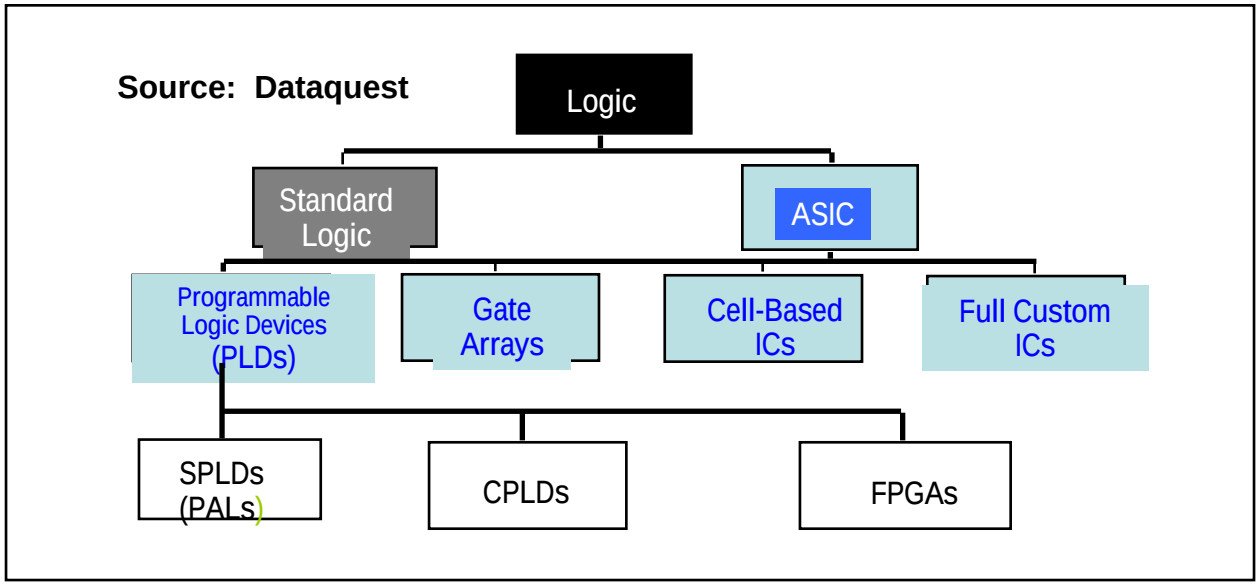
Ch.8 Introduction to PLD/FPGA

- Introduction to PLD Families
- Two kinds of the Basic FPGA Architecture
- The Programmable sources of FPGA
- CPLD or FPGA?
- Design of ASIC and FPGA Flow
- Design examples

1. Introduction to Programmable Logic Device Families



Acronyms

SPLD = Simple Prog. Logic Device
PAL = Prog. Array of Logic
CPLD = Complex PLD
FPGA = Field Prog. Gate Array

Common Resources

Configurable Logic Blocks (CLB)
- Memory Look-Up Table
- AND-OR planes
- Simple gates

Input / Output Blocks (IOB)
- Bidirectional, latches, inverters, pullup/pulldowns

Interconnect or Routing
- Local, internal feedback, and global

ASIC Types

- Full-Custom ASICs
- Semi-Custom ASICs
  - Standard-Cell ASICs
  - Gate-Array ASICs
- Programmable ASICs
  - Field Programmable Gate Array (FPGA)
  - Complex Programmable Logic Devices (CPLD)

The world famous FPGA Corp.

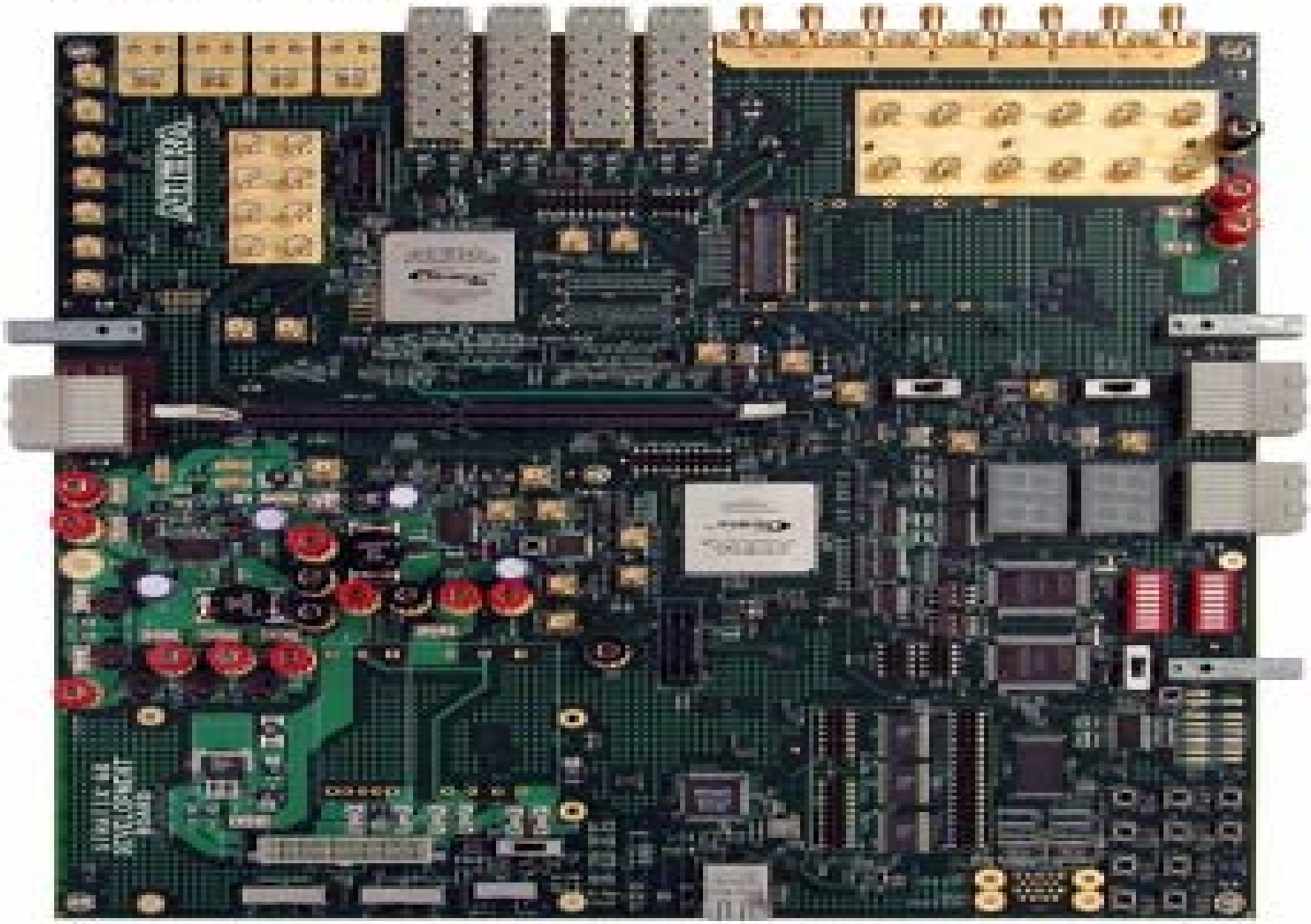
The Stratix GX Device Family

- Second-Generation Transceiver Product
  - 3.125 Gbps
  - Signal Integrity
  - Flexible Feature-Rich High-Speed Solution
- Complete Set of Efficient Hard IP Functions
  - Significantly Reduces Resource Consumption

Lattice

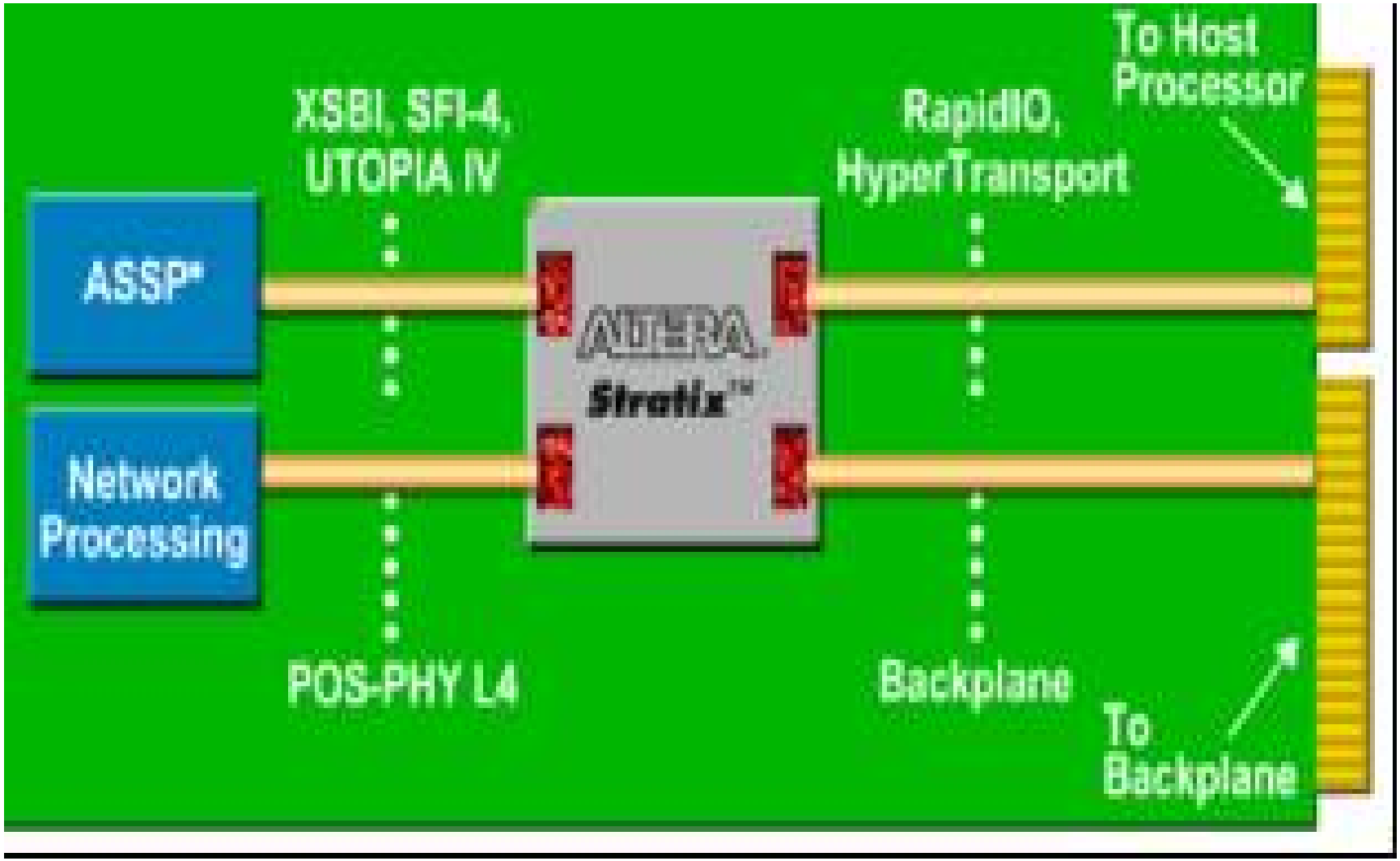

ALTERA
Stratix
Nios

Figure 1. Stratix GX Development Board

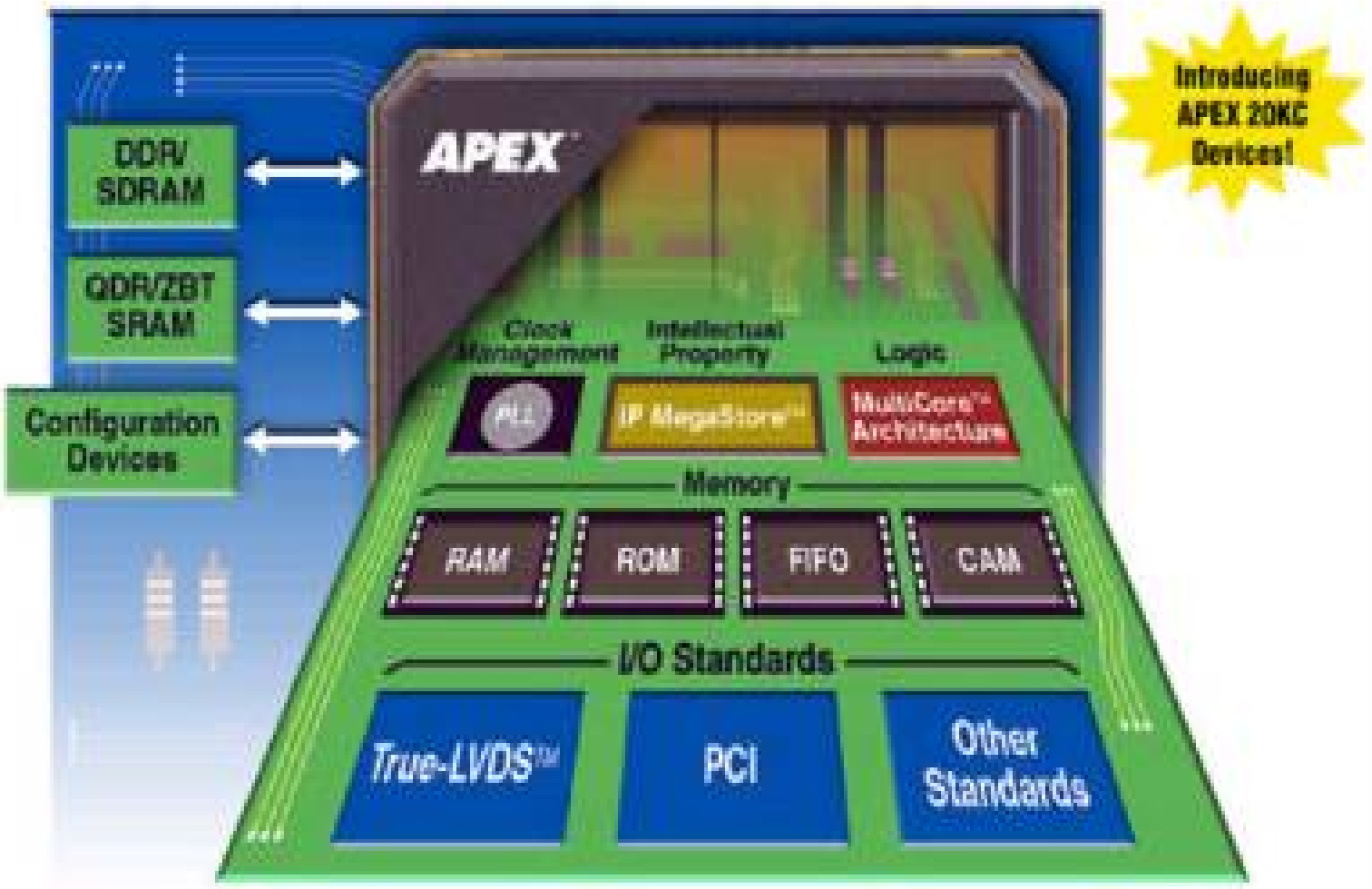


Nios embeded system development

Stratix -Application



Achitecture



Technology Selection

- Time-to-Market
- Performance
- Cost
- Risk
- Technology Migration

Time-to-Market

- **FPGA has shorter time-to-market than ASIC**
  - Faster prototype – no manufacturing lead time
  - Fast turn-around for design changes and bug fixes
  - Able to stage drops for incremental feature set to give SW an earlier start in system integration
  - Facilitate SW and HW co-design
- Off-the-shelf product responds better to the fluctuation of market demand

Performance

- FPGA has comparable **embedded RAM and processor** performance
- FPGA vendors offer custom-made **IP and high-speed cells** to help achieving the performance
- FPGA uses state-of-art **65nm process technology**
- Programmability makes **FPGA inherently slower than ASIC**
- **ASIC has lower power product**

Cost

- **Cost calculation**

  Product Cost =

  Fixed Part Cost + Variable Cost Per Part * Sales Volume

- FPGA has lower fixed cost than ASIC

  No NRE. Lower EDA tool cost. Lower design cost

- FPGA has higher variable part cost than ASIC
- FPGA shorter time-to-market helps capturing market opportunity. Delay to market could mean diminishing sales
- FPGA is more cost effective for medium and low volume products than ASIC

Risk

- **FPGA has lower risk than ASIC**
- ASIC has high silicon re-spin cost therefore much higher risk in committing to prototype
  - Evolving standards
  - Lack of in-system testing for interoperability
  - Months' long turn cycle for bug fixes
- **ASIC has higher risk** in getting return back on investment due to
  - Higher development cost
  - Longer development cycle
  - Not as adaptive to market change

Technology Migration

- Both FPGA and ASIC offer future technology roadmap
- Technology migration to **ASIC means smaller process geometry**
- FPGA technology migration is non-linear
- Build-in feature in Altera FPGA P&R tool to ensure correct migration
- Altera offers hardcopy for further cost reduction & performance boost from FPGA

ASIC Design v. FPGA : Primary Building Elements

	ASIC	FPGA
Logic Cells	Library of Standard Cells	Adaptive Logic Module with Programmability
Memory	Generated through Memory Compiler	Memory Blocks
I/O	I/O Cell Library	Programmable IOs & Dedicated Special IO Circuitry
Routing	Detailed	Segmented, Distributed & Dedicated
Dedicated Resources	None; but can Embed any Custom Blocks	PLL DSP

Memory

- **ASIC**
  - Generate memory using Memory Compiler
  - Instantiate the memory modules
- **FPGA**
  - Support synchronous modes only
  - Altera TriMatrix Memory consists of memory block sizes of 512, 4k and 512k bits per block
  - Memory blocks are organized in columns, up to 9Mbits
  - Memory depth and width ratio are configurable
    - 32x16 to 512x1 for M512 block
    - 128x32 to 4kx1 for M4k block
    - 4kx128 to 64kx8 for M-RAM block

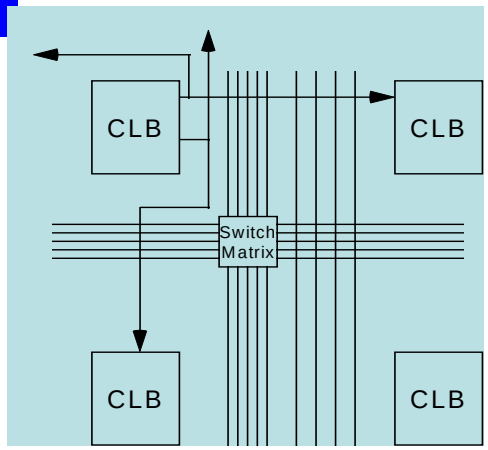
I/O

- **ASIC**
  - Instantiate the IO pads **provided in IO cell library**
- **FPGA**
  - IO structured in banks, up to 1,508 usable IOs
  - Support wide variety of I/O standards
    - Default standard is LVTTL
  - Has programmable driving strength
  - Dedicated circuitry for high speed IO support
  - No need for pad instantiation. IO pads are defined in synthesis constraint file

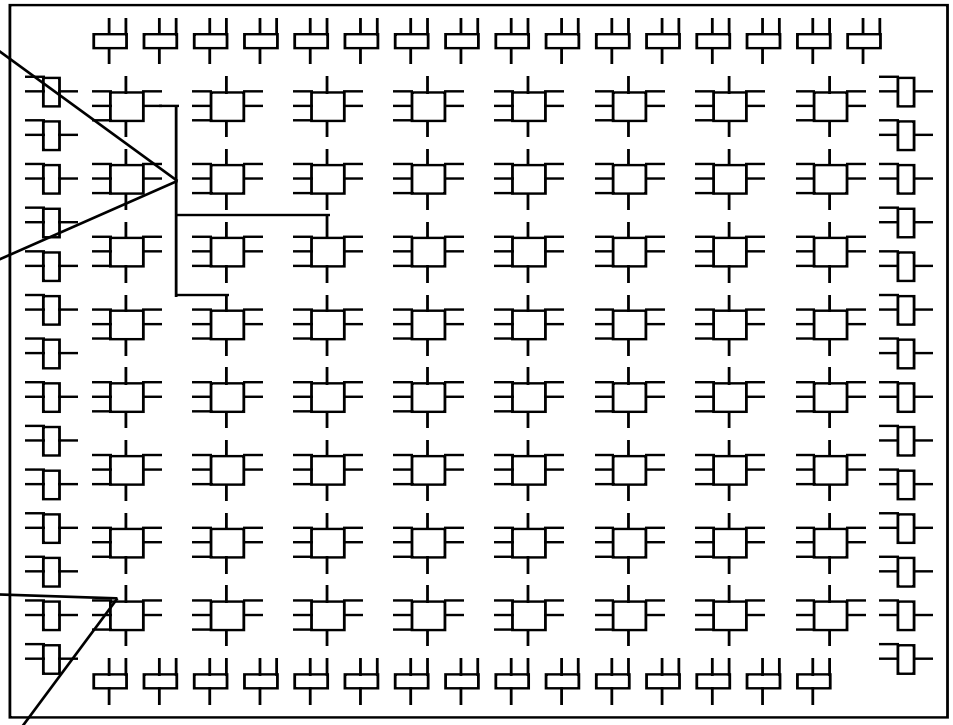
Routing

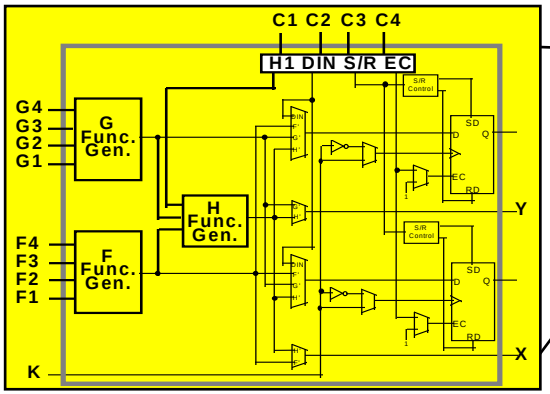
- **ASIC allows detailed routing among cells**
- FPGA uses a combination of programmable and dedicated routing lines
- FPGA high-speed interconnects
  - Register chain for fast shift function
  - Carry chain for high speed arithmetic operation
- FPGA general interconnects

Routing of FPGA



Programmable
Interconnect





Configurable
Logic Blocks (LE)

Performance Comparison

- **Combinatorial logic implemented in ASIC is typically faster than FPGA**
  - ASIC's fine-grain architecture allows more detailed routing and more predictable performance
  - FPGA's performance is more routing dependent
  - FPGA programmable routing structure limits levels of logic as apposed to ASIC detailed routing
- Critical paths that include IO, Memory & DSP have equivalent performance
  - FPGA dedicated resources make the implementation of these functions equivalent to ASIC

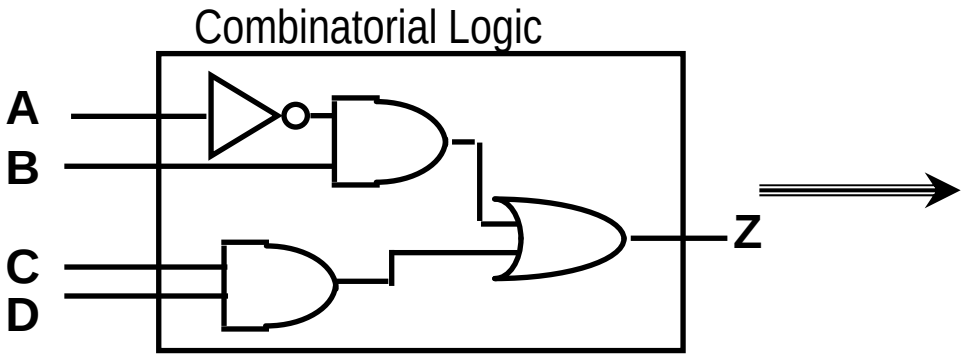
2. Two kinds of FPGA Architecture

- How do FPGA's work programmble logic circuits?

- **Look Up Table**
- **Multiplexer**

① Look Up Tables

- **Combinatorial Logic is stored in 16x1 SRAM Look Up Tables (LUTs) in a CLB**
- **Example:**



Look Up Table

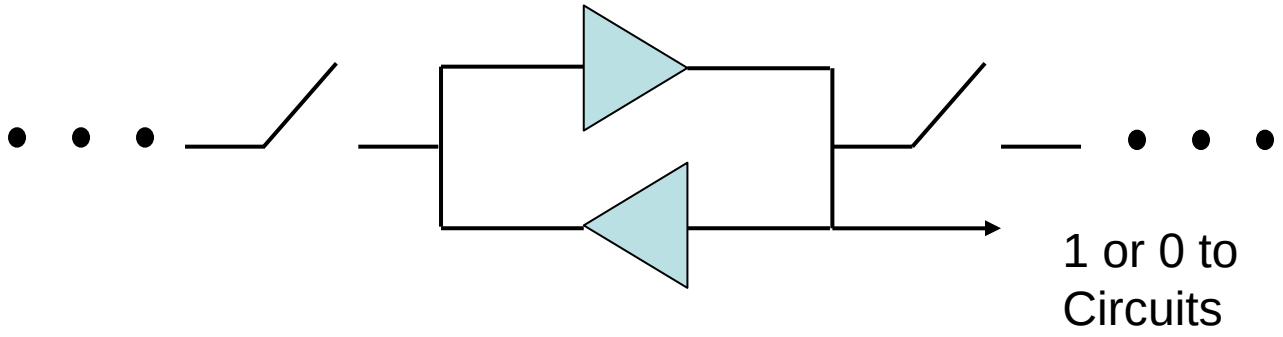
4-bit address

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

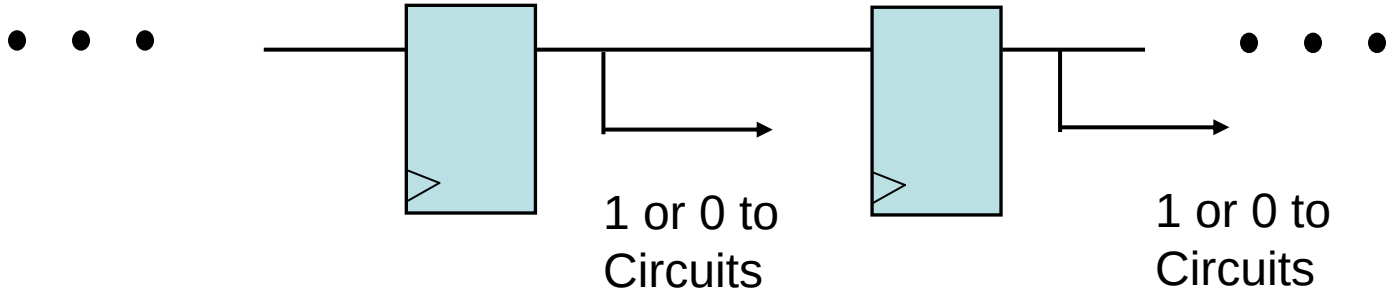
- Capacity is limited by number of inputs, not complexity
- Choose to use each function generator as 4 input logic (LUT) or as high speed sync.dual port RAM

Configuration Bits

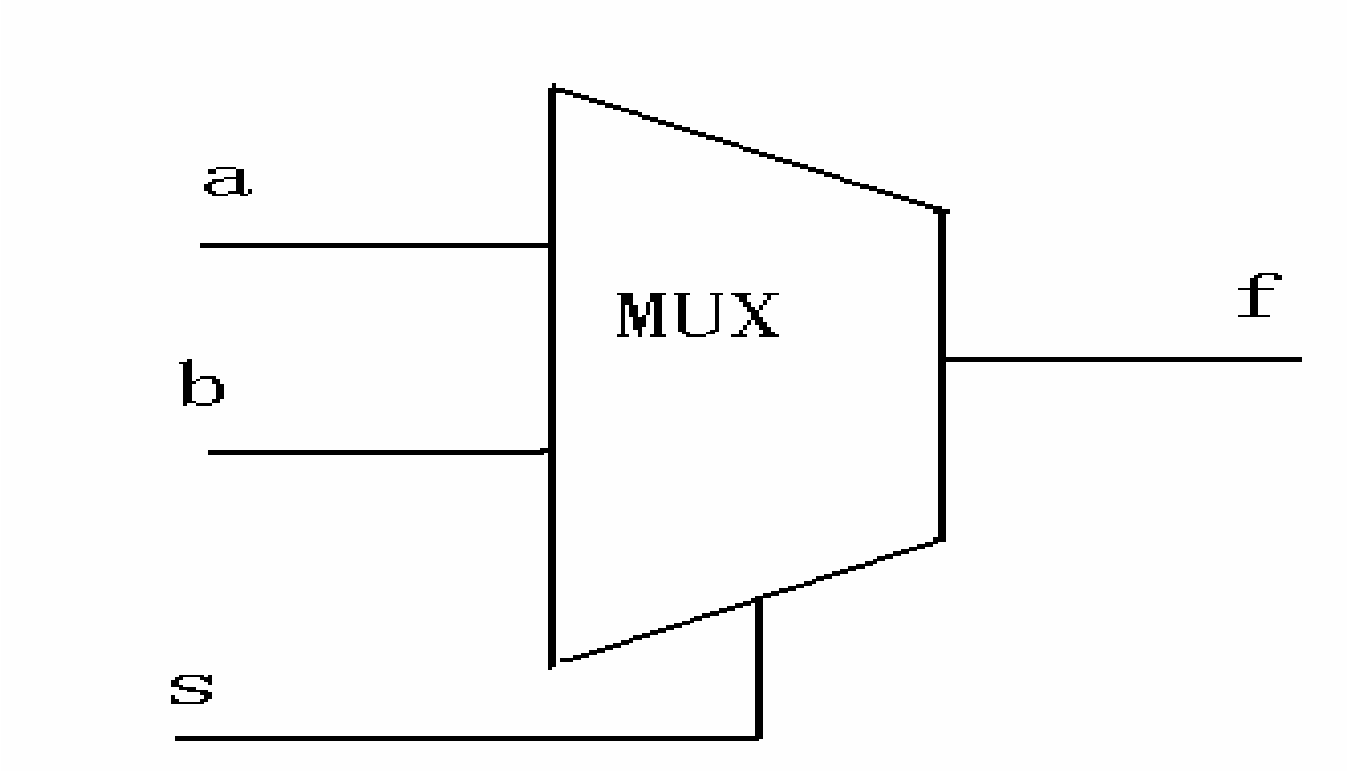
- There are Two Types of Configuration Bits
  - ✓ SRAM Cell (Majority of Bits)



  - ✓ Register (Allows Powered-up Reconfiguration)



② The another kind of FPGA Architecture



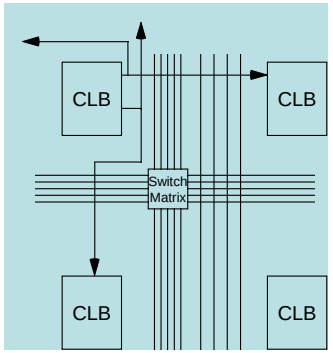
Example :

- A Mux with select *s and input a,b*

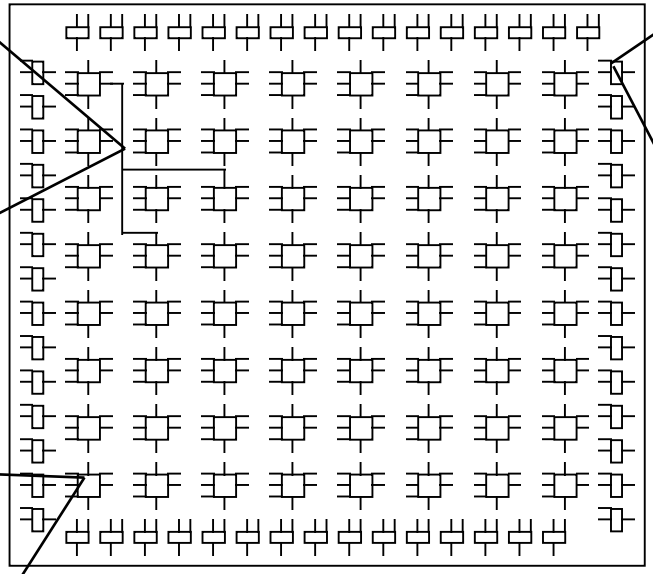
$$f = sa + \bar{s}b$$

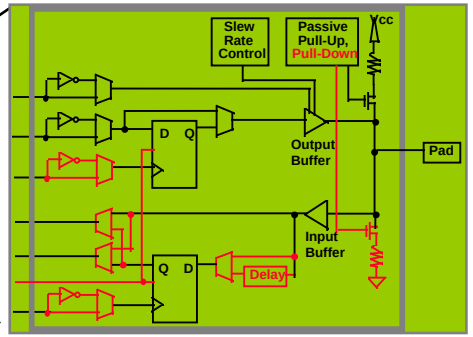
- when b=0,
- output $f = sa$
- The function is logic and
- when a=1,
- Output $f = s + b$
- The function is logic or

3. The Programmable Sources of FPGA

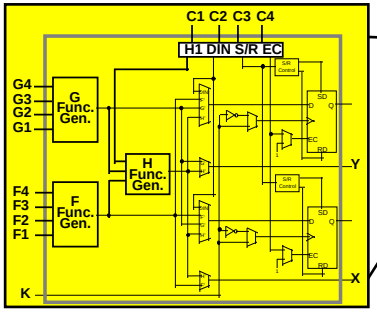


Programmable Interconnect





I/O Blocks (IOBs)



Configurable Logic Blocks (LE)

The sources of FPGA

① Configurable Logic locks (CLB) or Logic Element (LE)

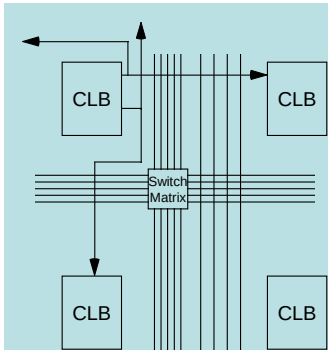
② Programmable I/O

③ Programmable interconnect

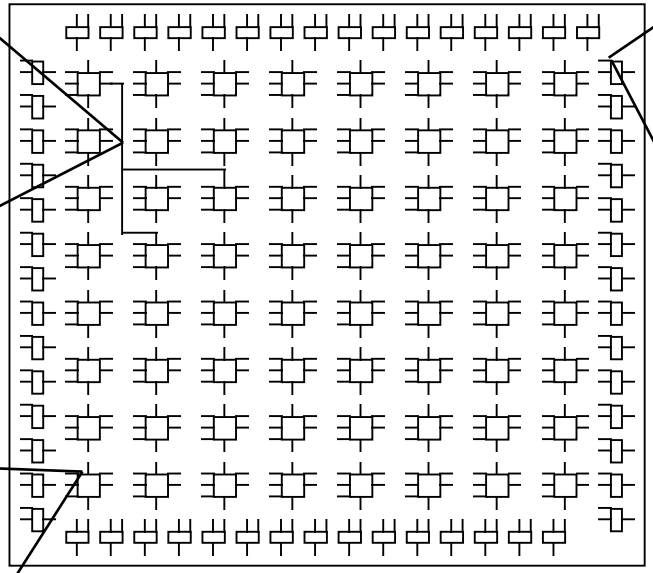
④ Hard Core in Chip

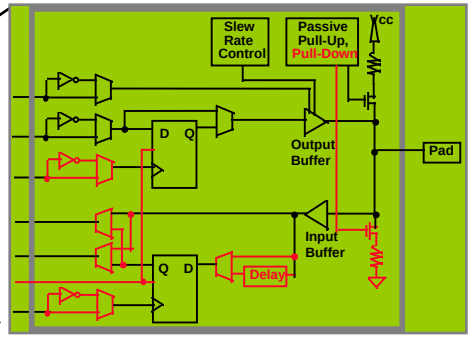
⑤ IP

The one of the Basic FPGA Architecture (Look Up Tables)

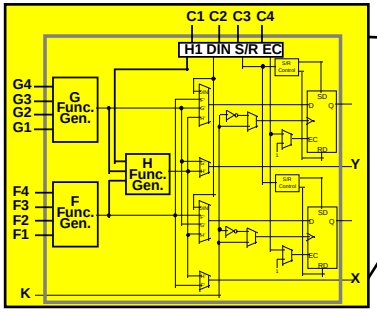


Programmable Interconnect





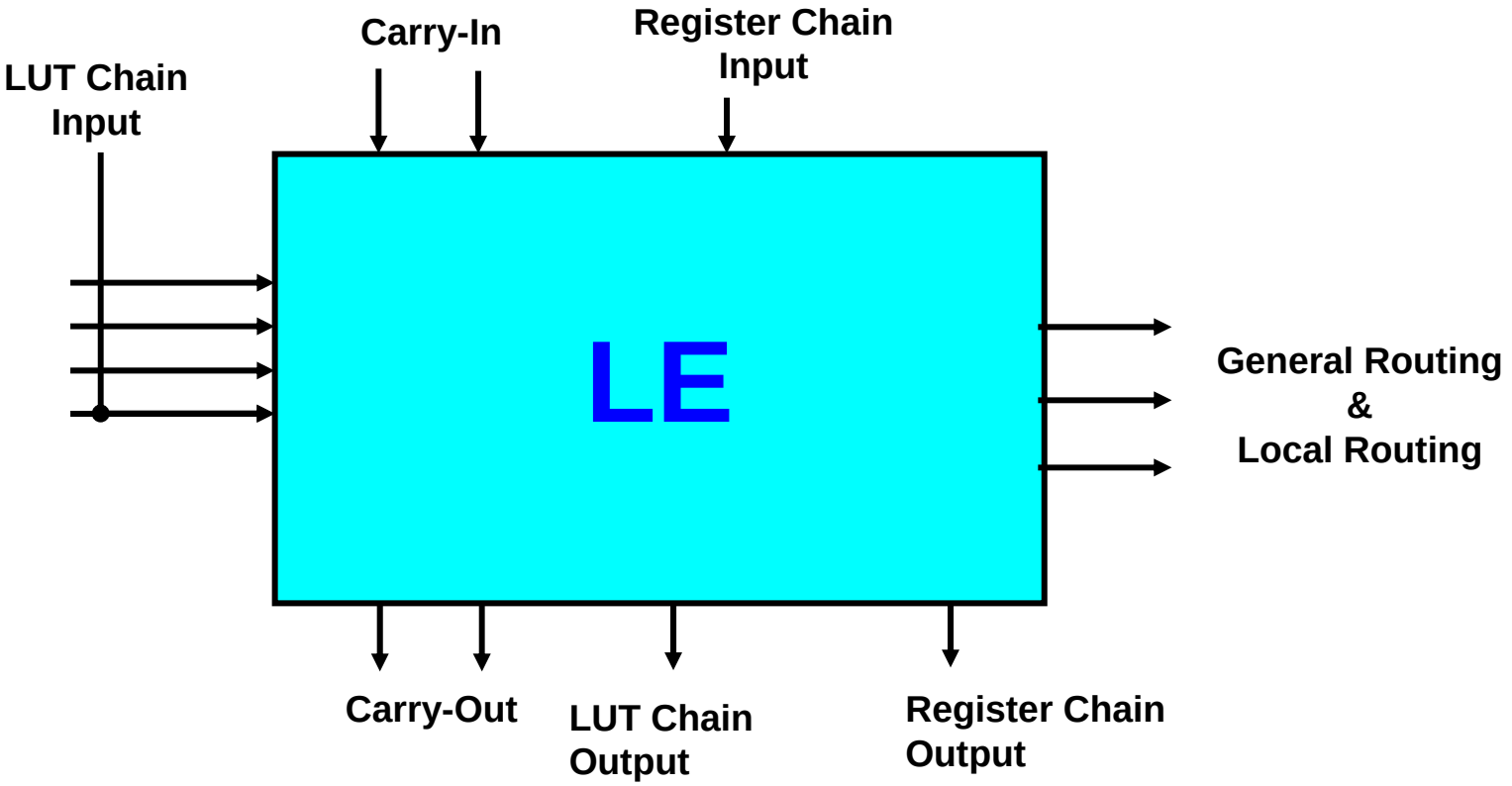
I/O Blocks (IOBs)



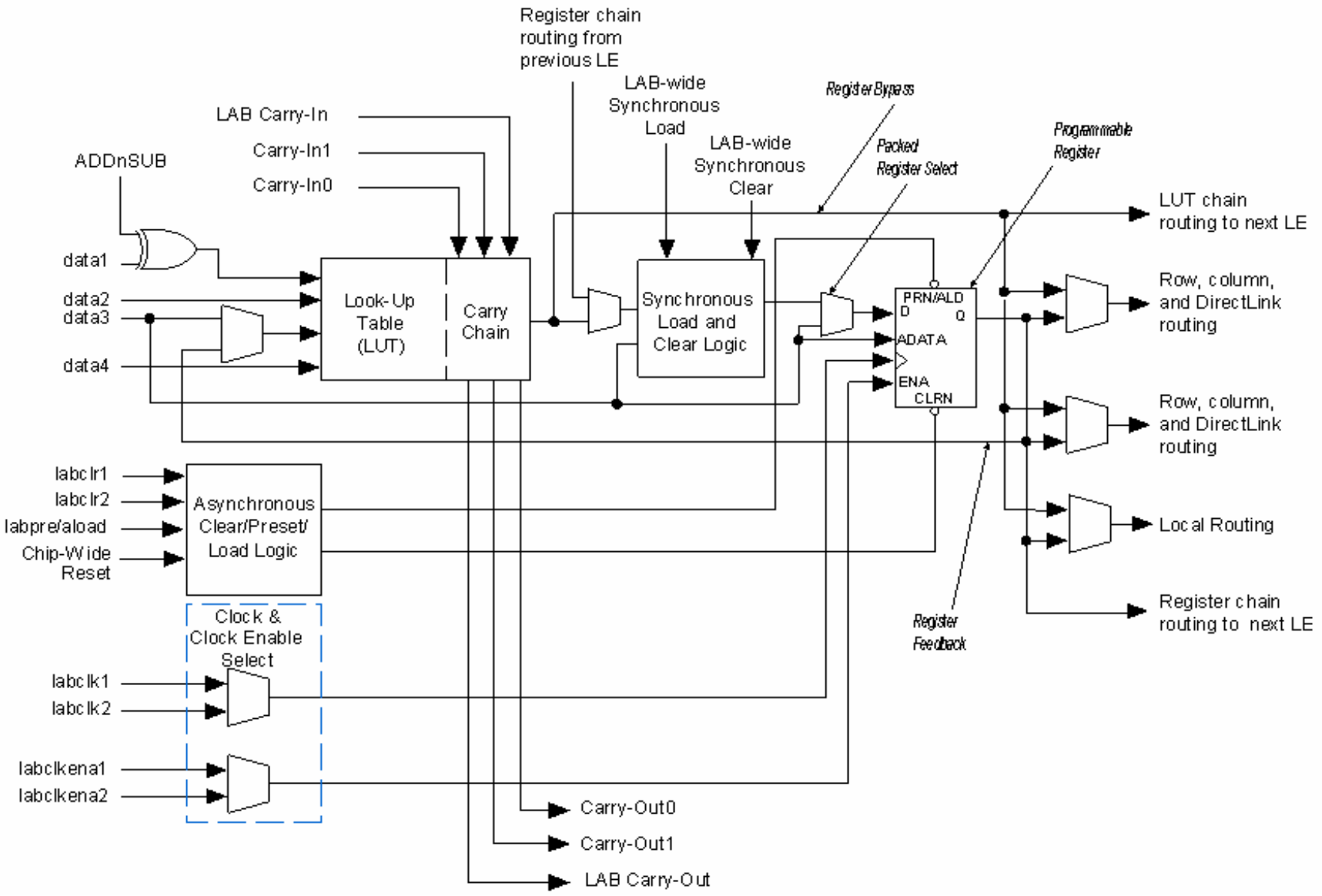
Configurable Logic Blocks (LE)

① Logic Elements

- Smallest Units of Logic
- Used for Combinatorial/Registered Logic



LE Datasheet Image



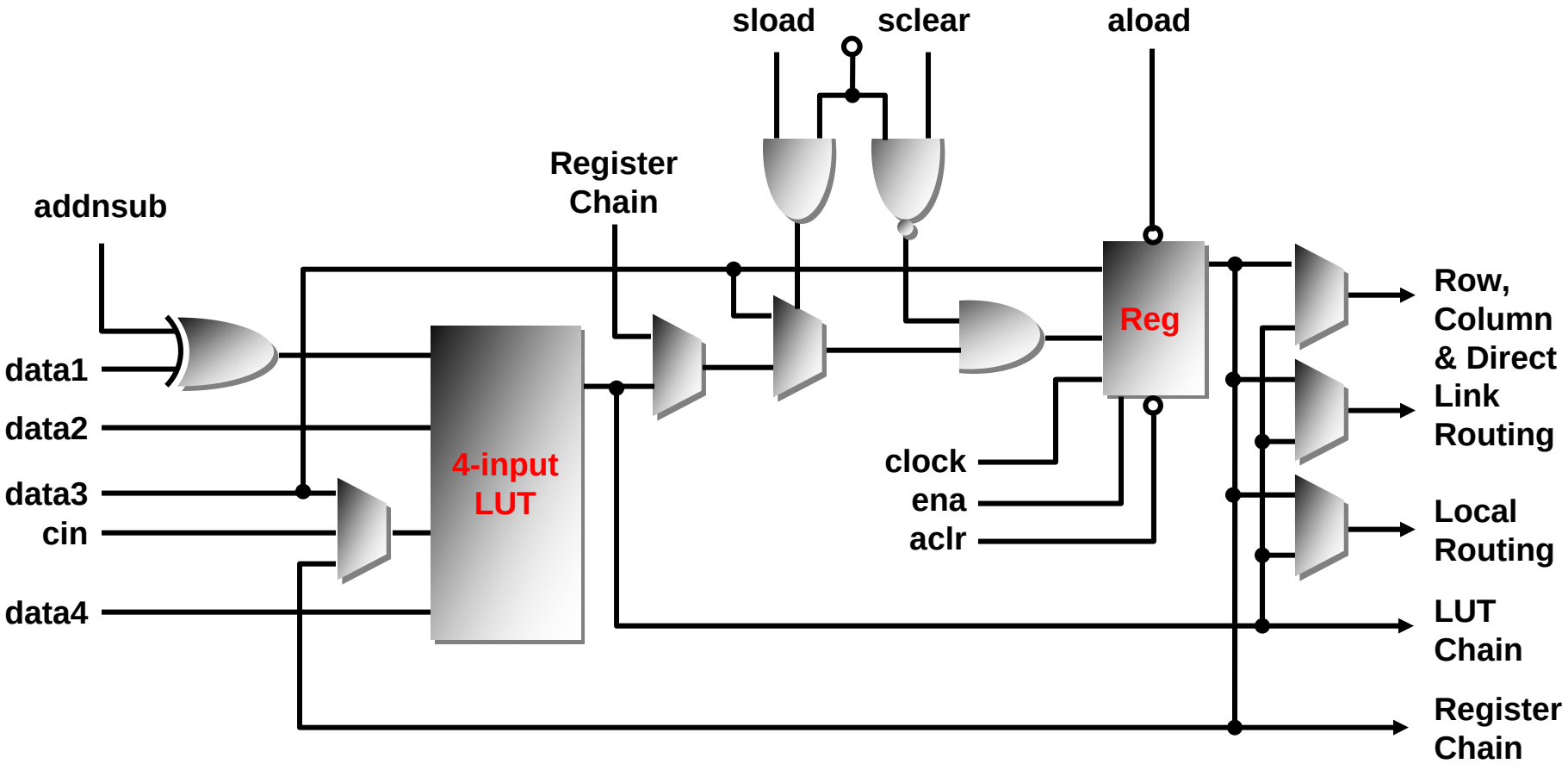
LE Features

- 4-input Look-Up Table (LUT)
- Configurable Register
- 2 Operation Modes
- Dynamic Add/Subtract Control
- Carry-Select Chain Logic
- Performance-Enhancing Features
  - LUT & Register Chain
- Area-Enhancing Features
  - Register Packing & Feedback

LE inputs/Outputs

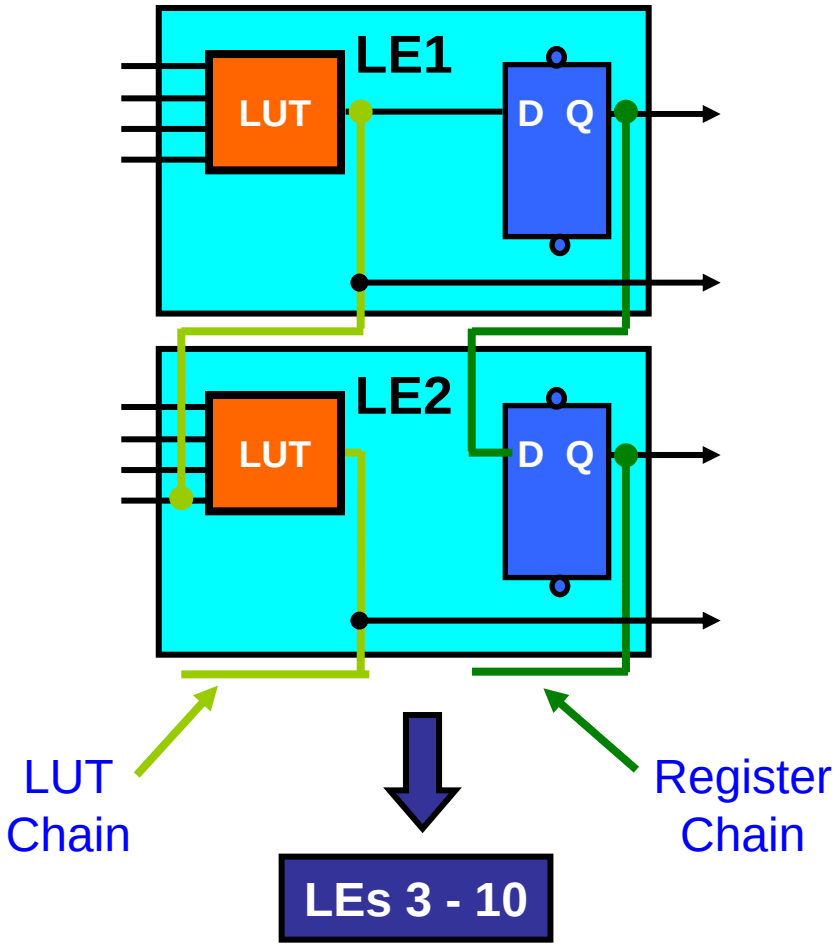
- inputs
  - 4 Data
  - 2 LE Carry-ins & 1 Lab Carry-in
  - 1 Dynamic Addition/Subtraction Control
  - Register Controls
- Outputs
  - 2 LE Carry-Outs
  - 2 Row/Column/DirectLink Outputs
  - 1 Local Output
  - 1 LUT Chain & 1 Register Chain

Altera Cyclone LE (Logic Element)

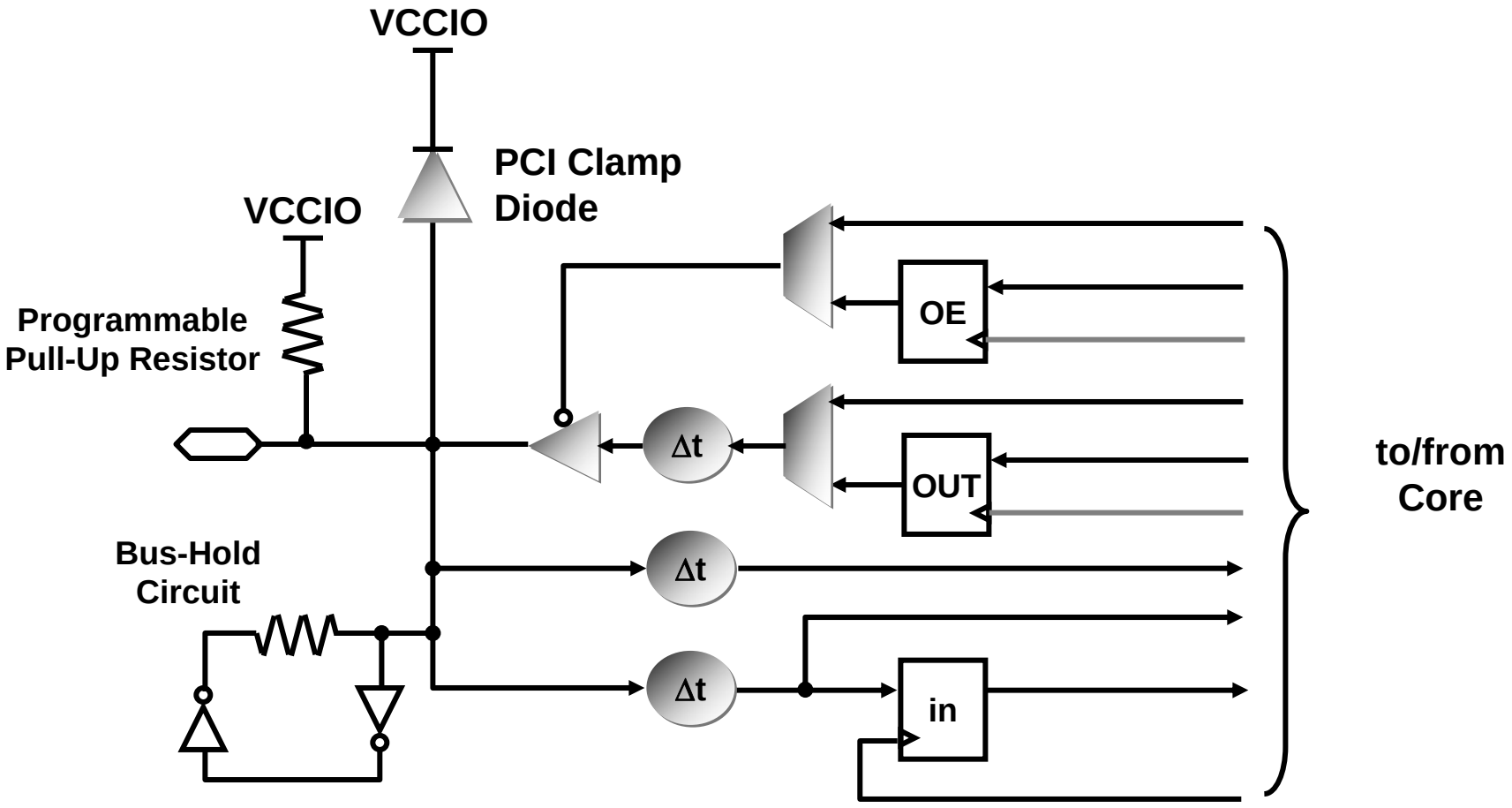


LUT & Register

- LUT Chain
  - Output of LUT Connects Directly to LUT Below
  - Available Only In Normal Mode
  - Ex. Wide Fan-In Functions
- Register Chain
  - Output of Register Connects Directly to Register Below (Shift Register)
  - LUT Can Be Used for Unrelated Function
  - Ex. LE Shift Register
- Both Chains End at LAB Boundary

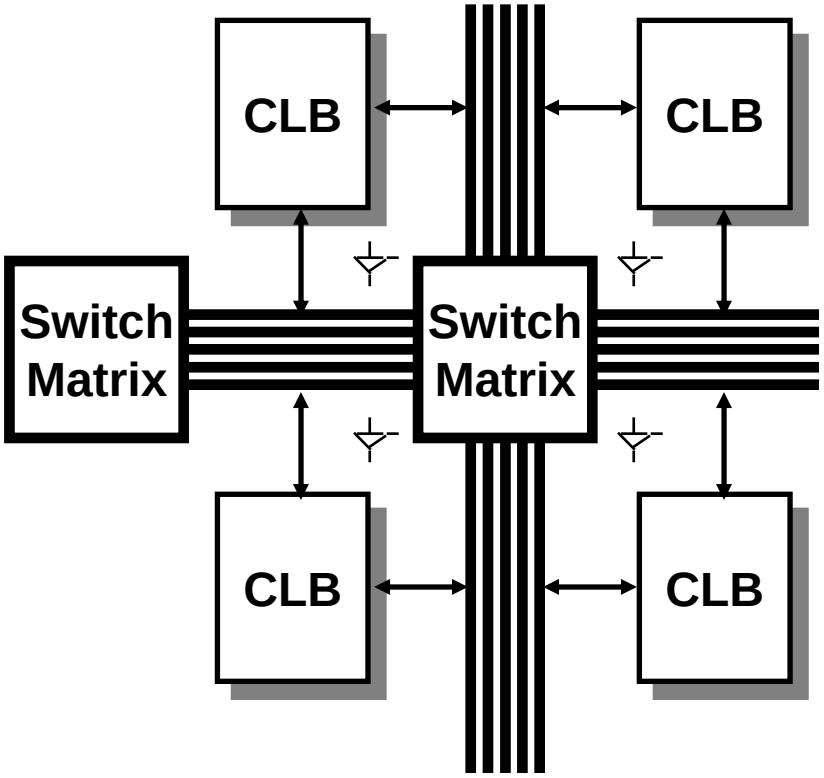


② I/O Element Structure



③ FPGA Routing

- 1) Fast Direct Interconnect - CLB to CLB
- 2) General Purpose Interconnect - Uses switch matrix
- 3) Long Lines
  - Segmented across chip
  - Global clocks, lowest skew
  - 2 Tri-states per CLB for busses
- Other routing types in CPLDs

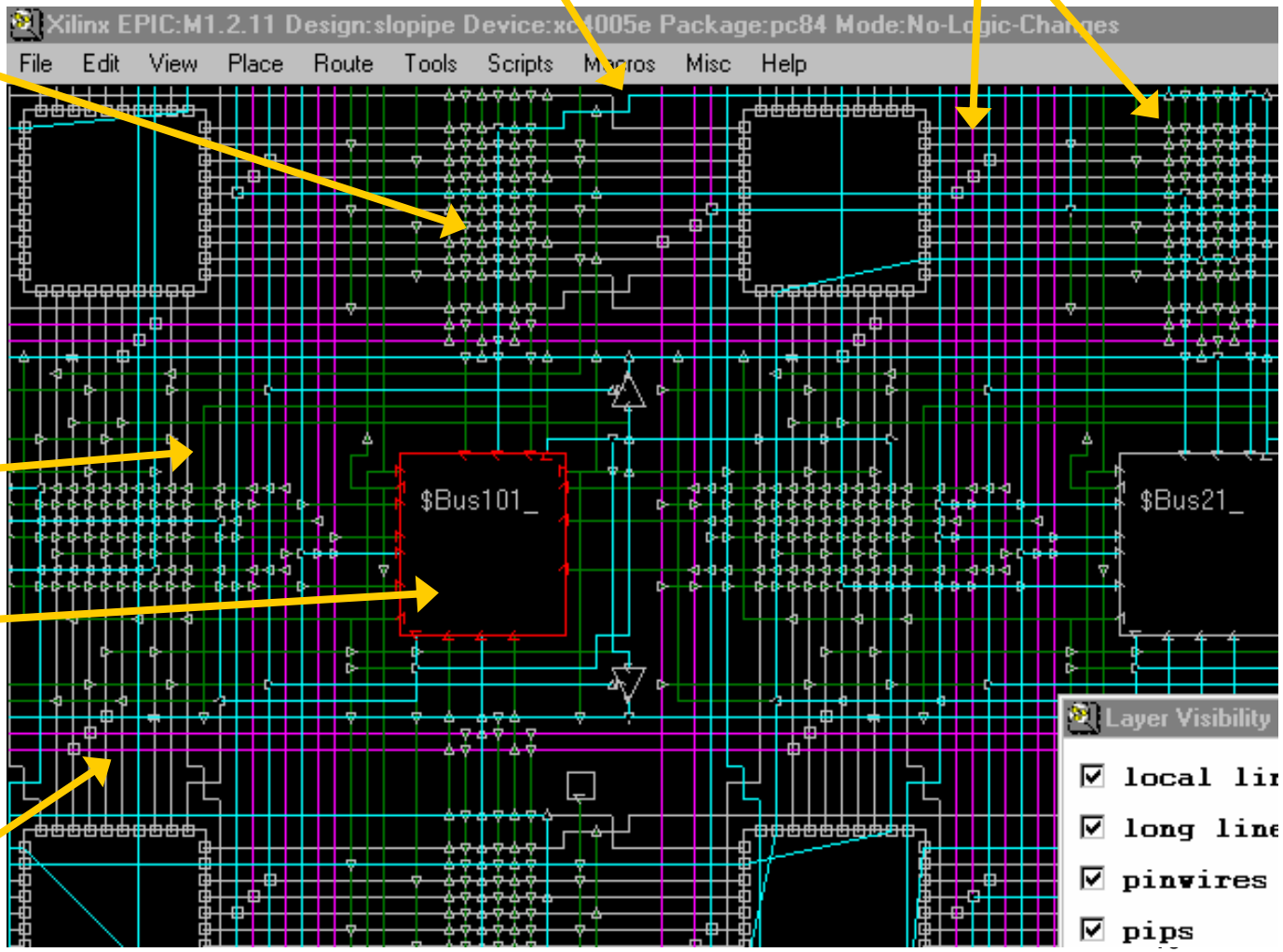


What's Really In FPGA

Programmable Interconnect Points, PIPs (White)

Switch Matrix

Routed Wires (Blue)



Direct Interconnect (Green)

LE (Red)

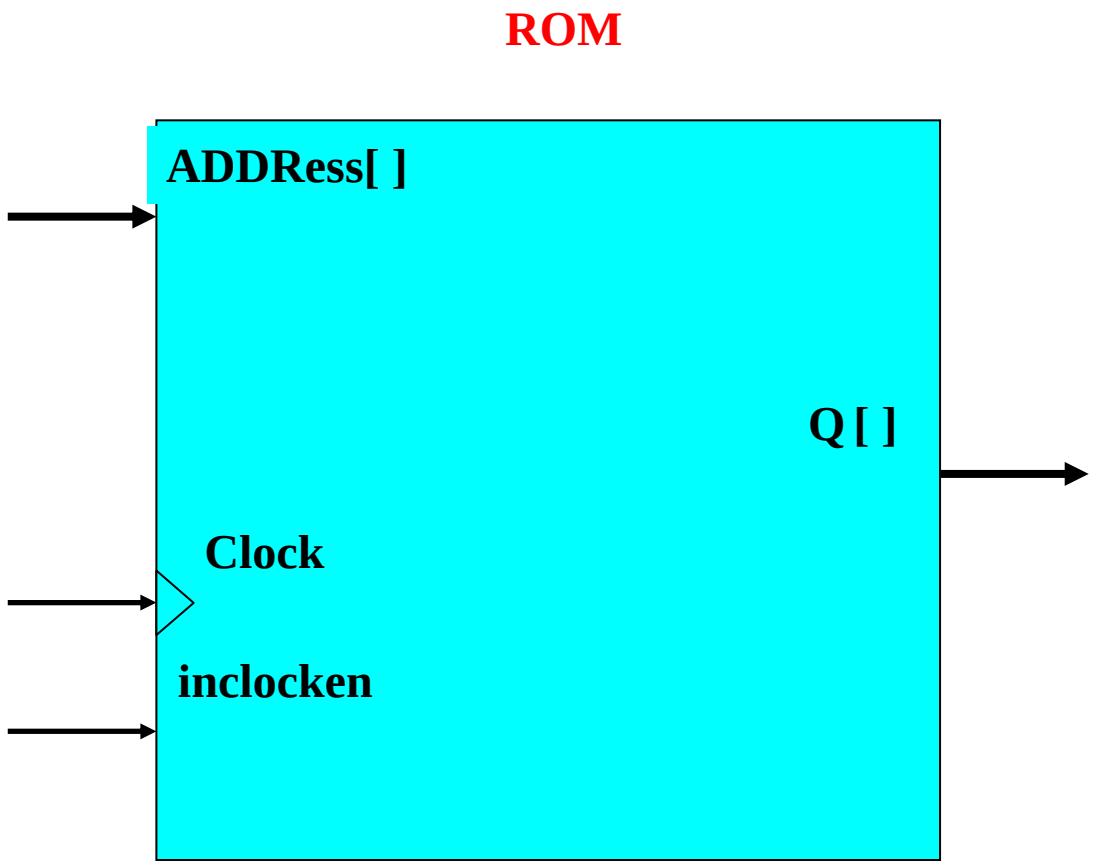
Long Lines (Purple)

④ Hard Core in Chip

- ROM
- RAM
- CPU
- DSP
- PLL
- CLOCK
- .....

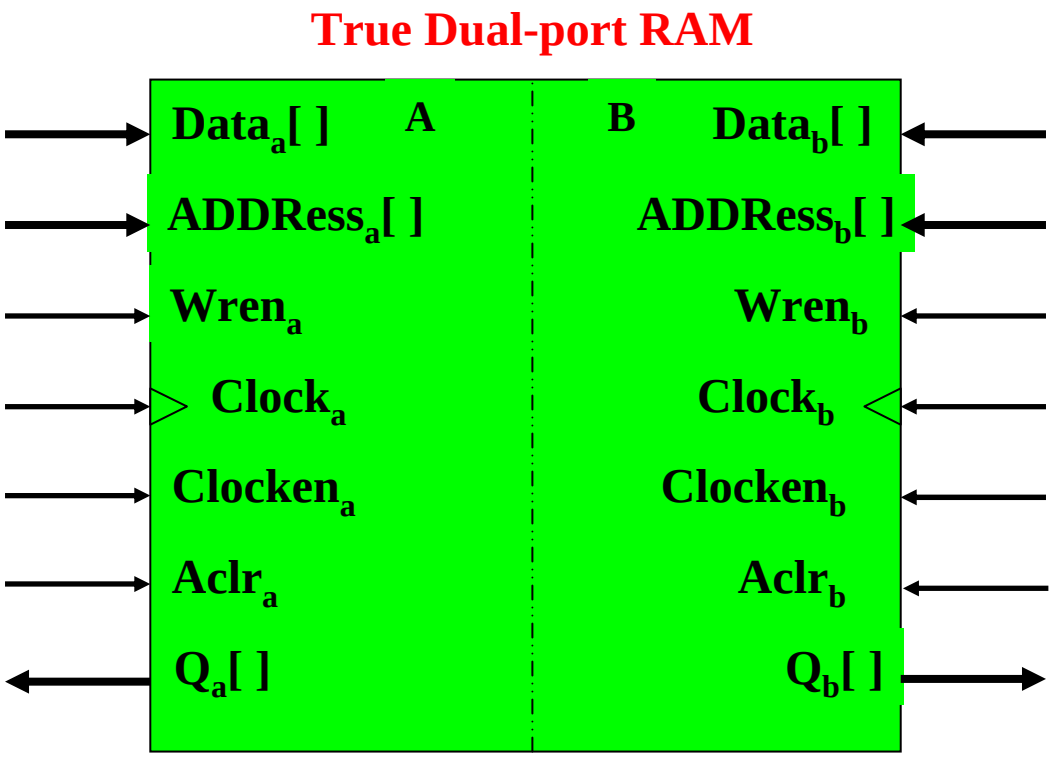
ROM

- Read-only Memory
  - Data Embedded in Configuration File
  - Loaded at Configuration Time
  - Inputs Are Registered
  - Data Cannot Be Altered



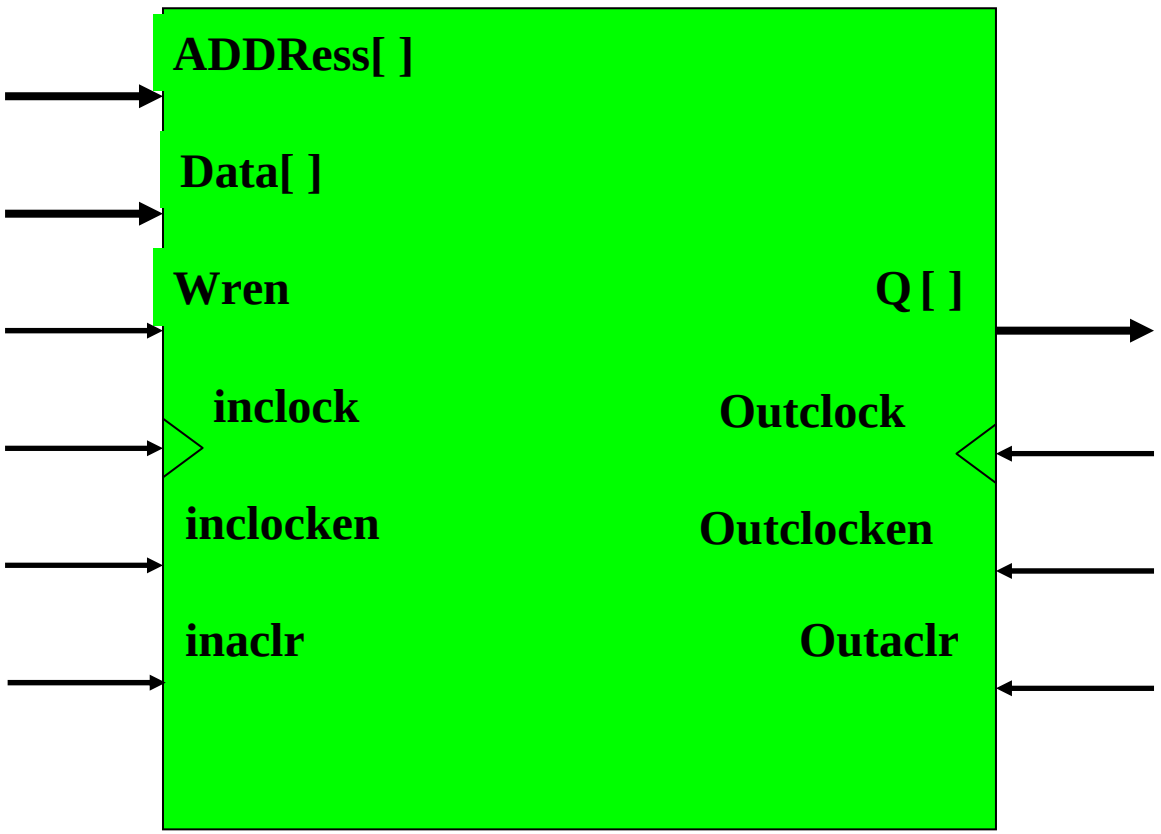
True Dual-port RAM

- True Dual-port RAM
  - Port A & Port B
  - All inputs Must Be Registered
  - Mixed Width Capability
- Clocking Options
  - Input/Output Clock Mode
  - Independent Clock Mode



Single-port RAM

- Single-port RAM
  - Read & Write Addresses Are the Same Port
  - All inputs Are Registered
- Clocking Options
  - Single Clock
  - Input/Output Clock Mode



⑤ IP CORE

- Hard IP：硬IP
- Soft IP:　软IP
- Firm IP:　固IP

IP Design Flow

Simulate RTL in
Third-Party Software

Now
Free!

Purchase After
RTL and H/W
Evaluation

Download

Parameterize

Evaluate in
Software

Purchase
License

Ship
Product

BUY

OpenCore™

MegaWizard™
Plug-In

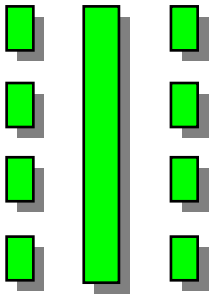
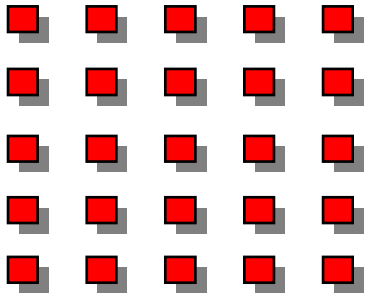
QUARTUS™ II

OpenCore®
Plus

Now
Free!

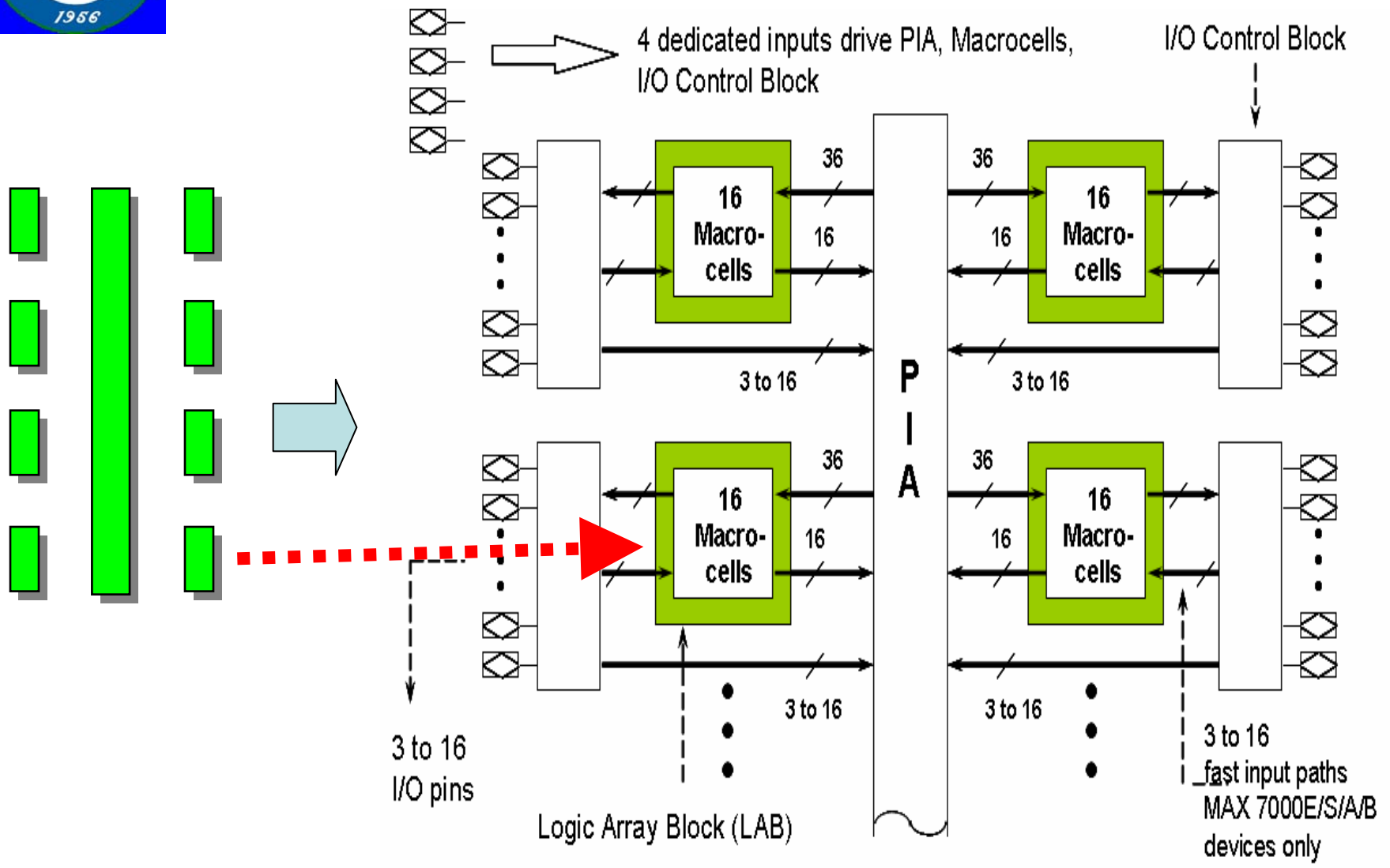
Evaluate in
Hardware

4. CPLD or FPGA?

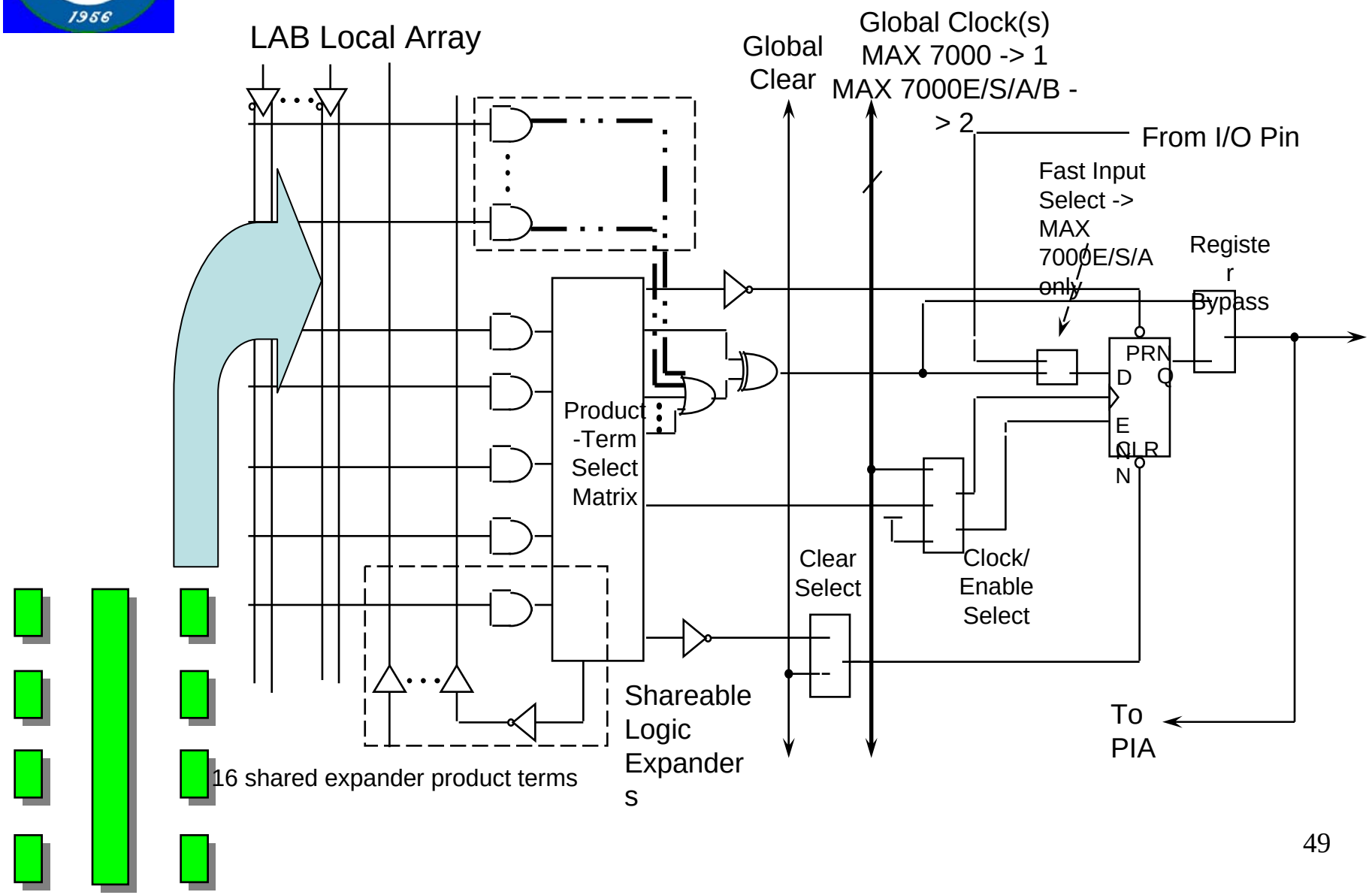
	Complex Programmable Logic Device	Field-Programmable Gate Array
	CPLD	FPGA
		
Architecture	PAL/22V10-like More Combinational	Gate array-like More Registers + RAM
Density	Low-to-medium 0.5-10K logic gates	Medium-to-high 1K to 20M system gates
Performance	Predictable timing Up to 250 MHz today	Application dependent Up to 400MHz today
Interconnect	“Crossbar Switch”	Incremental

Not shown: Simple PLD (SPLD) Architecture

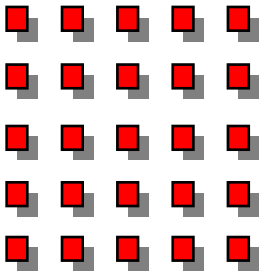
CPLD

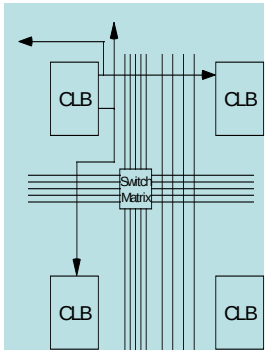

4 dedicated inputs drive PIA, Macrocells, I/O Control Block
I/O Control Block
16 Macro-cells
36
16
3 to 16
PIA
3 to 16 I/O pins
Logic Array Block (LAB)
3 to 16 fast input paths
MAX 7000E/S/A/B devices only

Macrocell

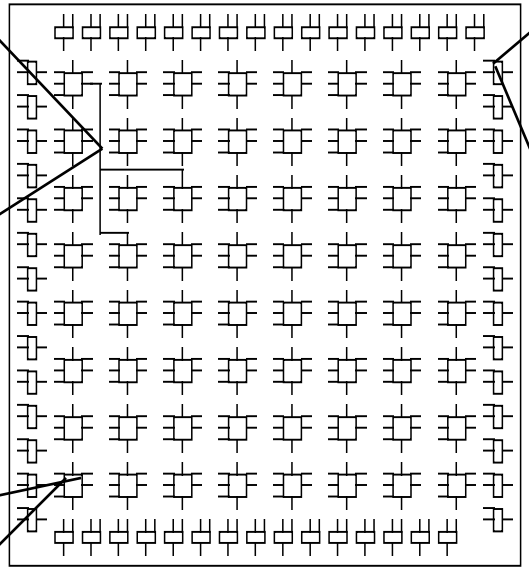

LAB Local Array
Global Clear
Global Clock(s)
MAX 7000 -> 1
MAX 7000E/S/A/B -
> 2
From I/O Pin
Fast Input
Select ->
MAX
7000E/S/A
only
Registe
r
Bypass
PRN
D
Q
E
CLR
N
Product
-Term
Select
Matrix
Clear
Select
Clock/
Enable
Select
To
PIA
Shareable
Logic
Expander
s
16 shared expander product terms

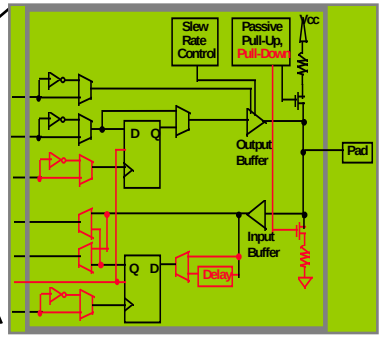
FPGA



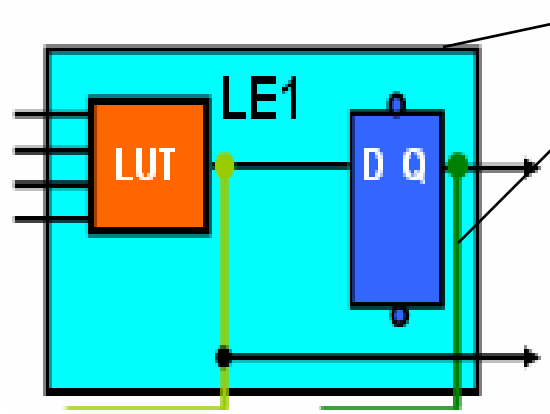


Programmable Interconnect





I/O Blocks (IOBs)



Configurable Logic Blocks (LE)

CPLD or FPGA?

CPLD

- Non-volatile
- JTAG Testing
- Wide fan-in
- Fast counters, state machines
- Combinatorial Logic
- Small student projects, lower level courses
- Control Logic

FPGA

SRAM reconfiguration

Excellent for computer architecture, DSP, registered designs

ASIC like design flow

Great for first year to graduate work

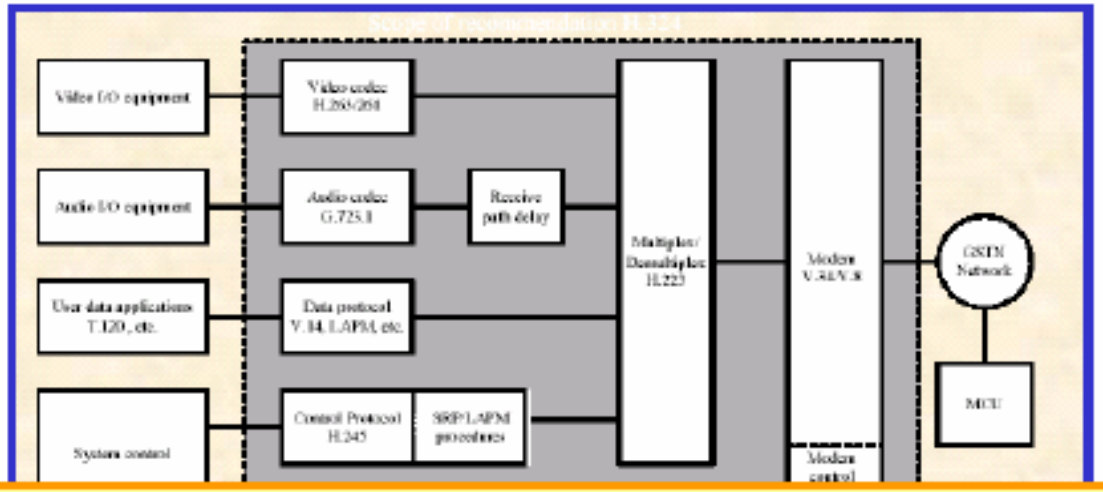
More common in schools

PROM required for non-volatile operation

5. Design of ASIC and FPGA Flow

- ASIC and FPGA have compatible yet moderately different design flow
- Both ASIC and FPGA use RTL based modular design methodologies
- FPGA design flow can be considered as a simplified version of full ASIC design flow
  - **No DFT required, Deep Sub-Micron (DSM) verification completed by vendor, no waiting for prototype**
- FPGA allows for reduced design cycles, faster bug-fixes and feature additions at no extra cost
- For high performance designs, FPGA code may require optimization

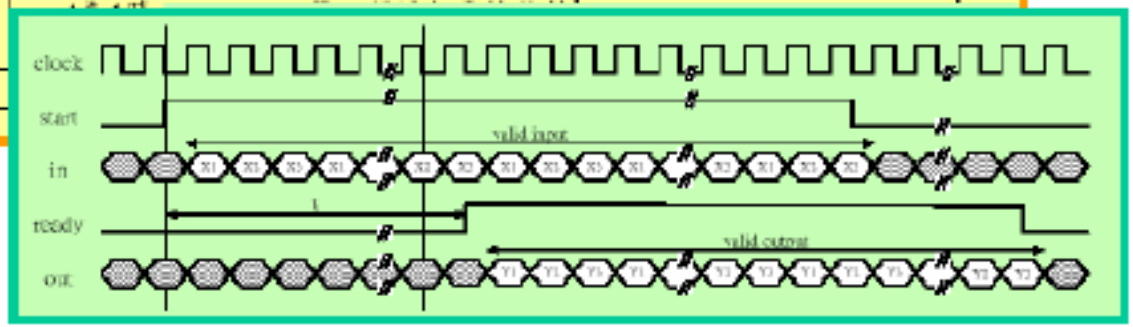
System Specification



Partition

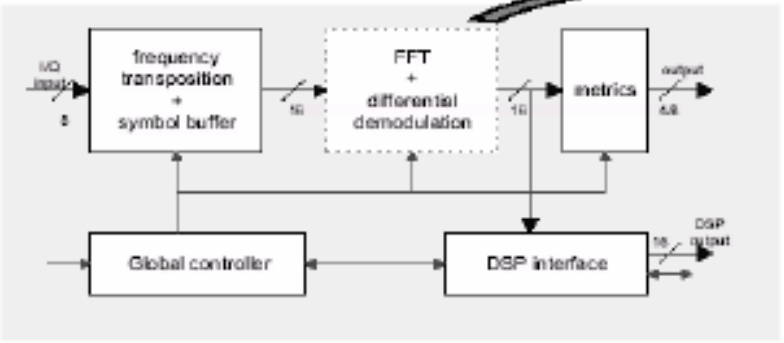
腳位名稱		描述	Drive Strength/Output Load
clk	輸入	系統時脈	assume infinite
reset	輸入	系統重置訊號，high active	1 ns/pf
din	輸入	每個clock cycle輸入一個16-bit 正整數	1 ns/pf
ready	輸出		
dout	輸出		

IO Spec.



IO Timing Spec.

Algorithm Mapping

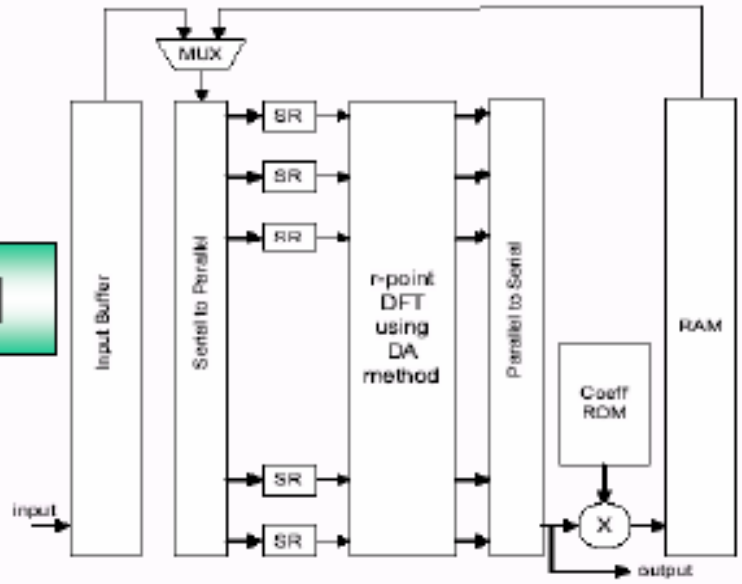


System/Algorithm Level

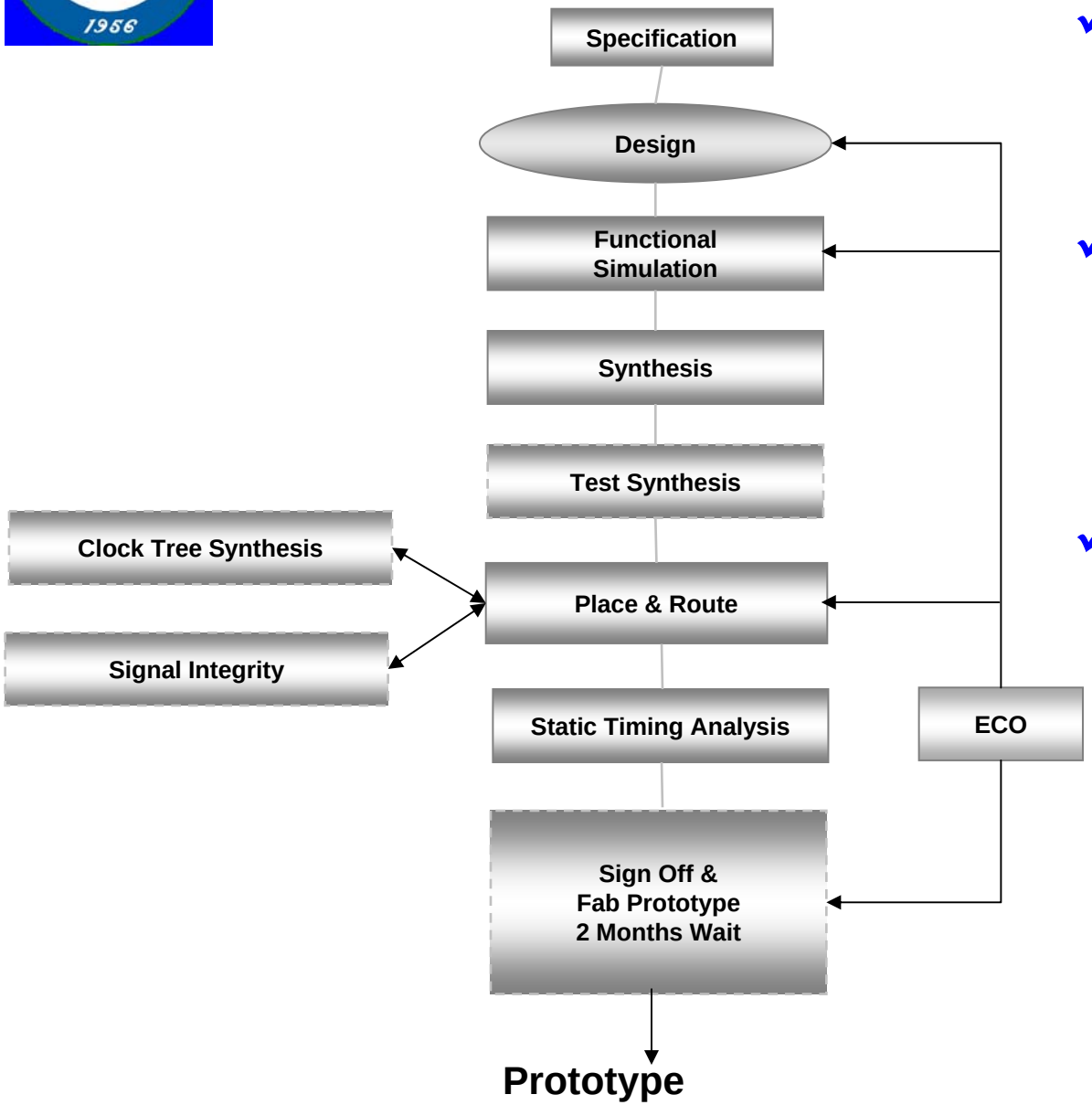
$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad k = 0,1,\ldots,N-1$$

$$X(k) = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{nk}$$

RTL Level

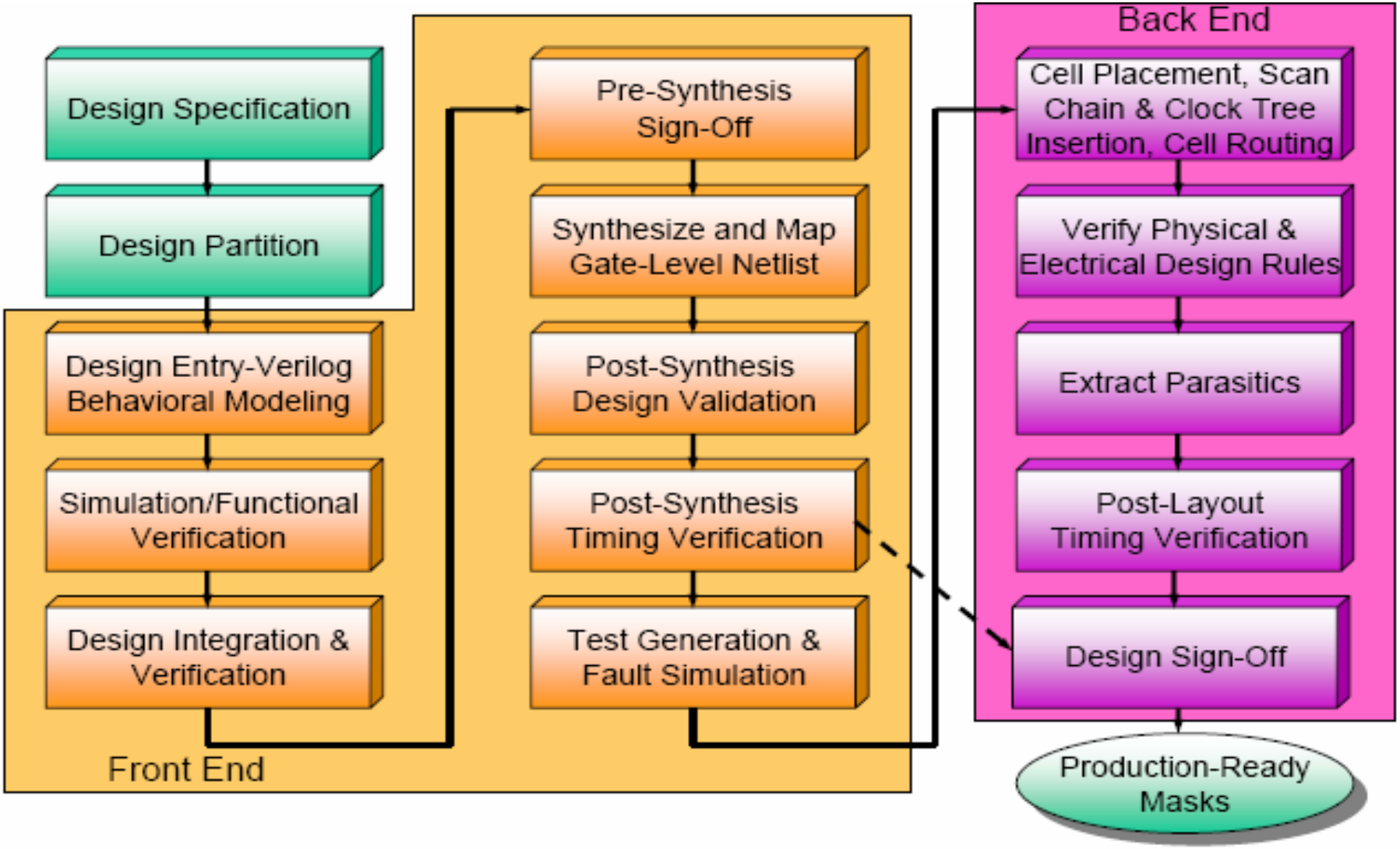


ASIC Design Flow

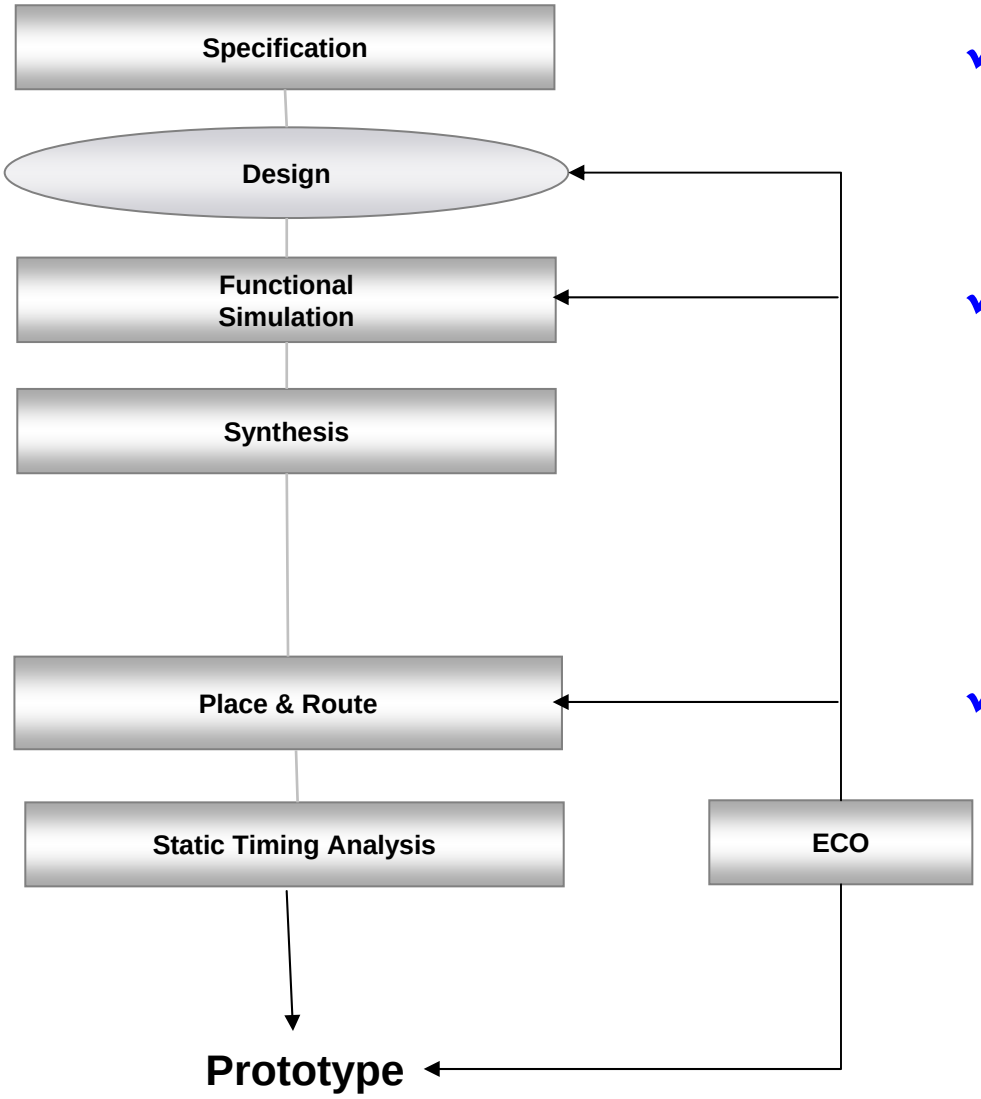


- ✓ ASIC tools are generally scripts driven
- ✓ Clock tree synthesis and signal integrity need to be performed during Place & Route
- ✓ Post-layout Static Timing Analysis (STA) and Equivalency Checking are musts for sign-off to foundry

Digital IC Design Flow



FPGA Design Flow



- ✓ FPGA tools are generally GUI-driven but do have scripting capability
- ✓ Once design passes simulation and STA, it can be immediately downloaded to FPGA for in-system verification
- ✓ Bug fixes and ECO are matter of minutes/hours

ASIC Implementation

- RTL Design
  - Optimized for ASIC technology and area
- Synthesis
  - Synopsys Design Compiler, primarily driven by scripts
  - DFT logic insertion including BIST, scan and JTAG
- Place and Route
  - Generally done in Foundry, requires Foundry interface
  - Cadence, AVANT!

FPGA Implementation

- RTL Design
  - Targeted at FPGA IO, memory, DSP and PLL
  - Optimized to FPGA architecture for performance
- Synthesis
  - SynplifyPro, Quartus， ISE,….
  - FPGA Compiler II, Push-button flow with scripting capabilities using Tcl
- Place and route
  - Completely done internally
  - FPGA P&R tool - Altera Quartus ,ISE
  - Push-button flow with scripting capabilities using Tcl

ASIC Verification

- RTL functional simulation
- Post-synthesis Static Timing Analysis
- Post-synthesis Equivalency Checking
- Post-Place & Route Static Timing Analysis
- Post-Place & Route Equivalency Checking
- Post-Place & Route Gate Level Simulation
- DSM verification
- In-system verification

FPGA Verification

- More simplified over ASIC verification because there is no expensive tape out and bug fixes
- Use same ASIC discipline for RTL development and functional verification. Ad hoc approach must be avoided

- Key verification points
  - RTL functional simulation
  - Post-Place & Route STA
  - In-system Verification
    - In-system verification may start as soon as basic set of functionality has been verified in simulation

Design for Test (DFT)

- FPGA design does not need to perform DFT
- **FPGA device is 100% tested by vendor**
- FPGA already contains JTAG
- FPGA saves time on test logic insertion and therefore provides more time for system testing

FPGA In-System Debugging Tool

- FPGA offers powerful, easy-to-use in-system debugging tools
- Altera Place & Route tool supports two integrated debugging tools
  - **SignalProbe** - Route internal nodes to unused I/O pins. External logic analyzer can be used to monitor these pins
  - **SignalTap II** – Embed logic analyzer, capture the data of internal nodes and transfer the data real time, via a download cable, to Quartus II software to be viewed in a waveform
    - Up to 1024 data channels. Up to 128K sample per channel
    - Circular or Segmented data acquisition methods
    - Up to 10 trigger level per channel

Engineering Change Order (ECO)

- ECO to prototype can be done in minutes for FPGA rather than months for ASIC
- Chip Editor
  - Altera integrated tool for performing ECO
  - Make minor modifications to LE, IO, PLL and routing
  - Good for quick turn-around for
    - Correcting design flaw
    - Doing ECO
    - Tweaking timing
- Need to make the permanent change to the source and recompile

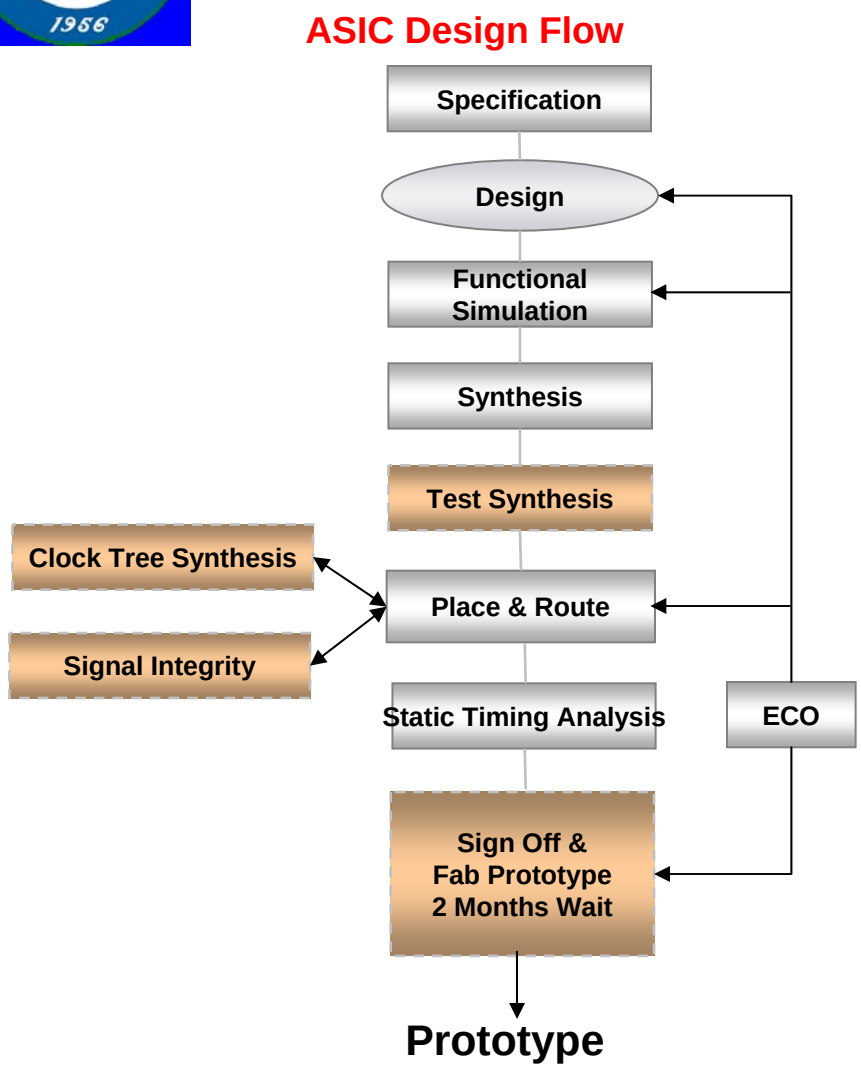
Equivalency Checking

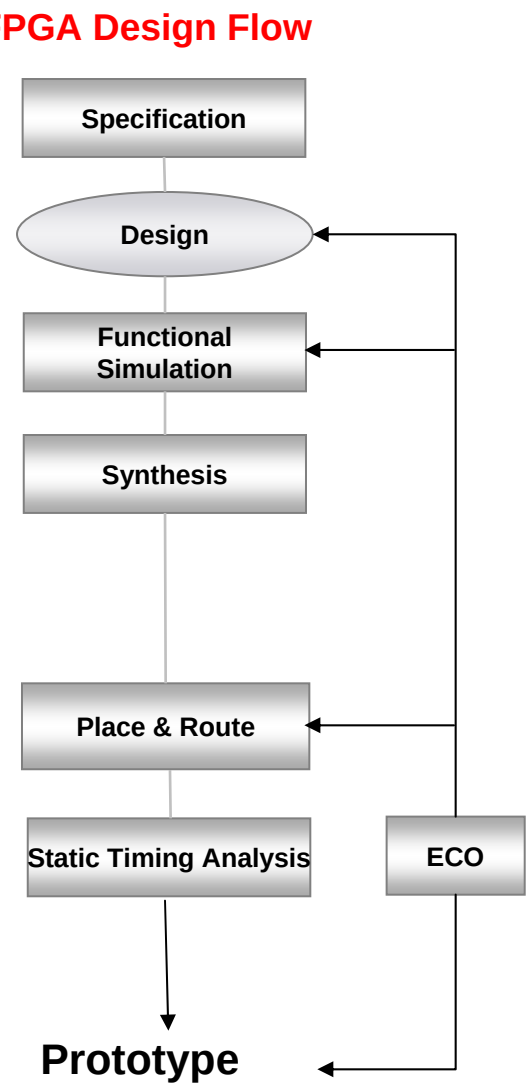
- Used to guarantee other “views” of the design match the golden RTL description
- Verplex Conformal LEC, Synopsys Formality
- Within Altera FPGA design flow, formal verification can be performed between the SynplifyPro synthesized netlist and the Quartus II post-fit netlist

Firmware Development

- Firmware development starts much earlier in the development cycle for FPGA than for ASIC
- Firmware development can start even earlier if prototypes can begin with basic feature set and increment the feature content in the follow-on drops

ASIC vs. FPGA Design Flow





Not Needed for FPGA

FPGA Design Flow Advantages

- No need for DFT logic insertion
- No need for post-synthesis STA
- Place & Route always in house
- No need for clock tree insertion and signal integrity check during Place & Route
- Earlier prototyping and volume production
- Earlier firmware development
- Faster turn-around for bug fixes and feature additions

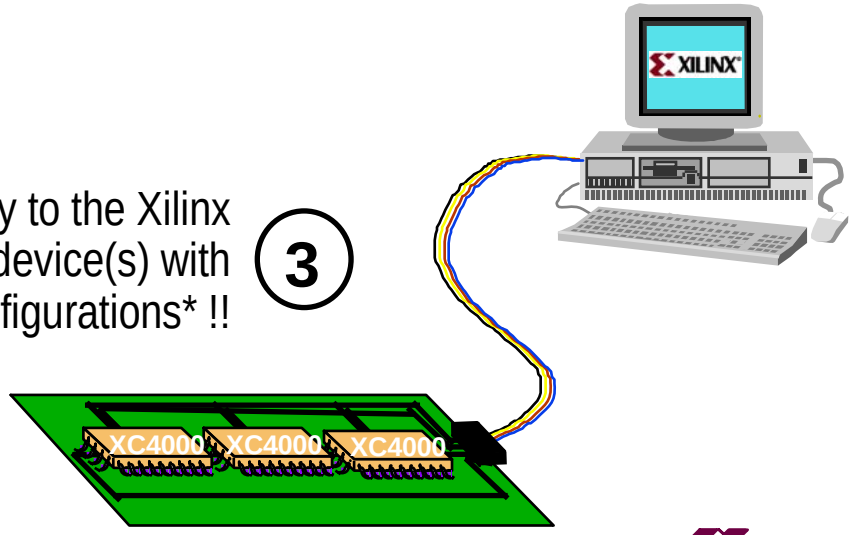
Design Flow

1. Design Entry in schematic, VHDL, and/or Verilog. Vendors include Synopsys, Aldec (Xilinx Foundation), Mentor, Cadence, Viewlogic, and others.

2. Implementation includes Placement & Routing and bitstream generation using EDA Technology. Also, analyze timing, view layout, and more.

M1 Technology

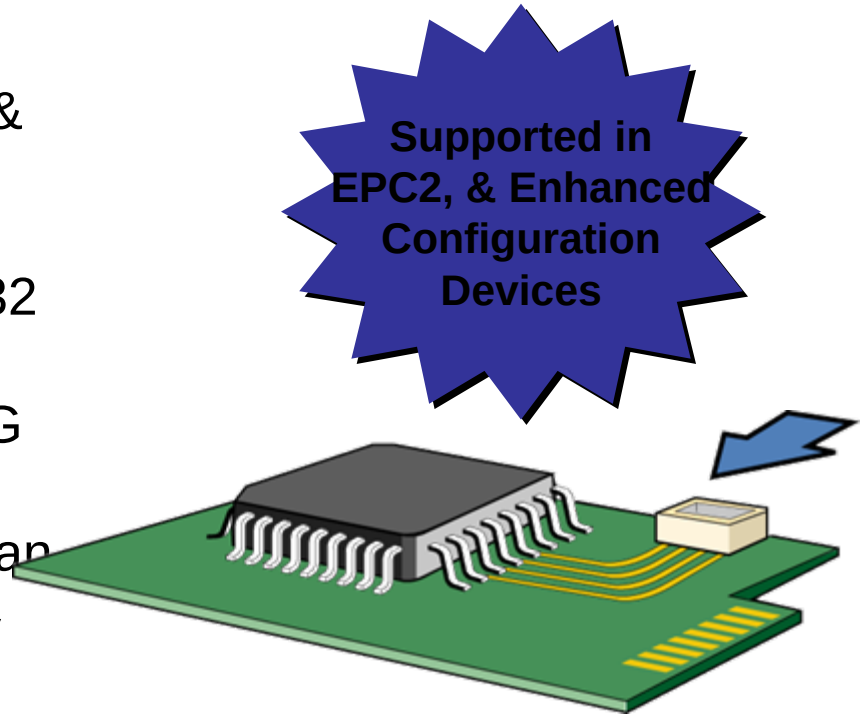
3. Download directly to the Xilinx hardware device(s) with unlimited reconfigurations* !!



*XC9500 has 20,000 write/erase cycles

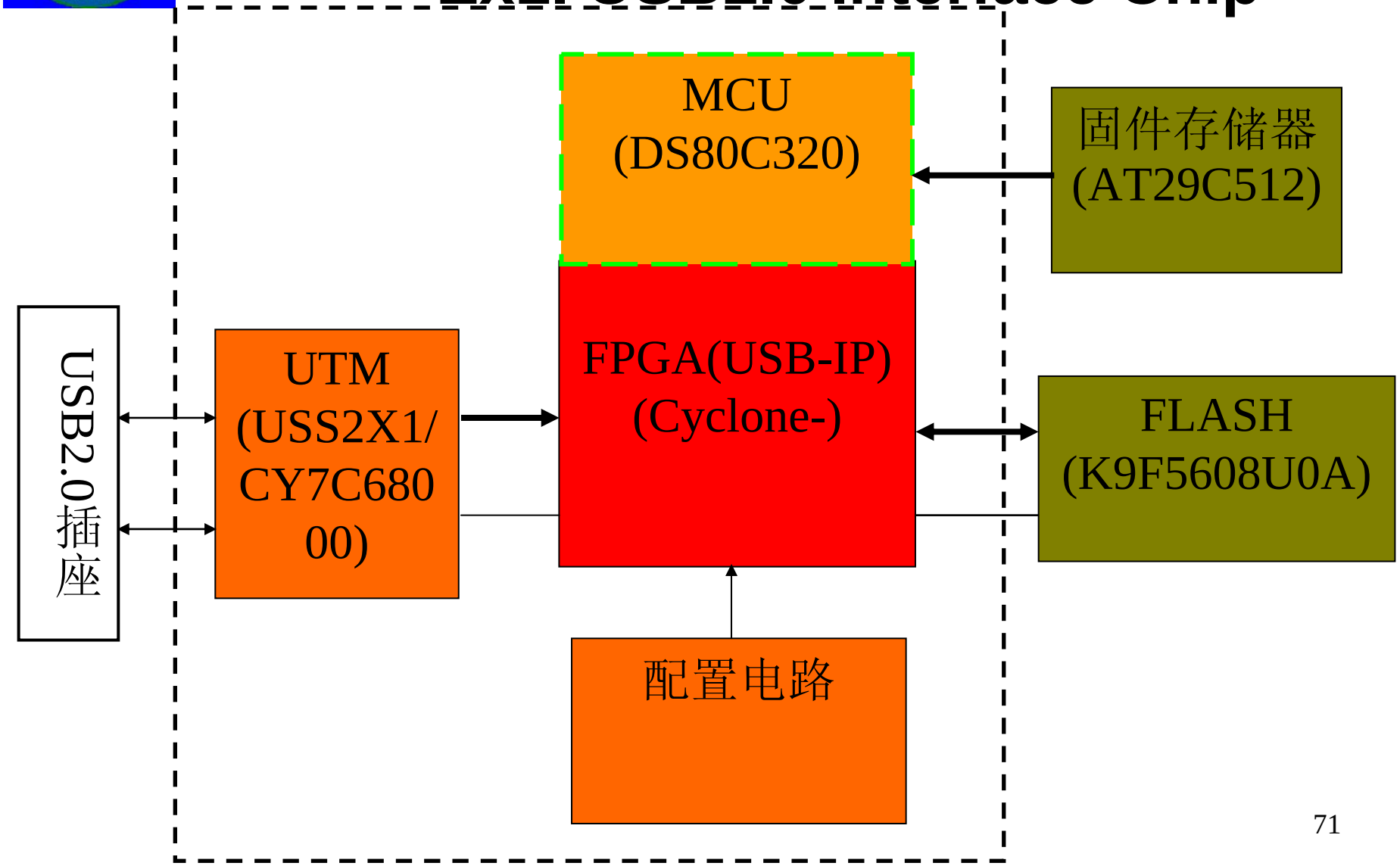
In-System Programmability (ISP)

- ISP Support in EPC2 and Enhanced Configuration Devices
  - Through Jam Standard Test & Programming Language (STAPL)
  - Compliant with IEEE Std. 1532 Draft 2.0 Specification
  - Industry-Standard 4-Pin JTAG Interface
  - Supports JTAG Boundary Scan
- Increases Design Flexibility
  - Update Design In-System
- Streamlines Manufacturing
  - Reduce Costs



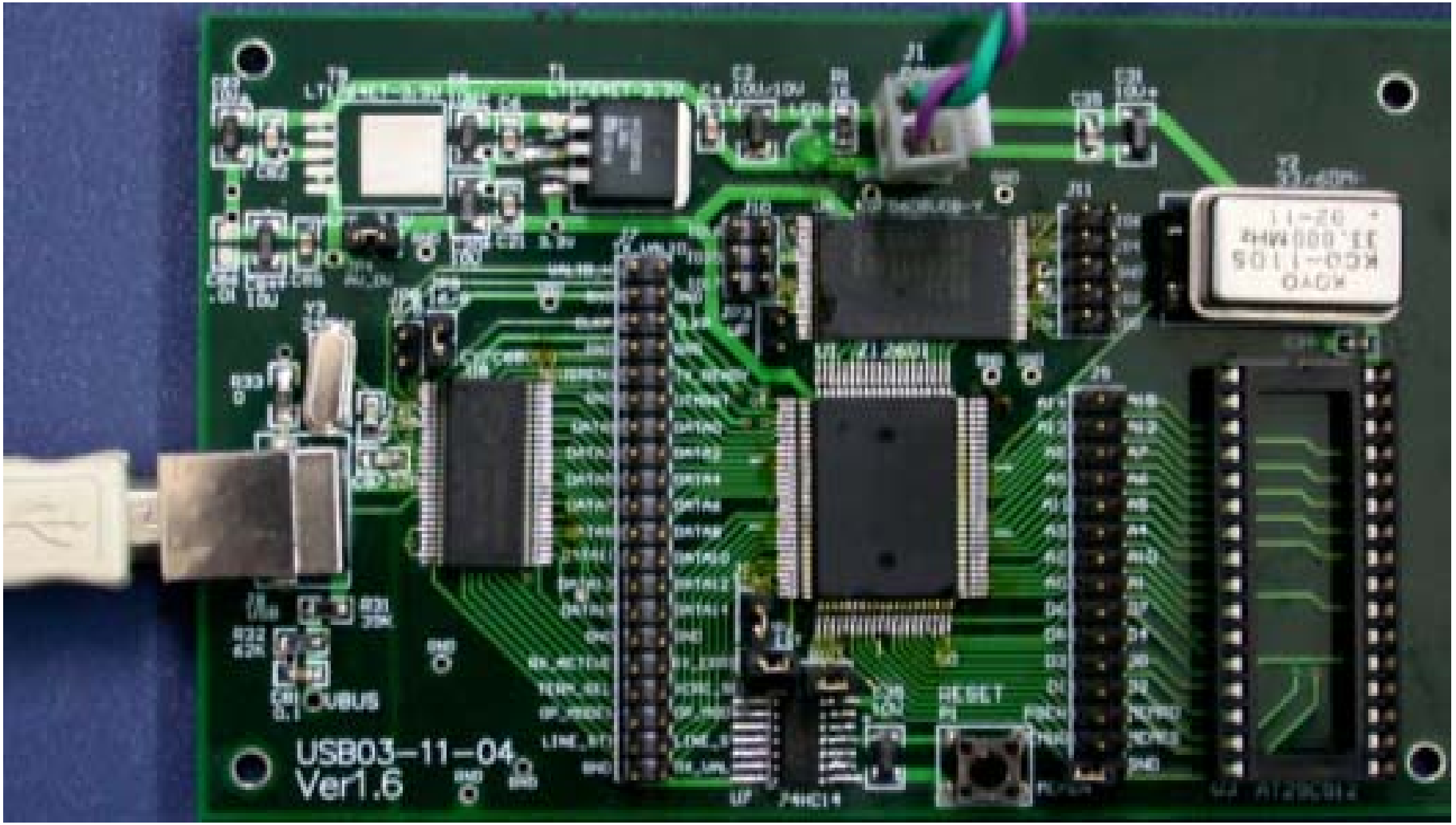
6. Design examples

Ex1. USB2.0 Interface Chip

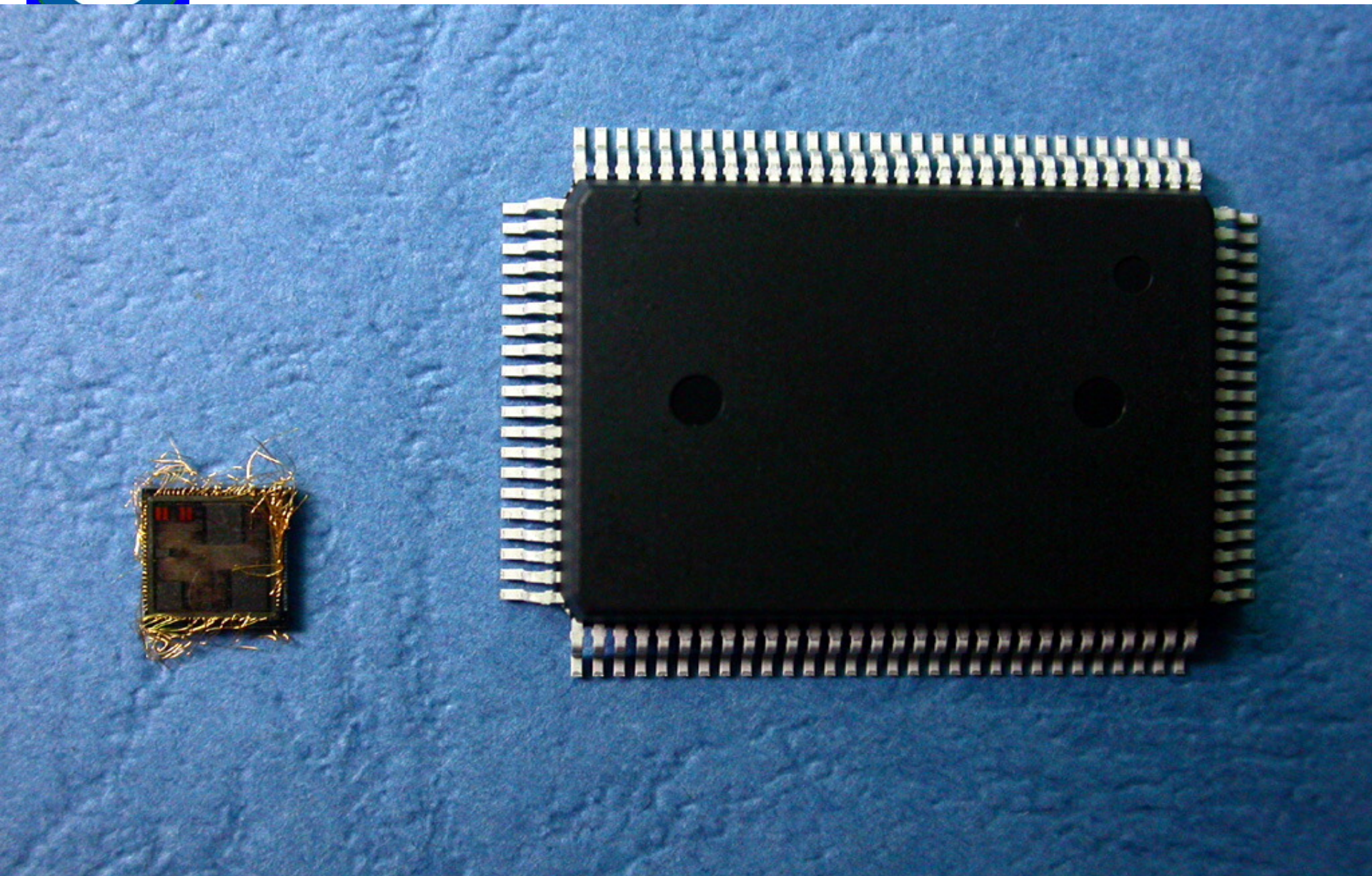


USB2.0 Interface Chip

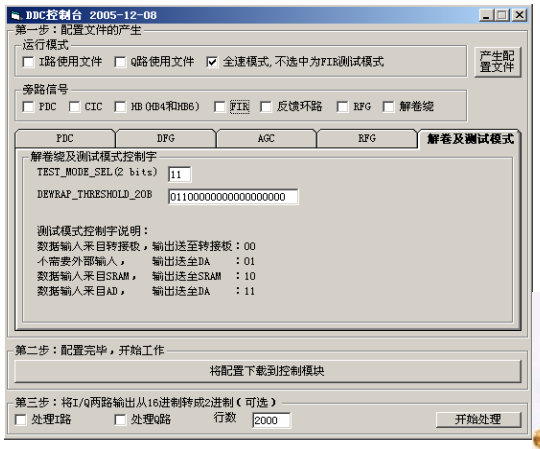
USB2.0 Interface Chip(FPGA)



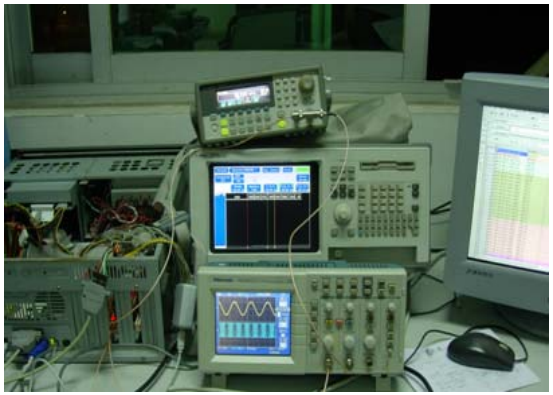
USB2.0 Interface Chip(ASIC)



Ex2. Digital Down Converter(DDC)

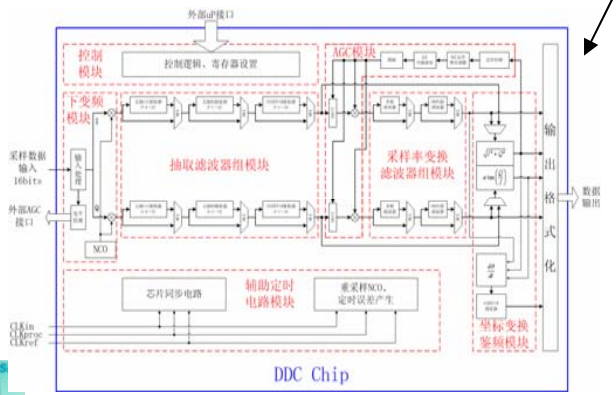


Simple PCI platform

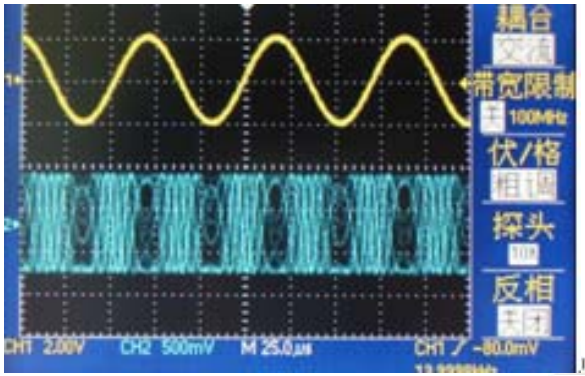


Integrated Verification Platform

Clear Structure Diagram



Fine Experiment Waveform



Die , Chip & Test Board of DDC