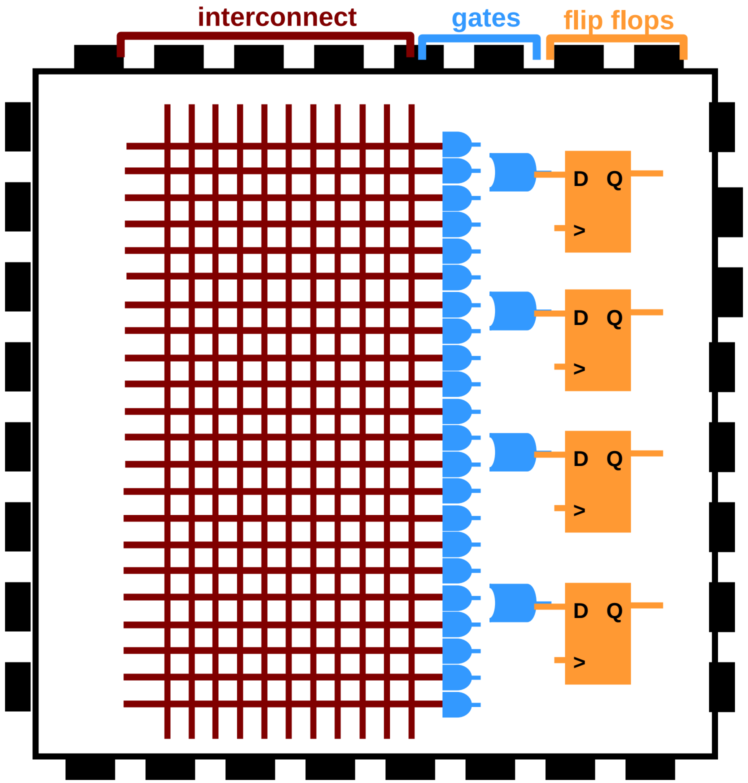
FPGAs!

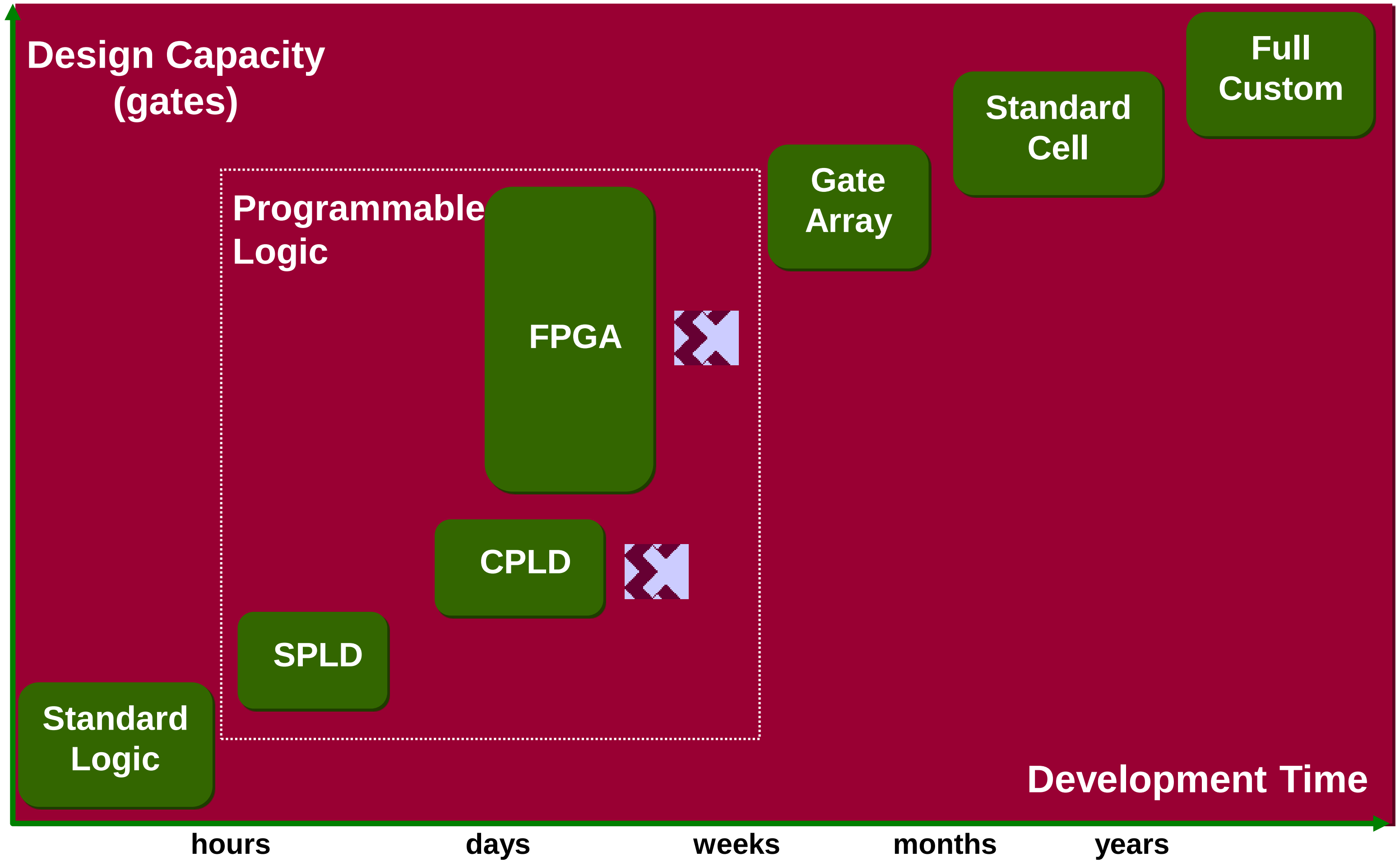
Basic Concepts – Building Blocks

- There are (3) fundamental building blocks found in digital devices
  - Gates
  - Flip-Flops
  - Interconnect (or routing)



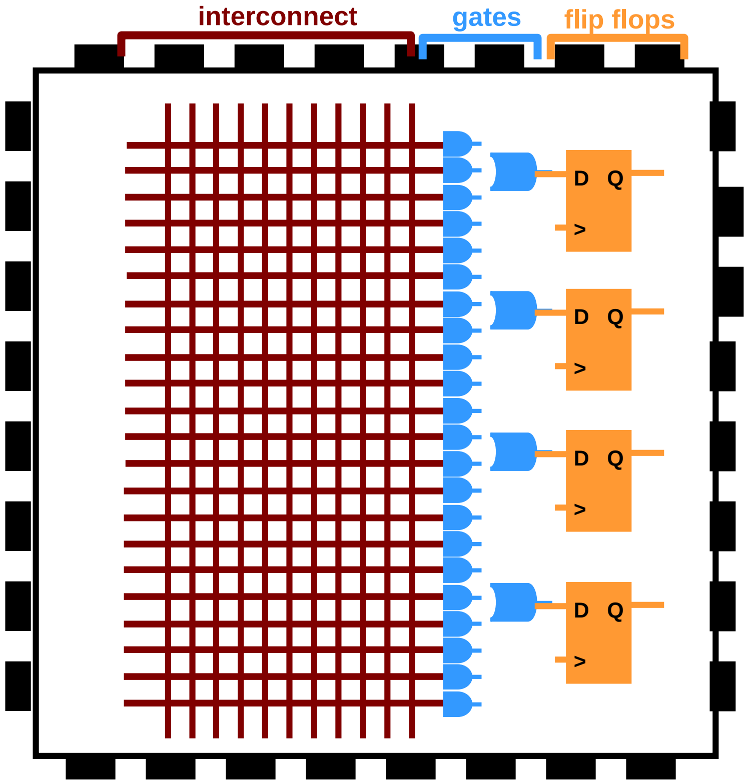
Digital Logic Landscape

The following slides provide a history of the various logic devices

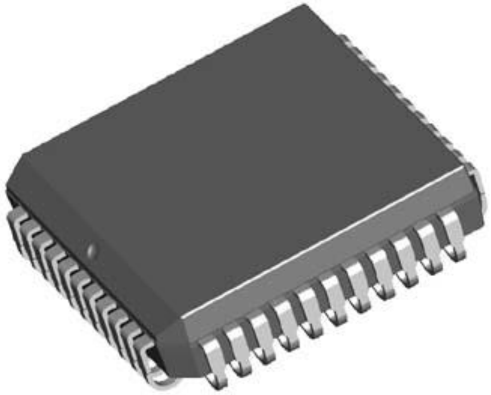


Digital Logic History - PLDs

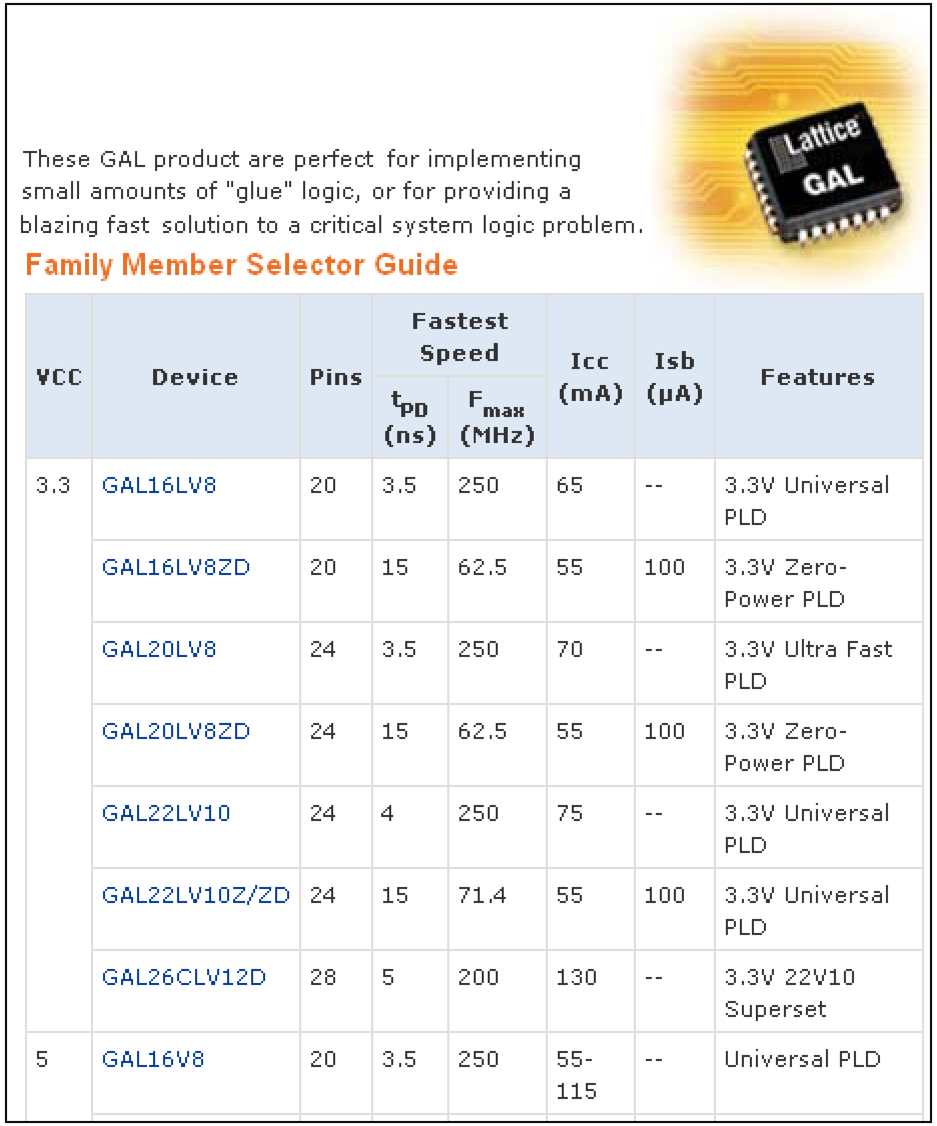
- Developed in the late 70s
- Major player today: Lattice
- First device that needs software
- 50 – 200 gates



A very common low cost IC package has pins on all 4 sides called a Plastic-Leaded Chip Carrier (PLCC)



PLD Example



These GAL product are perfect for implementing small amounts of "glue" logic, or for providing a blazing fast solution to a critical system logic problem.

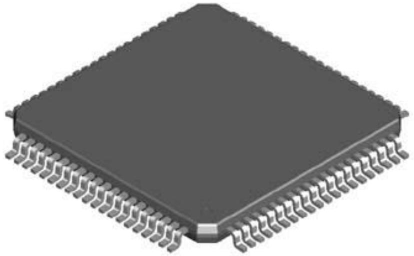
Family Member Selector Guide

VCC	Device	Pins	Fastest Speed		Icc (mA)	Isb (μA)	Features
			t_{PD} (ns)	F_{max} (MHz)			
3.3	GAL16LV8	20	3.5	250	65	--	3.3V Universal PLD
	GAL16LV8ZD	20	15	62.5	55	100	3.3V Zero-Power PLD
	GAL20LV8	24	3.5	250	70	--	3.3V Ultra Fast PLD
	GAL20LV8ZD	24	15	62.5	55	100	3.3V Zero-Power PLD
	GAL22LV10	24	4	250	75	--	3.3V Universal PLD
	GAL22LV10Z/ZD	24	15	71.4	55	100	3.3V Universal PLD
	GAL26CLV12D	28	5	200	130	--	3.3V 22V10 Superset
5	GAL16V8	20	3.5	250	55-115	--	Universal PLD

Digital Logic History - Gate Array

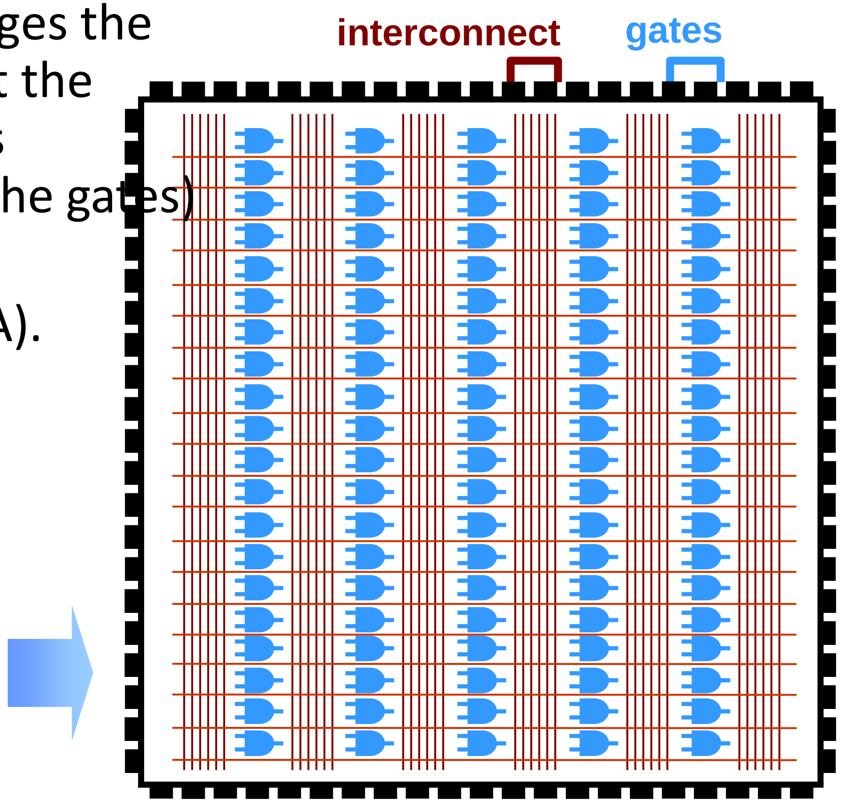
Definition: A pre-built IC consisting of a regular arrangement of gates and interconnect (routing) where the ***interconnect*** is modified to achieve a customer's desired functions.

- The customer designs the behaviors/functions
- The vendor manipulates/changes the metal interconnect to arrive at the customer's specified functions (that is, the vendor hooks up the gates)
- Sometimes called an Uncommitted Logic Array (ULA).



Gate Array in a TQFP package

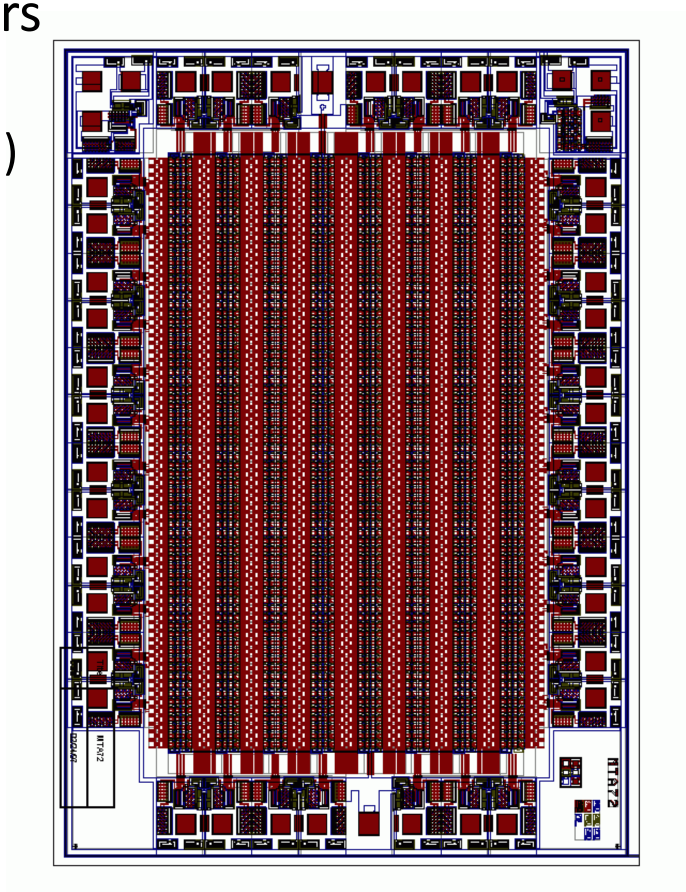
Packaging Enhancement: To increase the number of I/Os (Inputs/Outputs), the pin thickness and spacing (pitch) are dramatically reduced in this Thin Quad FlatPack package (TQFP).



1,000,000+ gates

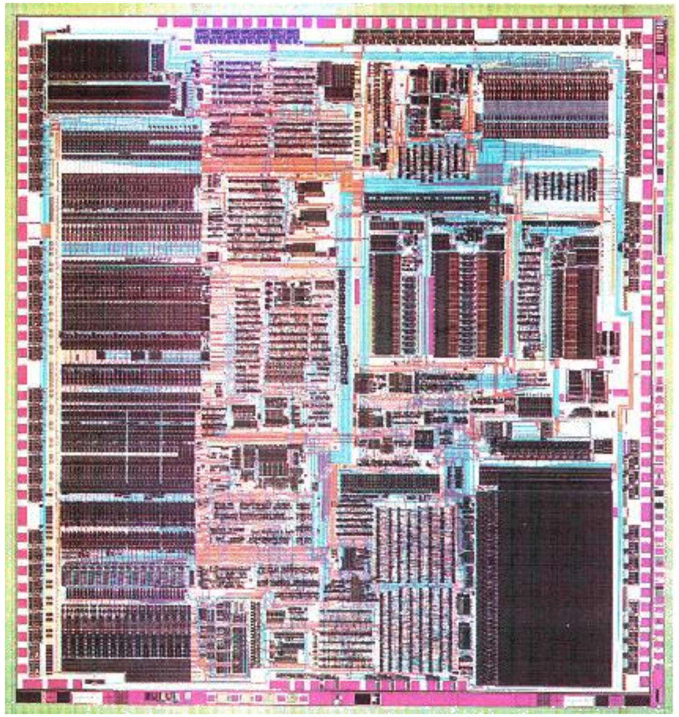
Gate Array

- The ultimate building tool set for digital designers
- Advantages
  - Very dense (today over 10,000,000 gates (10 million))
  - Fast performance (200 – 500 MHz)
  - Very low unit cost
- Disadvantages
  - Long turn around time (3 - 6 months)
  - $50K - $500K NRE
    - NRE = Non-Recurring Engineering charges, which are one-time “set-up” charges to ready the “fab” to build the custom part
      (“fab” = the “factory” where the ICs are manufactured;
      the “fabrication plant”)
  - Risk of re-spins



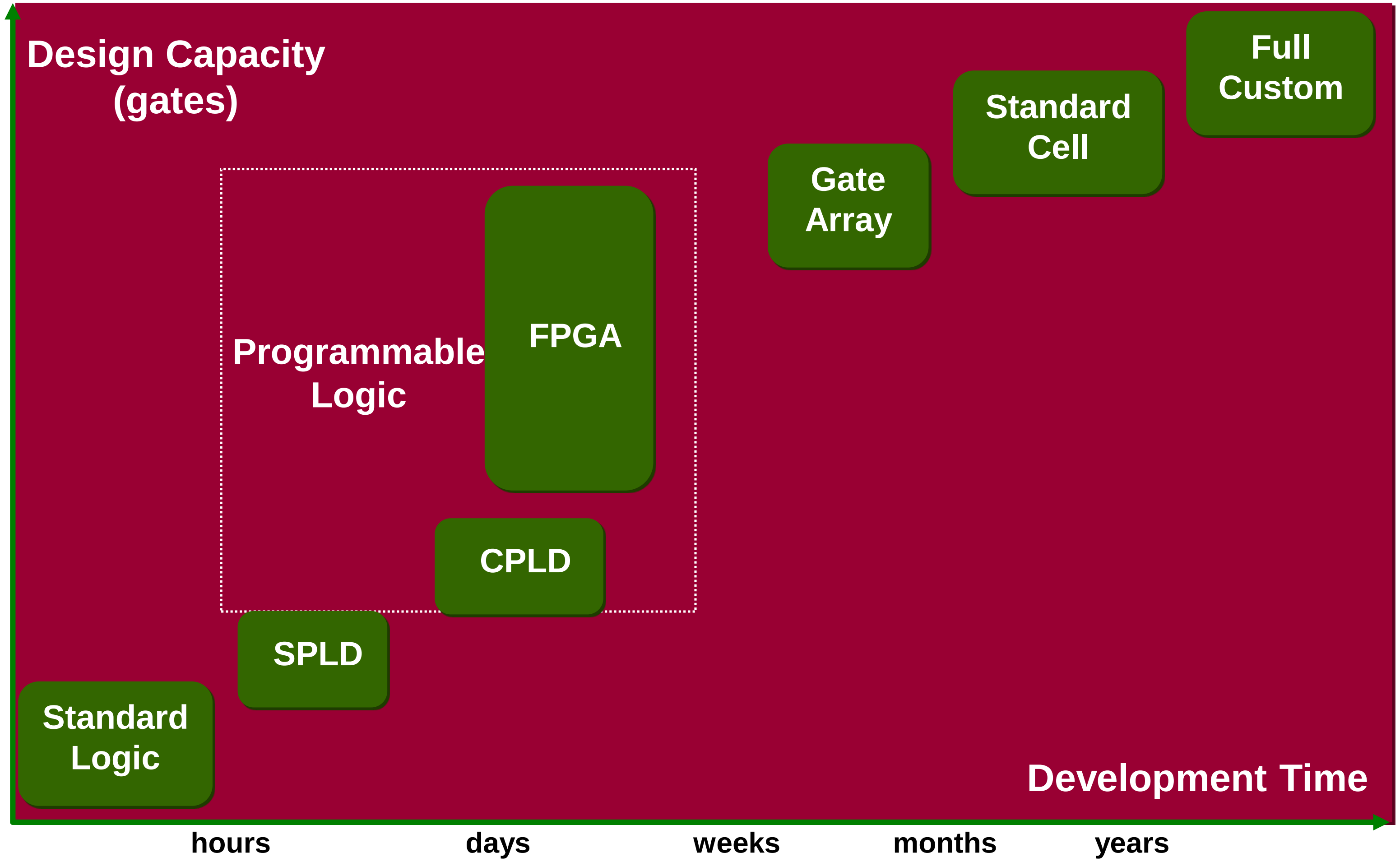
Digital Logic History - Standard Cell

- This device features a series of customized “cells”
  - Each cell is optimized for its “standard” function
- Cells are chosen form a library from the Standard Cell vendor, customized, and connected to the other cells and the routing on the part.
- There are no standard layers to the device; each layer is a unique design



- Advantages:
  - More optimized die size compared to GA
  - Cheaper device price compared to GA
  - Can add analog functions
- Disadvantages:
  - Extremely high NRE charges (up to $1M)
  - Requires >250k+ units/year
  - Much longer development time
  - Much higher risk (re-spins, etc.)

CPLDs, FPGAs


Design Capacity (gates)
Full Custom
Standard Cell
Gate Array
Programmable Logic
FPGA
CPLD
SPLD
Standard Logic
Development Time
hours
days
weeks
months
years

Digital Logic History - CPLD

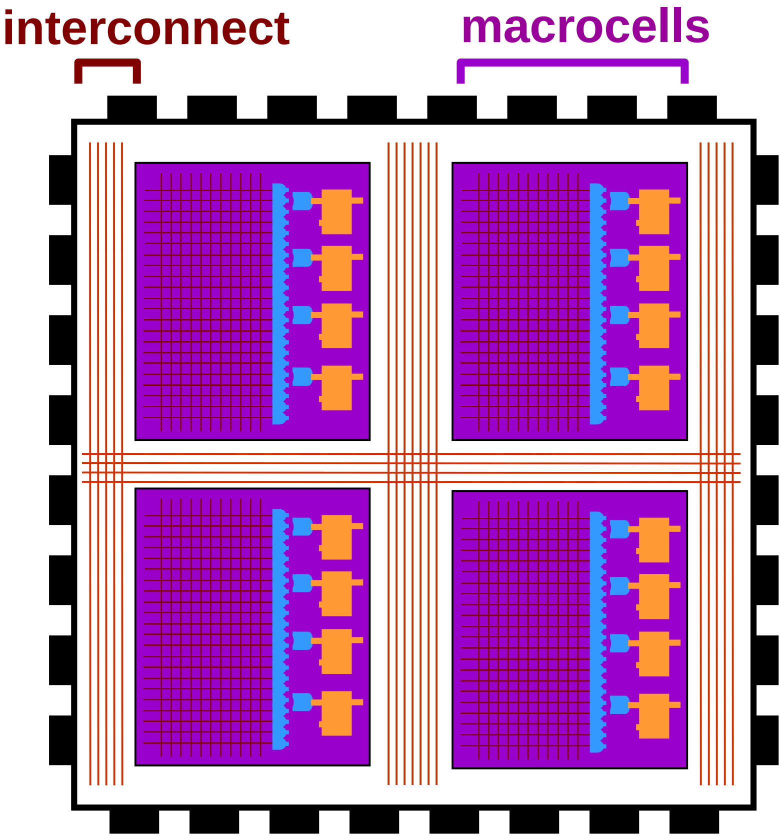
Complex Programmable Logic Device

Definition:

A CPLD contains a bunch of PLD blocks whose inputs and outputs are connected together by a global interconnection matrix.

CPLD has two levels of programmability:
--Each PLD block can be programmed
--The interconnection between the PLDs can be programmed.

CPLD technology was introduced in the late 80s



32-1024 macrocells

CPLDs

- Vendors: Altera, Lattice, Cypress, Xilinx
- 2 Primary Technologies
  - EEPROM
    (old technology)
  - FLASH
    (technology used by Xilinx CPLDs)

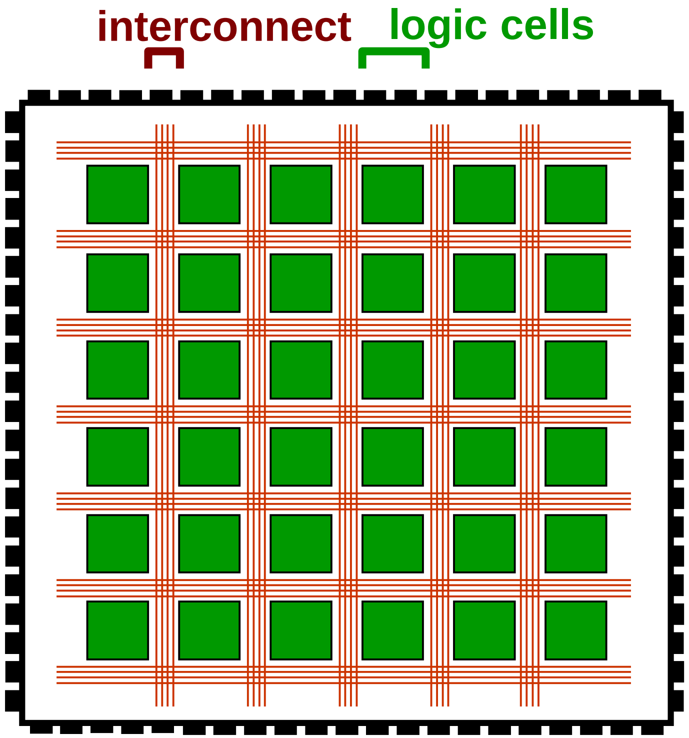
- FPGAs vs. CPLDs
  - FPGAs have much greater capacity
  - CPLDs are faster for some small applications
  - Both are easy to design

Digital Logic History - FPGA

Field Programmable Gate Array

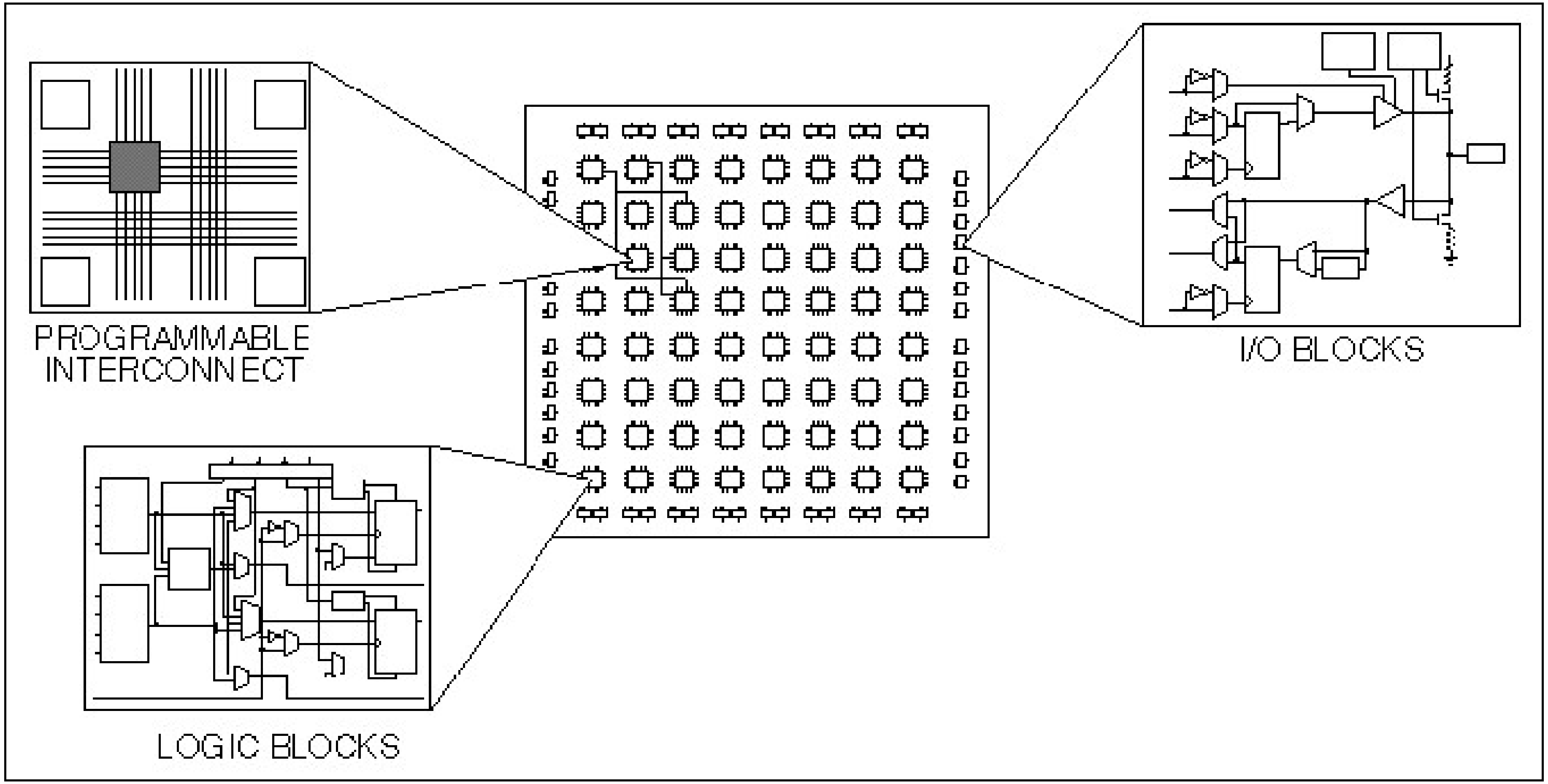
Definition:

- An array of “**logic cells**” surrounded by substantial routing, both of which are under the user’s control
- The CLB (Configurable Logic Block) is/was the fundamental building block of the logic cell, although today’s FPGAs use a very sophisticated collection of gates that goes beyond the original CLB design
  - The early Xilinx CLBs contained a (4) input look-up table (LUT), a flip-flop, and “carry logic”



>10 million gates

FPGA Building Blocks



An Early Xilinx CLB

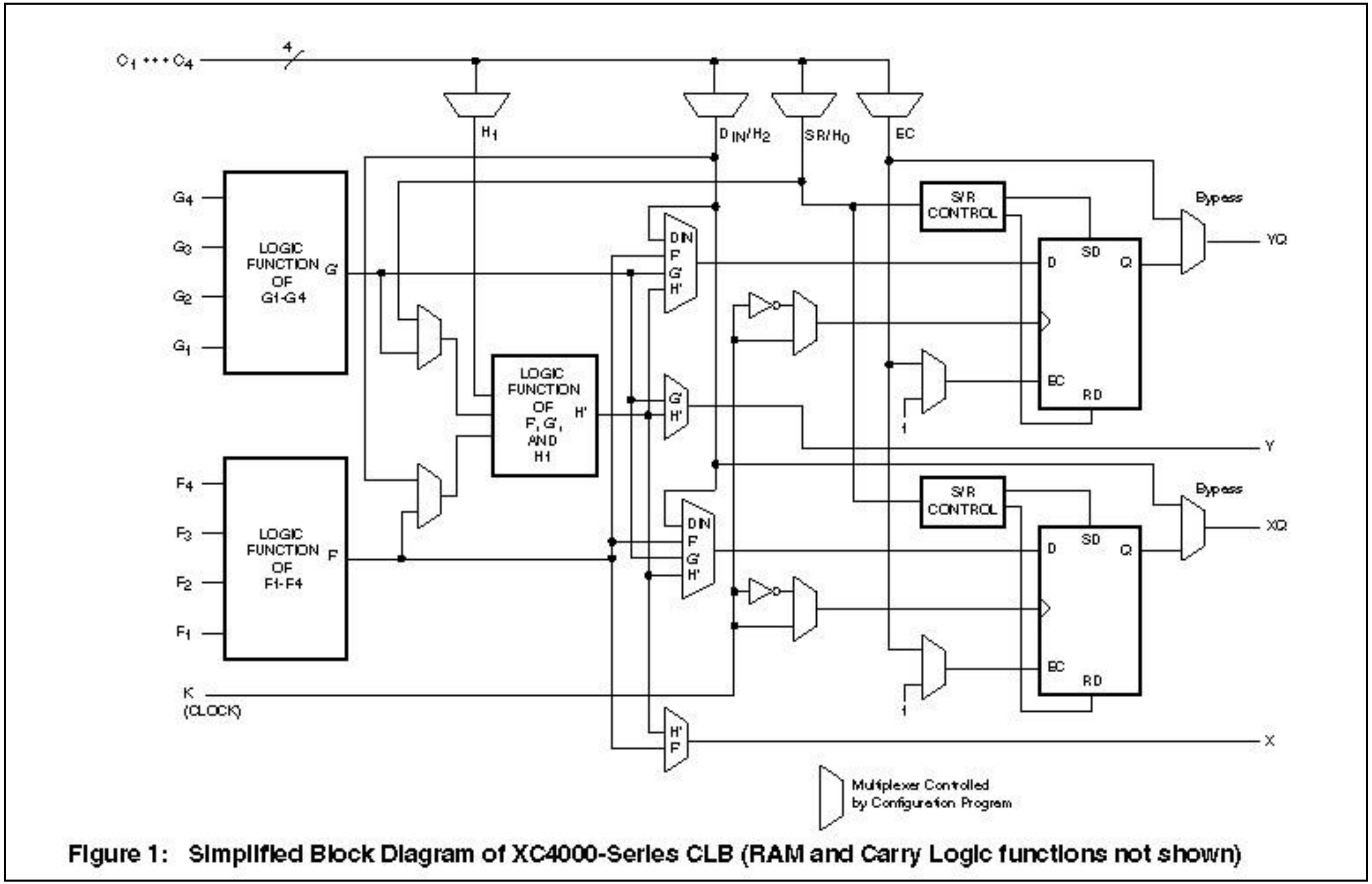


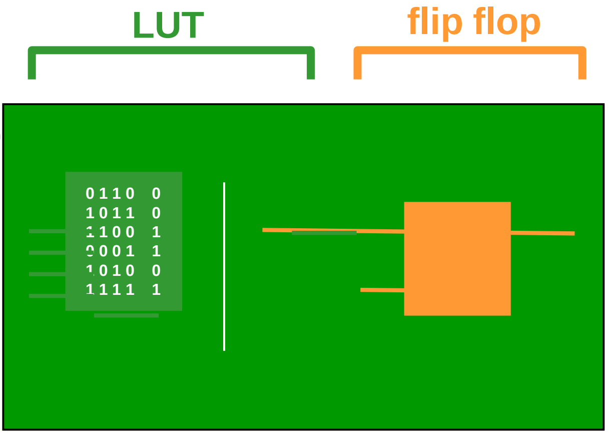
Figure 1: Simplified Block Diagram of XC4000-Series CLB (RAM and Carry Logic functions not shown)

Digital Logic History

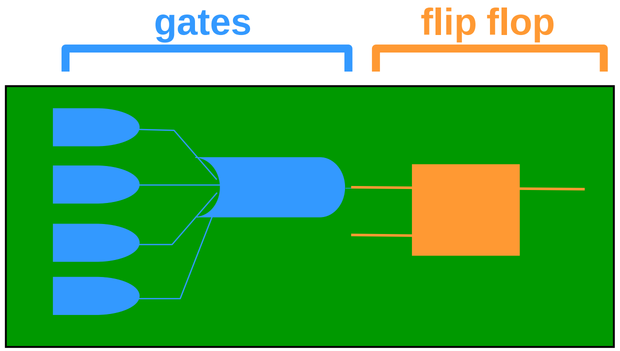
FPGA - Field Programmable Gate Array

2 types of FPGAs

- Reprogrammable (SRAM-based)
  - Xilinx, Altera, Lattice, Atmel
- One-time Programmable (OTP)
  - Actel, Quicklogic, EZchip



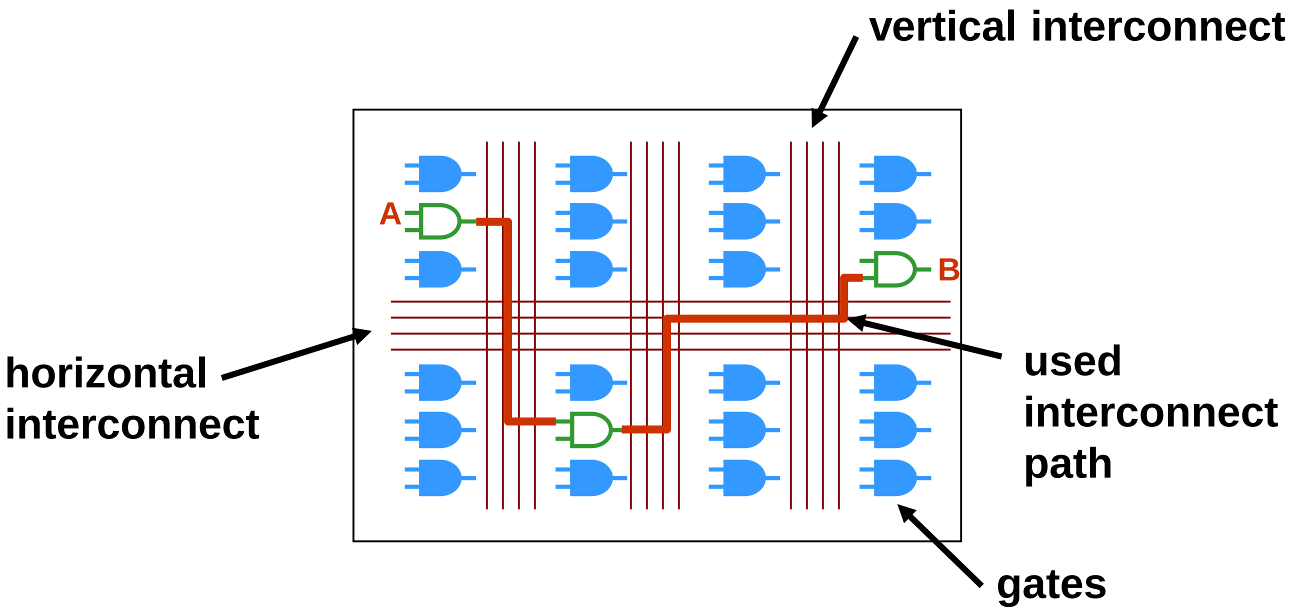
SRAM logic cell



OTP logic cell

Basic Concepts - Logic Interconnect

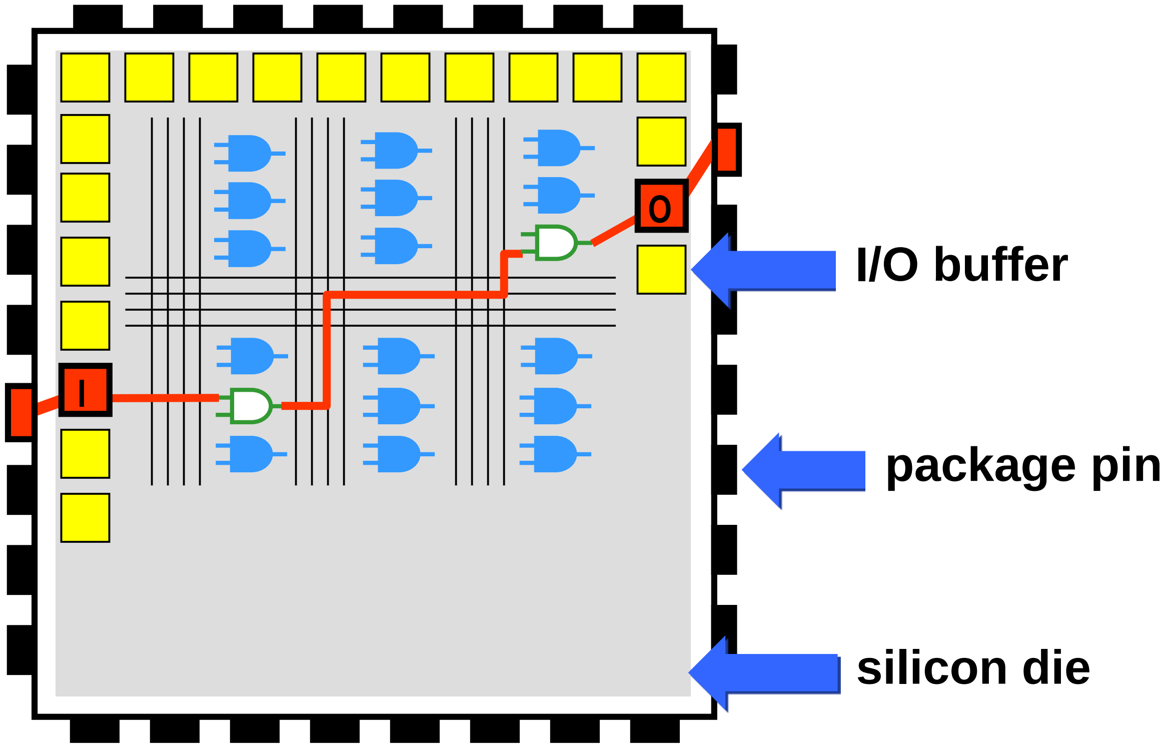
- Method to hook-up gates inside a single device
- Need to have enough routing to connect most gates
- Larger gate counts result in lots of routing, bigger die size, increased cost



Basic Concepts - I/Os

Inputs and Outputs

- All signals on & off chip must go through an I/O buffer
- User can choose many I/O buffer options

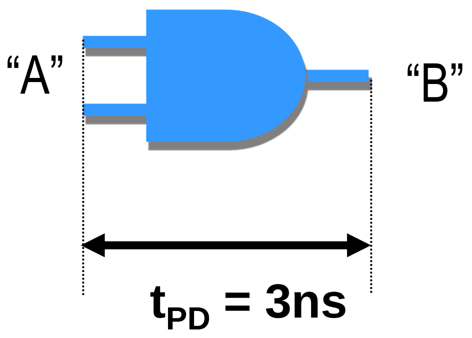


Basic Concepts

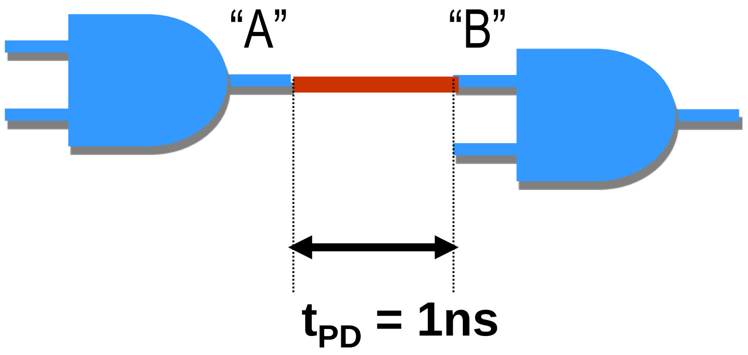
Propagation Delay (t_{PD})

Definition: The time required for a signal to travel from A to B, measured in nanoseconds (ns).

Gate Delay



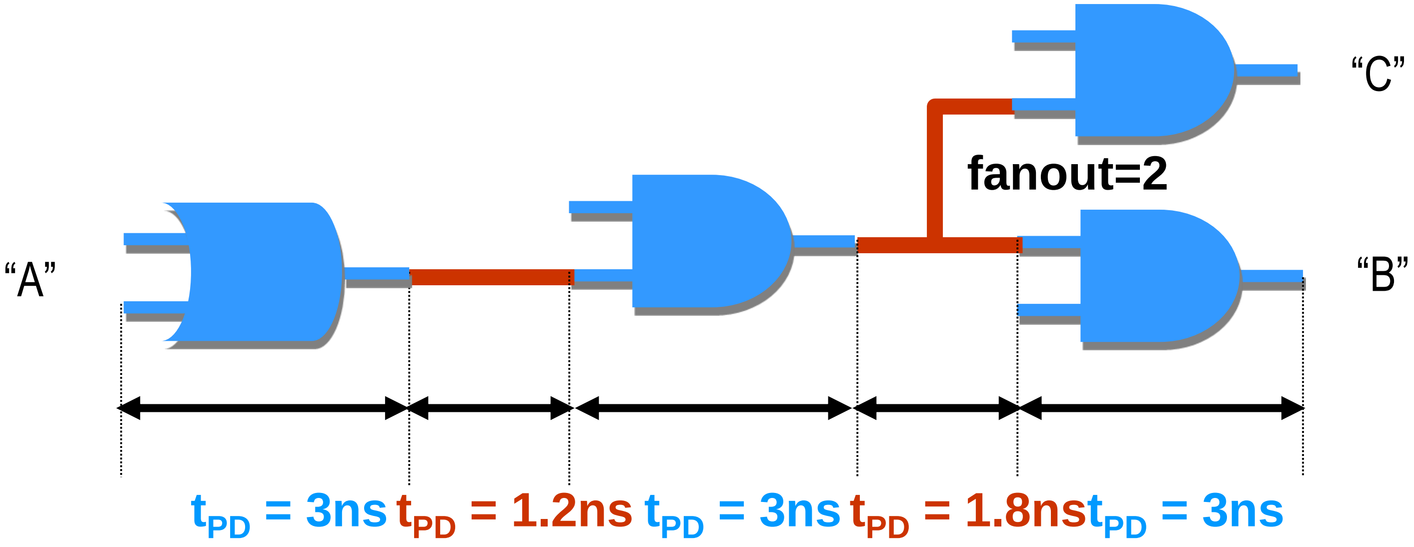
Interconnect Delay



Basic Concepts

Path Delay

Definition: The sum of all the gate and net delays from starting to ending point.



“C”

fanout=2

“A”

“B”

t_{PD} = 3ns t_{PD} = 1.2ns t_{PD} = 3ns t_{PD} = 1.8nst_{PD} = 3ns

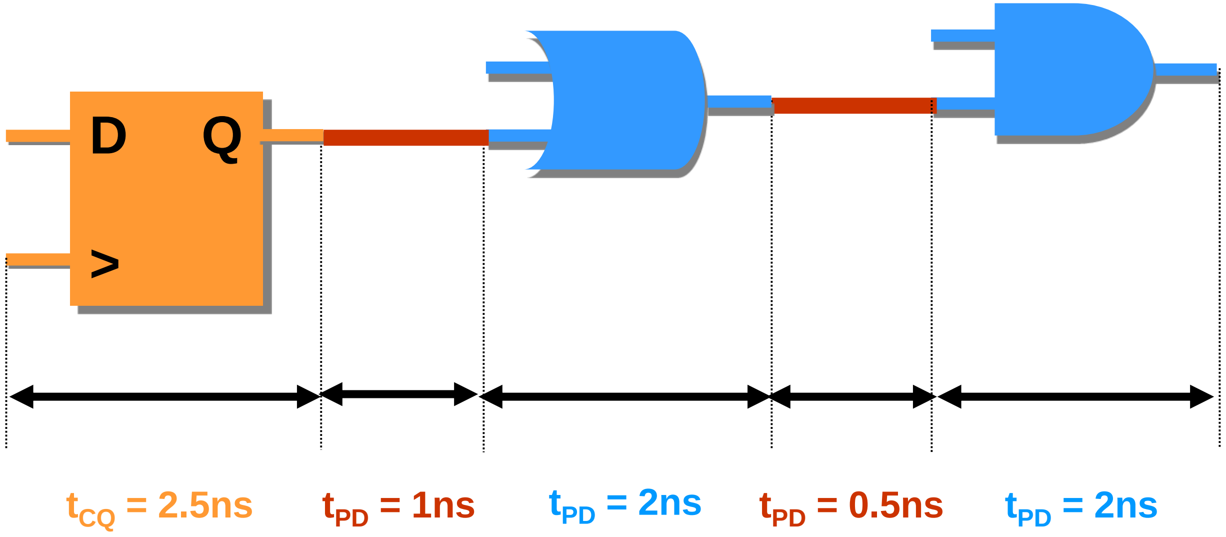
Path Delay “A” to “B” = sum of all gate + net delays

3ns + 1.2ns + 3ns + 1.8ns + 3ns = 12ns

Basic Concepts

Maximum System Performance (f_{MAX})

Definition: The fastest speed a circuit containing flip-flops can operate, measured In Megahertz (MHz).



Circuit Events per Second:
1 = 1 Hertz (Hz)
1,000 = kilo (kHz)
1,000,000 = mega (MHz)
1,000,000,000 = giga (GHz)

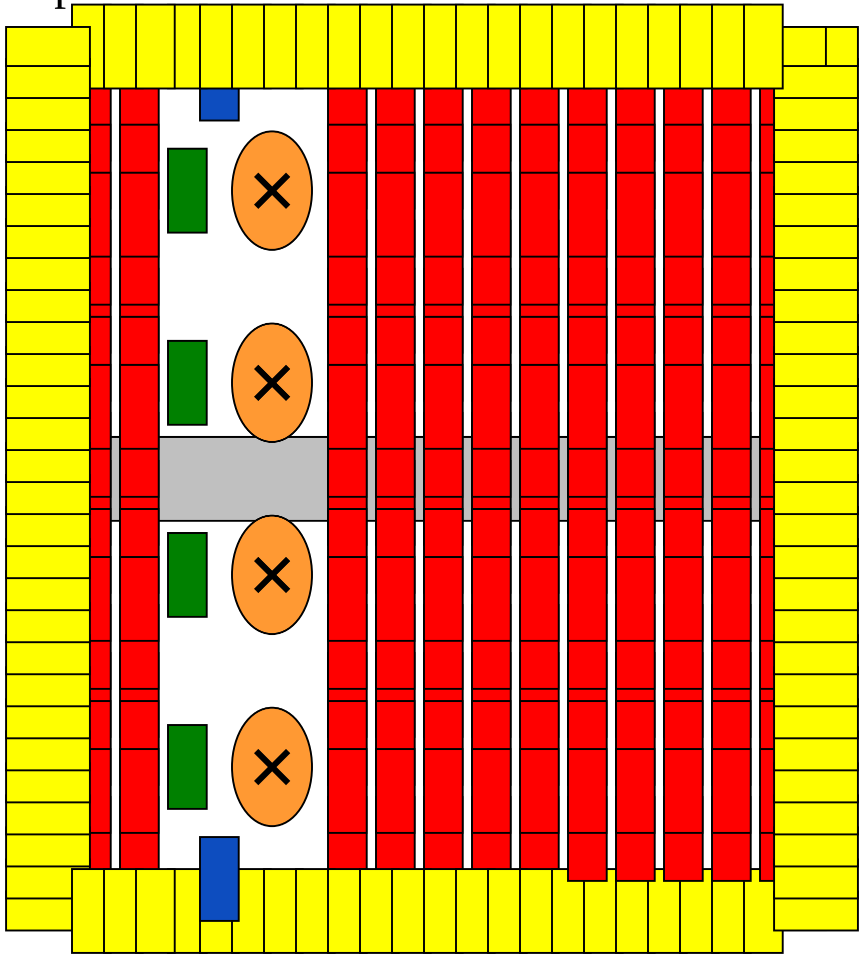
$$f_{MAX} = \frac{1}{\text{longest flip-flop path delay}}$$

f_{MAX} = 1/(flip-flop delay + gate delays + net delays)
= 1/(2.5 + 1 + 2 + 0.5 + 2)ns
= 125 MHz

Xilinx FPGA Architecture

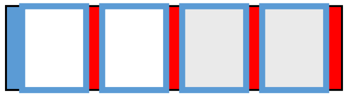
How are they arranged

Spartan 6



🟩 18Kbits
Dual Port RAM

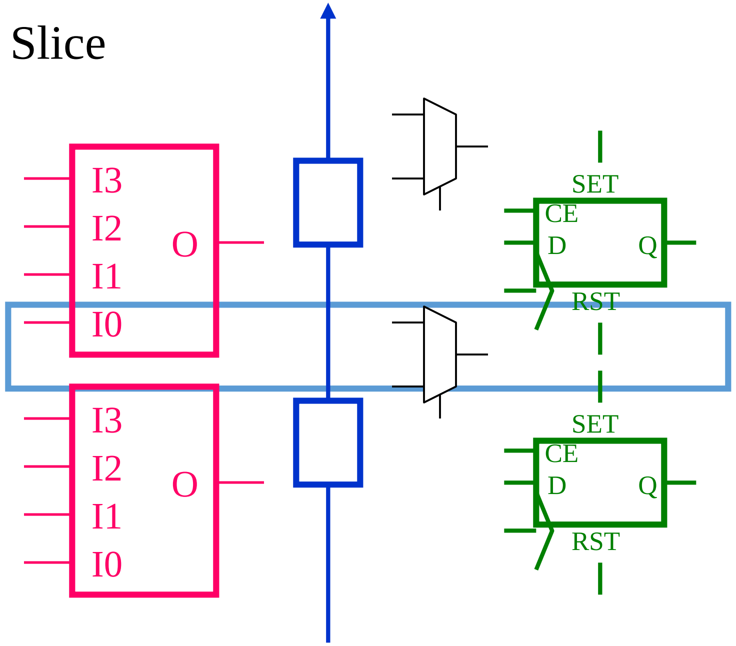
⊗ 18×18
Multiplier



CLB (Configurable Logic Block)

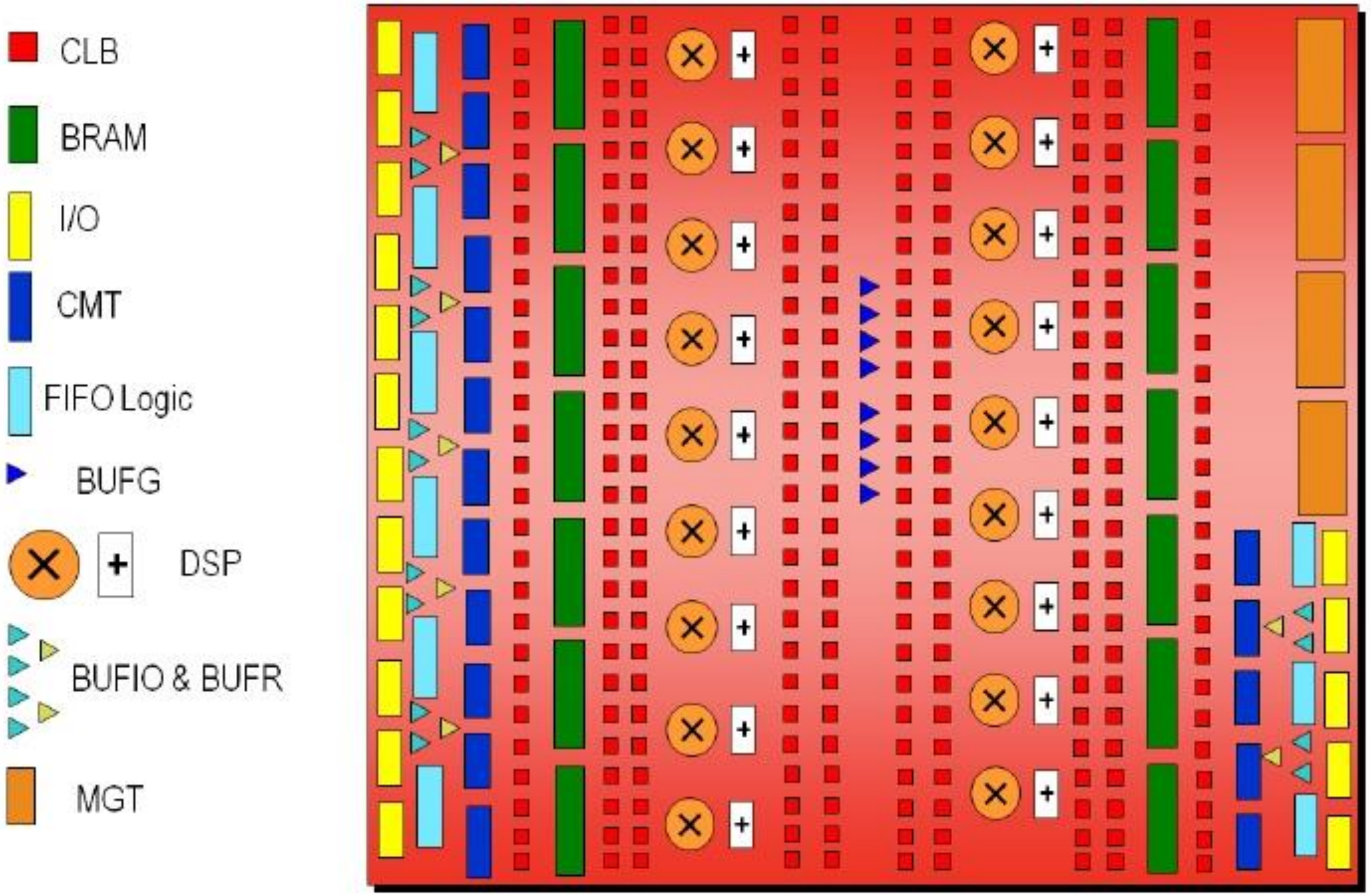
= 4 Slices

Slice



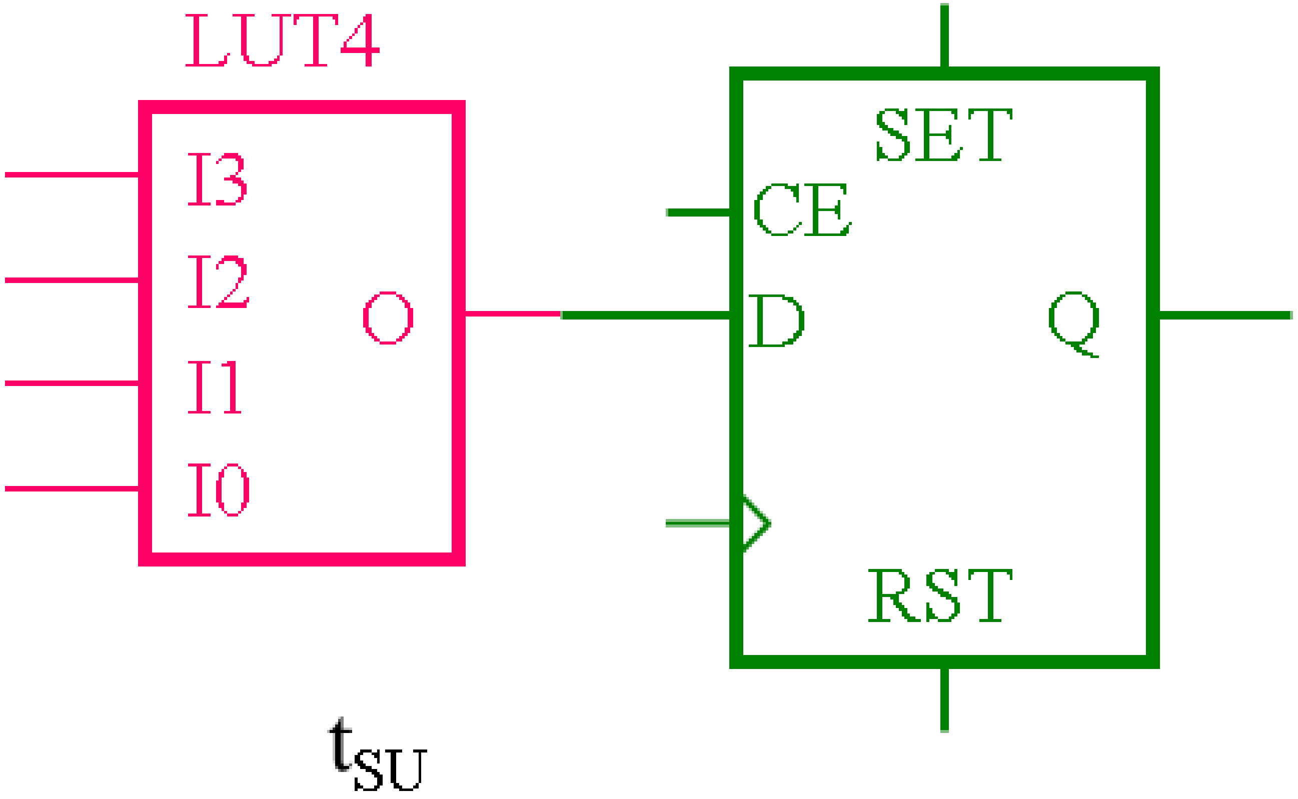
🟨 124 multi-standard I/O with JTAG

How they are arranged
Kintex-7 FPGA



Typical FPGA Logic Structure

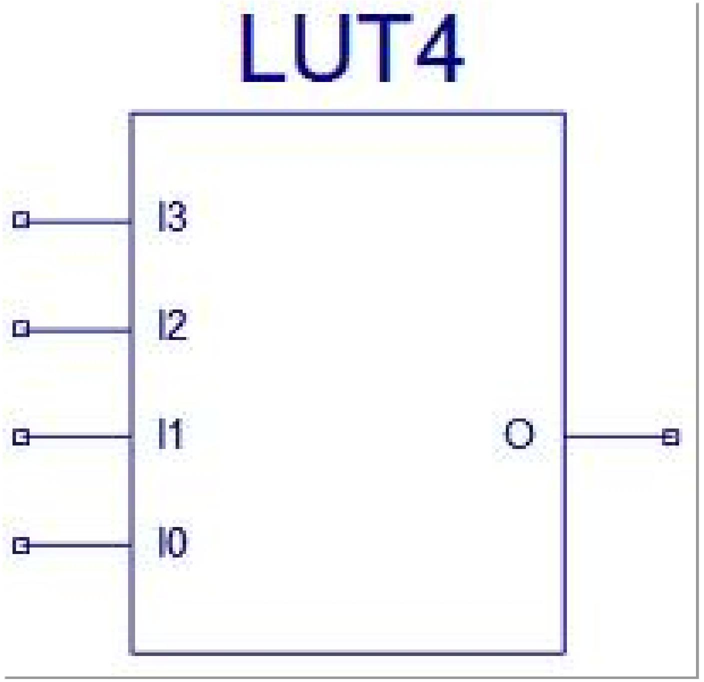
- LUT
- Flip flop



Typical 4 Input LUT

- 4 Inputs
- One Output

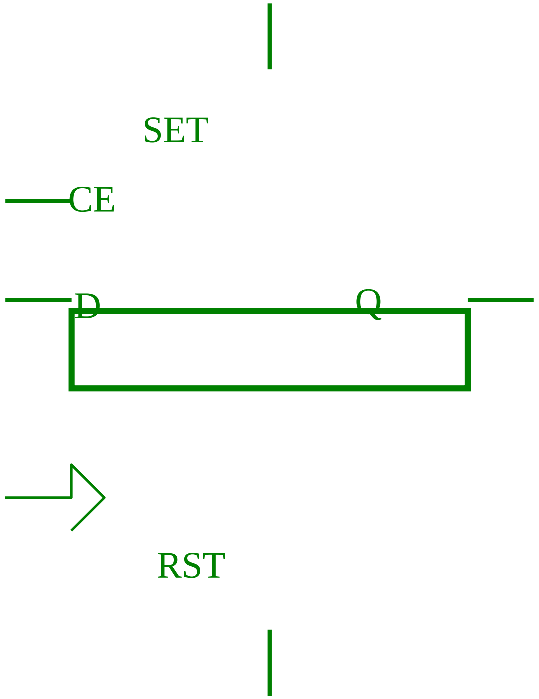
- Any 4 input Logic function can be implemented.



Flip Flop

- Input D
- Input Clock
- Input Clock Enable
- Input Set
- Input Reset

- Output Q

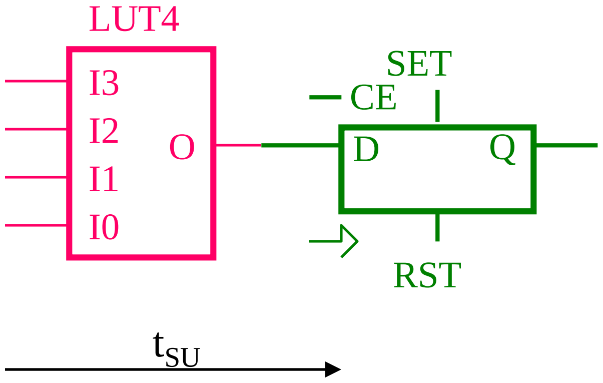


Making the Most of Controls

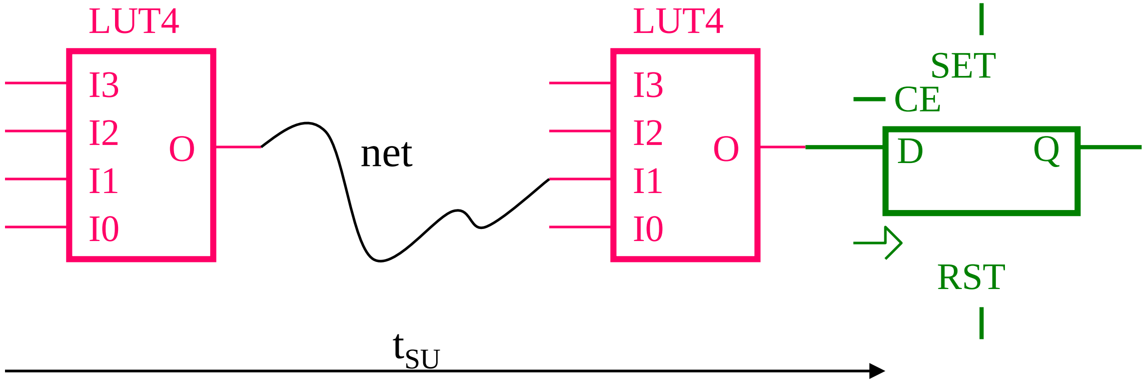
Dedicated Flip-Flop controls make designs smaller and faster.

1 level of logic - fast and small

Up to 4 data inputs plus 3 controls



2 levels of logic - significantly slower and twice the size (and cost)



Workshop - How can this be implemented?

This simple code describes a 4-input function followed by a Flip-Flop.
What size and performance is this function?

```
process (clk,reset)
begin
 if reset='1' then
    data_out <= '0';
   elsif clk'event and clk='1' then
    if enable='1' then
      if force_high='1' then
         data_out <= '1';
        else
         data_out <= a and b and c and d;
      end if;
    end if;
 end if;
end process;
```

reset

enable

set

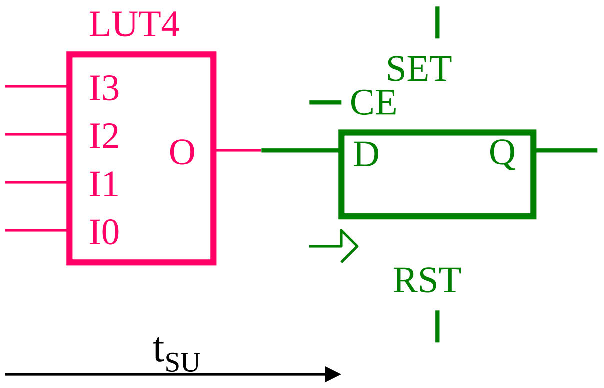
logic

Making the Most LUTs and FFs

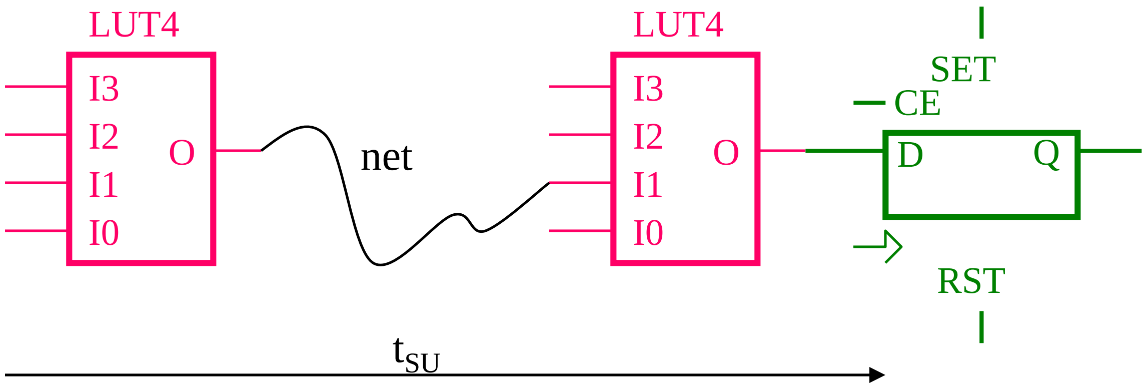
Dedicated Flip-Flop controls make designs smaller and faster.

1 level of logic - fast and small

Up to 4 data inputs plus 3 controls



2 levels of logic - significantly slower and twice the size (and cost)



Workshop - How can this be implemented?

This simple code describes a 4-input function followed by a Flip-Flop.
What size and performance is this function?

```
process (clk,reset)
begin
 if reset='1' then
    data_out <= '0';
   elsif clk'event and clk='1' then
    if enable='1' then
      if force_high='1' then
         data_out <= '1';
        else
         data_out <= a and b and c and d;
      end if;
    end if;
 end if;
end process;
```

reset

enable

set

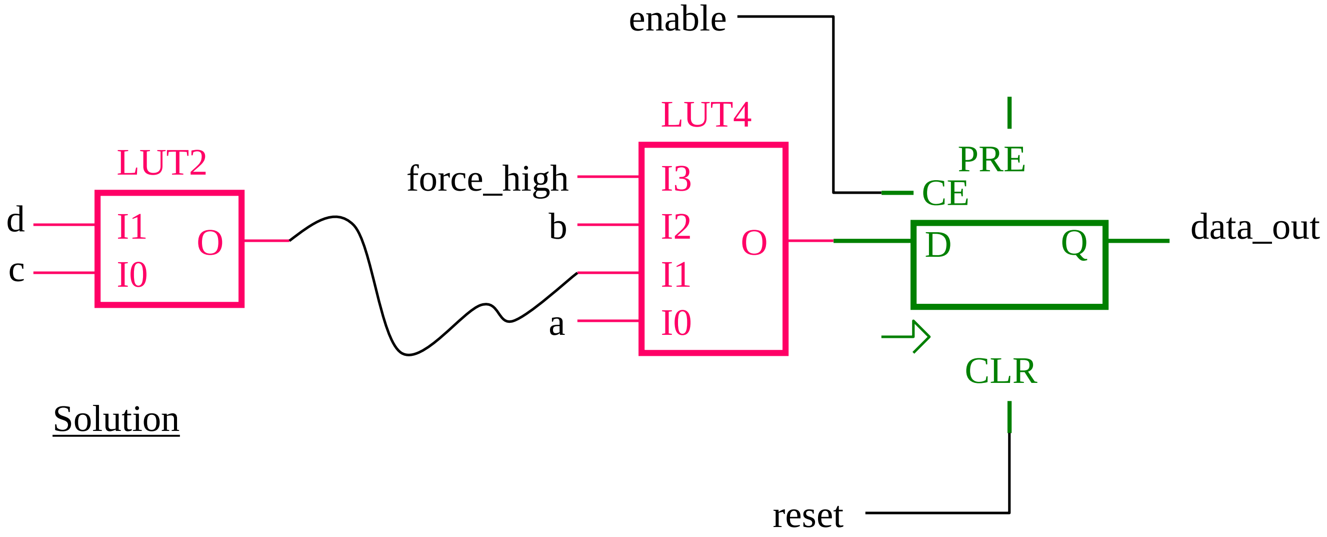
logic

TWICE the Cost and Half the Speed

TWICE as Big as it should be and Slow!

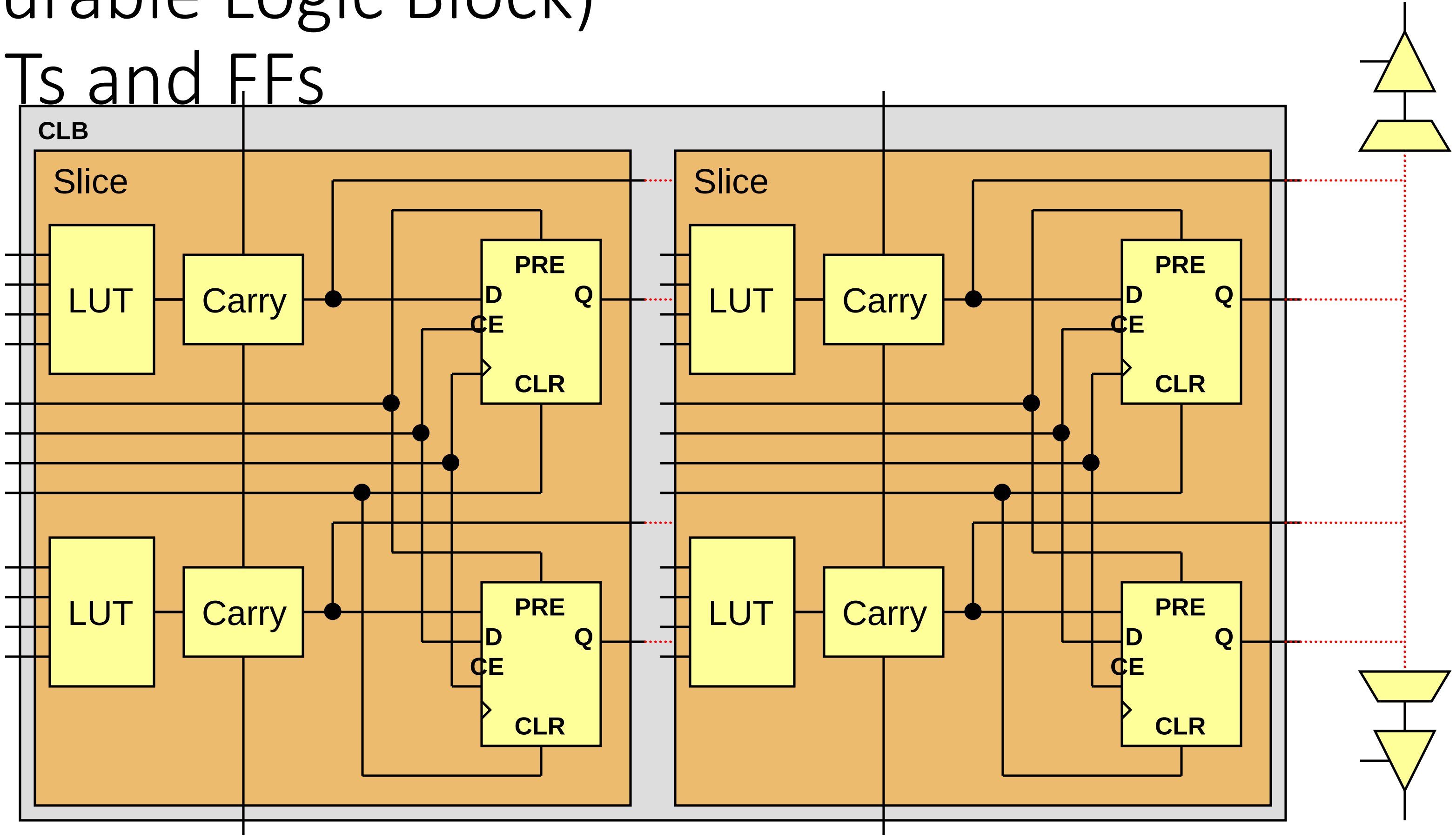
Report

```
Cell Usage :
# BELS                  : 2
#      LUT2             : 1
#      LUT4             : 1
# FlipFlops/Latches     : 1
#      FDCE             : 1
```



Solution

CLB (Configurable Logic Block)
Multiple LUTs and FFs

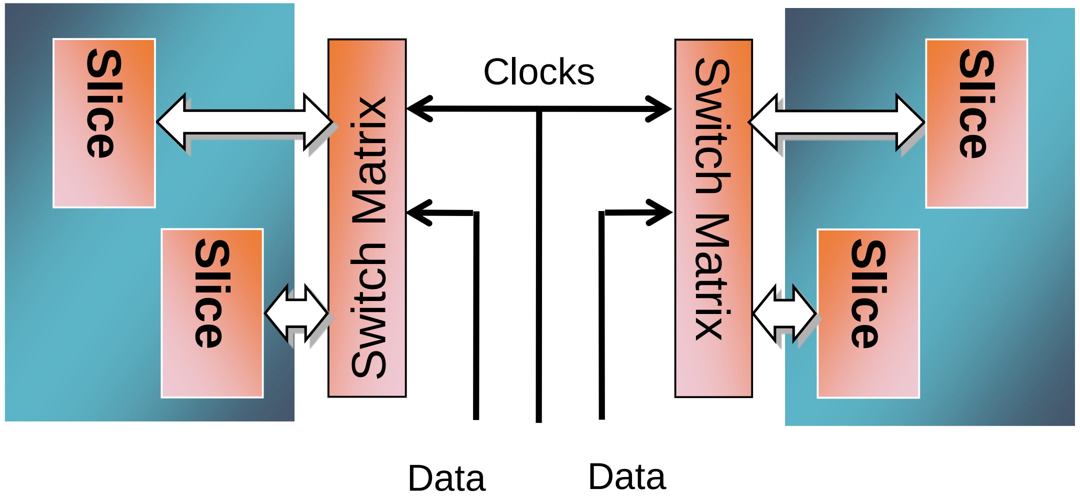


2 Slices in Each CLB

- Each Slice has Two LUTs and Two Flipflops

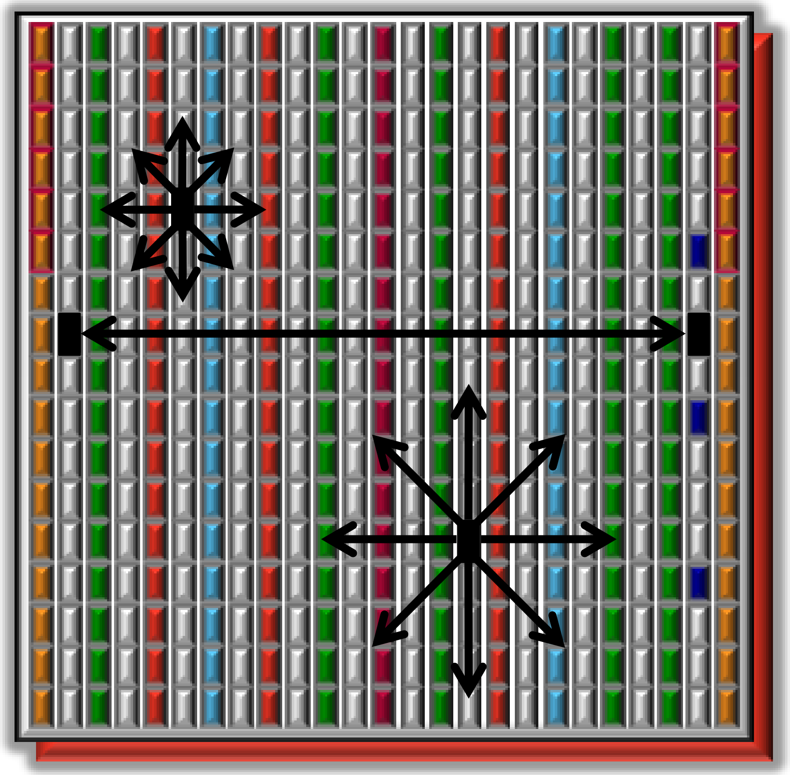
How do CLBs connect with each Other

- Pairs of CLBs are arranged symmetrically
- Connect via Switch matrix



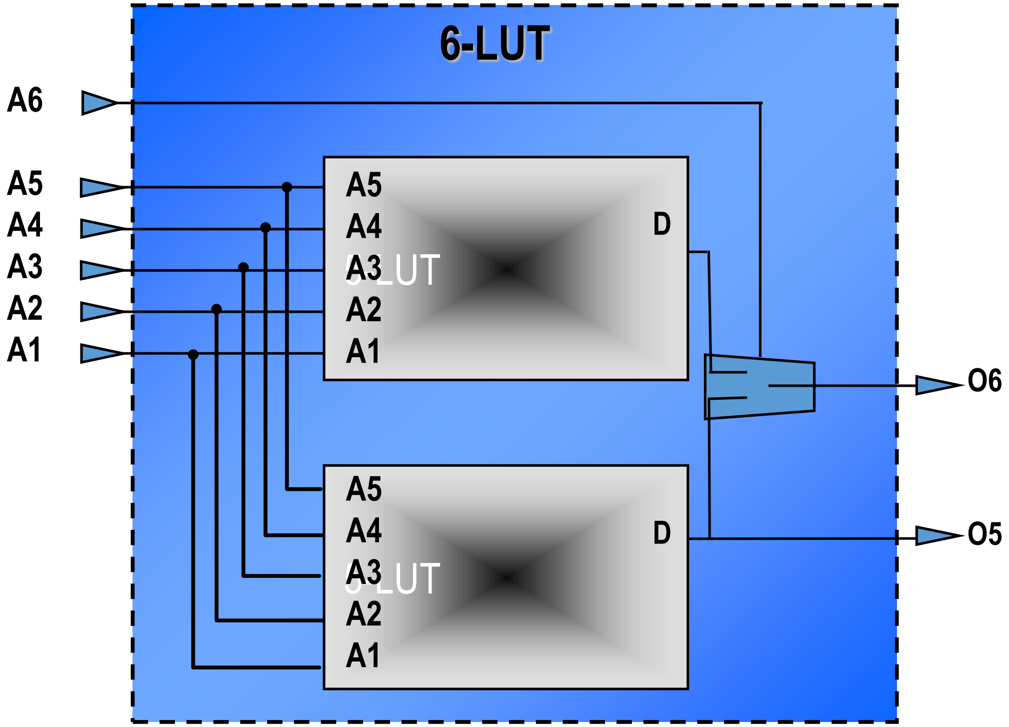
Fabric Routing

- Connections between CLBs and other resources use the fabric routing resources
  - Routing lines connect to the switch matrices adjacent to the resources
- Routes connect resources vertically, horizontally, and diagonally
- Routes have different spans
  - Horizontal: Single, Dual, Quad, Long (12)
  - Vertical: Single, Dual, Hex, Long (18)
  - Diagonal: Single, Dual, Hex



Different Architectures:
6 Input LUTs

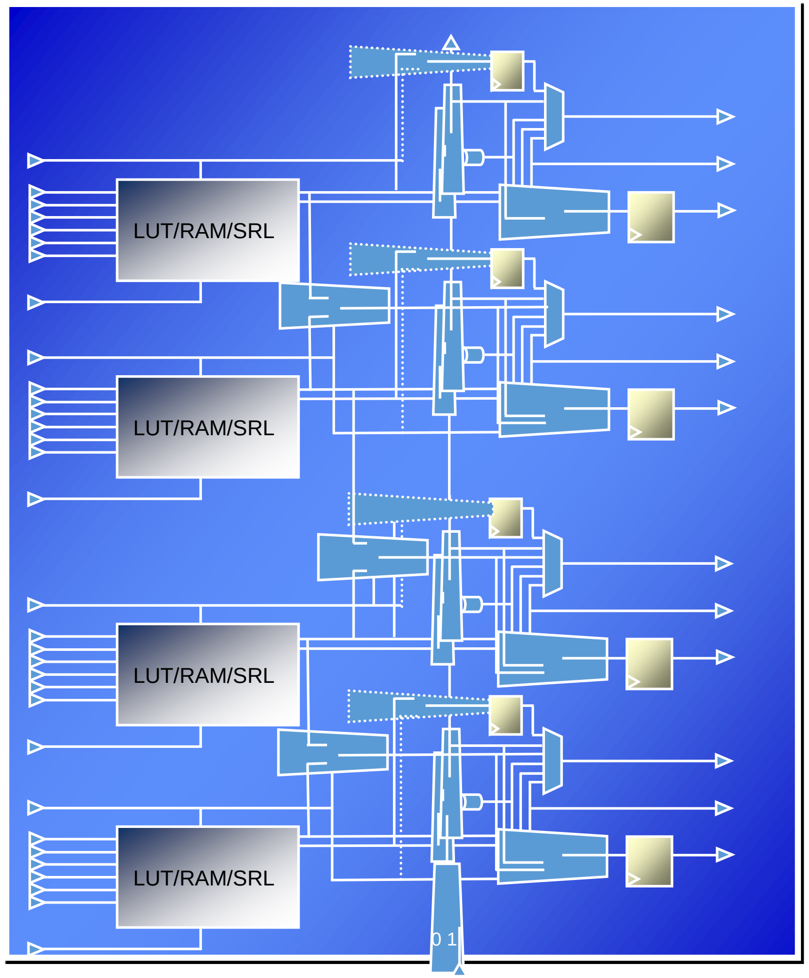
- 6-input LUT can be two 5-input LUTs with common inputs
  - Minimal speed impact to a 6-input LUT
  - One or two outputs
  - Any function of six variables or two independent functions of five variables



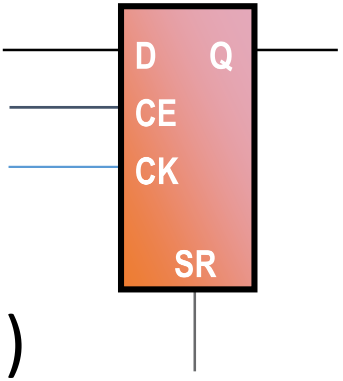
Different Architectures: Slice Structure with 4 LUTs

- Four six-input Look Up Tables (LUT)
- Wide multiplexers
- Carry chain
- Four flip-flop/latches
- Four additional flip-flops

- The implementation tools (MAP) are responsible for packing slice resources into the slice



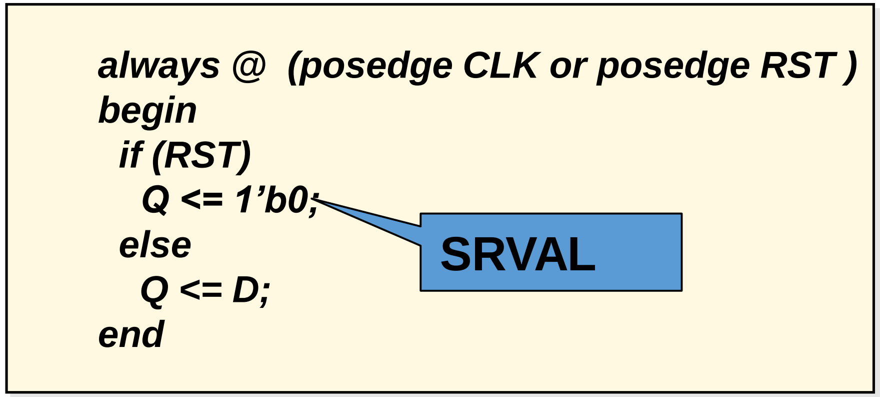
More Detailed Look at Flip Flops



- All flip-flops are D type
- All flip-flops have a single clock input (CLK)
  - Clock can be inverted at the slice boundary
- All flip-flops have an active high chip enable (CE)
- All flip-flops have an active high SR input
  - Input can be synchronous or asynchronous, as determined by the configuration bit stream
  - Sets the flip-flop value to a pre-determined state, as determined by the configuration bit stream

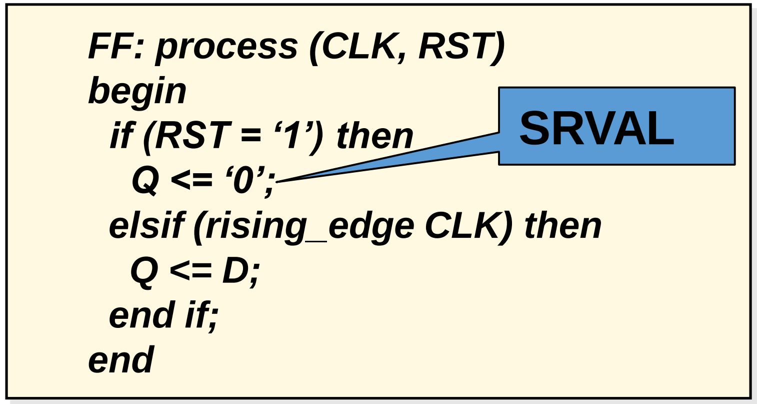
Asynchronous Reset

- To infer asynchronous resets, the reset signal must be in the sensitivity list of the process
- Output takes reset value immediately
    - Even if clock is not present
- SRVAL attribute is determined by reset value in RTL code



```
always @  (posedge CLK or posedge RST )
begin
  if (RST)
    Q <= 1’b0;
  else
    Q <= D;
end
```

SRVAL

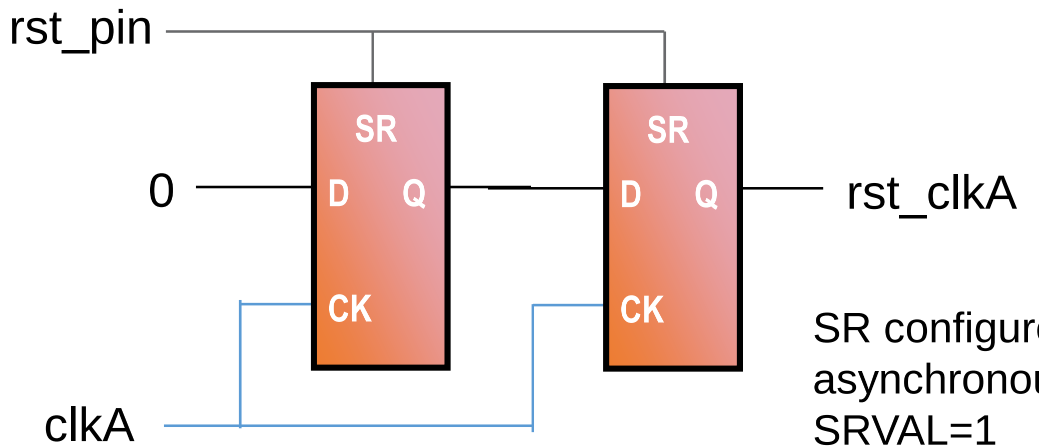


```
FF: process (CLK, RST)
begin
  if (RST = ‘1’) then
    Q <= ‘0’;
  elsif (rising_edge CLK) then
    Q <= D;
  end if;
end
```

SRVAL

Using Asynchronous Resets

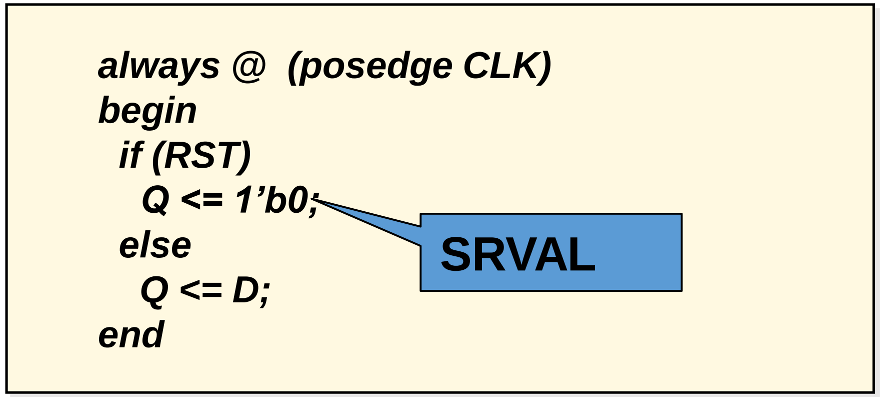
- Deassertion of reset should be synchronous to the clock
- Not synchronizing the deassertion of reset can create problems
  - Flip-flops can go metastable
  - Not all flip-flops are guaranteed to come out of reset on the same clock
- Use a reset bridge to synchronize reset to each domain



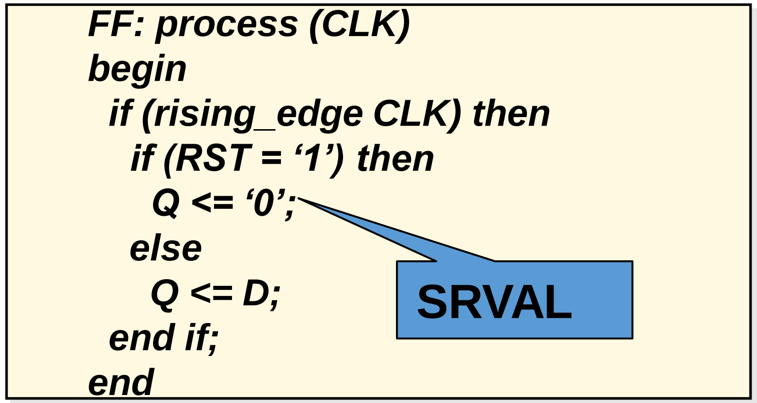
SR configured as asynchronous, SRVAL=1

Synchronous Reset

- A synchronous reset will not take effect until the first active clock edge after the assertion of the RST signal
- The RST pin of the flip-flop is a regular timing path endpoint
  - The timing path ending at the RST pin will be covered by a PERIOD constraint on the clock



```
always @  (posedge CLK)
begin
  if (RST)
    Q <= 1'b0;
  else
    Q <= D;
end
```



```
FF: process (CLK)
begin
  if (rising_edge CLK) then
    if (RST = '1') then
      Q <= '0';
    else
      Q <= D;
  end if;
end
```

Chip Enable

- All flip-flops in the 7 series FPGAs have a chip enable (CE) pin
  - Active high, synchronous to CLK
  - When asserted, the flip-flop clocks in the D input
  - When not asserted, the flip-flop holds the current value
- Inferred naturally from RTL code

```
always @  (posedge CLK )
begin
  if (CE)
    Q <= D;
end
```

```
FF: process (CLK)
begin
  if (rising_edge CLK) then
    if (CE = ‘1’) then
      Q <= D;
    end if;
  end if;
end
```

LUTs can also be used as RAM

Single Port	Dual Port	Simple Dual Port	Quad Port
32x2	32x2**D**	32x6**SDP**	32x2**Q**
32x4	32x4**D**	64x3**SDP**	64x1**Q**
32x6	64x1**D**		
32x8	64x2**D**		
64x1	128x1**D**		
64x2			
64x3			
64x4			
128x1			
128x2			
256x1			

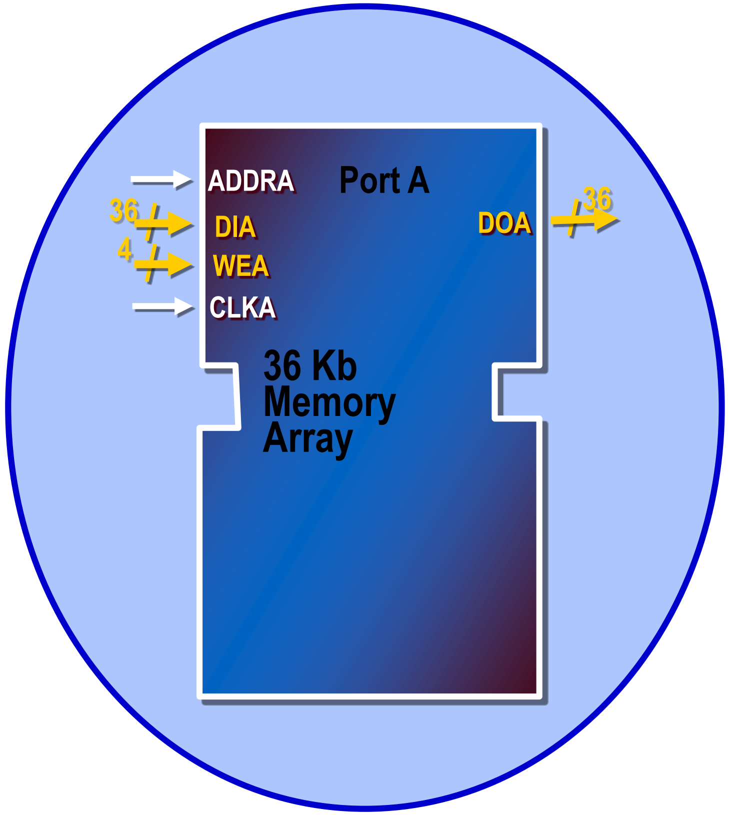
Each port has independent address inputs

- Uses the same storage that is used for the look-up table function
- Synchronous write, asynchronous read
  - Can be converted to synchronous read using the flip-flops available in the slice
- Various configurations
  - Single port
    - One LUT6 = 64x1 or 32x2 RAM
    - Cascadable up to 256x1 RAM
  - Dual port (**D**)
    - 1 read / write port + 1 read-only port
  - Simple dual port (**SDP**)
    - 1 write-only port + 1 read-only port
  - Quad-port (**Q**)
    - 1 read / write port + 3 read-only ports

Block RAMs
(In built Memory)

Single-Port Block RAM

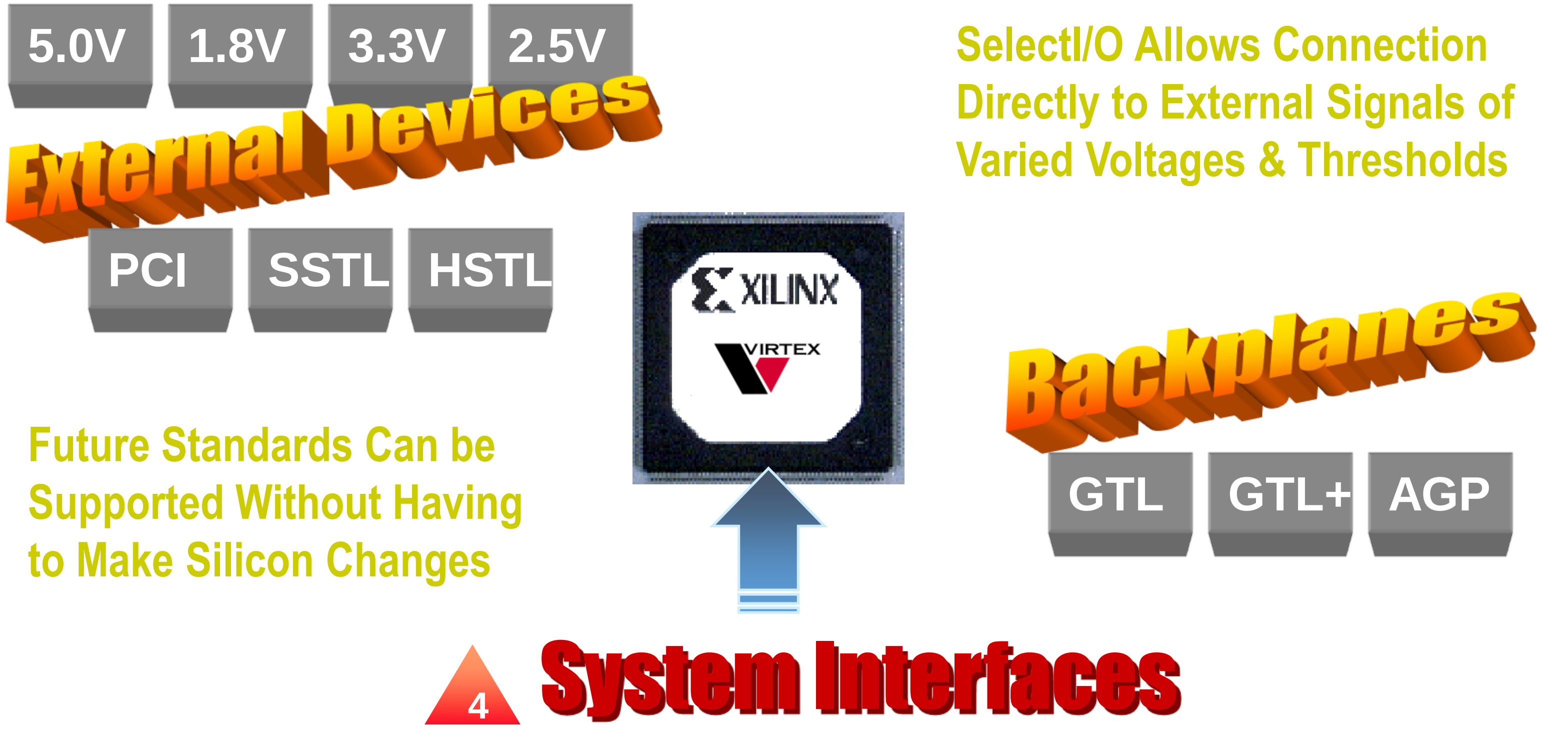
- Single read/write port
  - Clock: CLKA
  - Address: ADDRA
  - Write enable: WEA
  - Write data: DIA
  - Read data: DOA
- 36-kbit configurations
  - 32k x 1, 16k x 2, 8k x 4, 4k x 9, 2k x 18, 1k x 36
- 18-kbit configurations
  - 16k x 1, 8k x 2, 4k x 4, 2k x 9, 1k x 18, 512 x 36
- Configurable write mode
  - WRITE_FIRST: Data written on DIA is available on DOA
  - READ_FIRST: Old contents of RAM at ADDRA is presented on DOA
  - NO_CHANGE: The DOA holds its previous value (saves power)



Summary of Block RAM Configurations

	18kbit	36kbit	
Single Port	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18	32k x 1, 16Kx2, 8Kx4, 4Kx9, 2Kx18, 1Kx36	▪ 1 read/write port ▪ Read OR write in 1 cycle
True Dual Port	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18	32Kx1, 16Kx2, 8Kx4, 4Kx9, 2Kx18, 1Kx36	▪ Two fully independent read/write ports ▪ Any two operations in 1 cycle
Simple Dual Port	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18, 512x36	32K x 1, 16Kx2, 8Kx4, 4Kx9, 2Kx18, 1Kx36, 512x72	▪ 1 read port and 1 write port ▪ Read AND write in 1 cycle

SelectI/O



SelectI/O Allows Connection Directly to External Signals of Varied Voltages & Thresholds

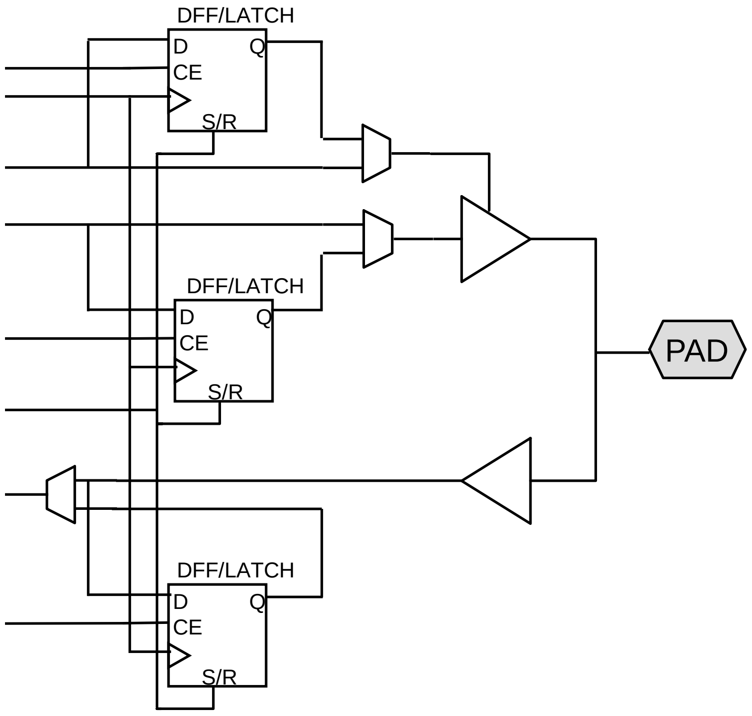
Future Standards Can be Supported Without Having to Make Silicon Changes

SelectI/O

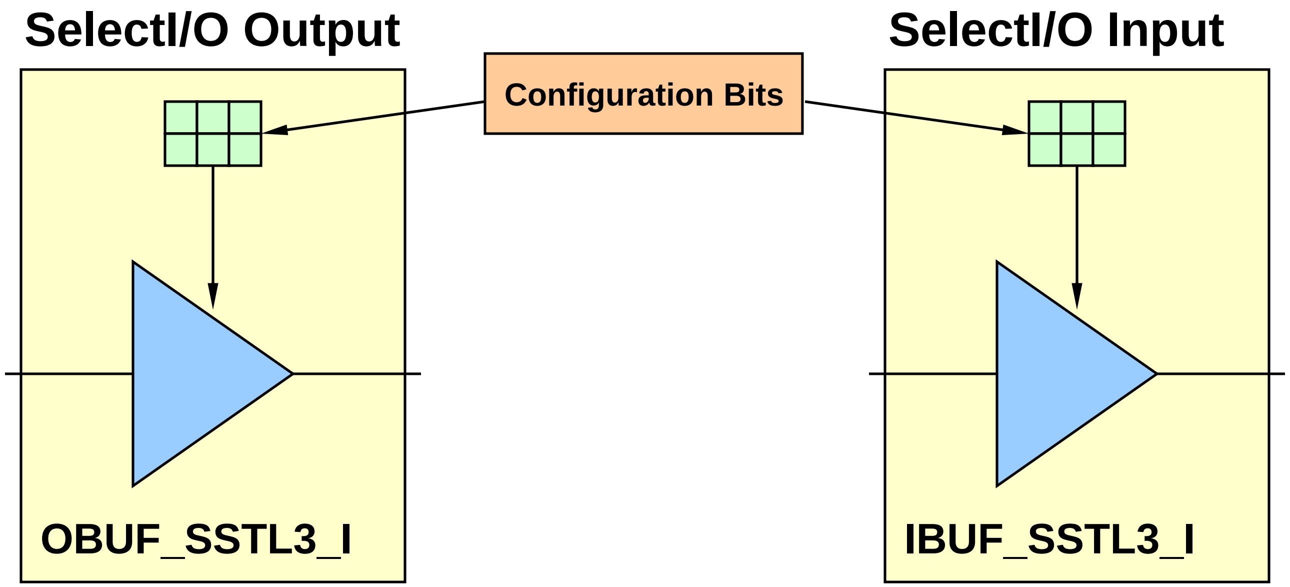
- Allows Connection & Use of a Wide Variety of Devices
  - Processors, Memory, Bus Specific Standards, Mixed Signal...
  - Provides Industry Standard IEEE/JDEC I/O Standards
  - Maximizes Speed/Noise Tradeoff - Use Only What is Needed
  - Can Connect to or Create High Performance Backplanes
    - PCI, GTL$^+$, HSTL
    - DIY - Virtex Based Backplane Design in Progress
- Define I/O by Simply Placing Desired Input And/Or Output Buffers Into the Design
  - Special IBUF and OBUF Components Provided in Schematic Based and HDL Based Design Flows
  - For Example: SSTL3, Class I Output Buffer - OBUF_SSTL3_I

Simplified IOB Structure

- Fast I/O Drivers
- Separate Registers for Input, Output & Three-State Control
    - Asynchronous Set or Reset Available on Each Flip-flop
    - Common Clock, Separate Clock Enables
- Programmable Slew Rate, Pullup, Input Delay, Etc
- Selectable I/O Standard Support
- Supported Standards List can be Updated After Testing



How It Works



SSTL3 Class1
Output Driver

SSTL3 Class1
Input Receiver

Xilinx 7 Series

ARTIX 7™

Lowest Power
and Cost

Compared to Spartan-6

- 30% more performance
- Lower system cost
- 50% less power
- 30% smaller footprint

KINTEX 7™

Industry's Best
Price-Performance
"New Class of FPGA"

Compared to Virtex-6

- Comparable performance with 50% lower cost for 2x better price-performance
- 50% less power

Compared to Spartan-6

- 3.3x larger
- Over 2x performance with 4x transceiver speed
- Superior price-performance

VIRTEX 7®

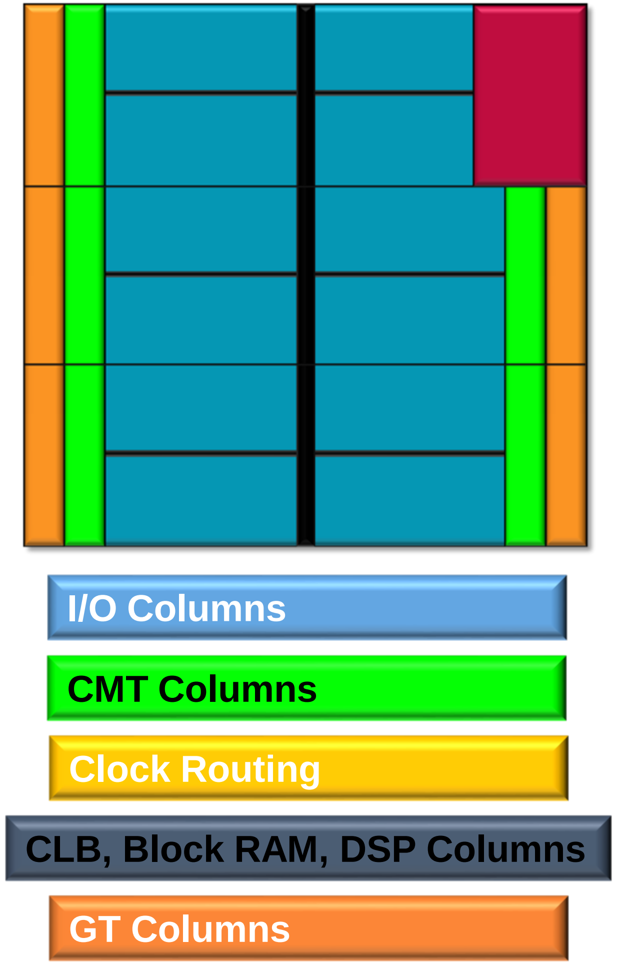
Industry's Highest
System Performance
and Capacity

Compared to Virtex-6

- 2.5x larger (2M LCs)
- 50% higher performance
- 50% lower power
- 2x line rate (28 Gb/s)
- Similar EasyPath™ cost reduction

7 Series FPGA Layout

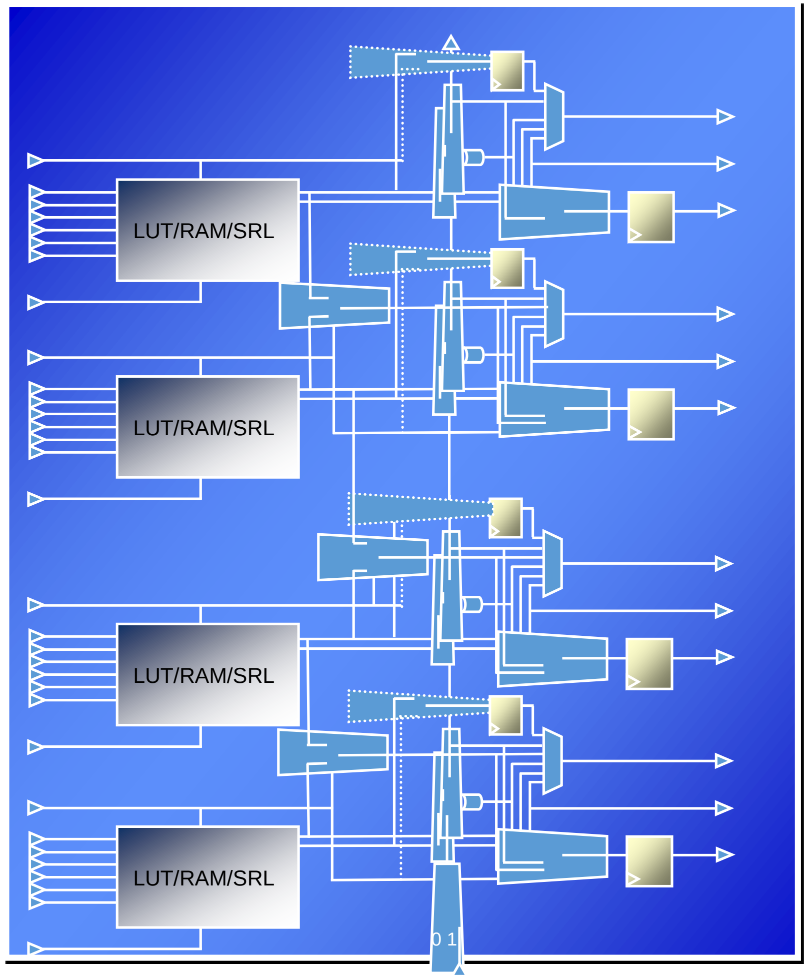
- Similar Floorplan to Virtex-6 FPGAs
  - Provides easy migration to 7 series FPGAs
- CMT columns moved from center of device to adjacent to I/O columns
  - No more inner vs. outer column performance difference
  - Support for higher performance interfaces
- Only one I/O column per half device
  - Uniform skew from center of device
- GT columns replace I/O and CMT in smaller devices
  - GT columns not always present



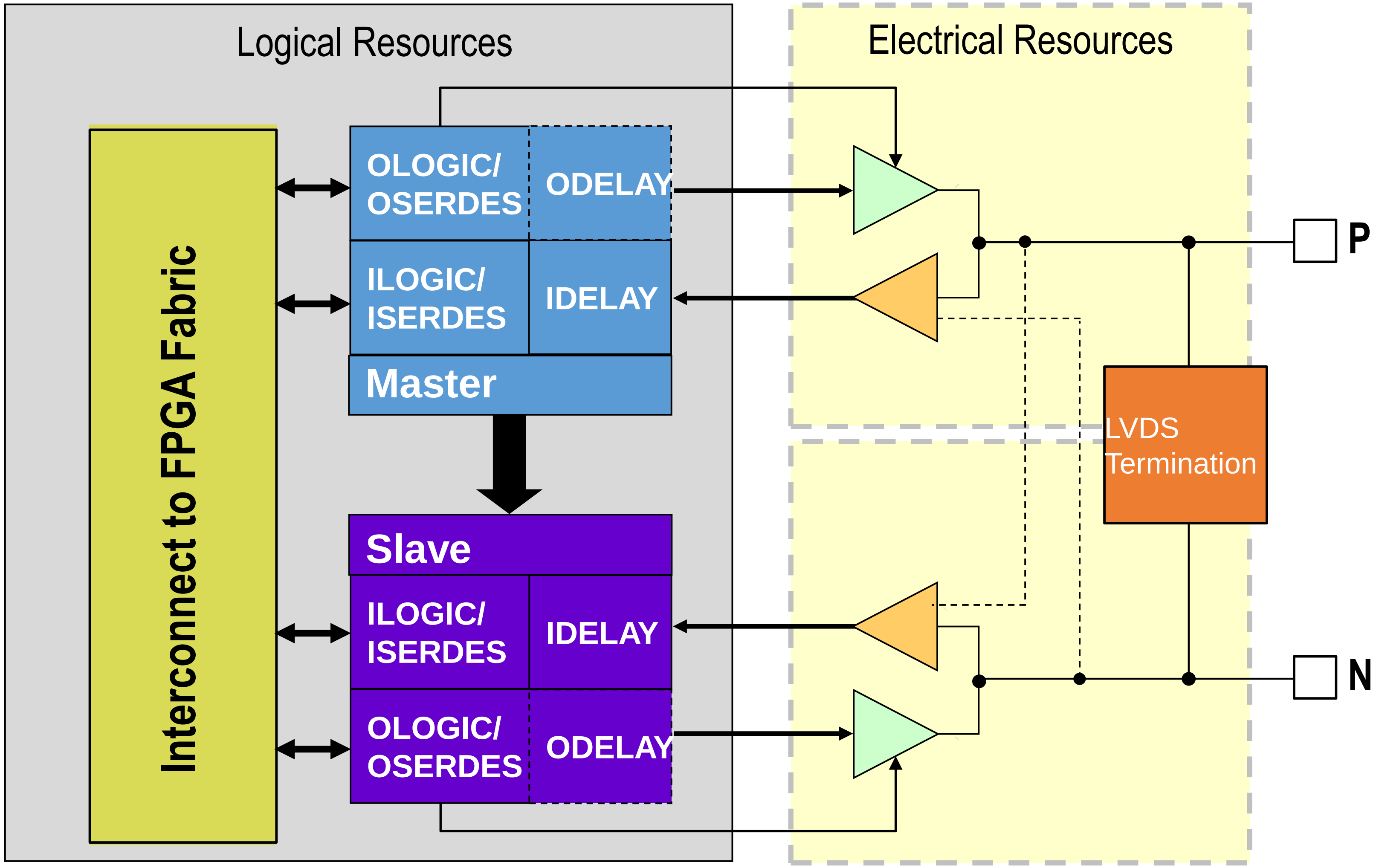
7 Series Slice Structure

- Four six-input Look Up Tables (LUT)
- Wide multiplexers
- Carry chain
- Four flip-flop/latches
- Four additional flip-flops

- The implementation tools (MAP) are responsible for packing slice resources into the slice

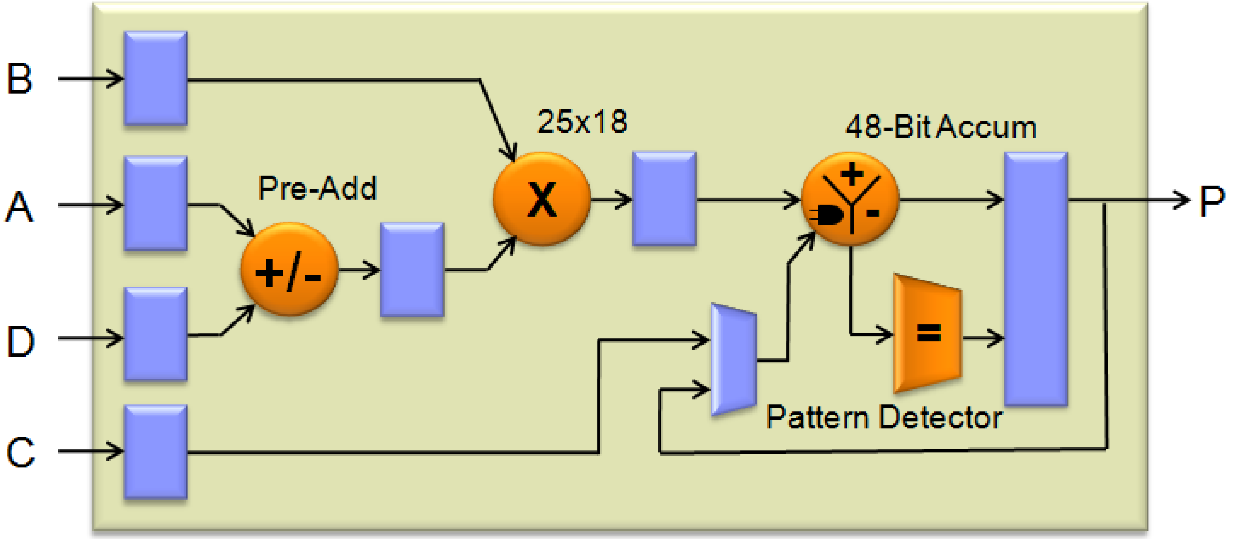


7-Series I/O Block Diagram



7 Series FPGAs DSP

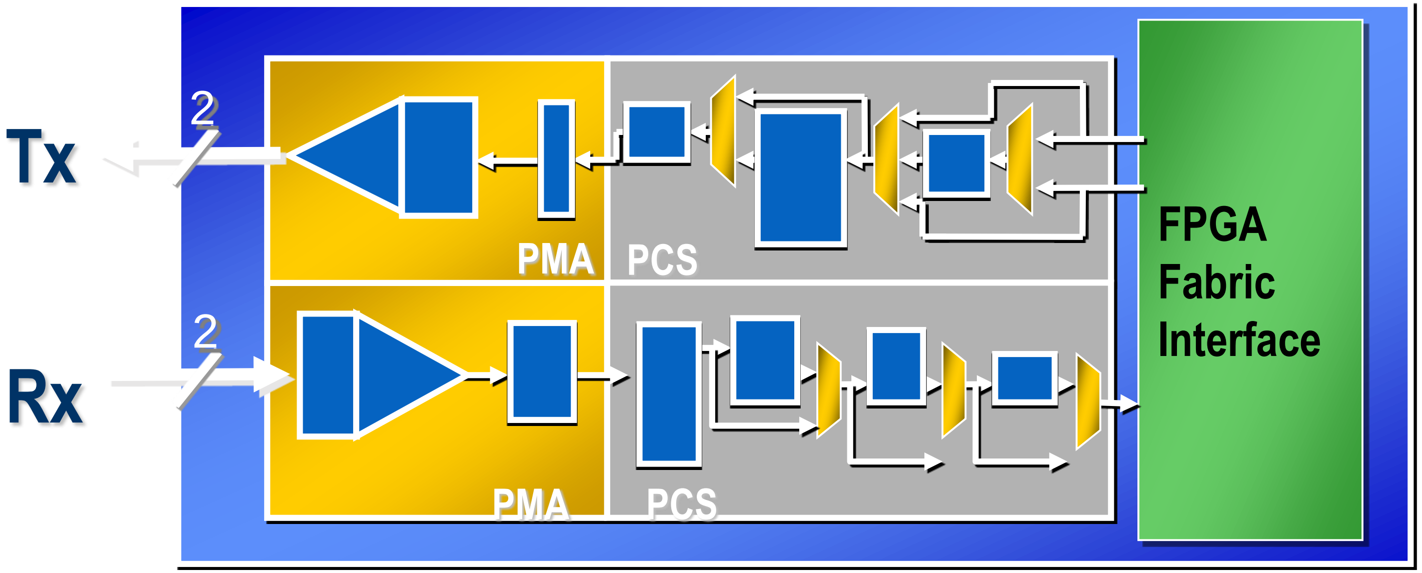
- 7 series FPGAs DSP slice 100% based on Virtex-6 FPGA DSP48E1
  - 25x18 multiplier
  - 25-bit pre-adder
  - Flexible pipeline
  - Cascade in and out
  - Carry in and out
  - 96-bit MACC
  - SIMD support
  - 48-bit ALU
  - Pattern detect
  - 17-bit shifter
  - Dynamic operation (cycle by cycle)



Programmable Systems Integration

Highly Capable, Dedicated DSP Logic in Every 7 Series FPGA

7-Series Gigabit Transceivers



- Dedicated parallel-to-serial transmitter and serial-to-parallel receiver
  - Unidirectional, differential bit-serial data I/O
  - Integrated PLL-based Clock and Data Recovery (CDR)
- Parallel interface to the FPGA internal fabric
  - Width varies by family, protocol, and line rate from 8 to 40 bits
- Serial interface to the printed circuit board (differential signaling)
  - Differential Current Mode Logic (CML)
  - Two traces for the transmitter and two traces for the receiver; removes common-mode noise