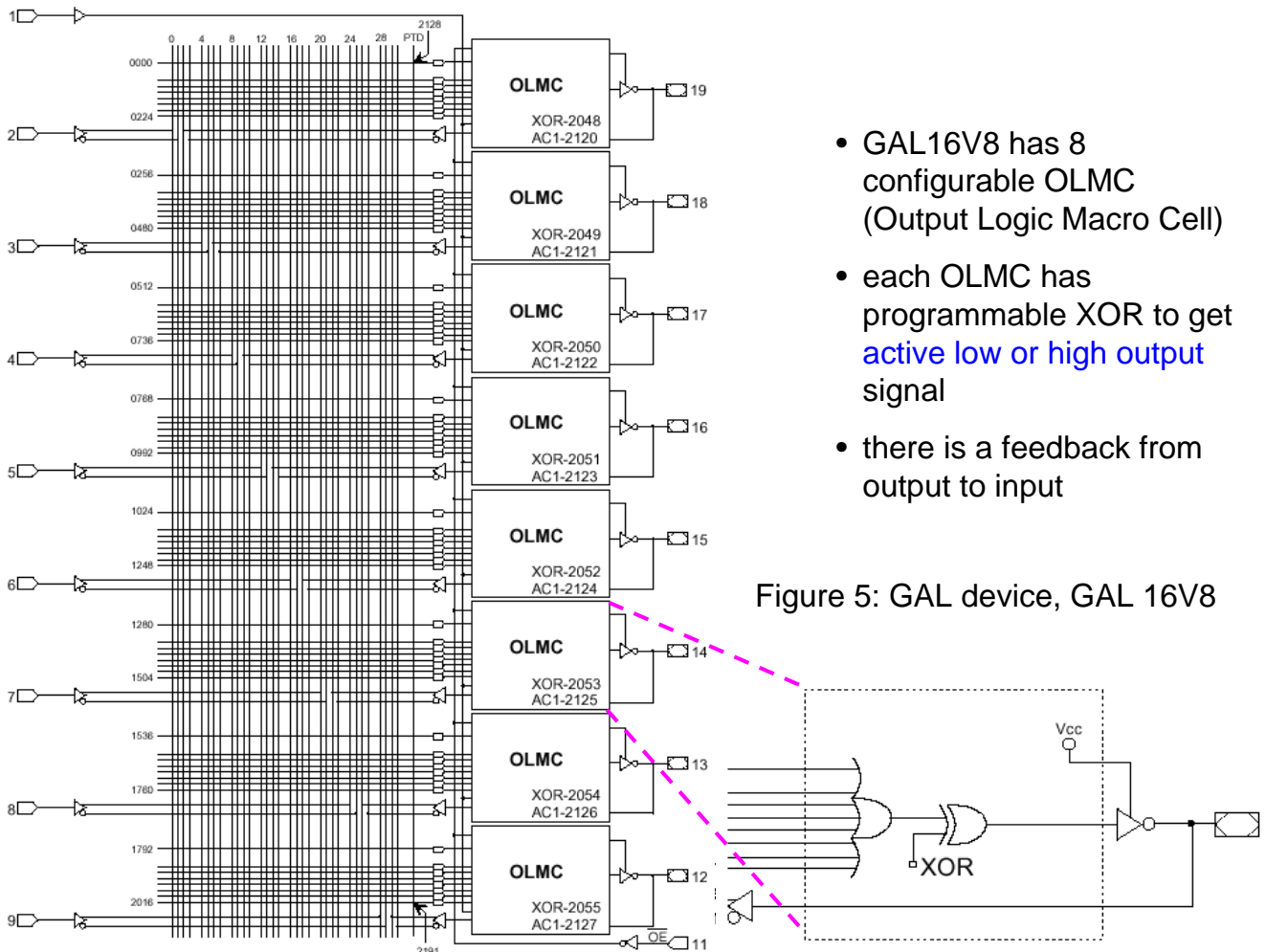


Figure 4:
Arithmetic PAL device, AMD PAL16A4



- GAL16V8 has 8 configurable OLMC (Output Logic Macro Cell)
- each OLMC has programmable XOR to get active low or high output signal
- there is a feedback from output to input

Figure 5: GAL device, GAL 16V8

Complex PLD (CPLD)

- is combination of multiple PAL or GAL type devices on a single chip
- CPLD architectures consists of
 - Macrocells
 - configurable flip-flop (D, T, JK or SR)
 - Output enable/clock select
 - Feedback select
- CPLD has predictable time delay because of hierarchical inter-connection
- easy to route, very fast turnaround
- performance independent of netlist
- devices is erasable and programmable with non-volatile EPROM or EEPROM configuration
- wide designer acceptance
- has more logic density than any classical PLDs device
- relatively mature technology, but some innovation still ongoing

Complex PLD (II)

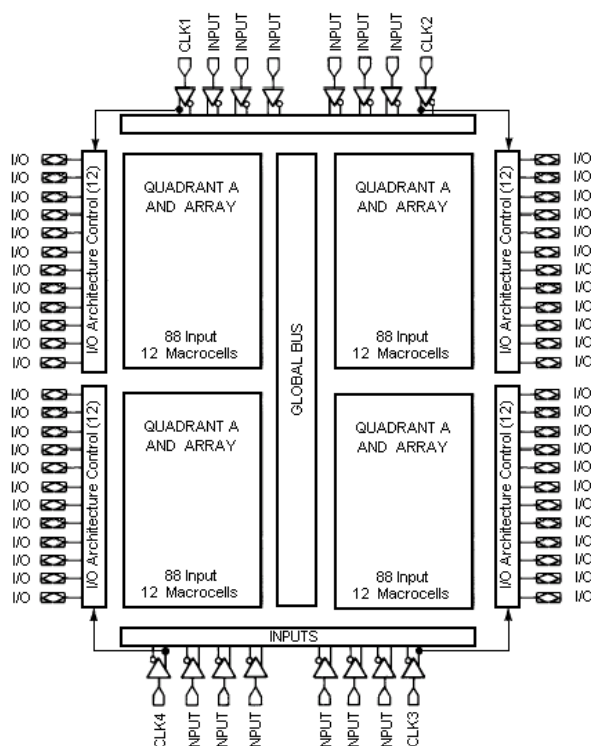


Figure 6:
Complex PLD device
Altera EP1800

Erasable CPLD

- EP1800 is erasable PLD device and has 48 macrocells, 16 dedicated input pins and 48 I/O pins.
- device is divided into four quadrants, each contains 12 macrocells and has local bus with 24 lines and a local clock
- out of 12 microcells, 8 are “local” macrocells and 4 are “global” macrocells

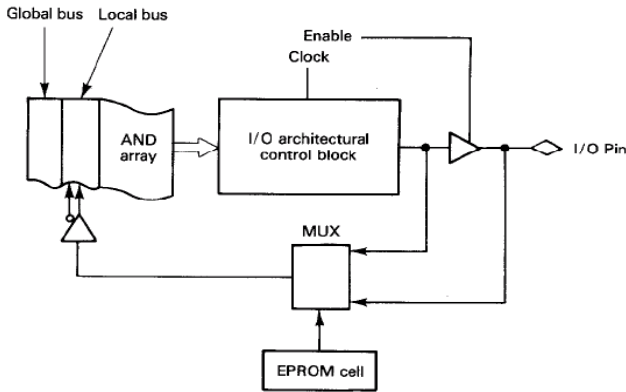


Figure 7: Local macrocell

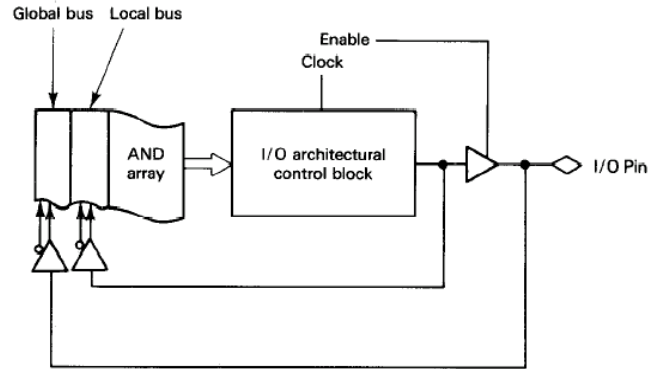


Figure 8: Global macrocell

Erasable CPLD (II)

- global bus has 64 lines and runs through all of the four quadrants (true and complement signals of 12 inputs (=24 lines) + true and complement of 4 clocks (=8 lines) + true and complement of I/O pins of the 4 global macro cells in each quadrant (=32 lines)
- macrocells: combinational or registered data output; the flip-flop is configurable as D, T, JK or SR type.

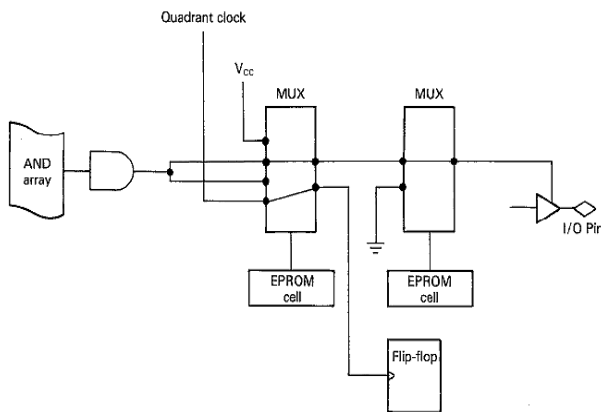


Figure 9: Synchronous clock, output enable by product term

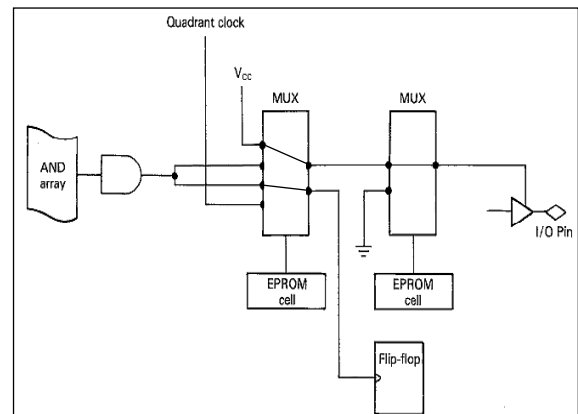
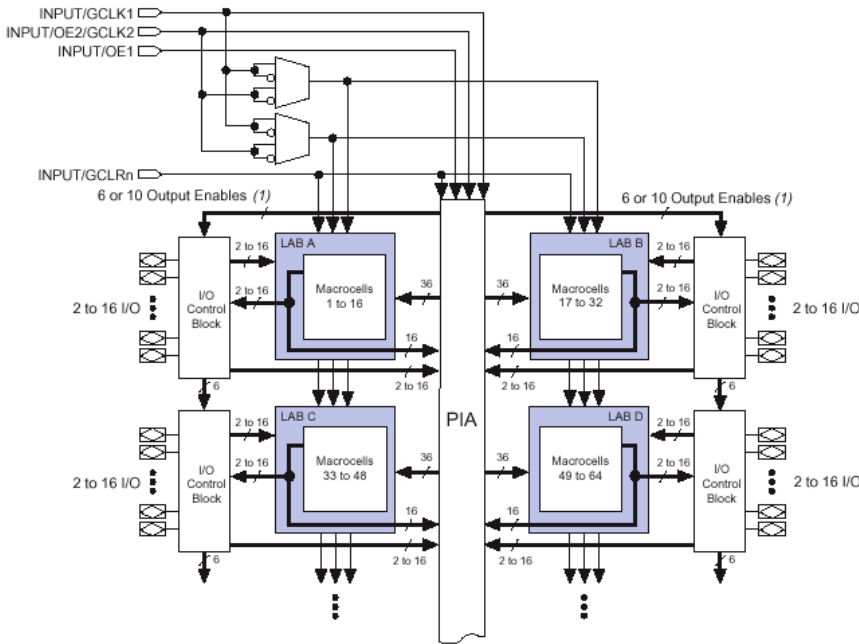


Figure 10: Asynchronous clock, output permanently enabled

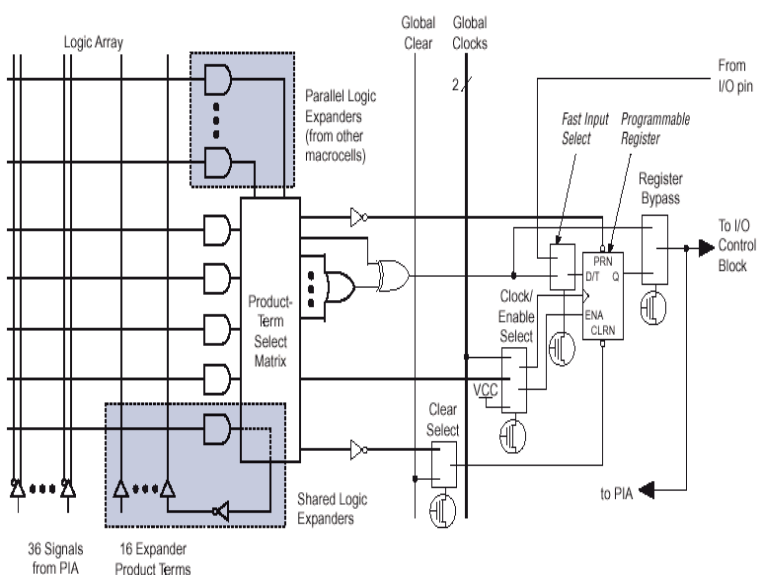
Electrically Erasable PLD



- MAX 7000 is EEPROM based programmable logic device
- it's architecture includes following elements,
 - Logic Array Blocks (LABs)
 - Macrocells
 - Programmable Interconnect Array (PIA)
 - I/O control blocks
- Pin to pin delay is about 5 ns
- predictable delay because of hierarchical routing structure of PIA

Figure 11: Block diagram of Altera MAX 7000 family

Electrically Erasable PLD (II)



- each Logic Array Block (LAB) has 16 macrocells
- each macrocell consists of logic array, product term select matrix and programmable register
- the product term select matrix allocates product terms from logic array to use them as either primary logic inputs to OR and XOR gate or secondary inputs to clear, preset, clock and clock enable control function for the register of macrocell

Figure 12: MAX 7000 device, macrocell

Electrically Erasable PLD (III)

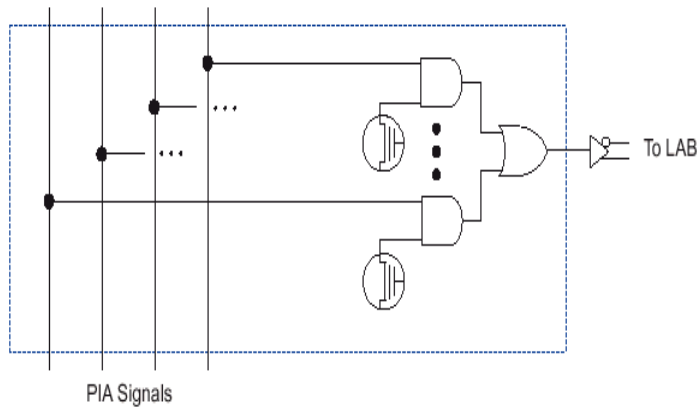





Figure 13:

MAX 7000 device, programmable Interconnect Array (PIA)

- logic is routed among LABs via the PIA.
- dedicated inputs, I/O pins, and macrocell outputs feed the PIA, which makes the signals available throughout the entire device
- only the signals required by each LAB are actually routed from the PIA into the LAB
- selecting of signal from PIA to LAB is done by an EEPROM cell

Field Programmable Gate Array

- FPGA is a general purpose, multi-level programmable logic device
- FPGA is composed of,
 - logic blocks  to implement combinational and sequential logic circuit
 - programmable interconnect  wire to connect input and output of logic blocks
 - I/O blocks  logic blocks at periphery of device for the external connection
- “The **routing resources** are both the greatest strength and weakness of the FPGA’s”

Field Programmable Gate Array (II)

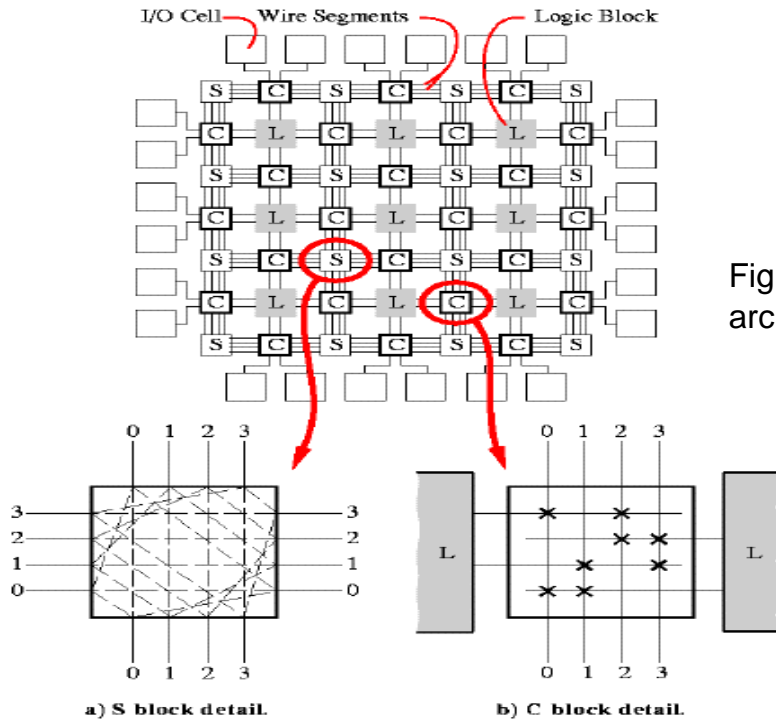
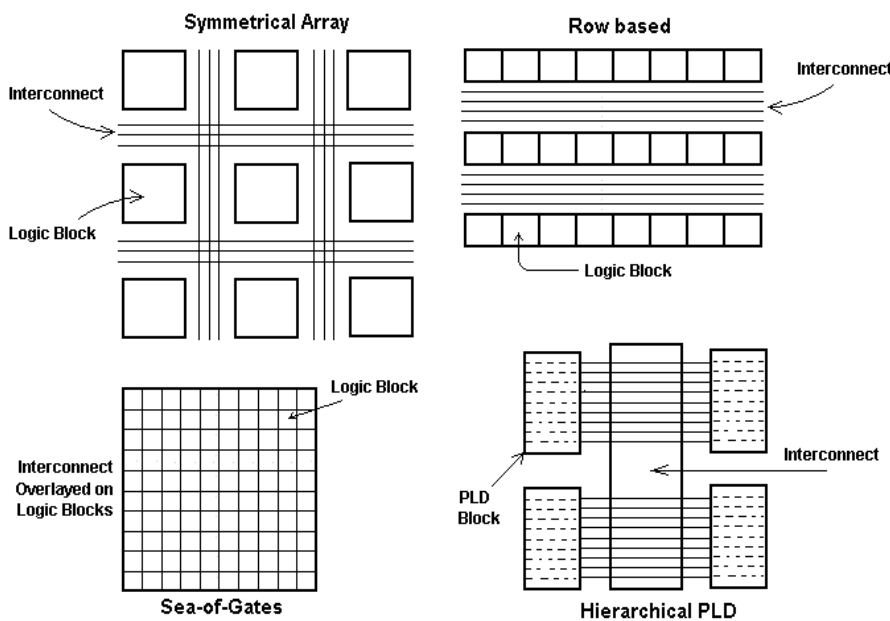


Figure 14: Symmetrical array architecture of FPGAs

Field Programmable Gate Array (III)



- There are four main categories of FPGAs available commercially,
 - symmetrical array
 - row - based
 - hierarchical PLD
 - sea of gates
- They differ to each other on their interconnection and how they are programmed

Figure 15: Category of different FPGA

Programming Technologies

- Currently, there are four programming technologies for FPGAs,
 - static RAM cells
 - anti fuse
 - EPROM transistor
 - EEPROM transistor

Static RAM programming technology:

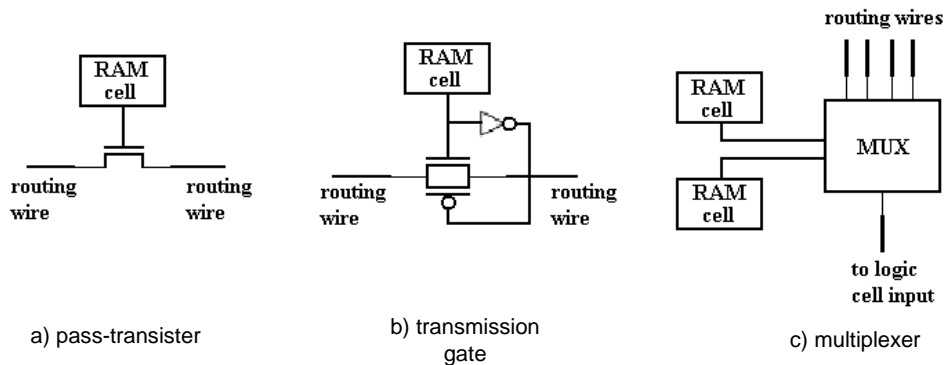


Figure 16: SRAM based programming technology

SRAM Programming technology

- completely reusable - no limit concerning **re-programmability**
- pass gate closes when a “1” is stored in the SRAM cell
- allows iterative prototyping
- **volatile memory** - power must be maintained
- **large area** - five transistor SRAM cell plus pass gate
- memory cells distributed throughout the chip
- fast re-programmability (tens of milliseconds)
- only **standard CMOS process** required

Anti-fuse Programming

- An **anti-fuse** is the opposite of normal fuse.
- Anti-fuse are made with a modified CMOS process having an extra step
- This step creates a very thin insulating layer which separates two conducting layers
- That thin insulating layer is fused by applying a high voltage across the conducting layer
- Such high voltage can be destructive for CMOS logic circuit
- **Non-volatile** (Permanent)
- Requires extra programming circuitry, including a programming transistor

Actel PLICE Anti-fuse programming technology

- The Actel PLICE anti-fuse consists of a layer of positively doped silicon (n^+ diffusion), a layer of dielectric (Oxygen-Nitrogen-Oxygen) and a layer of polysilicon
- it is programmed by placing a relatively high voltage (18V) across the anti-fuse terminals which results current of about 5 mA through it
- typical resistance of a fused contact is 300 to 500 Ω
- manufactured by **3 additional masks** to a normal CMOS process

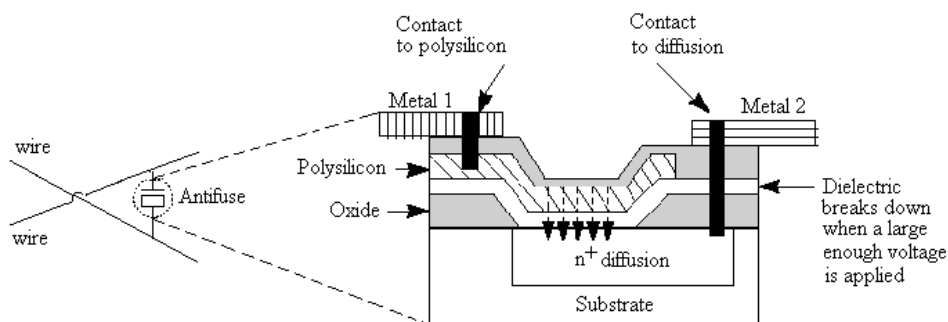


Figure 17: Actel PLICE anti-fuse structure

Quicklogic ViaLink Anti fuse programming technology

- amorphous silicon is used as an insulating layer
- direct metal to metal contact results path resistance below 50 Ω
- 10 V terminal voltage is required to fuse the amorphous silicon

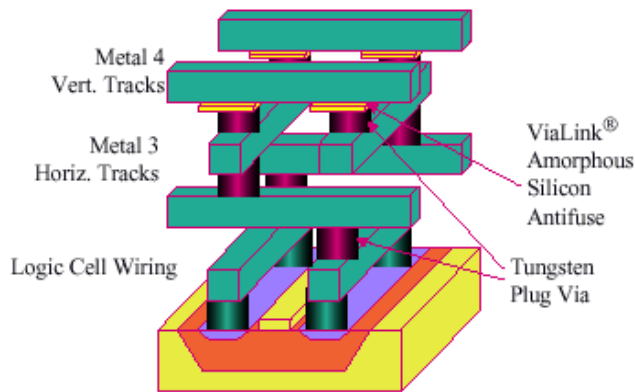


Figure 18 : Four layer Metal ViaLink structure

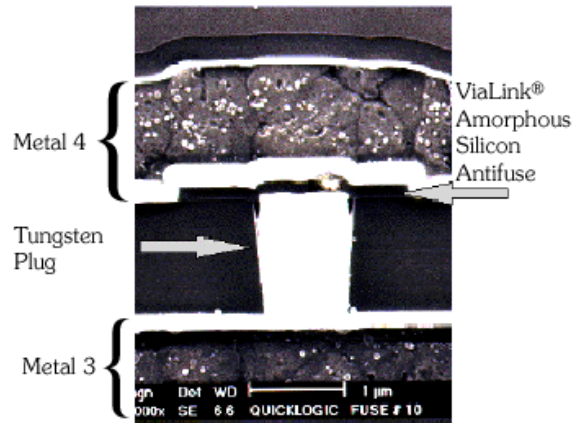


Figure 19: ViaLink element

EEPROM programming technology

- static charge on floating gate turns the transistor permanently off
- re-programmable
- non-volatile
- external permanent memory is not required
- slow re-configuration time
- floating-gate FET has relatively high on resistance
- higher static power consumption due to pull up resistor

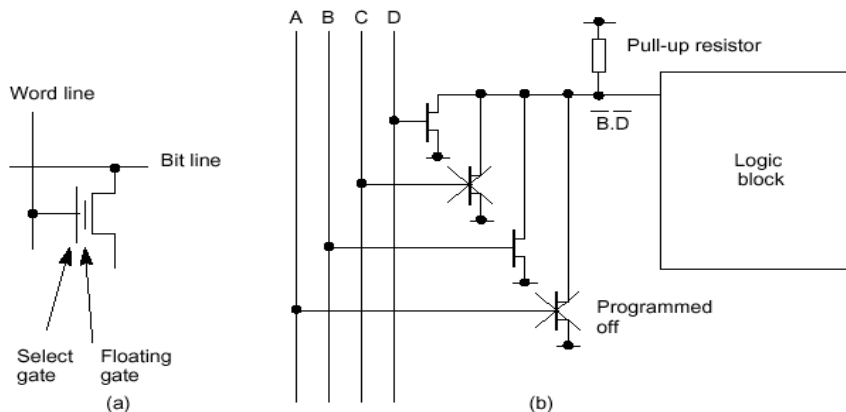


Figure 20: EEPROM programming technology

Commercially available FPGAs

Company	General Architecture	Logic Block Type	Programming Technology
Xilinx	Symmetrical Array	Look-up Table	Static RAM
Actel	Row-based	Multiplexer-Based	Anti-Fuse
Altera	Hierarchical PLD	PLD Block	EPROM/SRAM
Plessey	Sea-of-gates	NAND-gate	Static RAM
Plus	Hierarchical PLD	PLD Block	EPROM
AMD	Hierarchical PLD	PLD Block	EEPROM
Quicklogic	Symmetrical Array	Multiplexer-Based	Anti-Fuse
Algotronix	Sea-of-gates	Multiplexers and Basic Gates	Static RAM
Concurrent	Sea-of-gates	Multiplexers and Basic Gates	Static RAM
Crosspoint	Row-based	Transistor Pairs and Multiplexers	Anti-Fuse

Xilinx FPGA

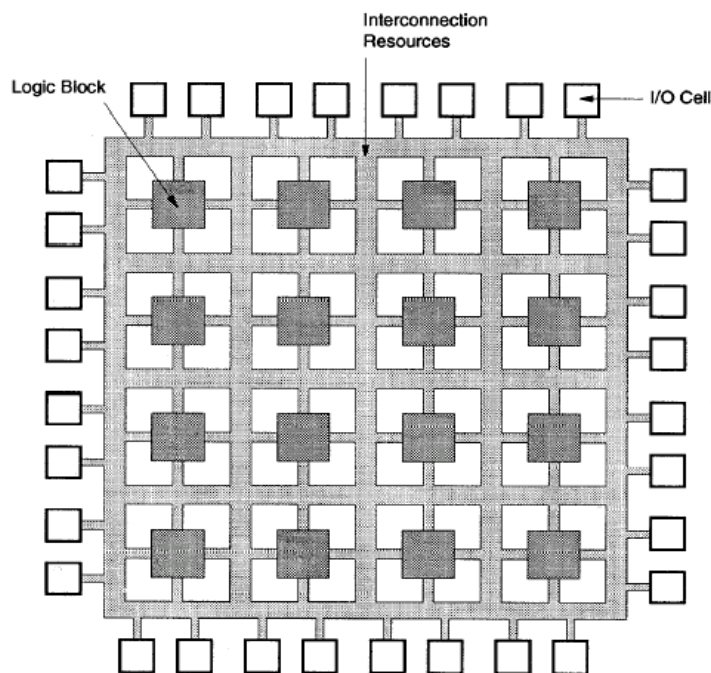


Figure 21: General architecture of Xilinx FPGA

- Xilinx architecture comprises of two dimensional array of logic block called as CLB.
- They are interconnected via horizontal and vertical routing channel
- I/O Blocks are user configurable to provide an interface between external package pin and input logic
- I/O can be configured as input, output and bi-directional signal

Xilinx FPGA (II)

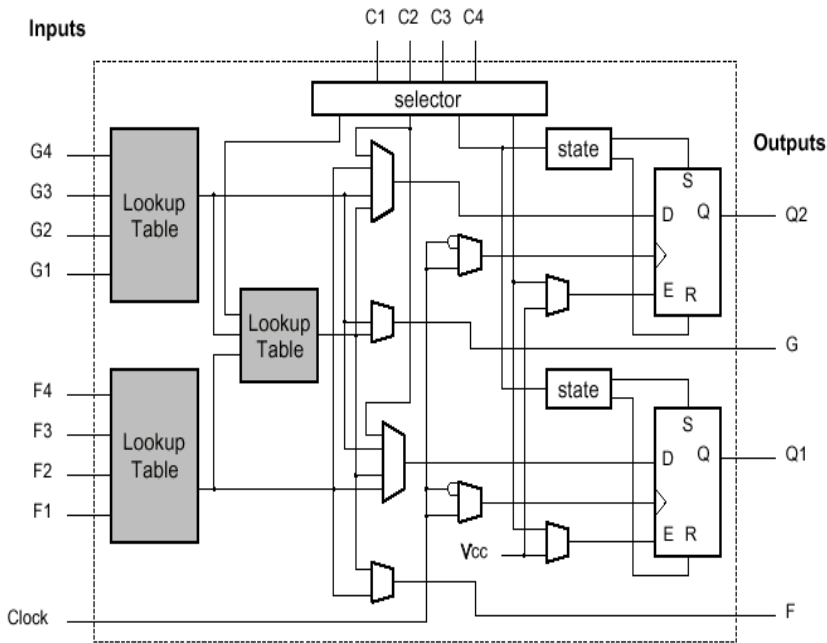


Figure 22: Xilinx XC4000 CLB

- Xilinx XC4000 is an SRAM based FPGA
- each CLB has three LUTs (Look Up Tables) and two flip-flops.
- result of combinatorial logic is stored in 16x1 SRAM LUTs
- LUTs can be also used as RAM
- combinatorial results of CLB is passed to the interconnect network or can be stored in flip-flops and pass to the interconnect network
- with two stage of LUTs, two functions of 4 variables or one function of 5 variables can be implemented ³¹

Xilinx FPGA (III)

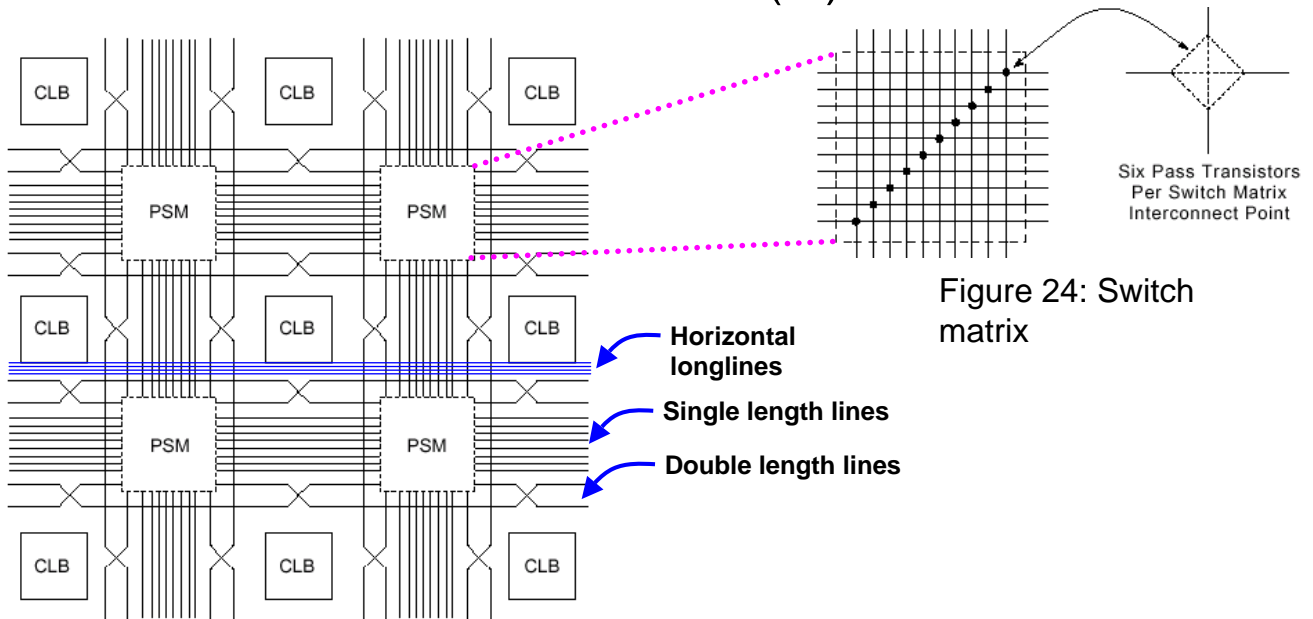


Figure 23: Programmable interconnect associated with XC4000 series CLB

Figure 24: Switch matrix

Xilinx FPGA (IV)

- interconnects of XC4000 device are arranged in horizontal and vertical channels
- each channel contains some number of wire segments
- They are,

Single length lines:

- they span a single CLB
- provide highest interconnect flexibility and offer fast routing
- acquire delay whenever line passes through switch matrix
- they are not suitable for routing signal for long distance

Double length lines:

- they span two CLB so that each line is twice as long as single length lines
- provide faster signal routing over intermediate distance

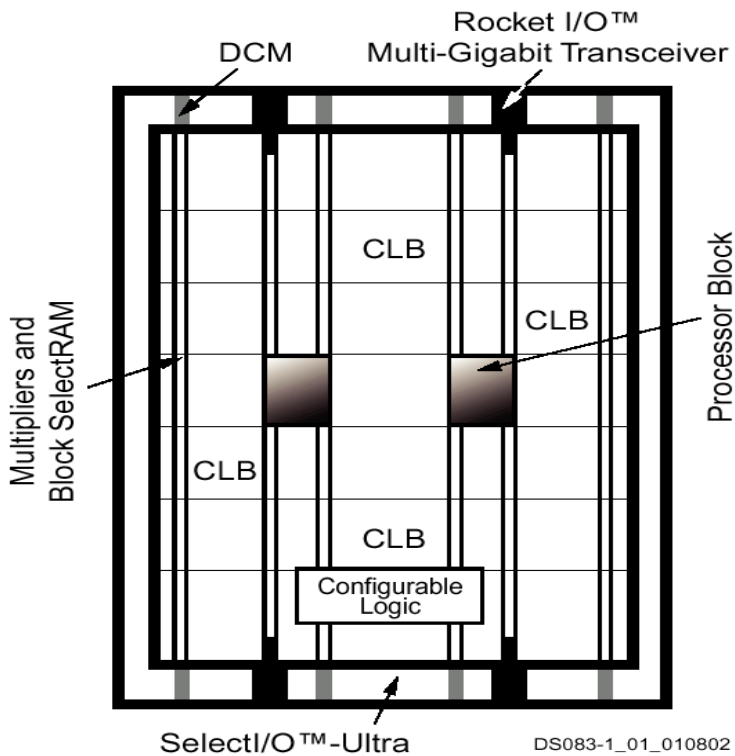
Longlines:

- Longlines form a grid of metal interconnect segments that run entire length or width of the array
- they are for high fan-out and nets with critical delay

Xilinx, Virtex-II Pro™ FPGA family

- The Virtex-II Pro Platform FPGA is the most technically sophisticated silicon and software product development in the history of the programmable logic industry.
- The Virtex-II Pro FPGAs are manufactured *in a 0.13-micron process*.
- It is capable of implementing high performance [System-On-a-Chip](#) designs with low development cost
- It can be used in the application such as system architectures in networking applications, deeply embedded systems and digital signal processing systems etc.
- Virtex-II Pro devices incorporates one to [four PowerPC 405 processor cores](#). The PowerPC 405 cores are fully embedded within the FPGA, where all processor nodes are controlled by the FPGA routing resources.
- Each PowerPC 405 core is capable of more than 300 MHz clock frequency.

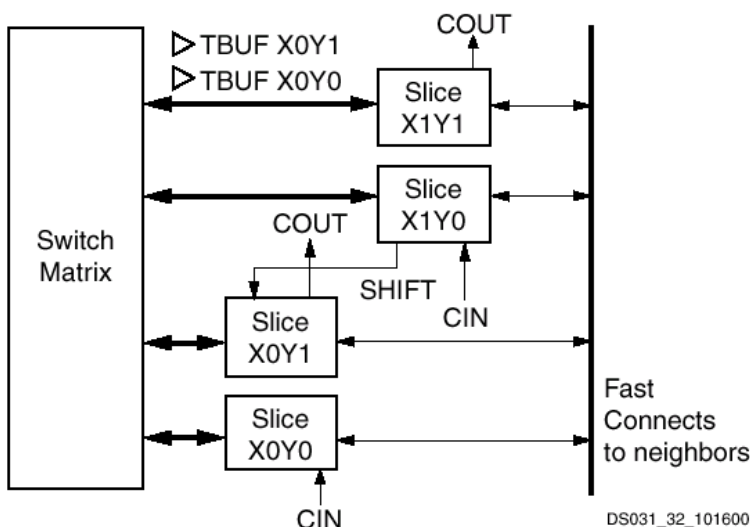
Xilinx, Virtex-II Pro™ FPGA family (II)



- The Virtex-II Pro FPGA consists of the following components:
 - Embedded Rocket I/O™ Multi-Gigabit Transceivers (MGTs)
 - Processor Blocks containing embedded IBM® PowerPC® 405 RISC CPU (PPC405) cores and integration circuitry
 - FPGA fabric based on Virtex-II architecture.

Figure 25: Virtex-II Pro Generic Architecture Overview

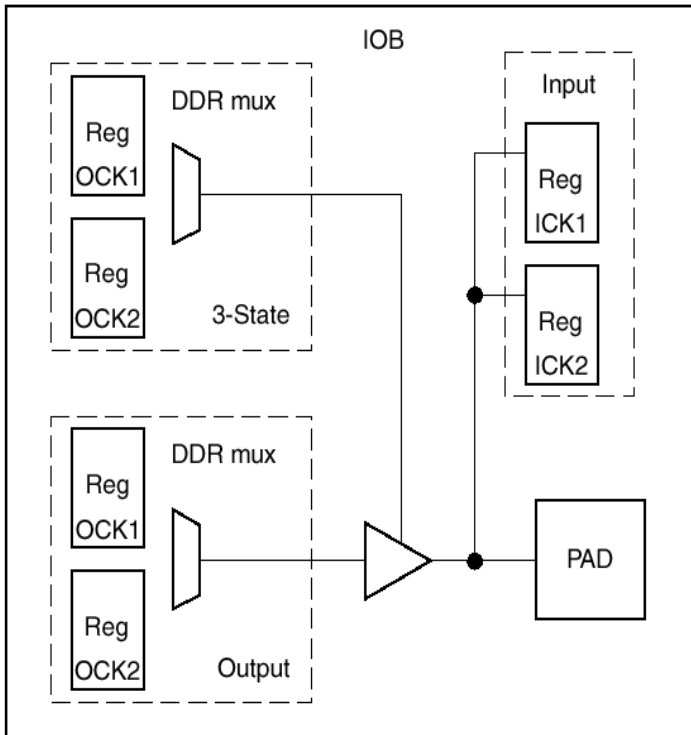
Xilinx, Virtex-II Pro™ FPGA family (III)



- CLB (Configurable Logic Block) include four slices and two 3-state buffers
- Each slice is equivalent and contains:
 - Two function generators (F & G)
 - Two storage elements
 - Arithmetic logic gates
 - Large multiplexers
 - Wide function capability
 - Fast carry look-ahead chain
 - Horizontal cascade chain (OR gate)

Figure 26: CLB (Configurable Logic Block) of Virtex-II Pro FPGA

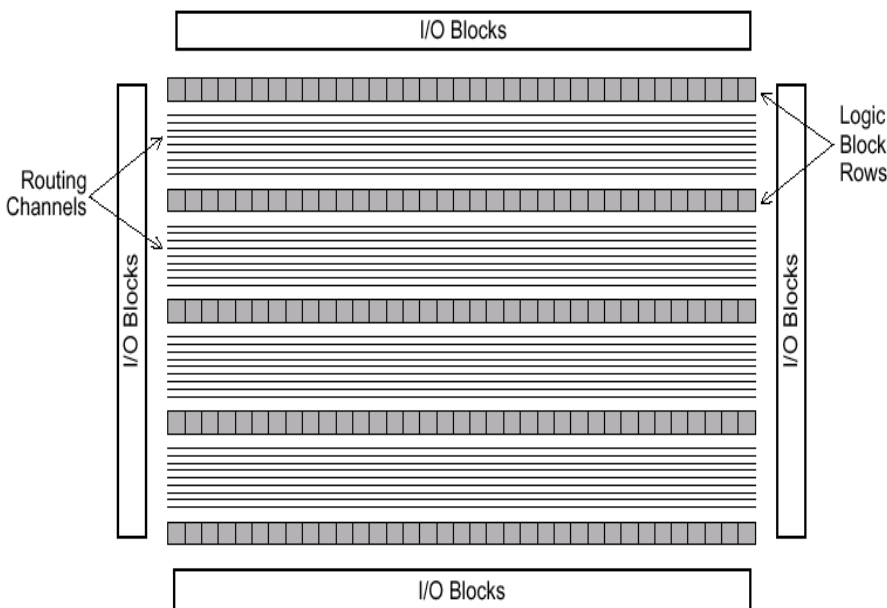
Xilinx, Virtex-II Pro™ FPGA family (IV)



- IOB blocks include six storage elements, as shown in Figure.
- Each storage element can be configured either as an edge-triggered D-type flip-flop or as a level-sensitive latch.
- On the input, output, and 3-state path, one or two DDR (Double Data Rate) registers can be used.
- Double data rate is directly accomplished by the two registers on each path, clocked by the rising edges (or falling edges) from two different clock nets.

Figure 27: IOB block of Virtex-II Pro FPGA

Actel/TI FPGA architecture



- Actel offers three main families:
 - Act 1, Act 2, Act 3
- programmable Logic blocks are arranged in row
- horizontal routing channels are arranged between the adjacent rows
- Actel FPGA are based on anti fused technology
- instead of LUTs, it has multiplexer

Figure 28: General architecture of Actel FPGA

Actel/TI FPGA architecture (II)

Act-1 Logic Module:

- The Act-1 logic module has 8 - input and 1- output logic circuit
- it has only combinatorial logic circuit module
- The Logic Module can implement the four basic functions which are NAND, AND, NOR and OR

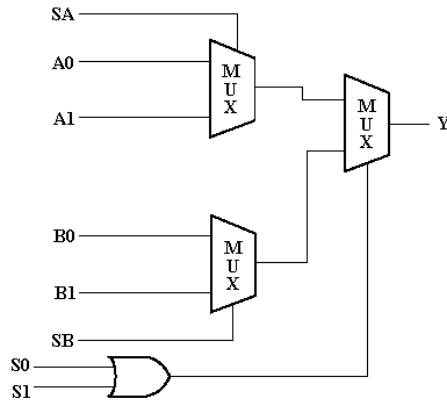


Figure 29: Act-1 logic module

Actel/TI FPGA architecture (III)

Act-2 Logic Module:

- Act-2 family has two module architecture, consisting of C module (Combinatorial) and S module (Sequential)
- the Logic Module is optimized for both combinatorial and sequential designs

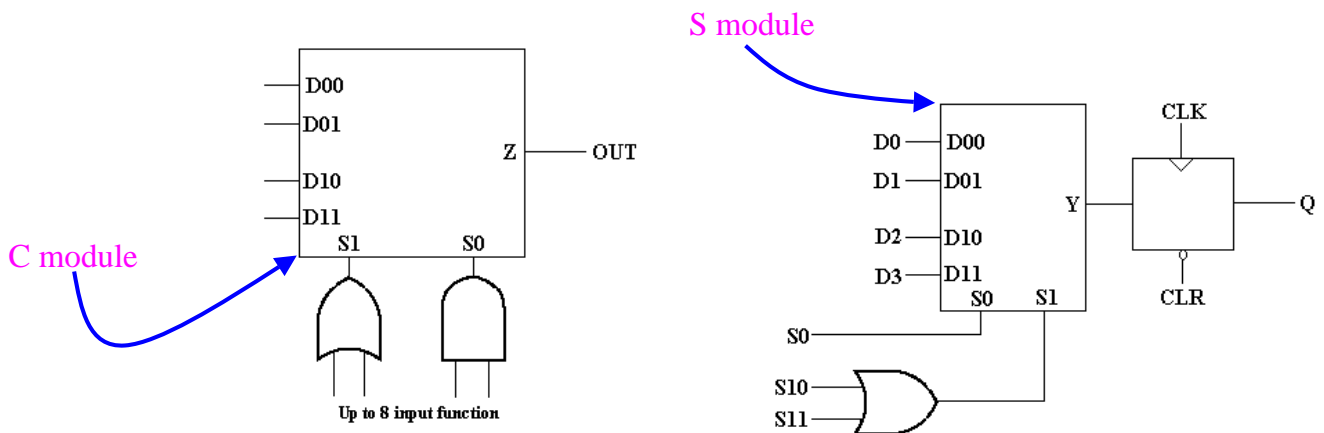


Figure 30: Act-2 logic module

Actel/TI FPGA architecture (IV)

Act-3 Logic Module:

- it comprises an AND and OR gate that are connected to a multiplexer-based circuit block.
- The multiplexer circuit is arranged such that, in combination with the two logic gates, a very wide range of functions can be realized in a single logic block
- about half of the logic blocks in an Act-3 device also contains a flip-flop

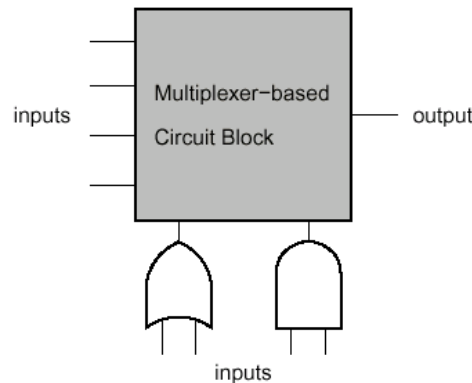


Figure 31: Act-3 Logic module

Actel/TI FPGA architecture (V)

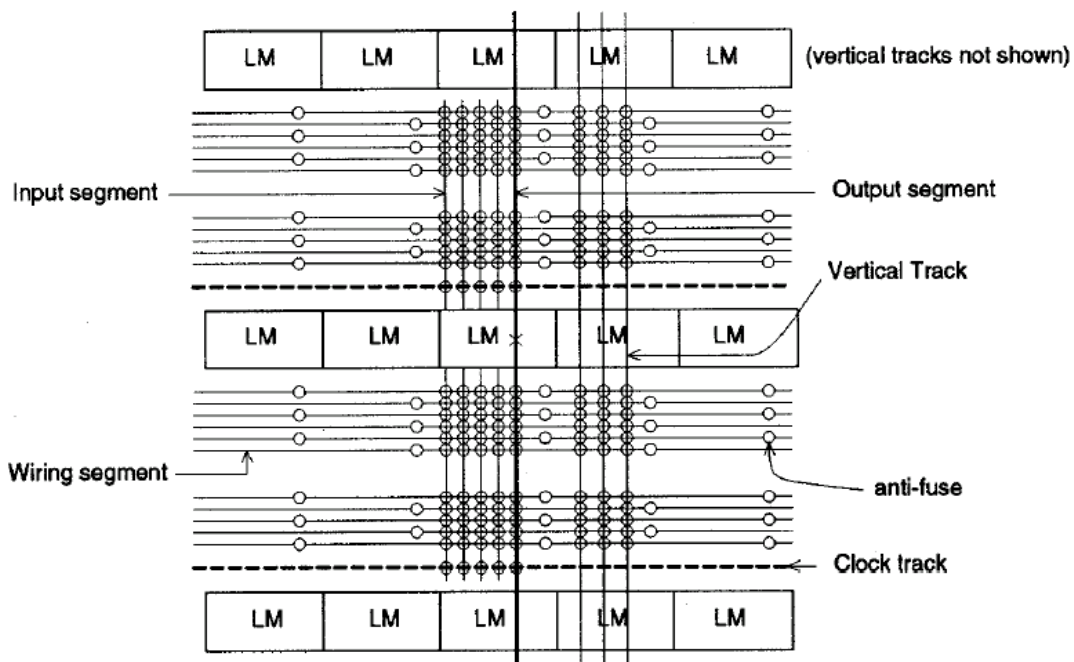


Figure 32: Act-1 programmable interconnection architecture

CAD for FPGAs

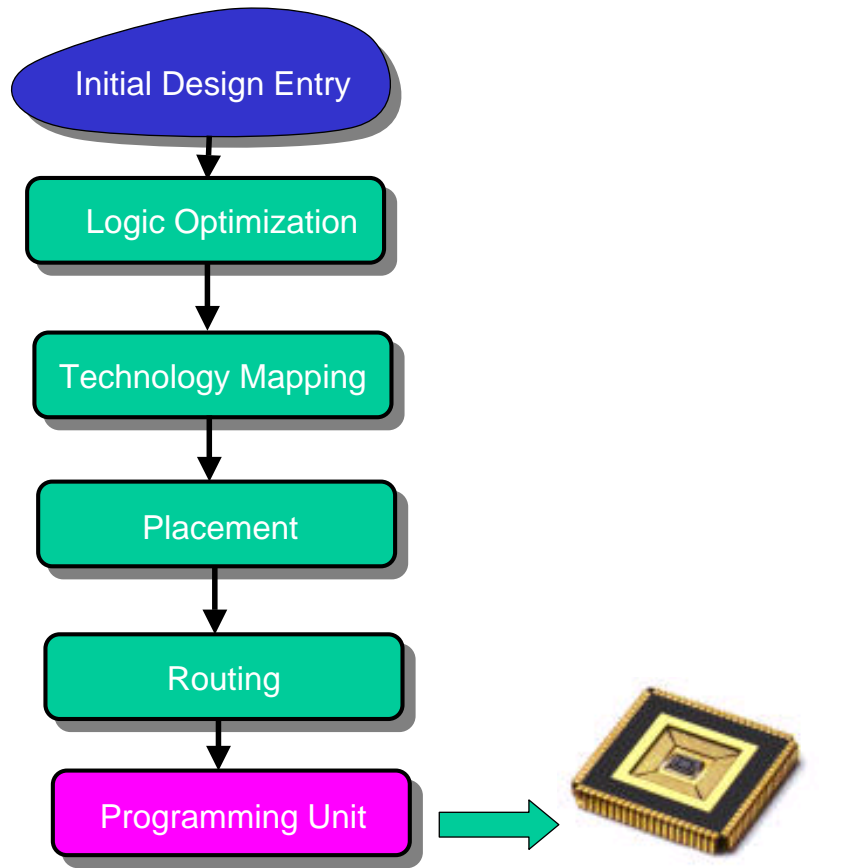
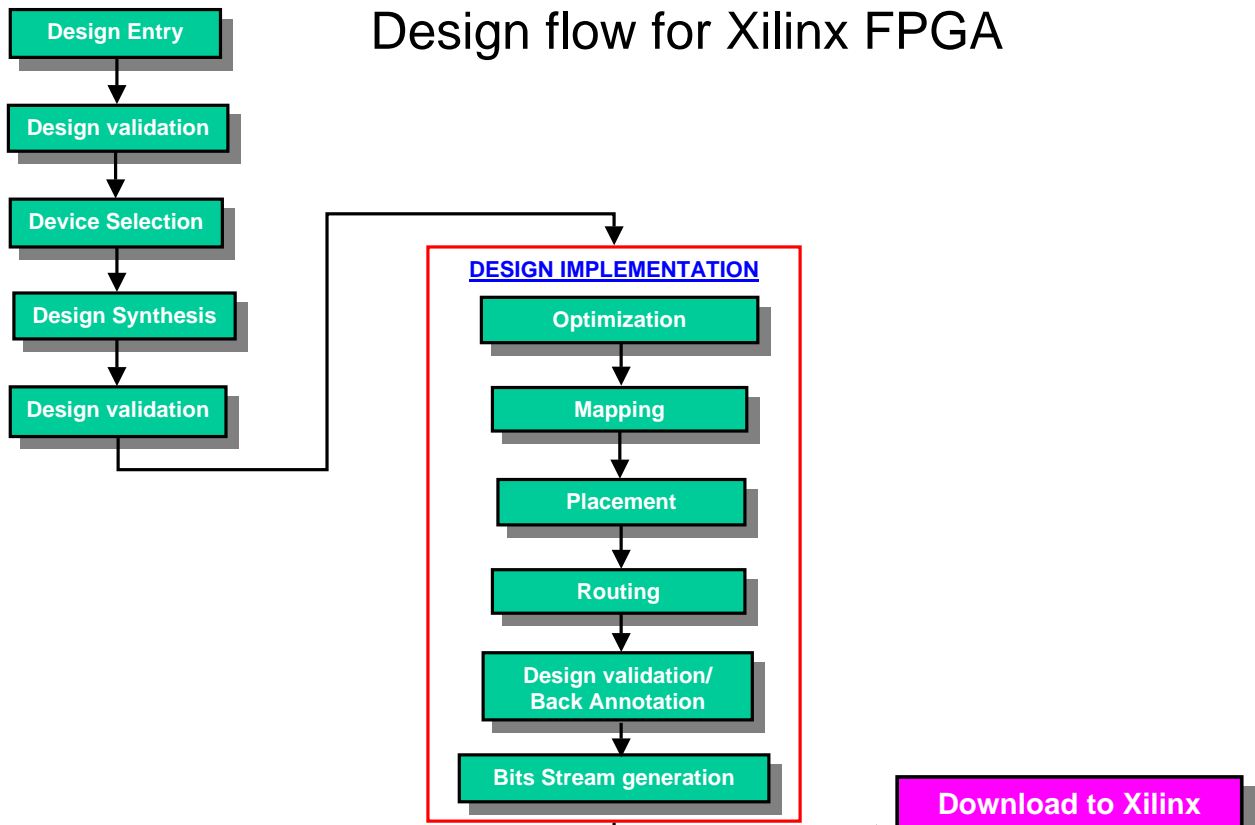


Figure 33: Design flow for FPGA

Design flow for Xilinx FPGA



Economical Considerations

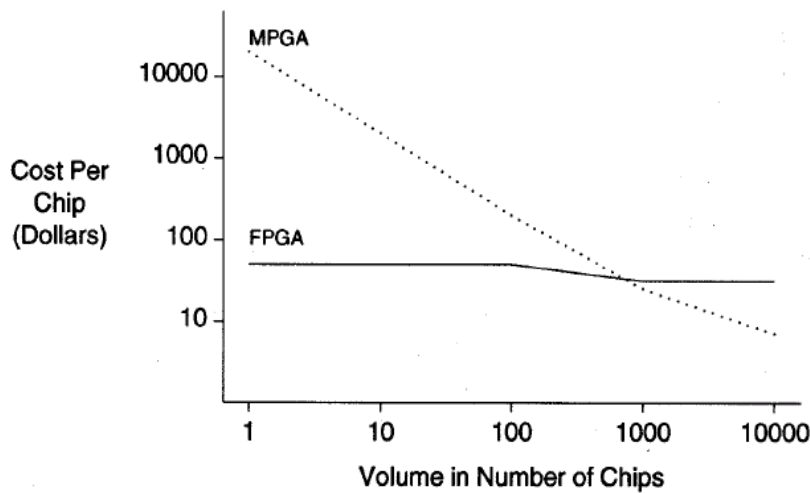


Figure 34: Cost per Chip

Economical Considerations (I)

FPGA	MPGA
<ol style="list-style-type: none"> 1. Cost per chip is less for low volumes (low fixed cost) 2. Short turnaround time 3. Design flexibility is high and cost for re-designing is low 4. Speed is relatively slow because of resistance and capacitance of the programmable switch 5. Programmable switches and configuration network require chip area, this results decreased in logical density 	<ol style="list-style-type: none"> 1. Less cost per chip for high volumes 2. Fabrication is done with hardwired metal connection layer, this results fast operation 3. High logic density 4. Very high costs for low volumes (high fixed cost) 5. No redesign flexibility

Logic design Alternatives

	SSI and MSI lcs	PLDs	Programmable gate arrays	Gate arrays	Custom ICs
Integration in gates	100s	< 500k	10k – 1M	100 –10M	1M – 100M
Speed	Fast	Slow to medium	Slow to medium	Slow to fast	Fast
Function defined by user	No	Yes	Yes	Yes	Yes
Time to costumize	-	Second s	Seconds	Months	Year
User programmable	No	Yes	Yes	No	No

Logic design Alternatives (I)

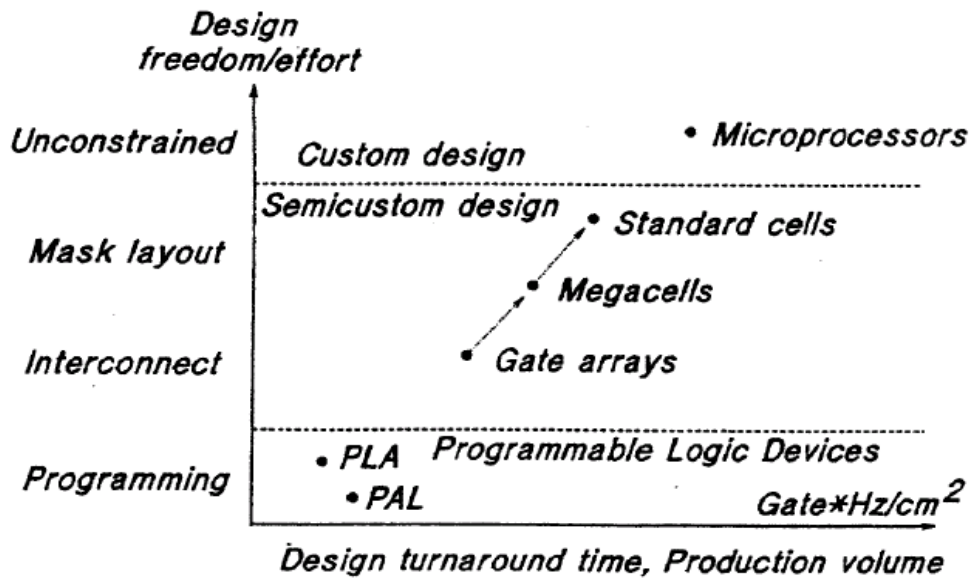
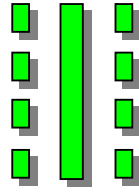


Figure 35: Relative merits of various ASIC implementation styles

CPLDs and FPGAs

Complex Programmable Logic Device (CPLD)



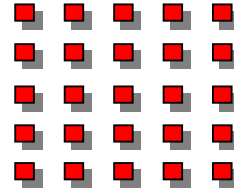
Architecture More Combinational

Density Low-to-medium
0.5-10K logic gates

Performance Predictable timing
Up to 250 MHz today

Interconnect “Crossbar Switch”

Field-Programmable Gate Array (FPGA)



Architecture Gate array-like
More Registers + RAM

Density Medium-to-high
1K to 3.2M system gates

Performance Application dependent
Up to 200 MHz today

Interconnect Incremental