# Introduction to FPGAs

H. Krüger

Bonn University

# Outline

1. History
2. FPGA Architecture
3. Current Trends
4. Design Methodology (short → see other lectures)

Disclaimer:

Most of the resources used for this lecture refer to *Xilinx* FPGAs. Other vendors (ALTERA, Lattice, Actel etc.) have similar function blocks – although with different names. Device specific drawings are taken from XILINX datasheets (© XILINX)

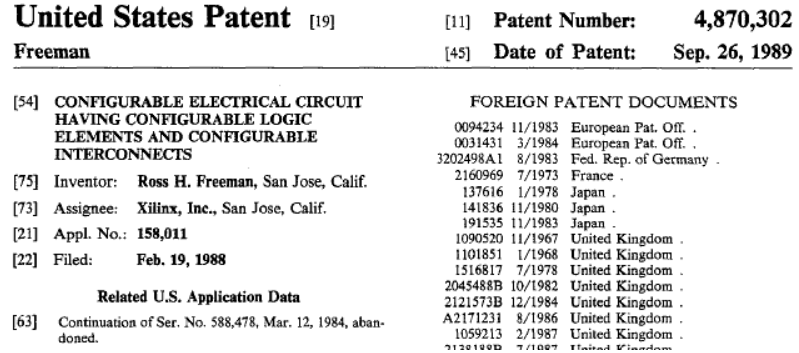
# History

- FPGA principle invented by Ross Freeman, co-founder of XILINX
- First commercial products in 1985 (XILINX XC2000 family)

Motivation

- Advancements in IC technology allowed to implement more complex digital circuits
  - but: IC manufacturing was (and still is) much more expensive than PCB design
- Therefore digital systems still consisted of a huge number of discrete logic gates
  - large **area** PCBs, high **power** consumption, low **speed**…

➔ Solve the application specific connection of logic functions to make use of the IC technology scaling for complex digital systems



**United States Patent** [19]
**Freeman**

[11] **Patent Number: 4,870,302**
[45] **Date of Patent: Sep. 26, 1989**

[54] **CONFIGURABLE ELECTRICAL CIRCUIT HAVING CONFIGURABLE LOGIC ELEMENTS AND CONFIGURABLE INTERCONNECTS**

[75] Inventor: **Ross H. Freeman**, San Jose, Calif.
[73] Assignee: **Xilinx, Inc.**, San Jose, Calif.
[21] Appl. No.: **158,011**
[22] Filed: **Feb. 19, 1988**

**Related U.S. Application Data**

[63] Continuation of Ser. No. 588,478, Mar. 12, 1984, abandoned.

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| 0094234 | 11/1983 | European Pat. Off. . |
| 0031431 | 3/1984 | European Pat. Off. . |
| 3202498A1 | 8/1983 | Fed. Rep. of Germany . |
| 2160969 | 7/1973 | France . |
| 137616 | 1/1978 | Japan . |
| 141836 | 11/1980 | Japan . |
| 191535 | 11/1983 | Japan . |
| 1090520 | 11/1967 | United Kingdom . |
| 1101851 | 1/1968 | United Kingdom . |
| 1516817 | 7/1978 | United Kingdom . |
| 2045488B | 10/1982 | United Kingdom . |
| 2121573B | 12/1984 | United Kingdom . |
| A2171231 | 8/1986 | United Kingdom . |
| 1059213 | 2/1987 | United Kingdom . |

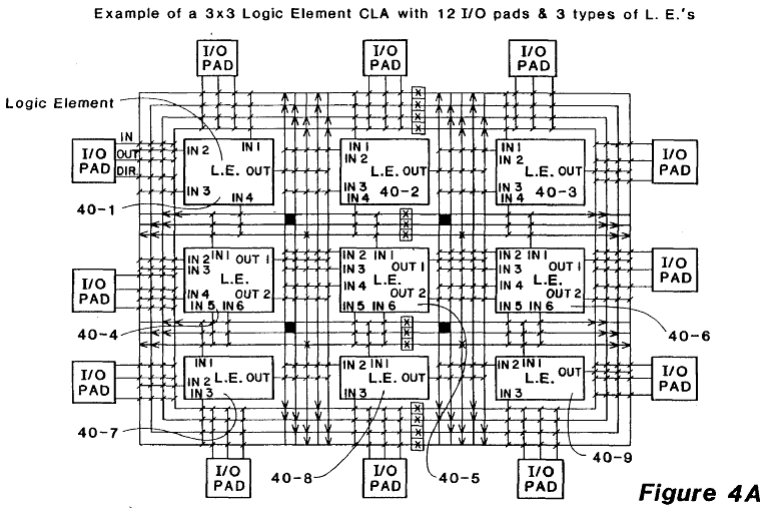
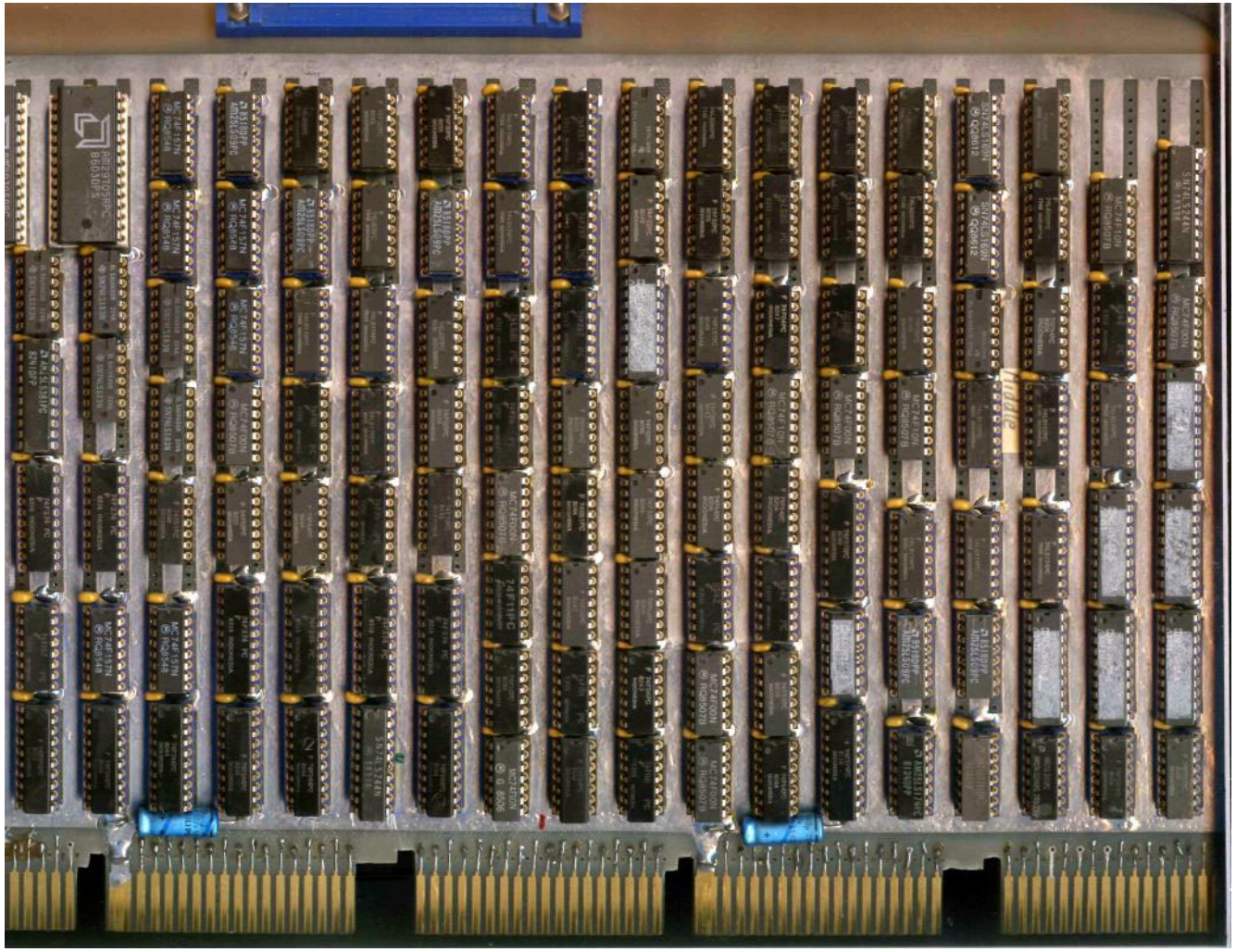


Figure 4A

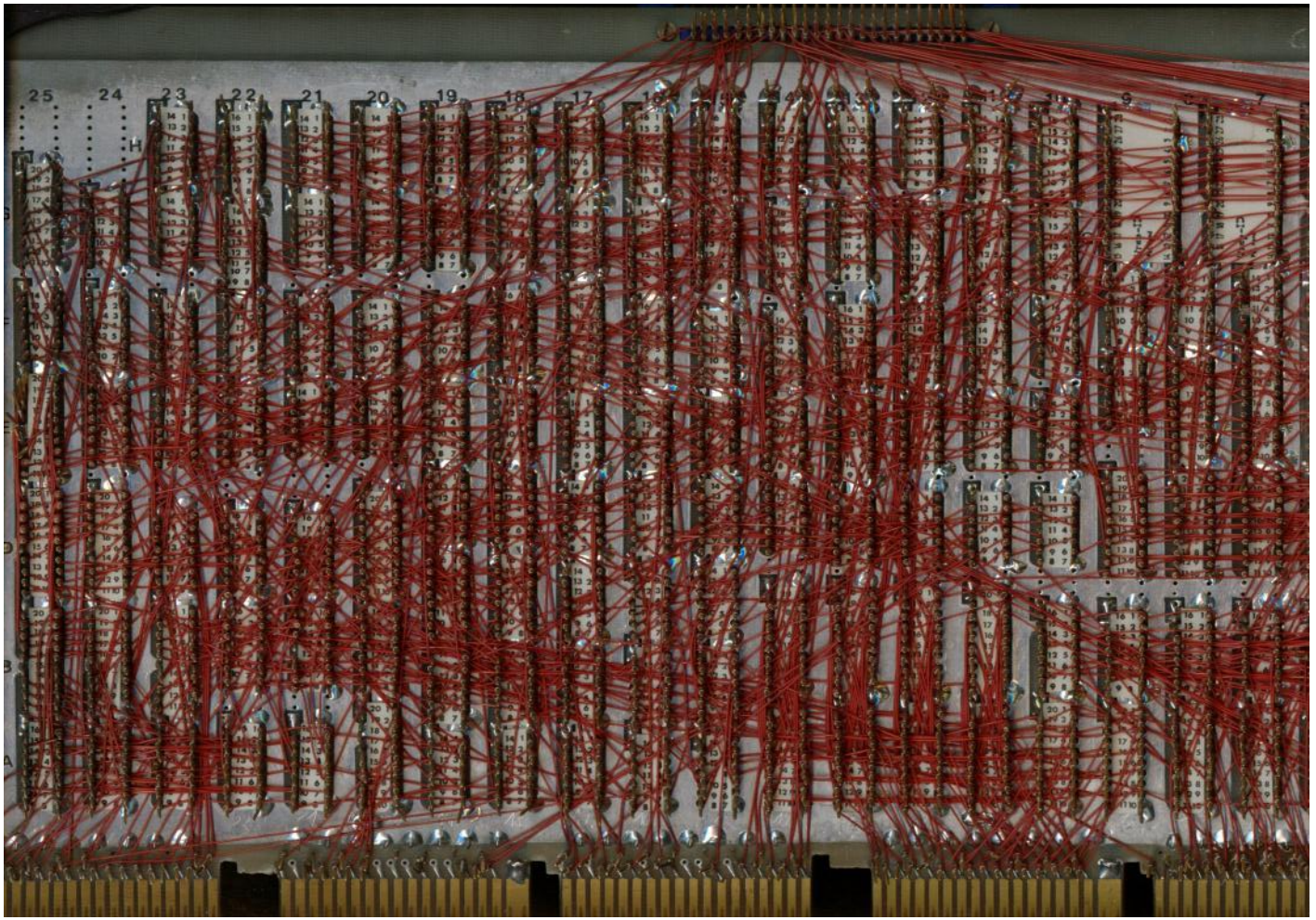
# Digital Logic ‘back in the days’

Placement of TTL ICs (~2-8 gates per component) → very low logic density



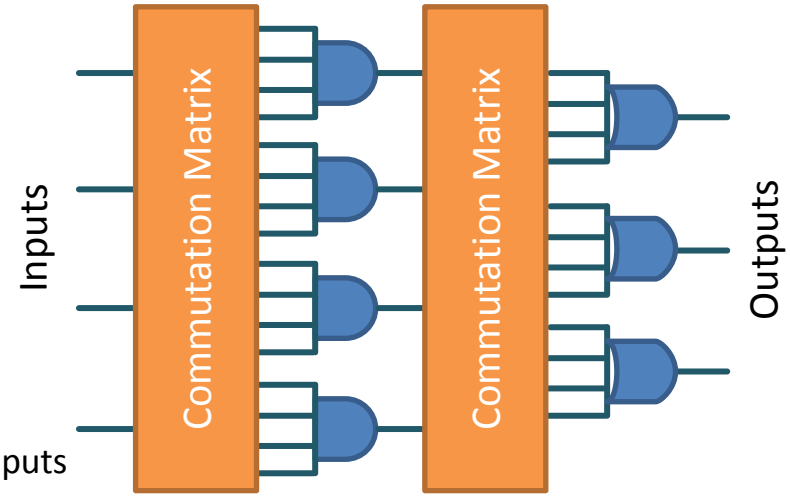
# Digital Logic ‘back in the days’

But routing those gates was even worse…



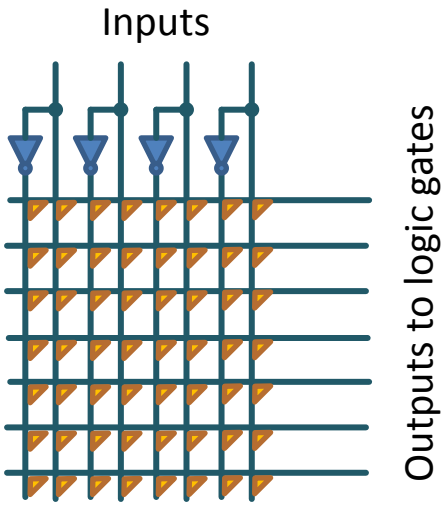
# ‘Early’ Programmable Logic

- Programmable Logic Array (**PLA**)
  - Early 70’s
  - Can implement any Boolean function
  - Fixed logic gates + Programmable interconnect
  - Interconnect via Commutation Matrix
    - Inverted and non inverted inputs
    - Outputs programmed to connect to any one of the inputs
  - One-time-programmable (**OTP**)
    - Fuse: all points connected with wires
      ➔ remove not wanted connection by fuse burning
    - Anti-Fuse: all connected with diodes in reverse bias
      ➔ establish connection by diode (non-reversible) breakdown



PLA Functional Diagram

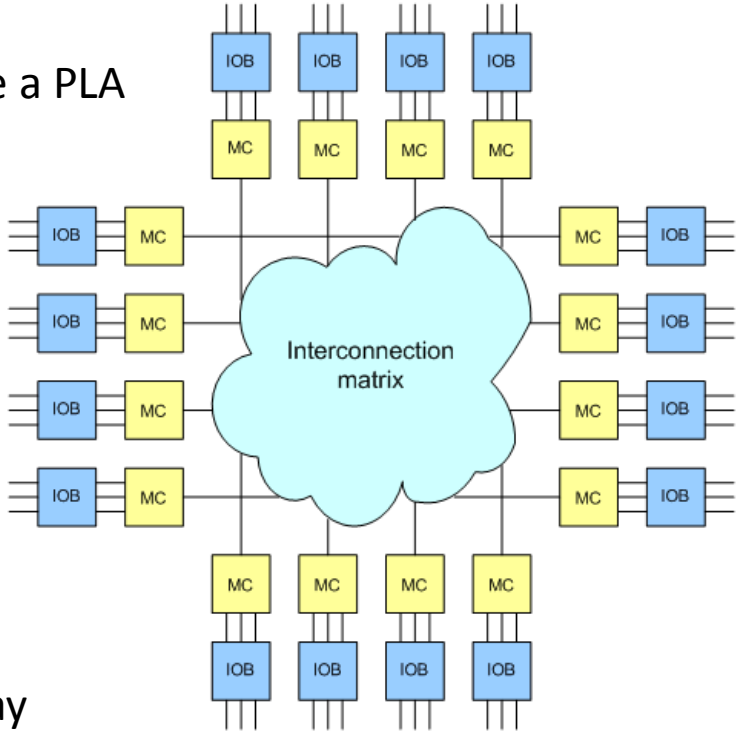
- Variant: Programmable Array Logic (**PAL**)
  - Hardwired second commutation → faster



Commutation Matrix

# ‘Early’ Programmable Logic

- Complex Programmable Logic Device (**CPLD**)
  - many macro cells (MC) each functioning like a PLA
  - configurable IO blocks (IOB)
  - Interconnect matrix
  - re-programmable (Flash memory)



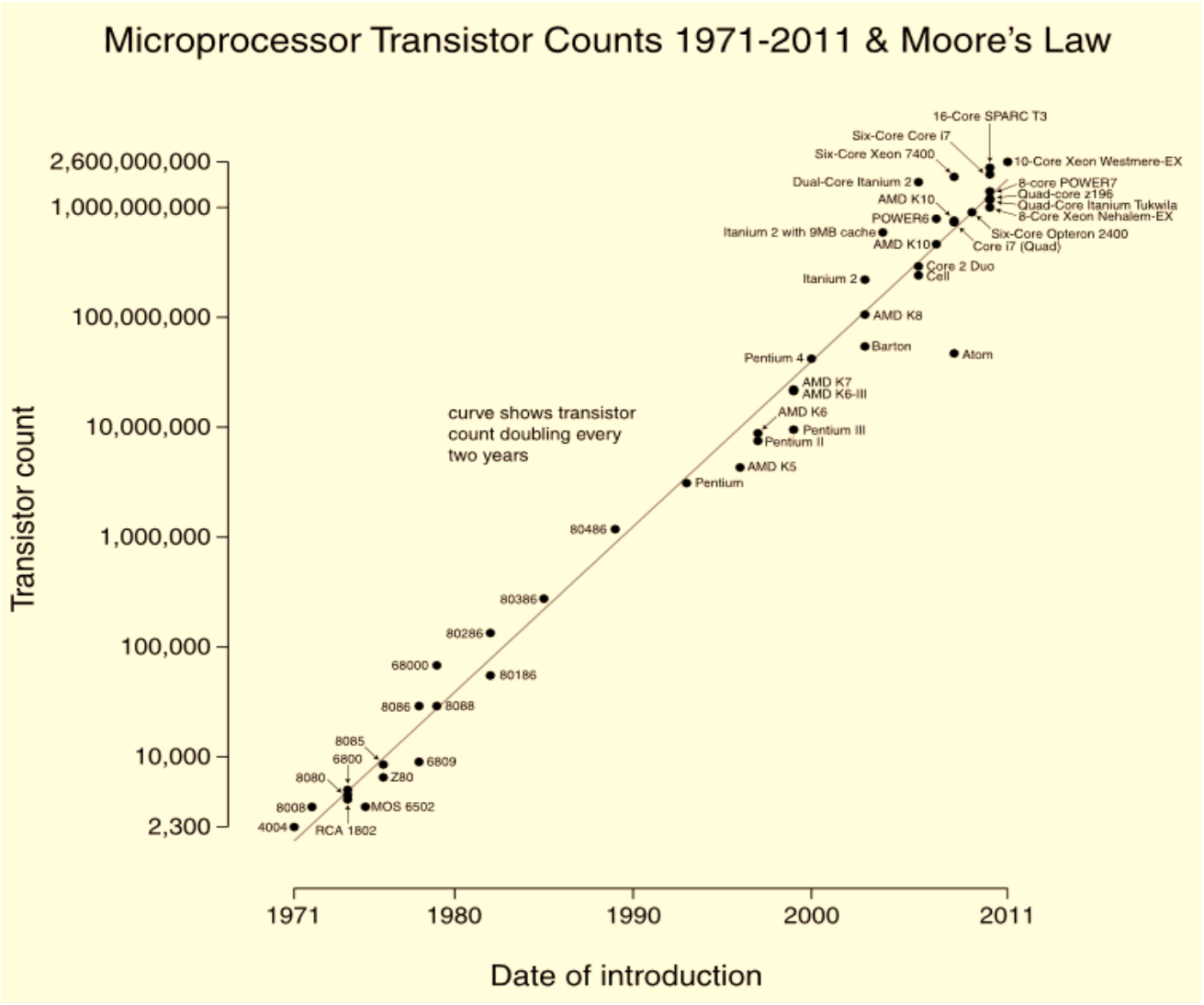

Not as complex as modern FPGAs but still used today

- low pad-to-pad delay, predictable
- non-volatile configuration
- cheap

➔ Interface between ASICs and/or peripherals and bus systems (glue logic)

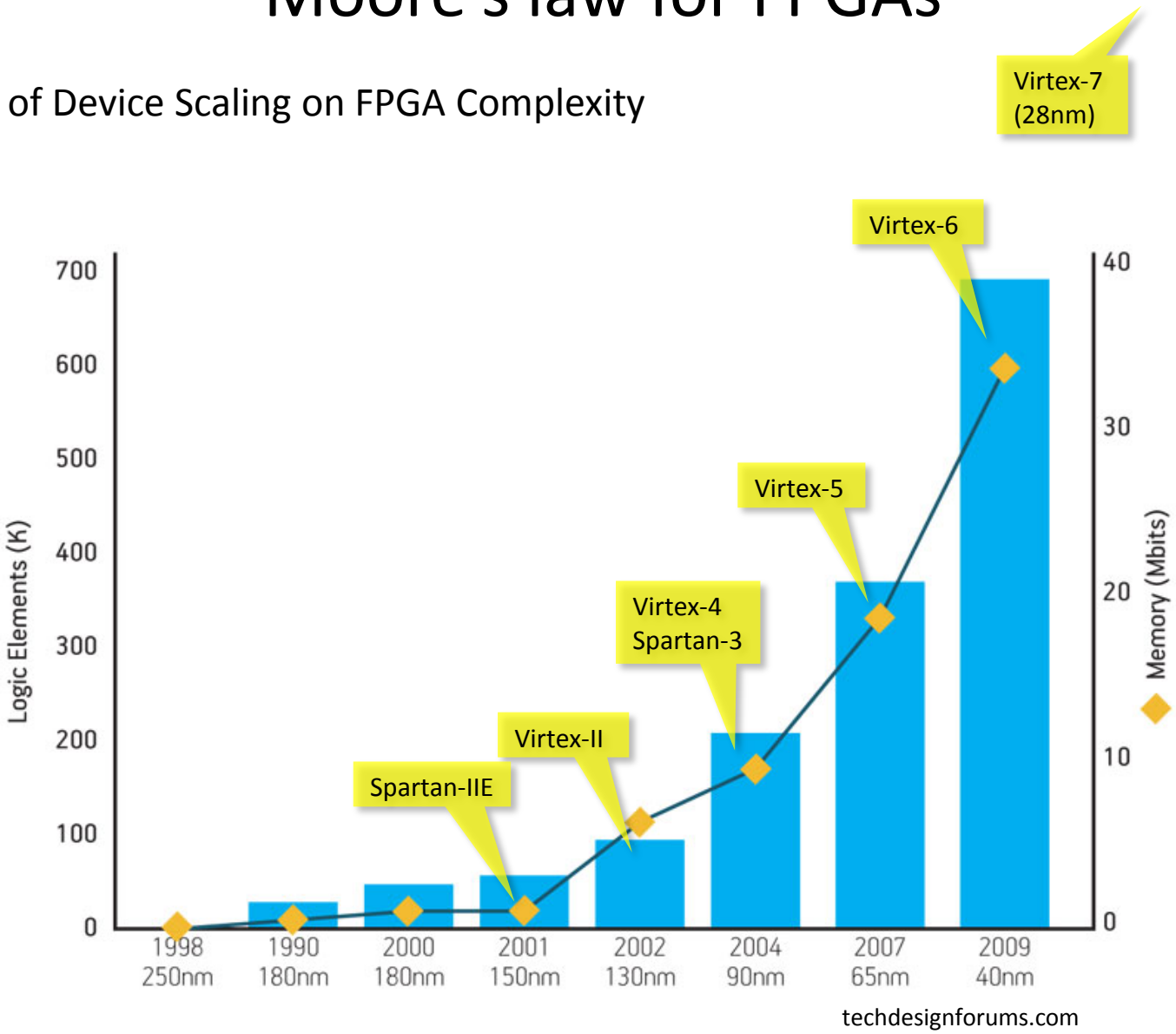
# IC Technology Development

- Doubling of gate density every 18 moths



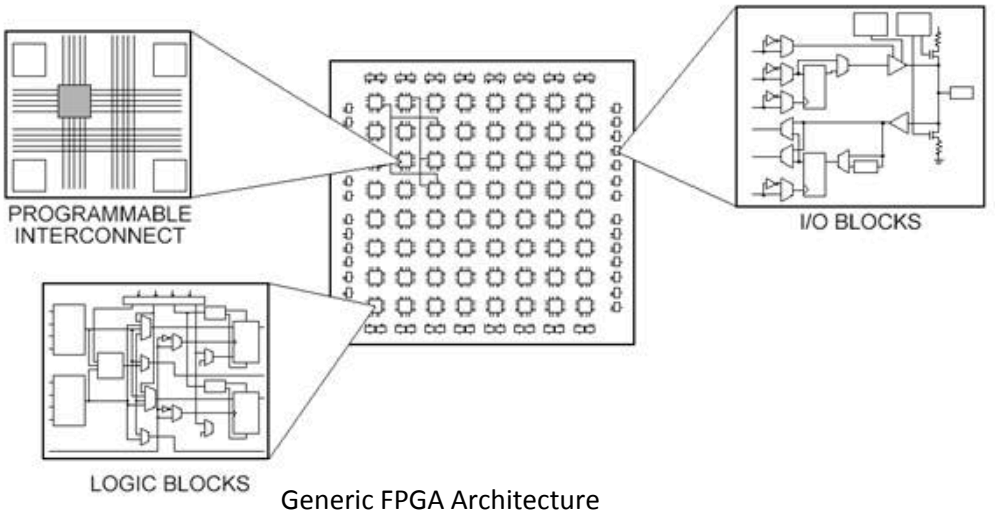
# Moore's law for FPGAs

- Impact of Device Scaling on FPGA Complexity



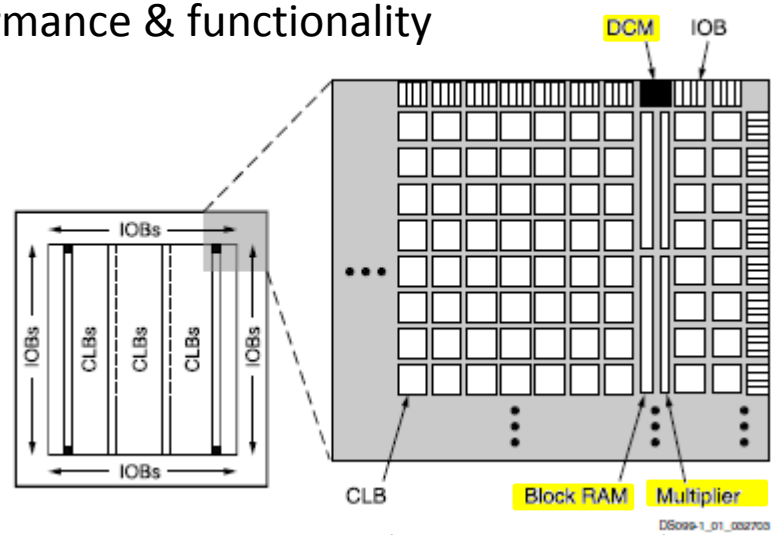
# FPGA Architecture

- Standard Ingredients
  - IO blocks (**IOB**)
  - Configurable Logic Blocks (**CLB**)
  - Programmable Interconnect



Generic FPGA Architecture

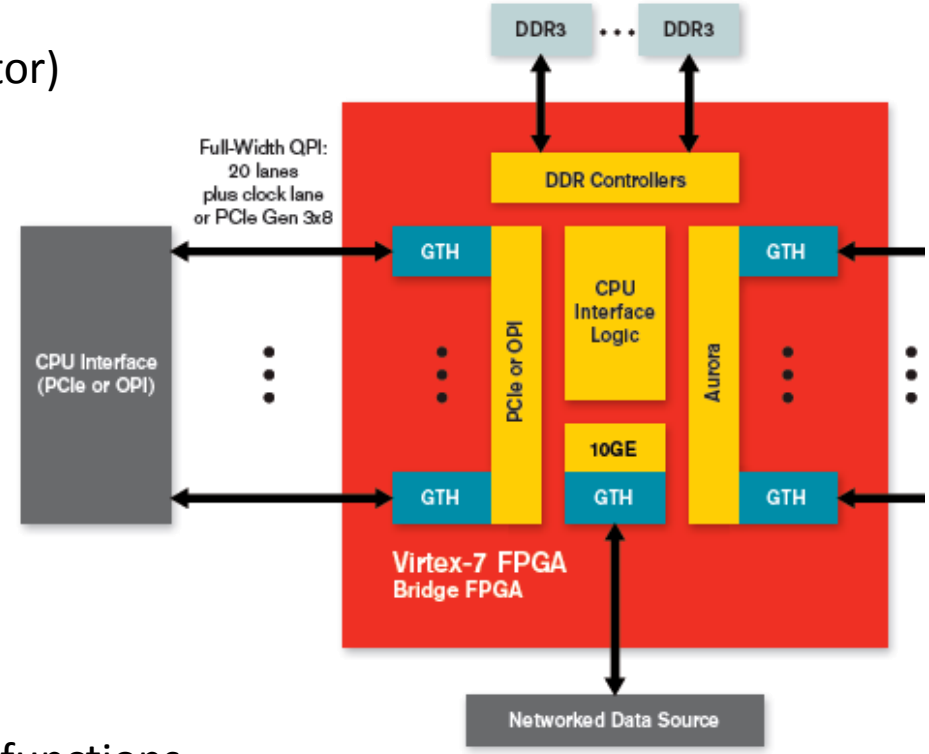
- Special Function Blocks → enhanced performance & functionality
  - Block Memories (**BRAM**)
  - Digital Clock Management (**DCM**)
  - Dedicated Multiplier
  - More IO standards (**SelectIO**)



Spartan-3 Architecture (2003, Xilinx Datasheet)

# Advanced FPGA Architecture

- (even more) Special Function Blocks
  - DSP slices (multiplier/adder/accumulator)
  - Multi-Gigabit Transceiver (GTP, GTH…)
  - PCIe interfaces
  - Ethernet MAC
  - Dedicated DDR memory controller
  - Processor (hard) cores: PowerPC, ARM
  - ADCs
  - Partial reconfiguration
  - …

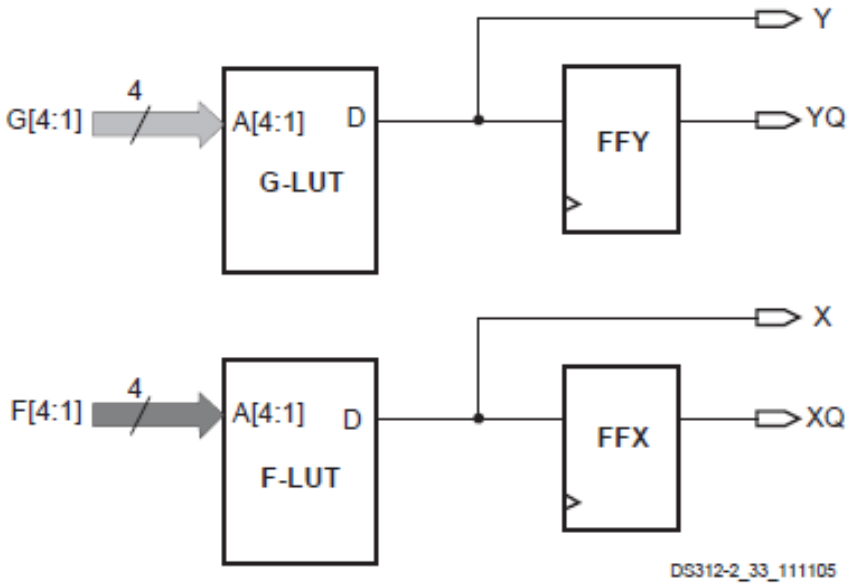



- More efficient implementation of complex functions
  - Area, speed, power consumption
  - Fixed function but still quite a lot of configuration options

# Configurable Logic Block (CLB)

- The **CLB** is the main resources to implement any synchronous or asynchronous logic functions. It consists of
  - Look-up table (**LUT**)
  - Storage element: flip-flop (**FF**)

- The **LUT** is a 16x1 bit memory, used as
  - 4-input logic gate (truth table)
  - ROM
  - synchronous RAM (**Distributed RAM**)



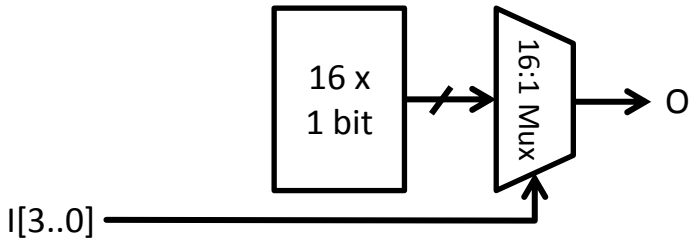

Simplified CLB / SLICE Block Diagram (Xilinx Datasheet)

- Used as a SRAM the **LUT** is fast an convenient to implement small FIFOs or register files (distributed RAM).
- Cascading of **LUT** to have wider inputs or larger LUT: 5, 6 inputs

# Look-up Table (LUT)

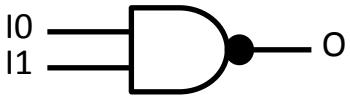
Resources need to implement a **LUT**

- 16 x 1 Bit memory
- 16:1 Multiplexer




LUT Functional Block Diagram

Example: 2-input NAND

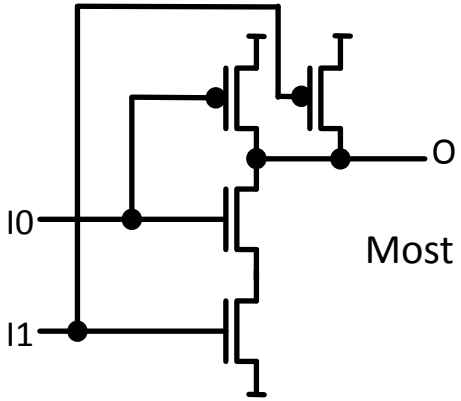



FPGA

| I1 | I0 | O |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Flexible but has some drawbacks
- large number of transistors
→ area, speed, power

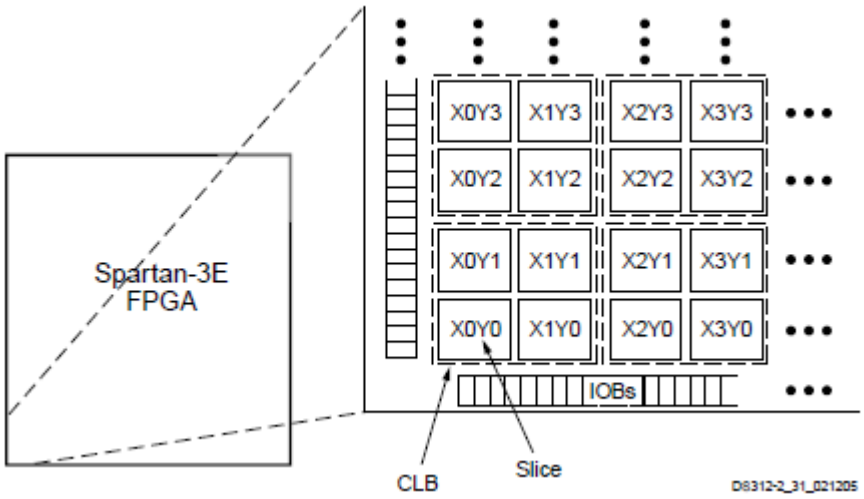
ASIC




Most efficient but fixed function

# Configurable Logic Block (CLB)

- Implementation
  - Two **LUT** and two **FF** make up one **SLICE**
  - Four **SLICEs** are combined within one **CLB**




Arrangement of SLICEs within a CLB (Spartan-3, Xilinx Datasheet)

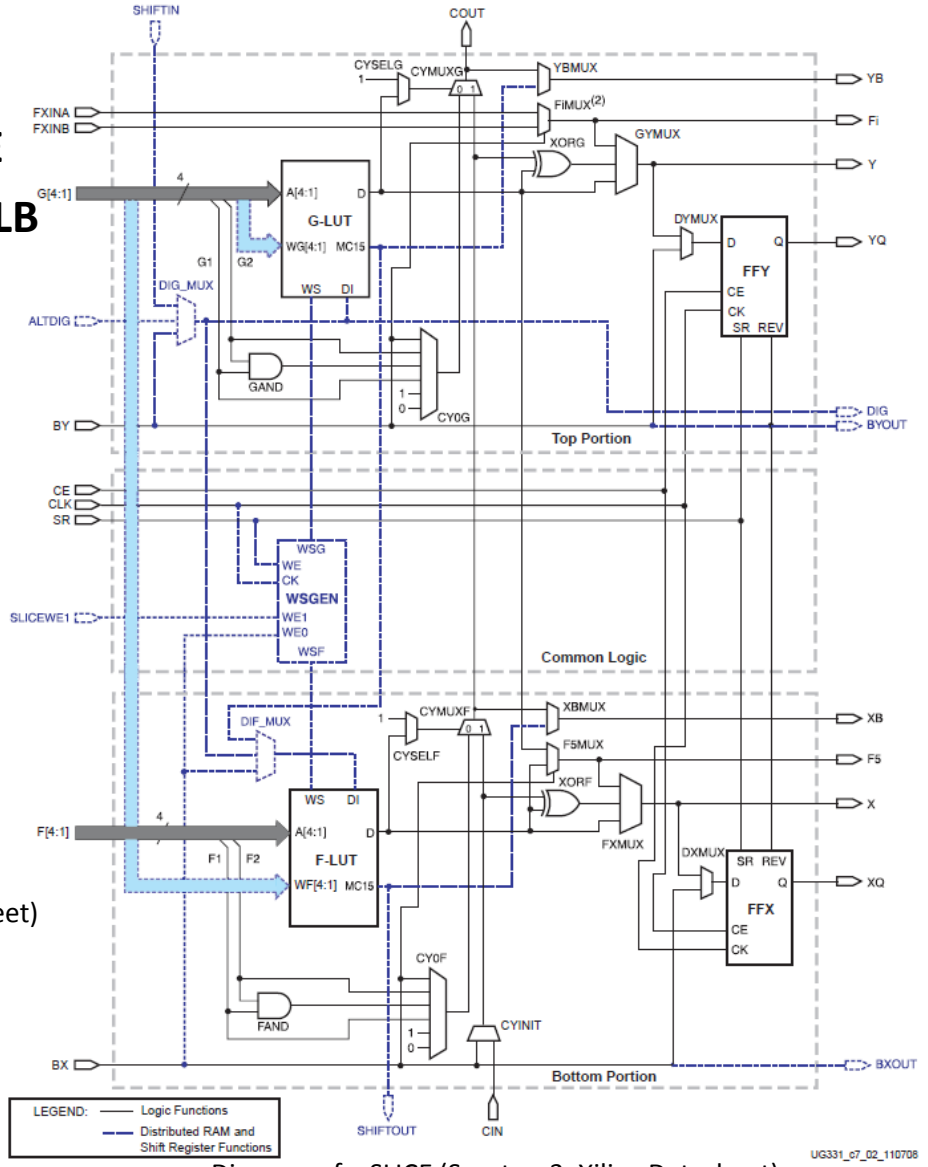
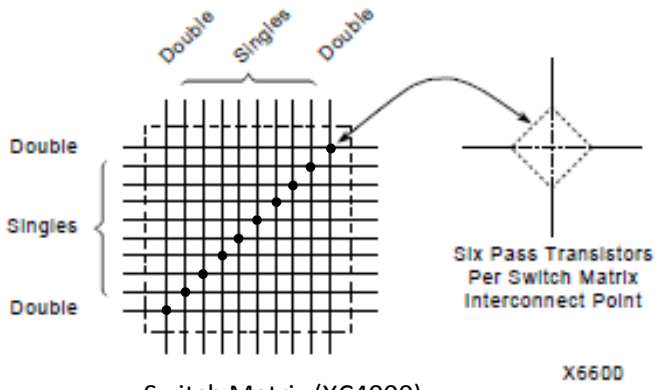


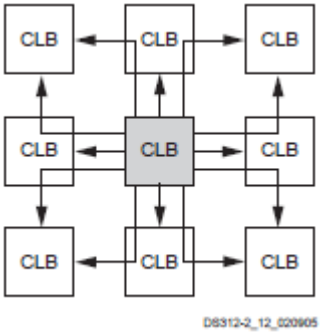

Diagram of a SLICE (Spartan-3, Xilinx Datasheet)

# Routing Resources

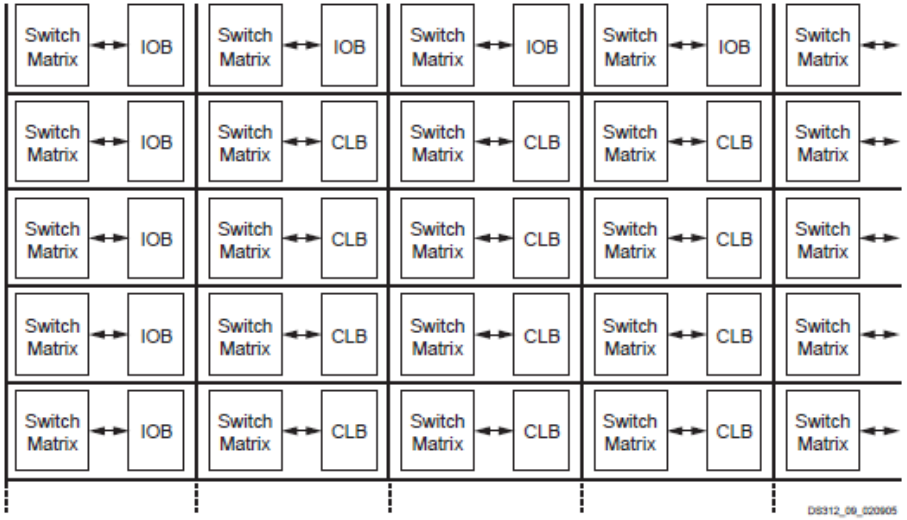
- **Switch Matrices**
  - Switch Matrix (Cross Point Switch)
  - Programmable Interconnect Point (PIP)
- **Wires**
  - Local Interconnect
  - Global interconnect (long lines)




Switch Matrix (XC4000)



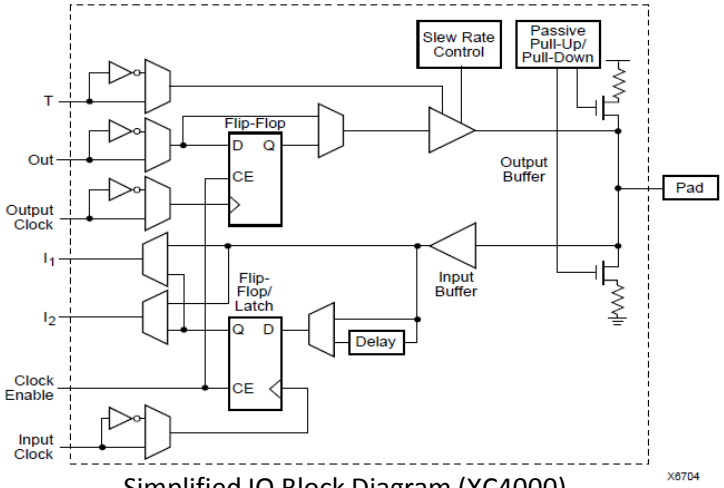

Local connect (Spartan-3)



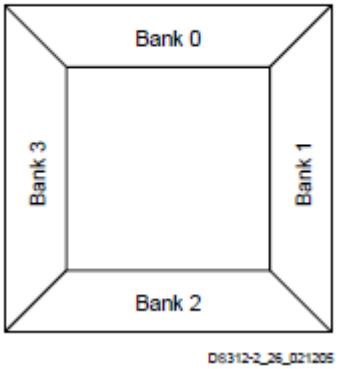

Global Interconnect Tile (Spartan-3)

# Input/Output Blocks

- Standard **I/O** configuration options
  - In, out or bi-directional
  - Sync. or async.
- Logic standard and levels
  - LVTTL, LVCMOS
  - Selectable drive current
  - Logic levels controlled by IO supply voltage
- I/O blocks grouped together in **banks**
  - Independent power supply connections
  - typ. 1.2V – 3.3V



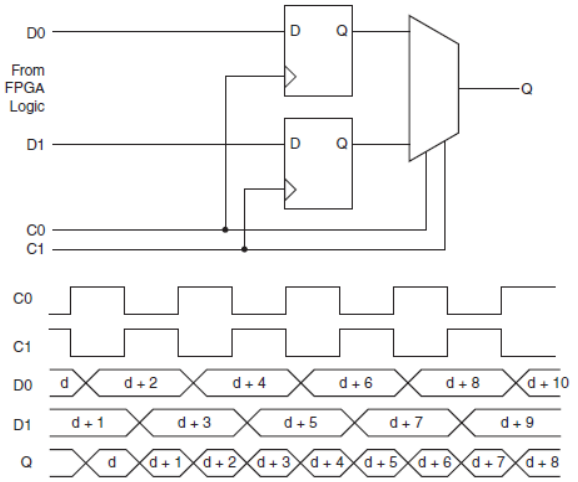

Simplified IO Block Diagram (XC4000)



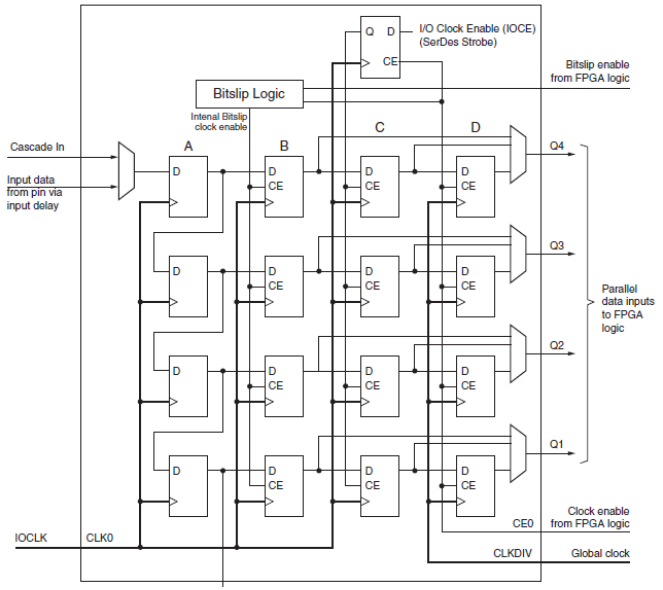

IO Banks

# Input/Output Blocks

- More advanced features
  - High speed single-ended: GTL, HSTL, SSTL …
  - Differential standards: LVDS, BLVDS, LVPECL …
  - On-chip termination, controlled output impedance (DCI)
  - Double data rate (DDR) registers
  - Programmable delay lines (IDELAY, ODELAY)
  - Serializer / Deserializer (ISERDES, OSERDES)



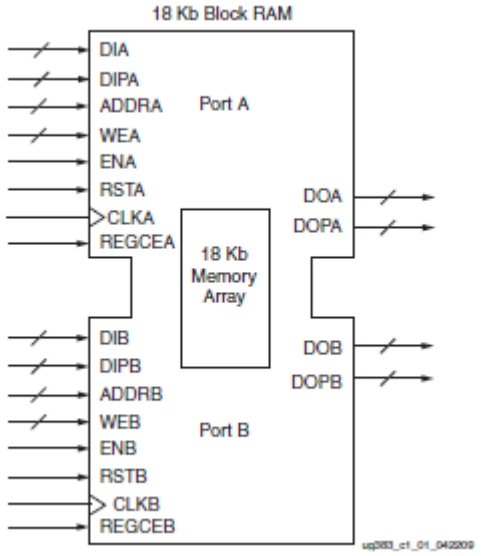

DDR Output Block (Xilinx Datasheet)




Input Serializer (ISERDES, Xilinx Datasheet)

# Block Memory

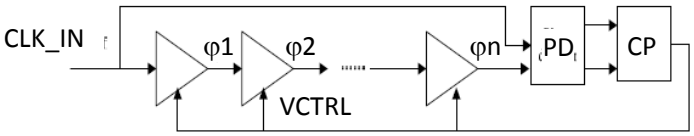
- Block RAM (**BRAM**)
  - Optimized SRAM memory block for medium size data storage
  - Typical 9 Kb blocks, configurable for different port widths
    - 8K x 1, 4K x 2… 256 x 32 (w/o parity)
    - 1K x 9, 512 x 18, 256 x36 (with parity)
  - Optional Dual-port (independent read / write)
    - Dual port supports **different port width** for A and B port.
    - Implementation of **FIFOs**
  - Larger memories by combining multiple **BRAM** blocks



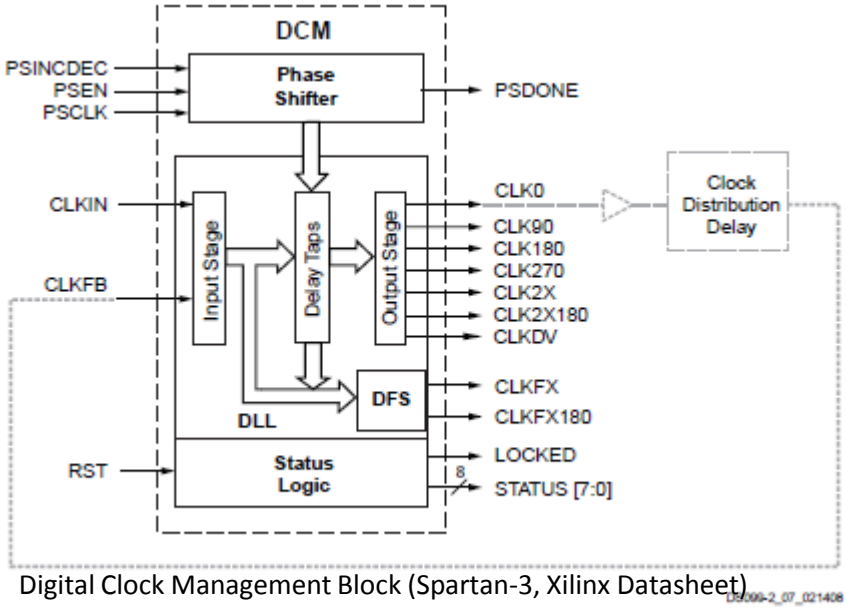

Dual Port RAM (Spartan-6, Xilinx Datasheet)

# Digital Clock Management (**DCM**)

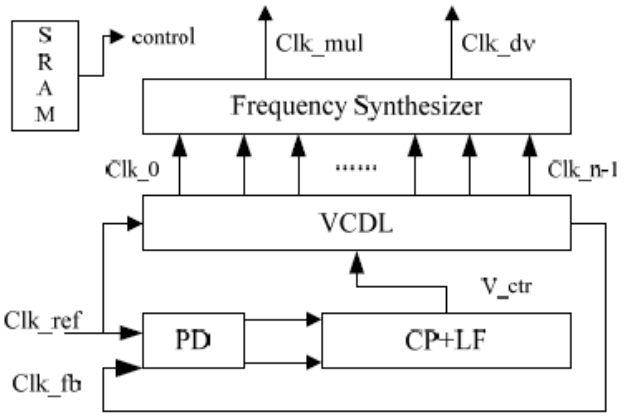
- Delay Locked Loop (**DLL**)
  - Voltage controlled delay line (VCDL)
  - Multiple output phases
- Control of phase delay
  - zero delay clock buffer
- Frequency synthesis (**DFS**)
  - Combinatorial function of the phases φn




Voltage Controlled Delay Line



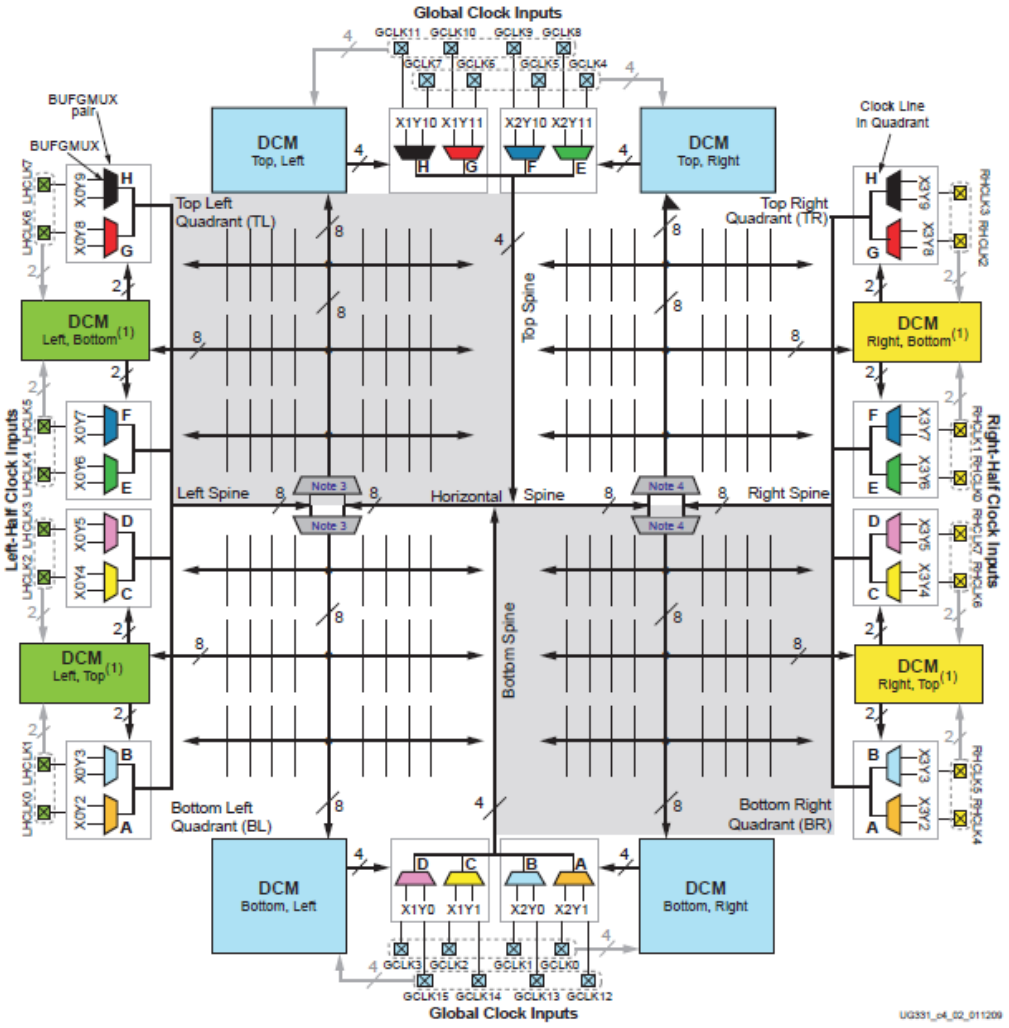

Digital Clock Management Block (Spartan-3, Xilinx Datasheet)




Frequency Synthesis with VCDL

# Clock Generation & Distribution

- Clock **distribution** via dedicated resources
  - Dedicated I/O blocks and buffers
  - Fast, low skew clock nets (clock trees)
  - Individual clock domains




Clock Structure (Spartan-3, Xilinx Datasheet)

# Gigabit Transceiver

- Need to **scale total IO bandwidth** (*number of IO pads x signal frequency)*
  but number of IO pads limited, did not increase at the same scale as the logic density.

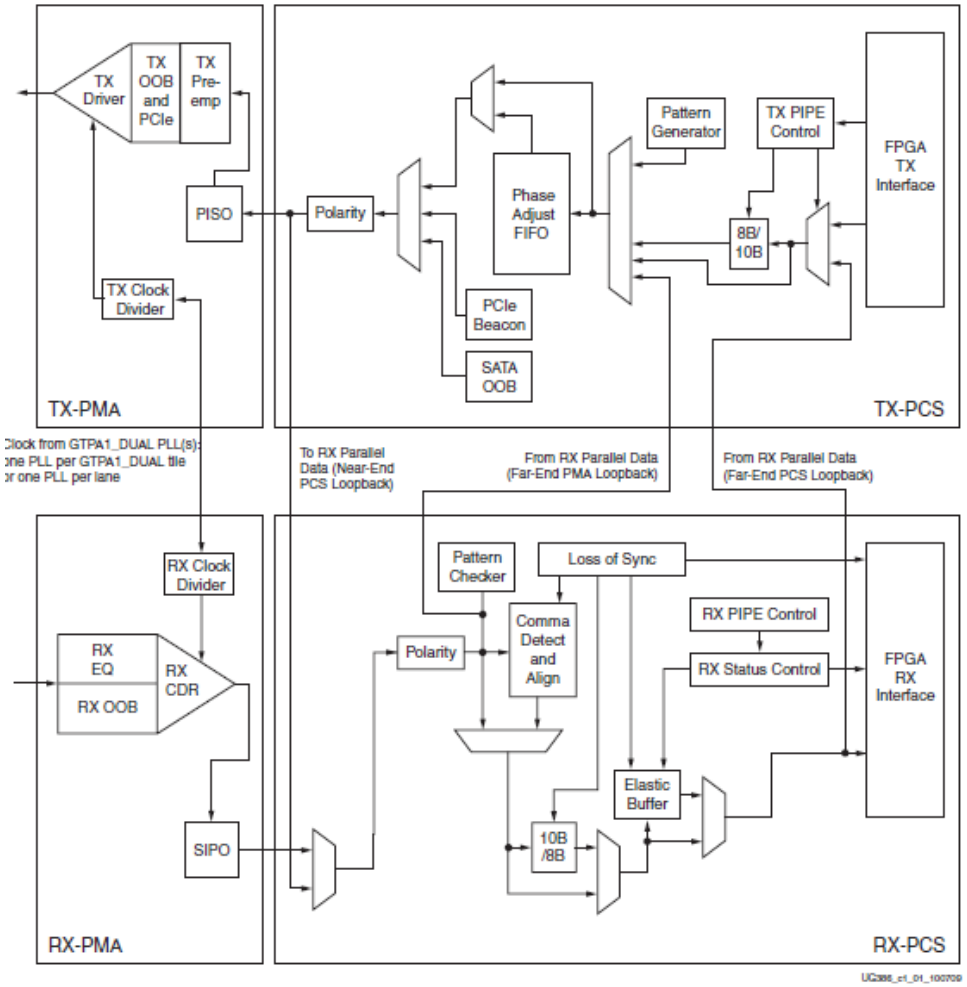
  ➔ **Increase signaling frequency**

- Maximum signal transmission rate (typical)
  - LVCMOS: <100 MHz
  - LVDS: 800 MHz
  - Multi-Gigabit Transceiver (GTP, GTX, GTH): 3.2 Gbps, 11.18 Gbps (up to 28 Gbps!)

- Examples
  - USB 3.0 (SuperSpeed): 5 Gbps
  - PCI Express 2.0: 5.0 Gbps
  - SATA-II: 3.0 Gbps

# Gigabit Transceiver

- Physical layer
  - Differential IO, Current Mode Logic (CML)drivers
  - Signal conditioning (pre-emphasis, equalization)
- Link layer
  - Clock & data recovery (CDR)
  - Synchronization, symbol detection
  - Protocol specific (de)framing
- Automatic configuration to support industrial standard serial protocols: SATA, PCIe, XAUI, 1G Ethernet, DisplayPort …



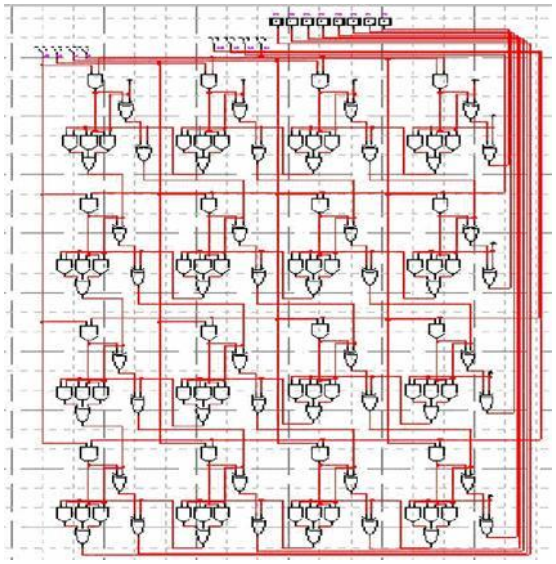

GTP Transceiver Pair Block Diagram (Spartan-6, Xilinx Datasheet)

Drawings © Xilinx

# Multiplier & DSP Slice

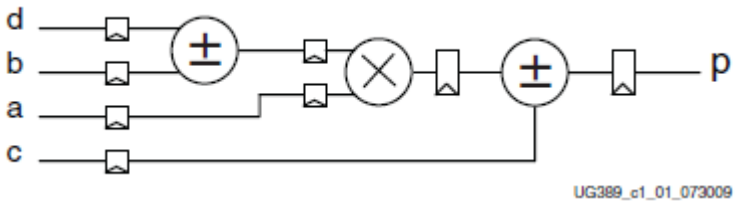
- **Multiplier**
  - 32 bit x 32 bit multiplier would need huge number of LUT/FF

  ➔ Multiplier implemented with optimized digital logic (typical 18 bit x 18 bit)



4 bit x 4 bit Multiplier (generic)

- **DSP Slice**
  - Combination of multiplier/ adder
  - 18 bit inputs
  - Pre-adder
  - Multiplier
  - Post-adder
  - 48 bit outputs



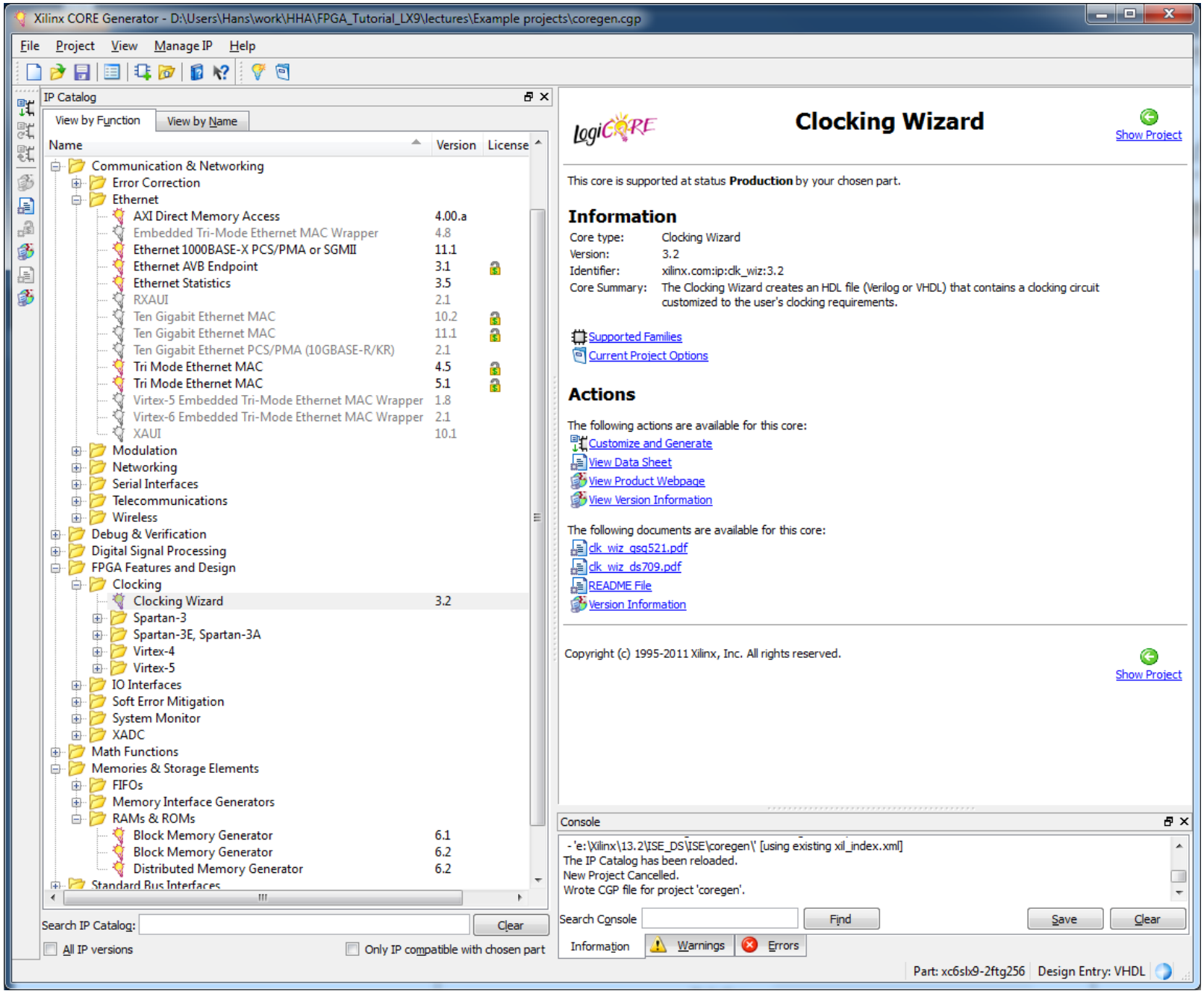

DSP Slice high level Diagram (Spartan-6)

➔ Efficient implementation of digital filters (> 300 MHz clock frequency)

# IP Cores

- Soft IP
  - Functionality implemented by use of existing logic resources (CLBs, BRAMs)
  - pre-placed & routed (hard macro)
  - Examples: **MicoBlaze** (32 bit) or **PicoBlaze** (8 bit) microprocessors
  - pro
    - Configurable functionality
    - Scalable
  - con
    - Efficiency issues (area, speed, power)
- Hard IP
  - Fixed function blocks
  - Examples: **PowerPC** CPU, **BRAM**, **DSP** slices…
  - pro
    - Optimum efficiency
  - con
    - Fixed function
    - Limited scalability

- Soft and Hard IP blocks are configured (generated) with the **CORE Generator** tool except microprocessor cores (Xilinx Platform Studio, **XPS**).

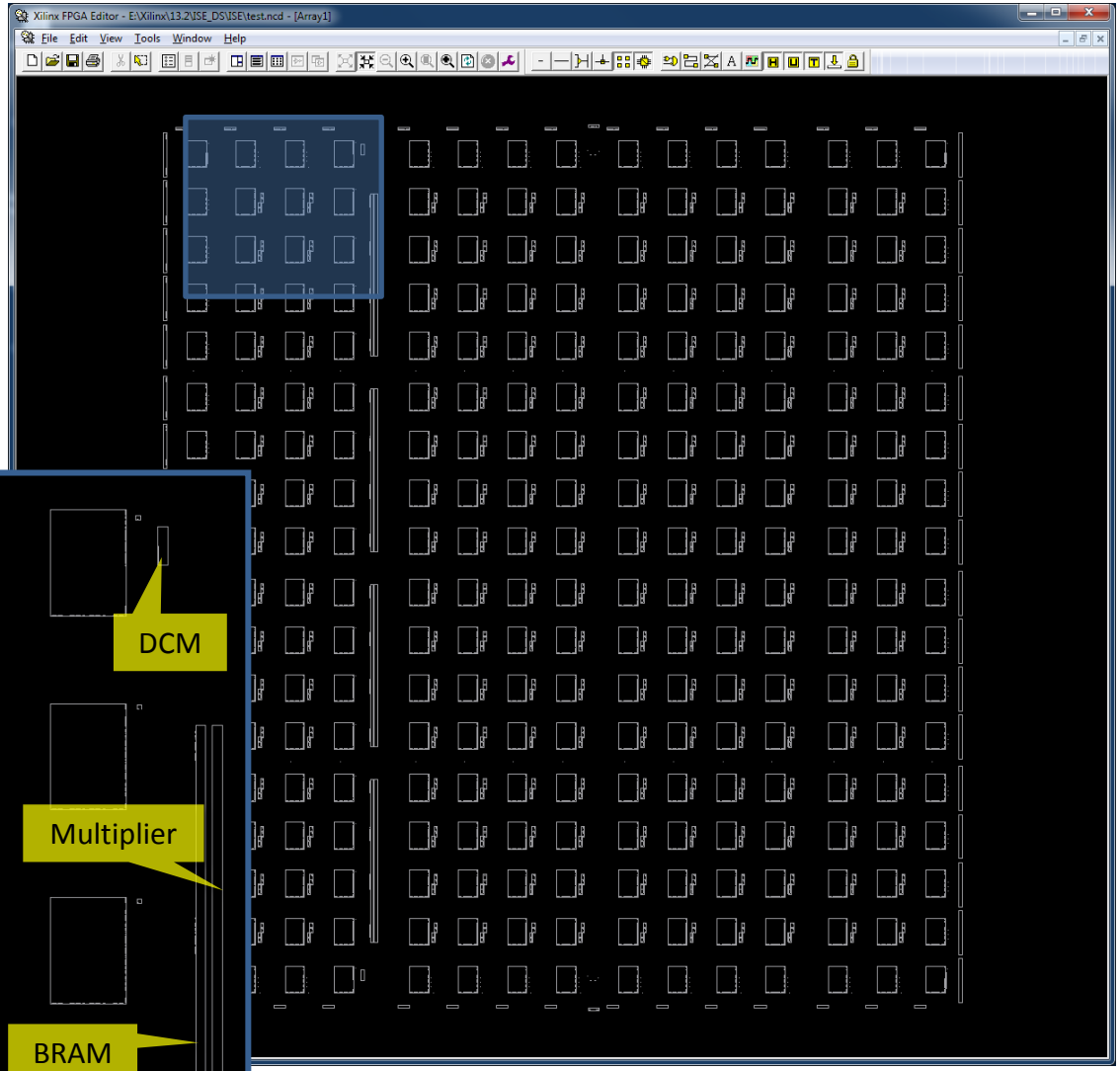

Xilinx CORE Generator - D:\Users\Hans\work\HHA\FPGA_Tutorial_LX9\lectures\Example projects\coregen.cgp
File Project View Manage IP Help
IP Catalog
View by Function View by Name
Name Version License
Communication & Networking
Error Correction
Ethernet
AXI Direct Memory Access 4.00.a
Embedded Tri-Mode Ethernet MAC Wrapper 4.8
Ethernet 1000BASE-X PCS/PMA or SGMII 11.1
Ethernet AVB Endpoint 3.1
Ethernet Statistics 3.5
RXAUI 2.1
Ten Gigabit Ethernet MAC 10.2
Ten Gigabit Ethernet MAC 11.1
Ten Gigabit Ethernet PCS/PMA (10GBASE-R/KR) 2.1
Tri Mode Ethernet MAC 4.5
Tri Mode Ethernet MAC 5.1
Virtex-5 Embedded Tri-Mode Ethernet MAC Wrapper 1.8
Virtex-6 Embedded Tri-Mode Ethernet MAC Wrapper 2.1
XAUI 10.1
Modulation
Networking
Serial Interfaces
Telecommunications
Wireless
Debug & Verification
Digital Signal Processing
FPGA Features and Design
Clocking
Clocking Wizard 3.2
Spartan-3
Spartan-3E, Spartan-3A
Virtex-4
Virtex-5
IO Interfaces
Soft Error Mitigation
System Monitor
XADC
Math Functions
Memories & Storage Elements
FIFOs
Memory Interface Generators
RAMs & ROMs
Block Memory Generator 6.1
Block Memory Generator 6.2
Distributed Memory Generator 6.2
Standard Bus Interfaces
Search IP Catalog: Clear
All IP versions
Only IP compatible with chosen part
LogiCORE
Clocking Wizard
Show Project
This core is supported at status Production by your chosen part.
Information
Core type: Clocking Wizard
Version: 3.2
Identifier: xilinx.com:ip:clk_wiz:3.2
Core Summary: The Clocking Wizard creates an HDL file (Verilog or VHDL) that contains a clocking circuit customized to the user's clocking requirements.
Supported Families
Current Project Options
Actions
The following actions are available for this core:
Customize and Generate
View Data Sheet
View Product Webpage
View Version Information
The following documents are available for this core:
clk_wiz_gsg521.pdf
clk_wiz_ds709.pdf
README File
Version Information
Copyright (c) 1995-2011 Xilinx, Inc. All rights reserved.
Show Project
Console
- 'e:\Xilinx\13.2\ISE_DS\ISE\coregen\' [using existing xil_index.xml]
The IP Catalog has been reloaded.
New Project Cancelled.
Wrote CGP file for project 'coregen'.
Search Console Find Save Clear
Information Warnings Errors
Part: xc6slx9-2ftg256 Design Entry: VHDL

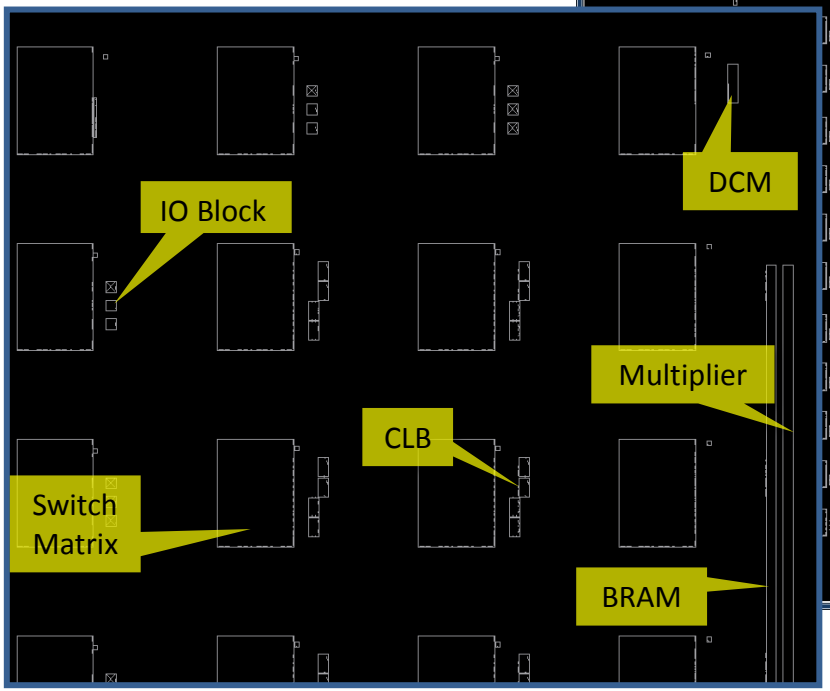
# FPGA Editor

Components (Sites)

- XC3S50 (Spartan-3)
  192 CLBs, 16 x 12 array

State-of-the-art:
XC7V2000 (Virtex-7) 305.400 Slices
just to compare…

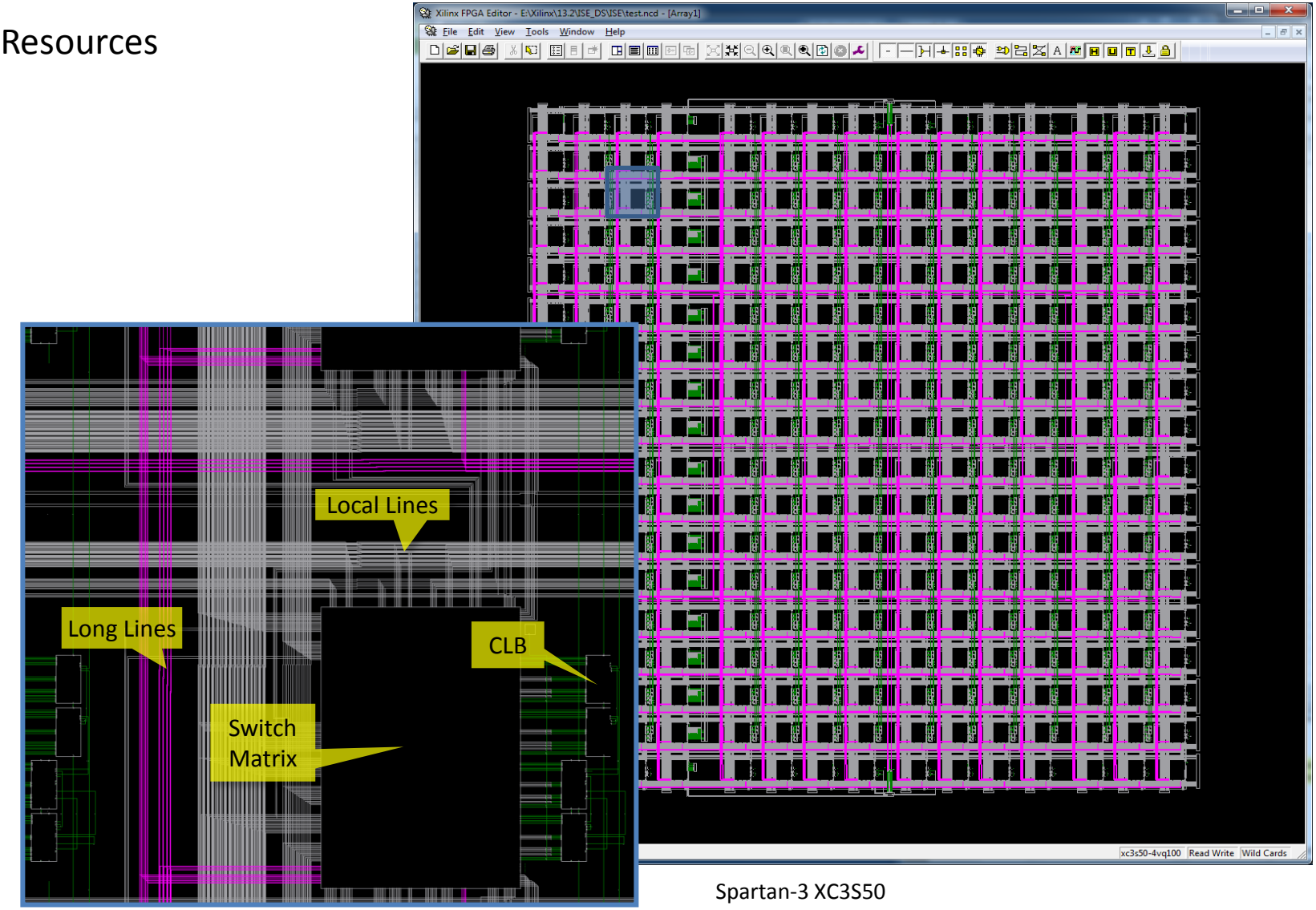



Spartan-3 XC3S50

# FPGA Editor

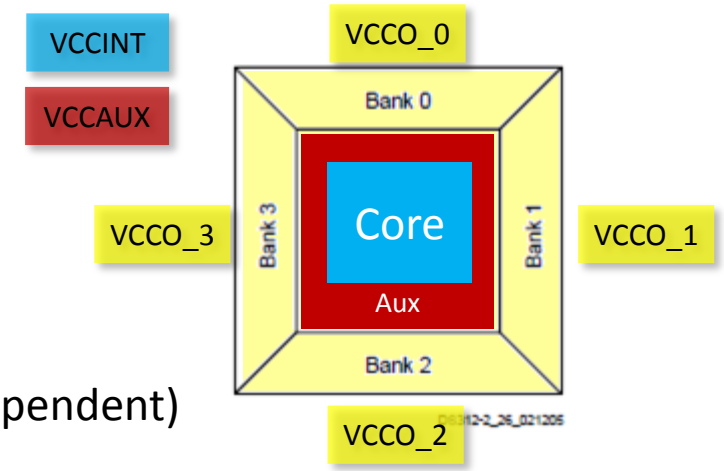
Routing Resources




Spartan-3 XC3S50

# FPGA Power Supplies

- VCCO_x
  - Adjustable 1.2V – 3.3V
  - Independent for every I/O bank




- VCCINT
  - Core supply, fixed: 0.9V .. 1.2V (family dependent)

- VCCAUX
  - typ. 2.5V (1.8V Virtex-7)
  - JTAG I/O
  - Special circuits: IOB pre-drivers, differential input amplifiers, LVDS bias generators, DCM delay lines, internal references

- Dedicated power supplies for Gigabit Transceiver (multiple)

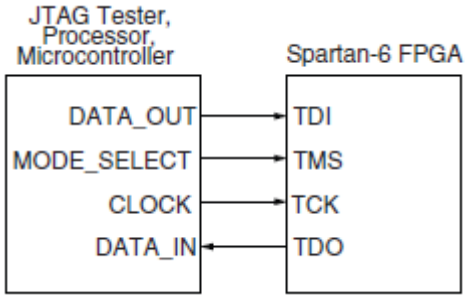
# IO Pin Classes

- General-purpose I/O
  - *IO_L<xy>*
  - Optional differential pair configuration: *IO_L<xy>**p** / IO_L<xy>**n***
  - some with additional capabilities
    - …*GCLK<xy>*: directly connect to global clock buffers
    - …*VRN / …VRP*: digital controlled impedance calibration input
    - …*VREF*: input comparator threshold voltage (certain IO standards)
    - …*D<y>*: Data input during configuration in parallel mode (plus control signals)
    - …
- Configuration & Test
  - Serial configuration (CCLK, DONE, PROG_B, M<x>)
  - JTAG (TDI, TMS, TDO, TCK)

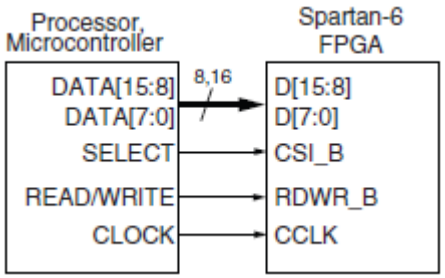
- Many additional special I/O with advanced FPGA families
  - Dedicated DDR memory controller interface
  - Gigabit Transceiver
  - SPI Flash Interface
  - ADC
  - …

# Configuration

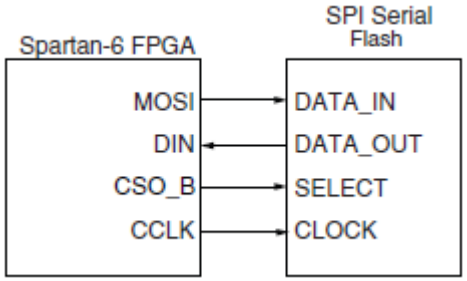
- FPGA configuration data stored in SRAM memories distributed across the chip
- Volatile memory ➔ needs to be written after power-up
- Different options:




a) JTAG (slave, serial)




b) SelectMAP (slave, parallel)

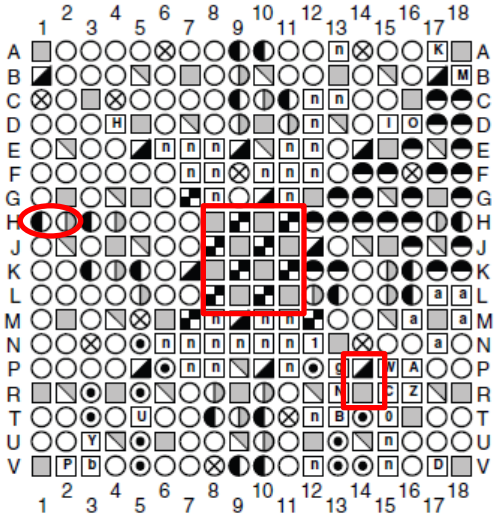


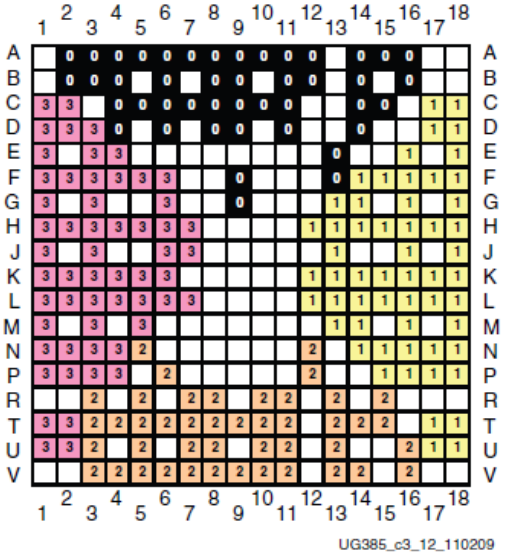

b) SPI Flash (master, serial)

- Data to download
  - Spartan-3 XC3S50 (smallest device): ~53kByte
  - Virtex-7 XCV2000T (largest device) : ~55MByte
  - Spartan-6 XC6SLX9 (device used in the tutorials): ~340kByte

# FPGA Packaging

- Spartan 3 LX9, CSG324 Package





UG385_c3_12_110209

I/O Bank Diagram

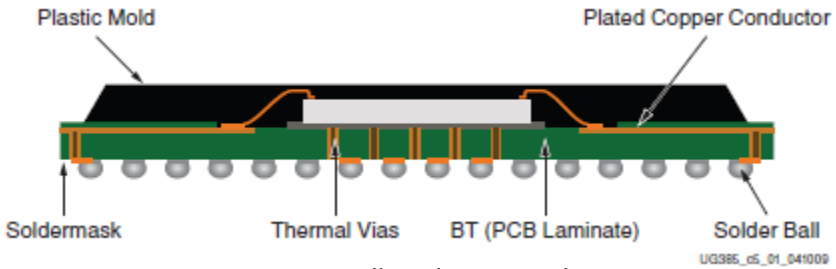
| User I/O Pins | Multi-Function Pins | | Dedicated Pins | Other Pins |
|---|---|---|---|---|
| IO_LXXY_# | VREF | CCLK | PROGRAM_B_2 | GND |
| | P_GCLK | CSI | TCK | VCCAUX |
| | N_GCLK | CSO | TDI | VCCINT |
| | D0 - D15 | DIN | TDO | VCCO |
| | A0 - A25 | DOUT_BUSY | TMS | NC |
| | FCS / FWE / FOE / HDC / LDC | HSWAPEN | DONE_2 | |
| | | INIT | SUSPEND | |
| | RDWR_B_VREF | M1, M0 | CMPCS_B_2 | |
| | | AWAKE | | |

UG385_c3_11_111909

Pinout Diagram

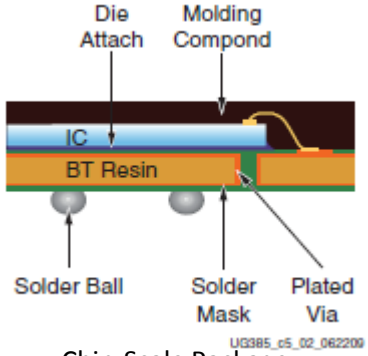
# Advanced FPGA Packaging

- Cavity-Up Ball Grid Array (**BGA**) Package
  - medium to high performance/density
  - low cost (chip is wire bonded)



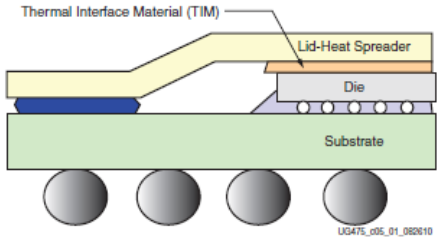

Cavity-Up Ball Grid Array Package

- Chip-Scale Package (**CSP**)
  - Like BGA with package size ~ die size




Chip-Scale Package

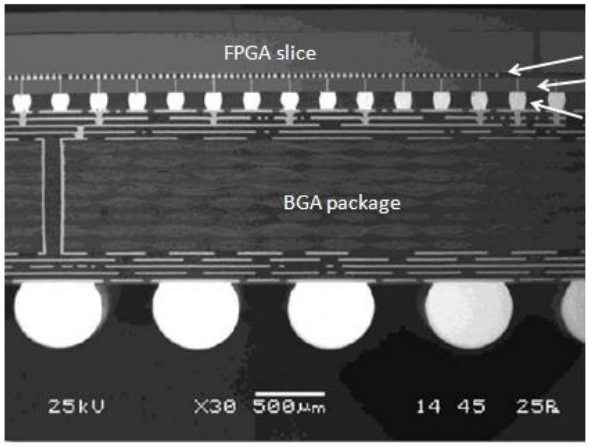
- Finned Heat Sink BGA
  - enhanced thermal ratings (up to 40W)
  - ultra high performance FPGAs
  - chip is bump bonded



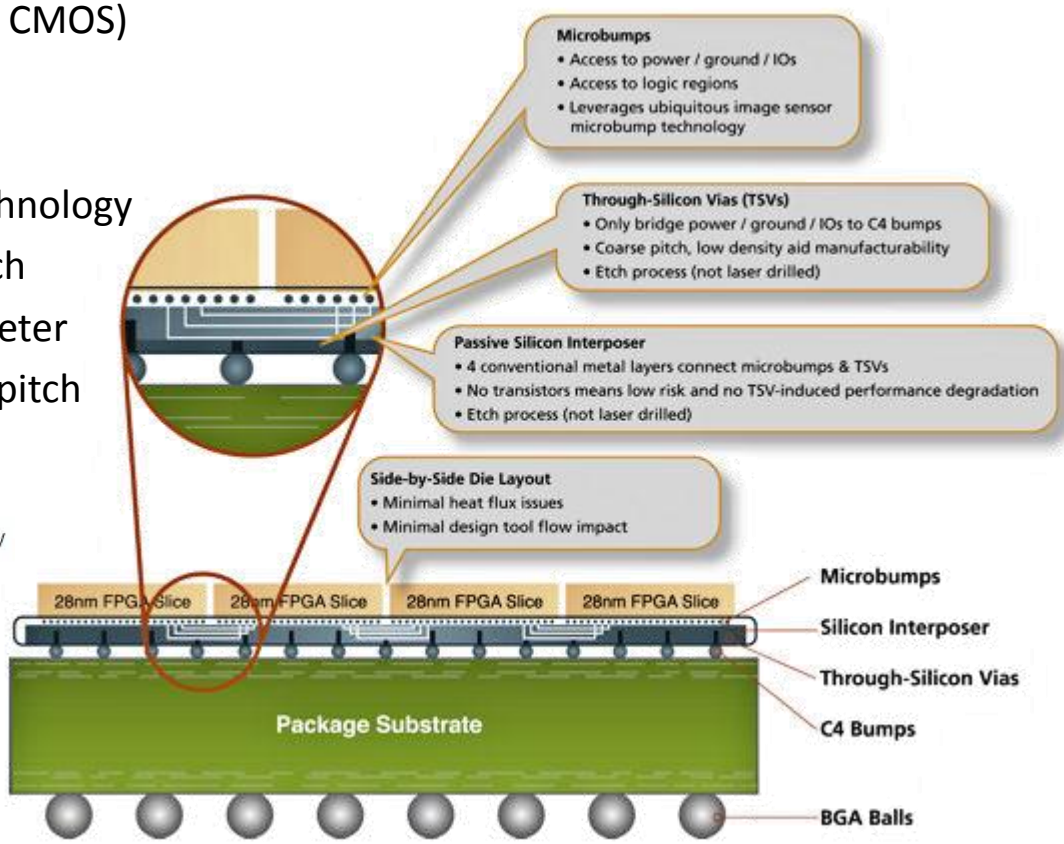

Heat Spreader with Thermal Interface Material

# Going to higher Densities…

- Stacked silicon interconnect (SSI, aka **2.5D** Integration)
  - Multiple FPGA chips in one package
  - Virtex-7: Up to four FPGA tiles (28nm CMOS)
- Common substrate
  - Passive silicon interposer, 100µ thick
  - Four metal layers based on 65nm technology
  - Microbumps to the FPGAs, 45µm pitch
  - Through silicon vias (**TSV**s), 10µ diameter
  - C4 bumps to the BGA package, 180µ pitch




Micro bumps
Si interposer with TSV
C4 bumps

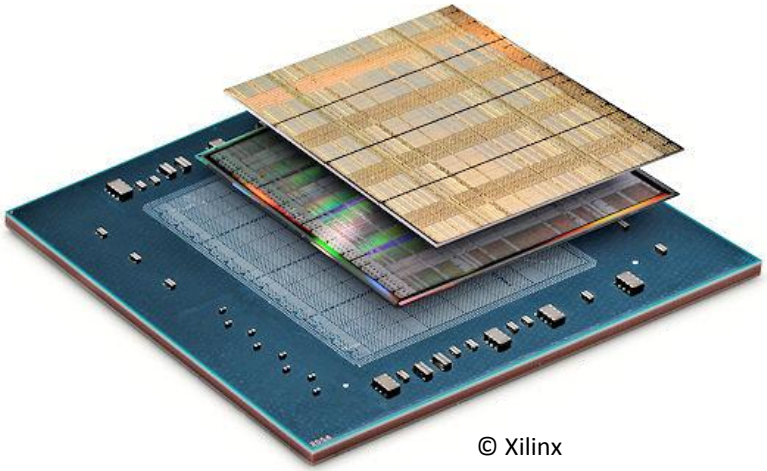



Drawings © Xilinx

# State-of-the-Art

Virtex-7 2000T

- Active area: 775 $mm^2$ (four tiles)
- 6.8 billion transistors
- Logic equivalent: 20 million logic gates
- 1200 user IO pins
- 26 Gbit transceiver
- Demonstration:
  - 3600 (!) PicoBlaze soft core processors implemented on one chip
  - 180 GIPS
  - < 20W



© Xilinx

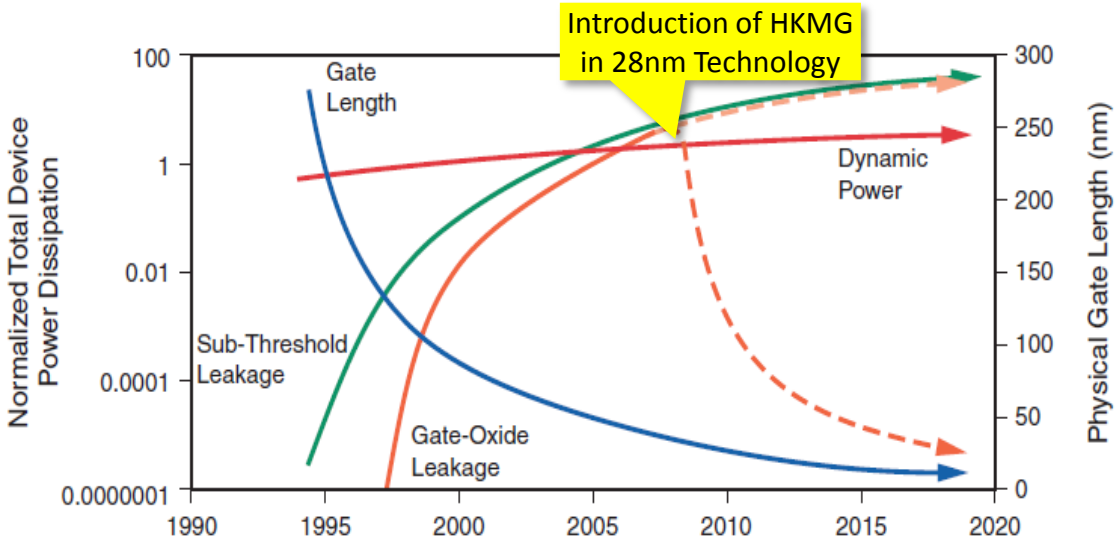
➔ still more to come: Integration of Dual ARM Cortex A9

# Aspects of Technology Scaling

- Finer process geometries lead to higher **static power consumption**
  - sub-threshold leakage ➔ use MOS with up-shifted $V_{THR}$: low power but also slower
  - gate leakage ➔ use of thicker gate oxide technological **not possible**

- Introduction of new MOS transistor gate configuration (HKMG) starting from 28nm
  - high *k* (ε = 25) insulator + metal gate instead of SiON (ε = 3.9) + poly silicon

  ➔ thicker gate isolation with no loss of current control ($g_m$)



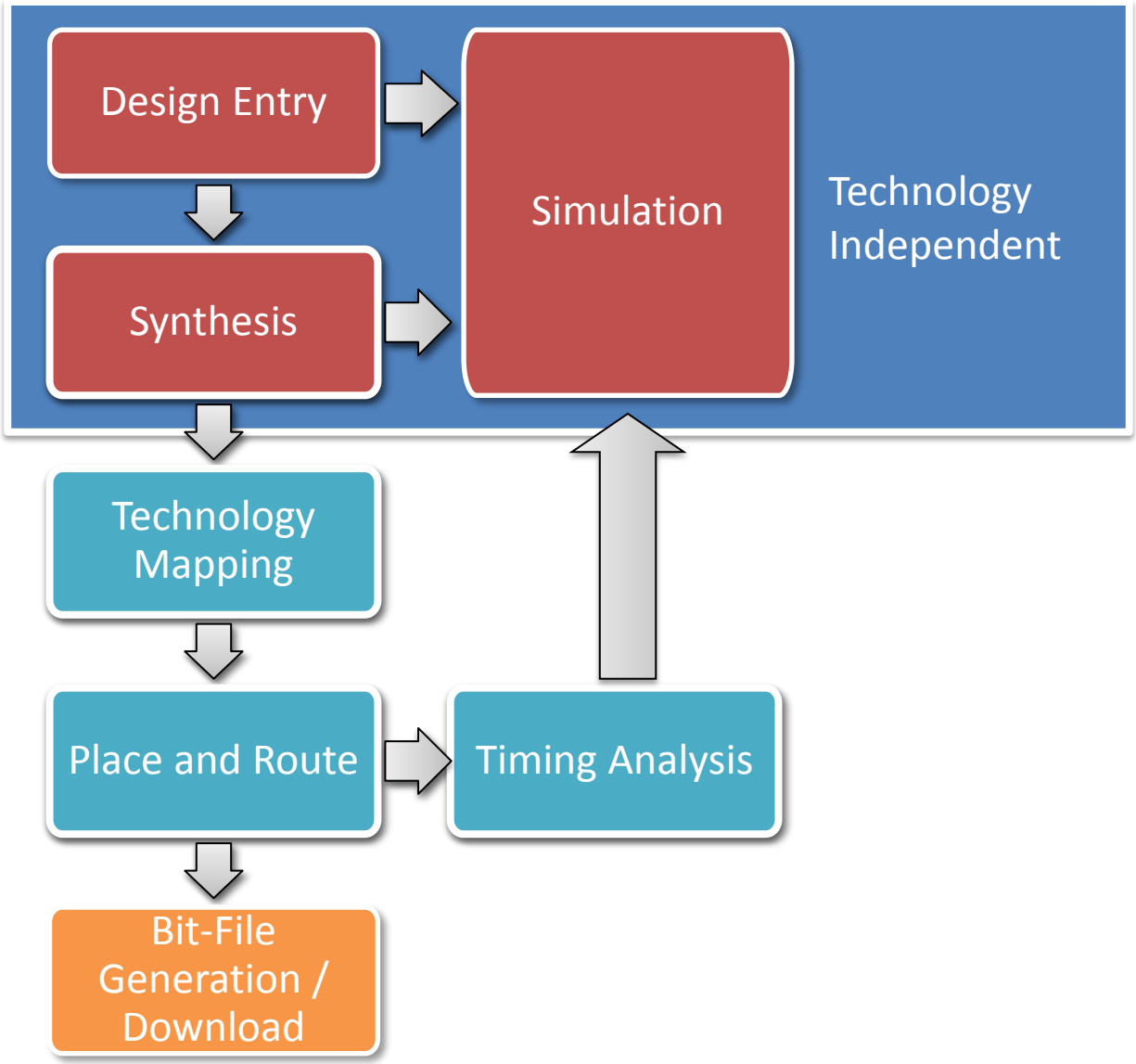

Source: Semiconductor Industry Association. The International Technology Roadmap for Semiconductors, 2002 Update. SEMATECH: Austin, TX, 2002.

WP312_01 _021810

# Trends in FPGA Development

- More hard IP (SOC)
  - Analog functions (ADC, DAC…)
  - Peripheral interfaces (USB…)
  - Processor cores (ARM Cortex)
- Multi chip packages
  - Higher logic densities
  - Less power
- Interconnect
  - Improve Logic to IO ratio
  - More (and) faster Serial Transceivers
- Device family variants optimized for
  - Digital filters (DSP blocks) → base stations, real-time image processing
  - High density logic → general purpose
  - Serial Connectivity (Multiple-Gigabit Transceivers) → network switch

# Design Methodology



# Design Methodology

**Design entry** options (here only as an overview, see next lectures and tutorials)

- Schematic Capture
  - Small designs
- Hardware Description Language (HDL)
  - Language: Verilog or VHDL
  - In addition: State Diagram Entry to generate Finite State Machines (FSM)
  - CoreGenerator to configure Hard- and Soft IP cores

➔ Lecture & Tutorial “From HDL to Physical Implementation”

- Embedded System Design
  - Microprocessor core (soft or hard)
  - User Application
  - User Logic

➔ Lecture & Tutorial “Embedded Systems Design”