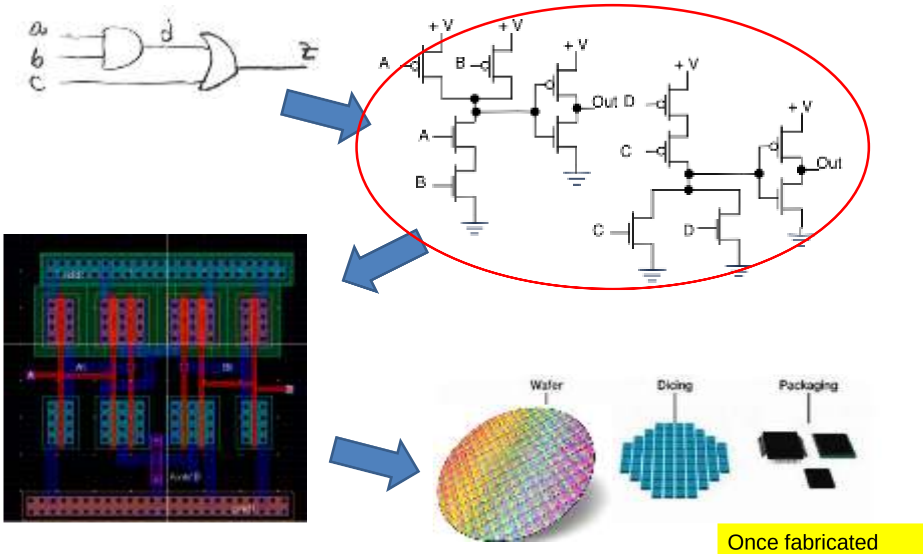
A Light Introduction to FPGAs (it’s not too late to learn about)

Cristinel Ababei
Dept. of Electrical and Computer Engineering
Marquette University

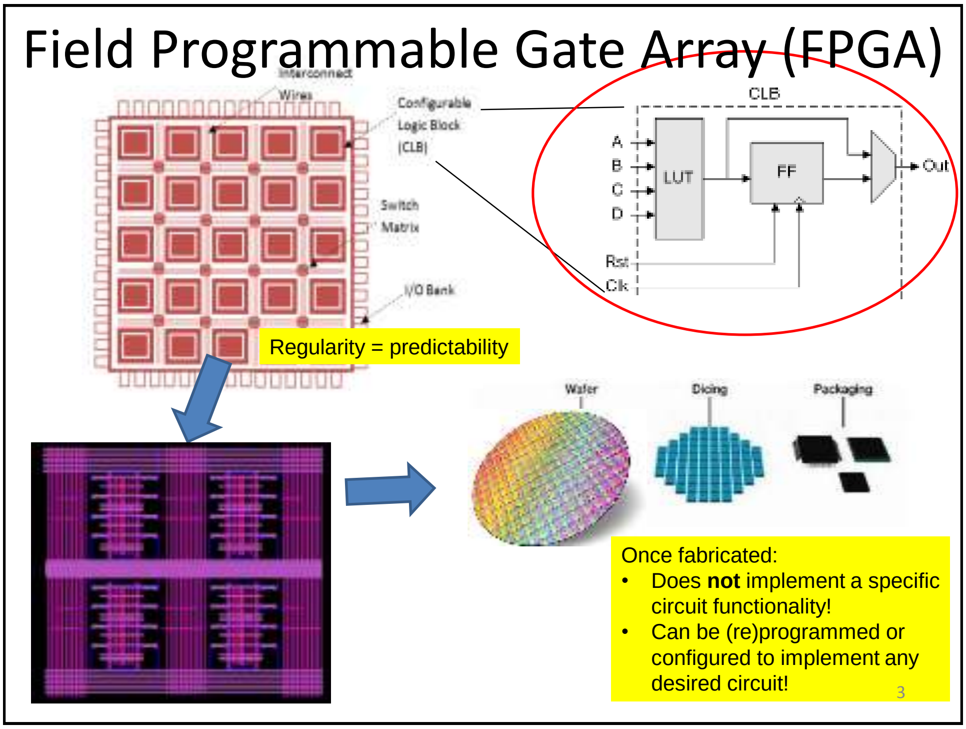
A presentation in the Senior Design Project (SDP) Course in the OPUS College of Engineering, Marquette University, Mar.29.2017

(Credits: the Internet, from where some figures have been borrowed…)

Traditional CMOS Circuits (think of application specific integrated circuits, ASICs)



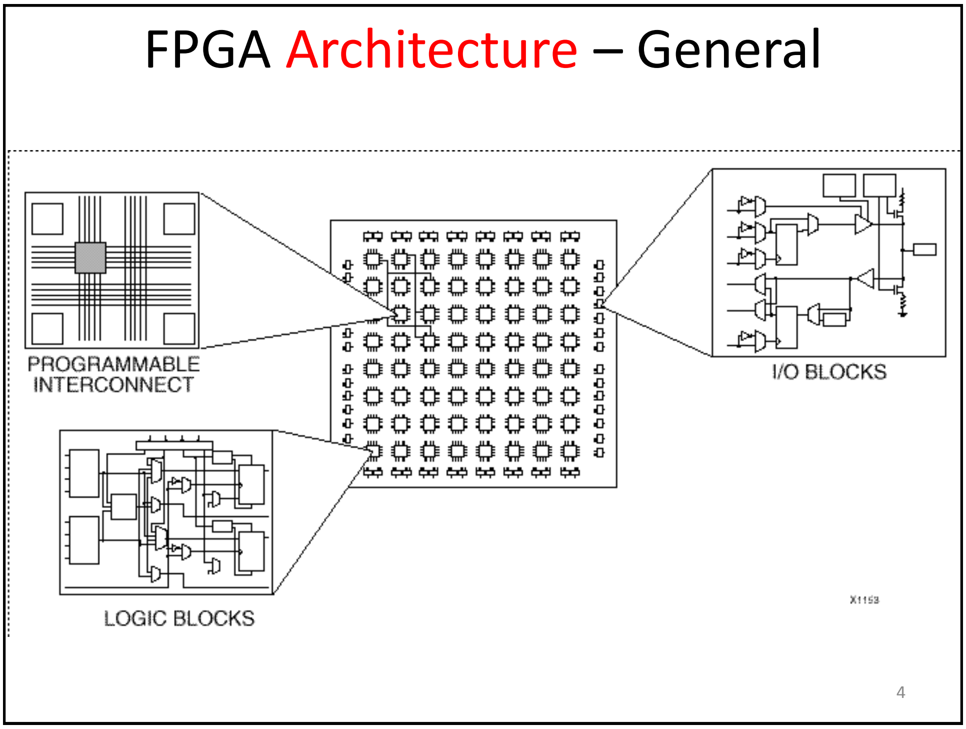
Once fabricated cannot be changed!



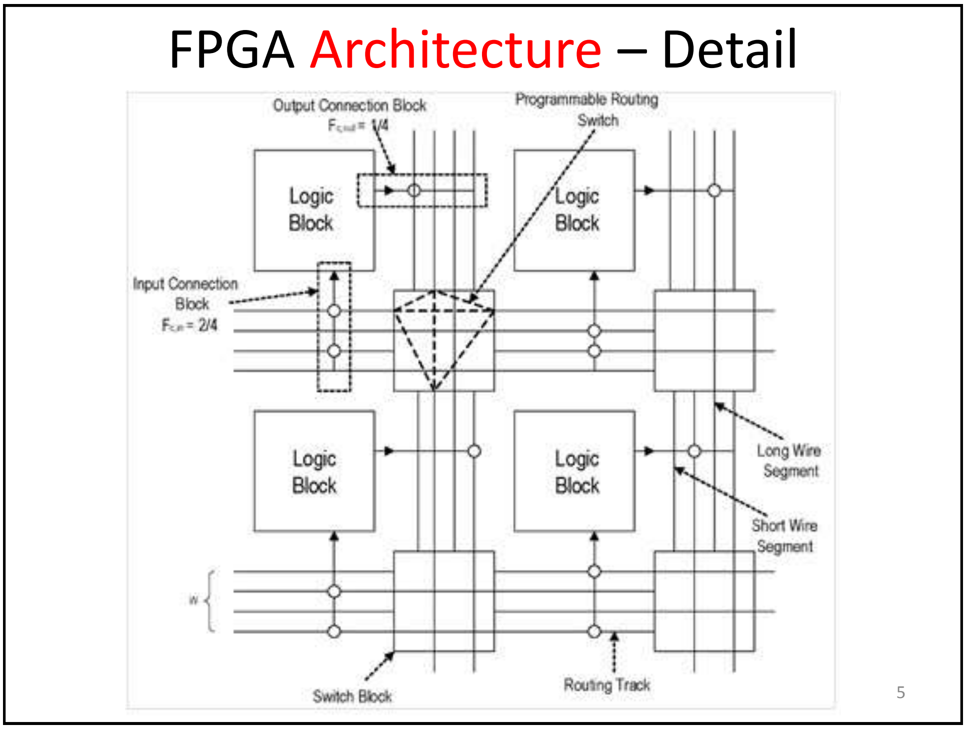
Field Programmable Gate Array (FPGA)

Regularity = predictability

Once fabricated:
- Does **not** implement a specific circuit functionality!
- Can be (re)programmed or configured to implement any desired circuit!

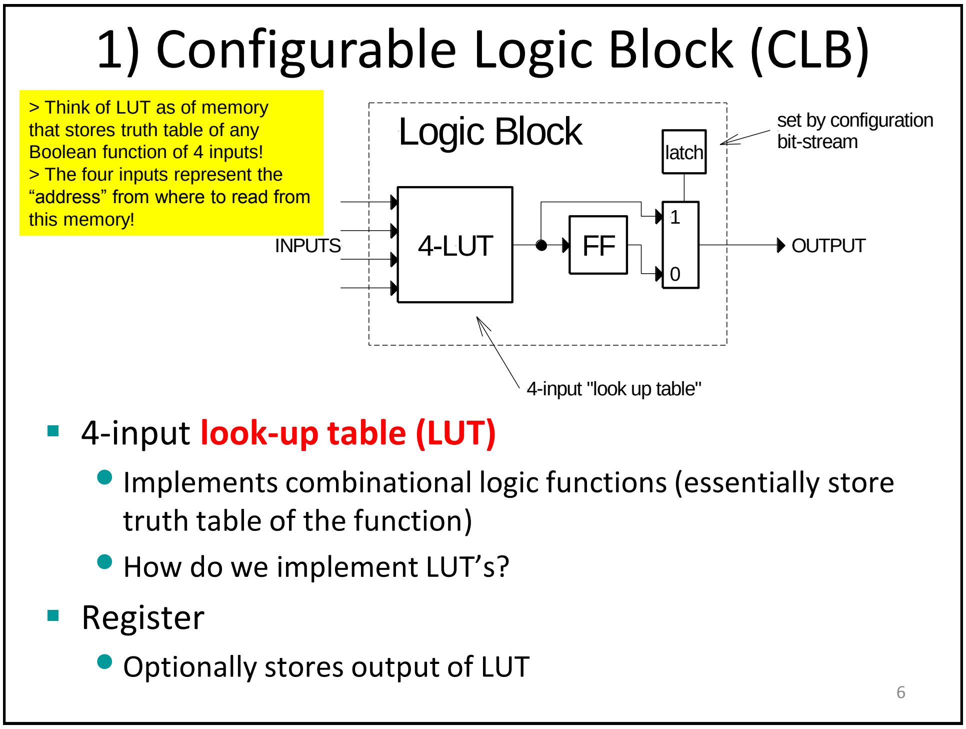


FPGA Architecture – General



FPGA Architecture – Detail

5

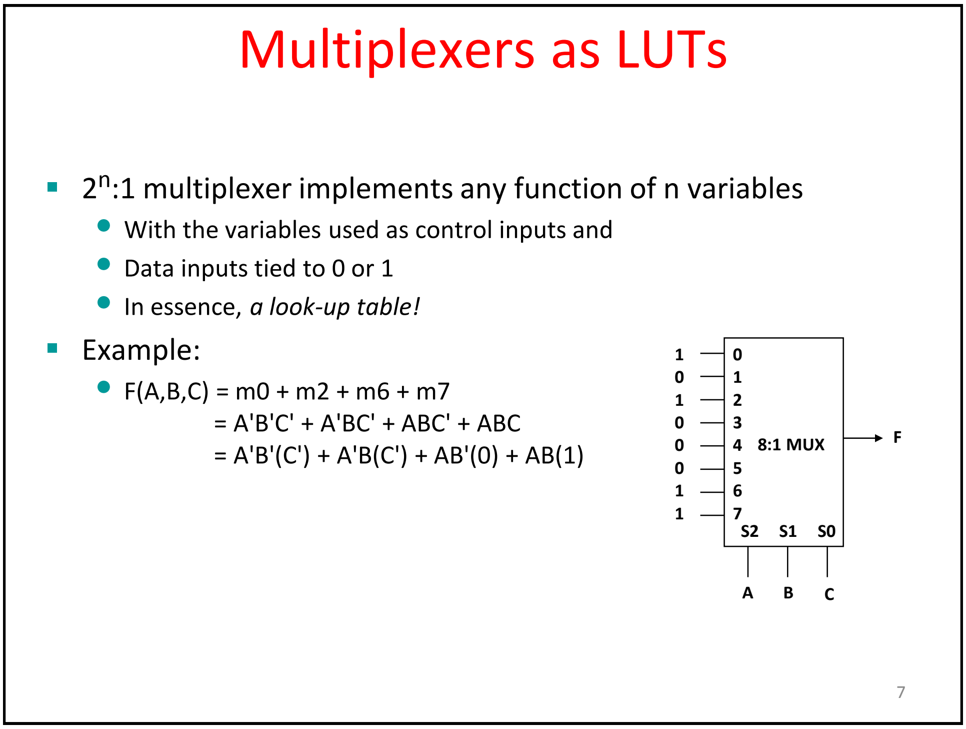


1) Configurable Logic Block (CLB)

> Think of LUT as of memory that stores truth table of any Boolean function of 4 inputs!
> The four inputs represent the "address" from where to read from this memory!

- 4-input **look-up table (LUT)**
  - Implements combinational logic functions (essentially store truth table of the function)
  - How do we implement LUT's?
- Register
  - Optionally stores output of LUT

6



Multiplexers as LUTs

- 2^n:1 multiplexer implements any function of n variables
  - With the variables used as control inputs and
  - Data inputs tied to 0 or 1
  - In essence, *a look-up table!*
- Example:
  - F(A,B,C) = m0 + m2 + m6 + m7
    = A'B'C' + A'BC' + ABC' + ABC
    = A'B'(C') + A'B(C') + AB'(0) + AB(1)

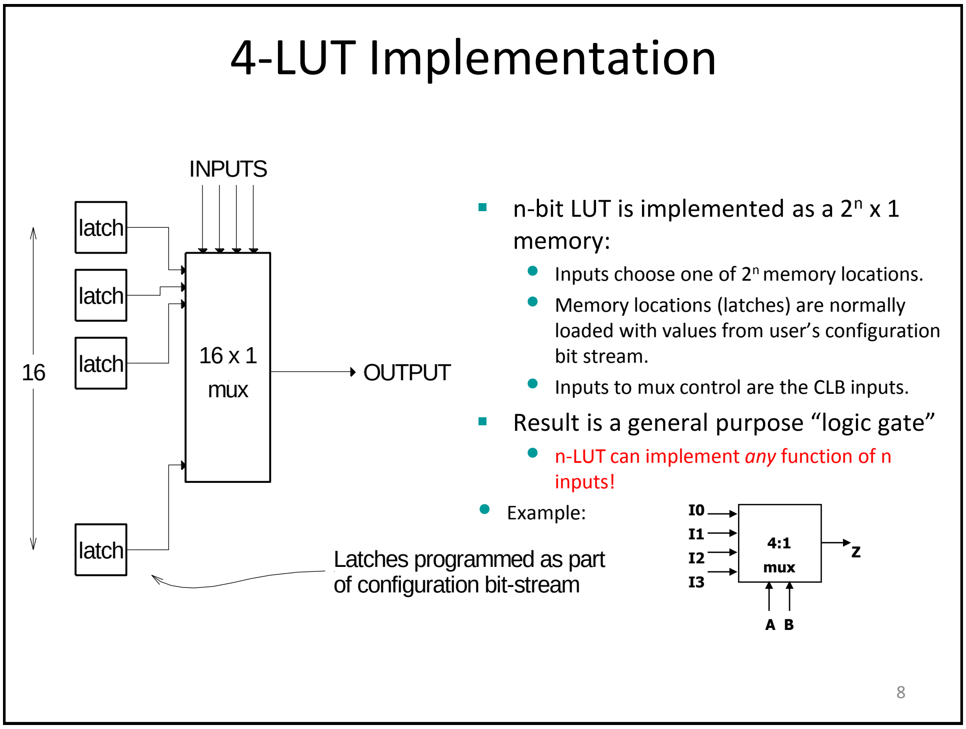
1 — 0
0 — 1
1 — 2
0 — 3
0 — 4
0 — 5
1 — 6
1 — 7

8:1 MUX

F

S2 S1 S0

A B C



4-LUT Implementation

INPUTS

latch
latch
latch
latch

16

16 x 1
mux

OUTPUT

Latches programmed as part
of configuration bit-stream

- n-bit LUT is implemented as a 2^n x 1 memory:
  - Inputs choose one of 2^n memory locations.
  - Memory locations (latches) are normally loaded with values from user's configuration bit stream.
  - Inputs to mux control are the CLB inputs.
- Result is a general purpose "logic gate"
  - n-LUT can implement *any* function of n inputs!
- Example:

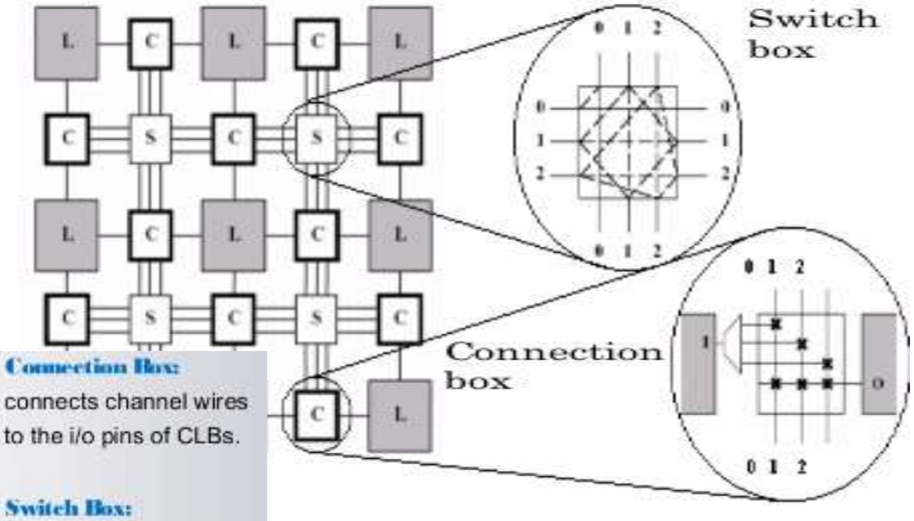
I0
I1
I2
I3

4:1
mux

Z

A B

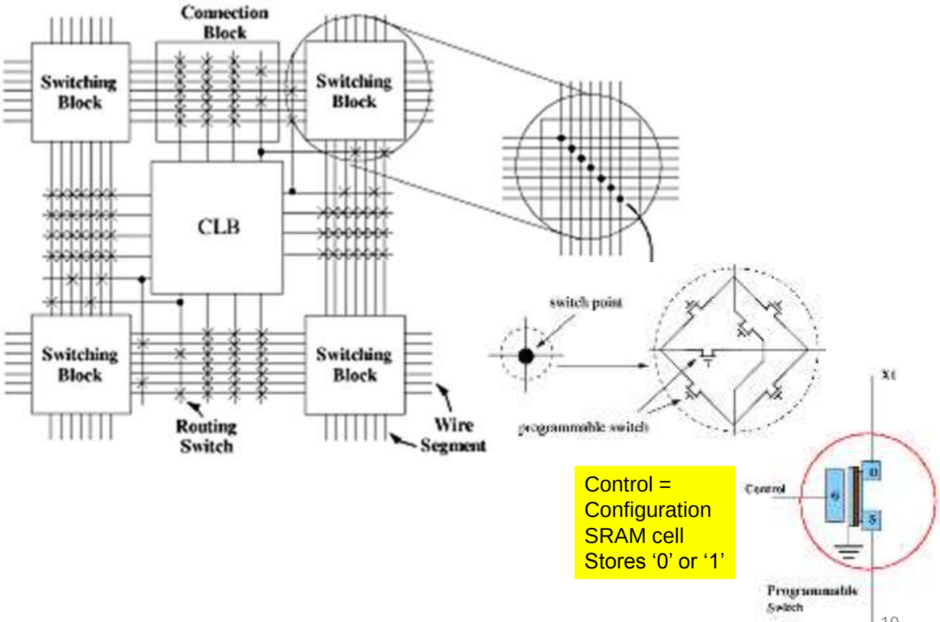
2) Switch Blocks and Connection Blocks



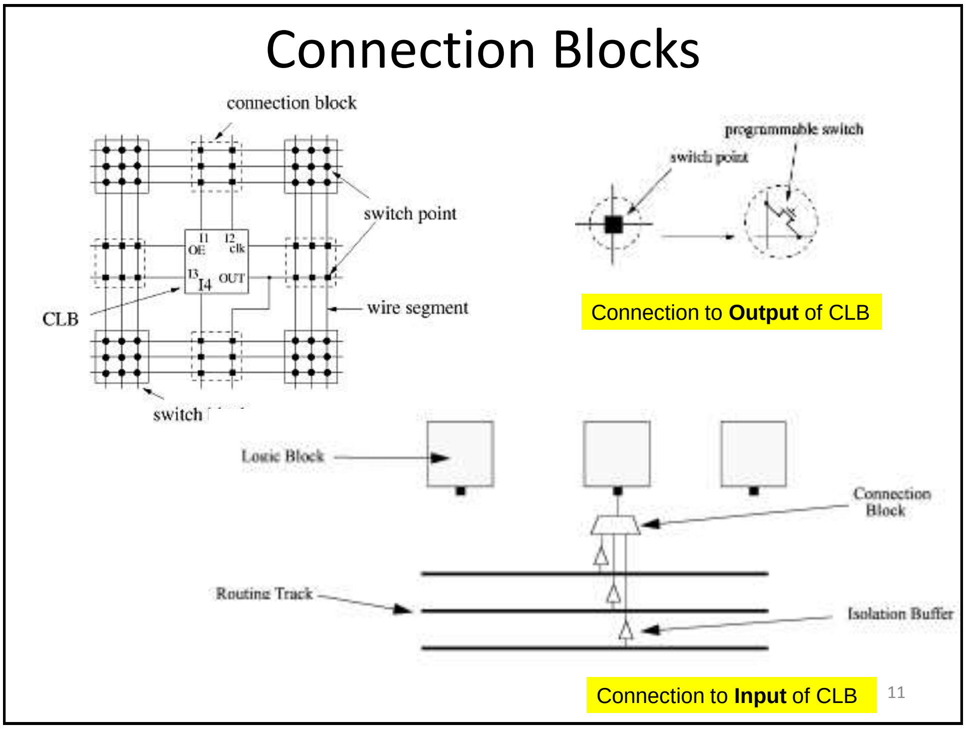
- **Connection Box:**
connects channel wires to the i/o pins of CLBs.

- **Switch Box:**
allow wires to switch between vertical and horizontal wires.

Switch Blocks



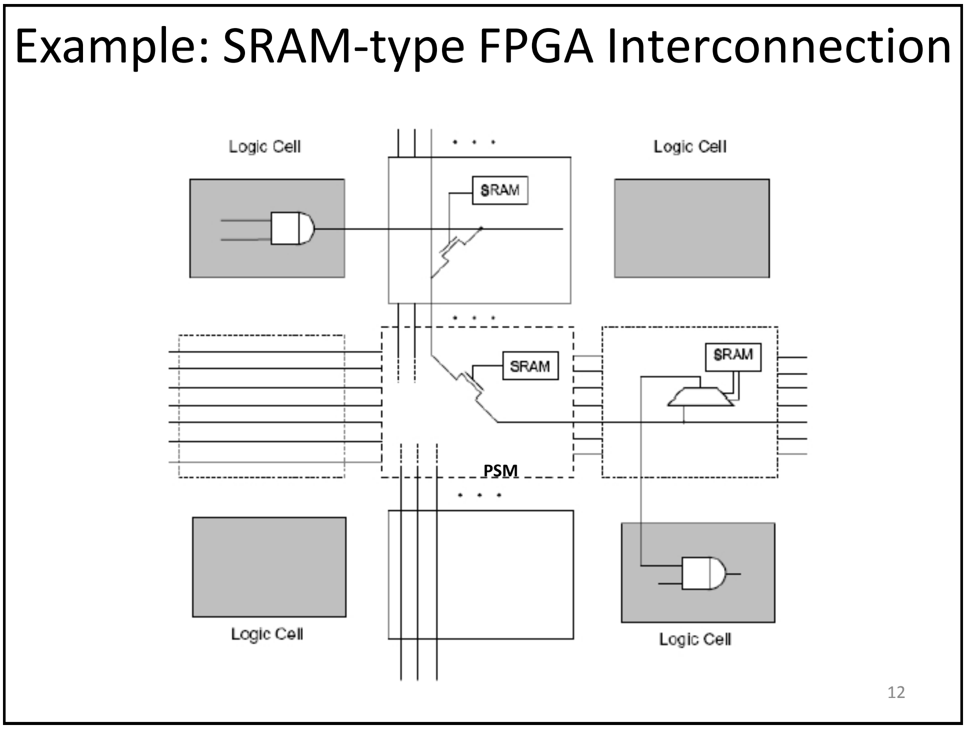
Control = Configuration SRAM cell Stores ‘0’ or ‘1’



Connection Blocks

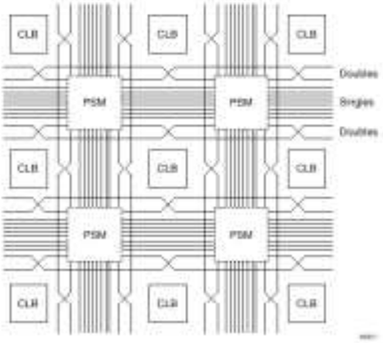
Connection to **Output** of CLB

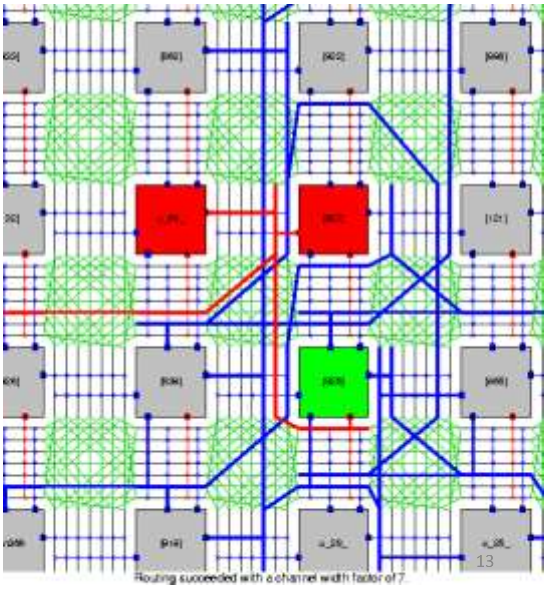
Connection to **Input** of CLB



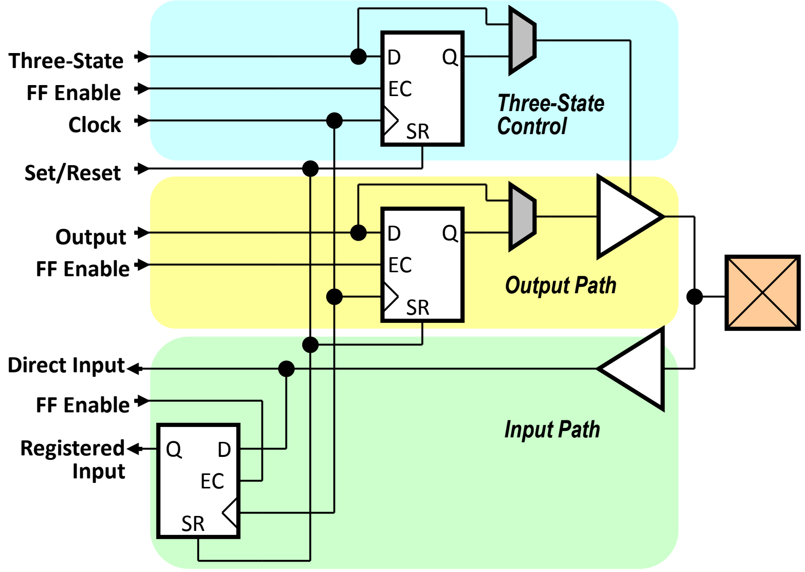
Example: SRAM-type FPGA Interconnection

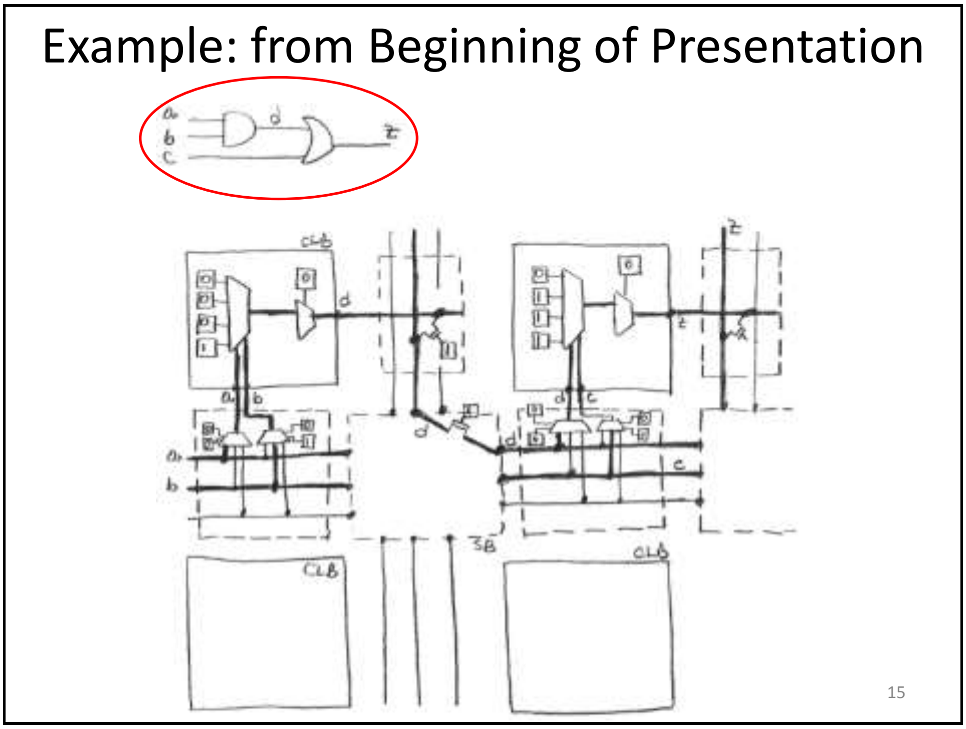
3) Wire Segments




Routing succeeded with a channel width factor of 7.

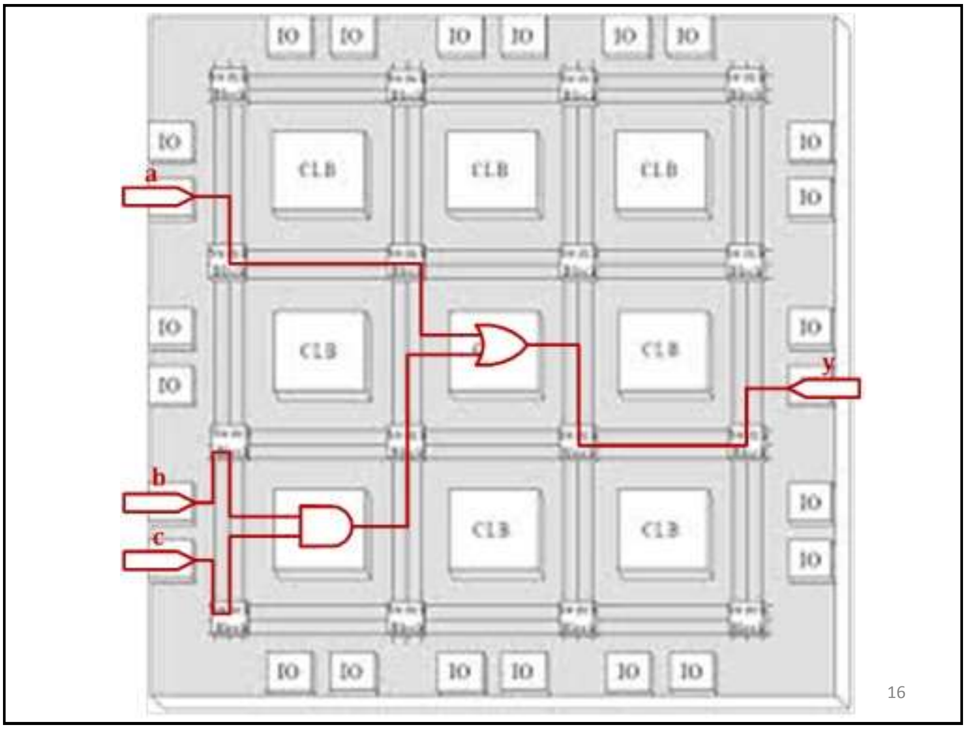
4) I/O Blocks





Example: from Beginning of Presentation

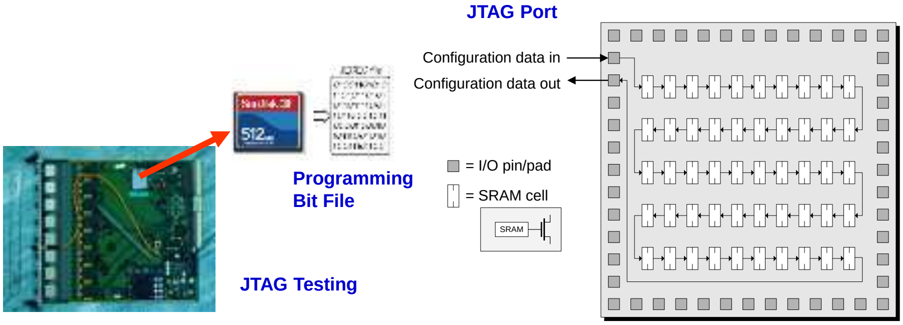
15



16

Configuring an FPGA

- Millions of SRAM cells holding LUTs and Interconnect Routing info
- Volatile Memory. Loses configuration when board power is turned off
- Keep Bit Pattern describing the SRAM cells in non-Volatile Memory, e.g., ROM or SD card
- Configuration takes ~ secs



ASIC vs. FPGA

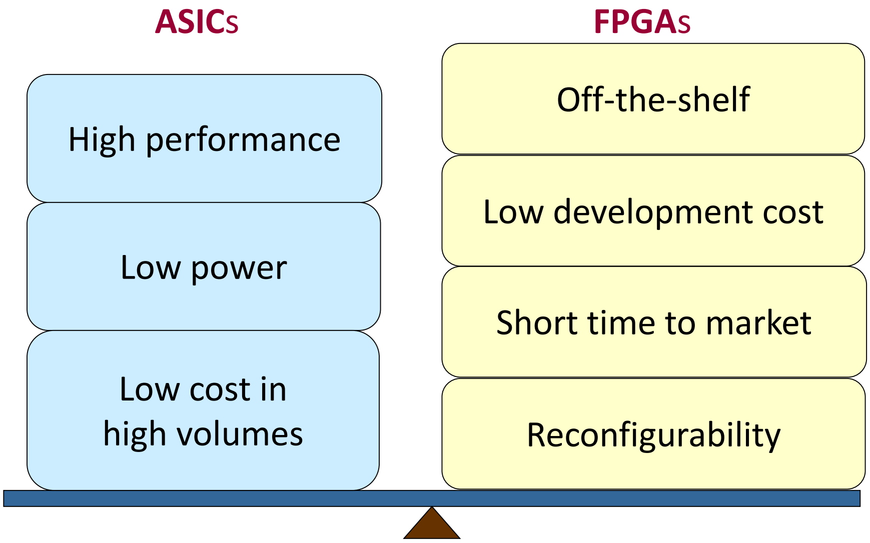
ASIC
Application **S**pecific
Integrated **C**ircuit

- designed all the way from behavioral description to physical layout
- designs must be sent for expensive and time consuming fabrication in semiconductor foundry

FPGA
Field **P**rogrammable
Gate **A**rray

- no physical layout design; design ends with a bitstream used to configure a device
- bought off the shelf and reconfigured by designers themselves

Which way to go?



Applications of FPGAs? (u kidding me?)

- Used to serve as "glue logic" and for prototyping. Now? Everywhere!
  - Communications, software-defined radio, digital signal processing, ASIC prototyping, computer hardware emulation, medical imaging, computer vision, automotive, speech recognition, cryptography, bioinformatics, financial, bitcoin, …
  - https://www.altera.com/products/fpga/arria-series/arria-10/applications.html
  - https://www.xilinx.com/applications.html
  - https://www.xilinx.com/about/customer-innovation/aerospace-and-defense/mars-exploration-rovers.html
  - HW accelerators in datacenter servers (Intel purchased Altera for $16 billion).

Outline

- FPGAs vs. ASICs
- Demo: Edge Detection Circuit
- Summary

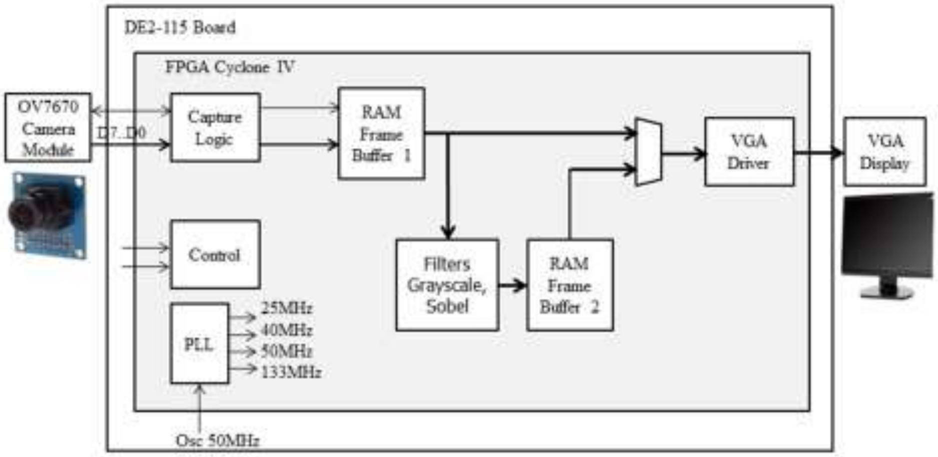
Realtime Edge Detection: From VHDL to FPGA

- Use VHDL (VHSIC hardware description language) to design an edge-detection algorithm, i.e., Sobel operator.
- Edge-detection is a basic algorithm, which is used to build more complex systems in various application domains (computer vision, robotics, medical imaging, etc.).
- Validate the design on an FPGA (field programmable gate array) chip: Cyclone IV E of Altera

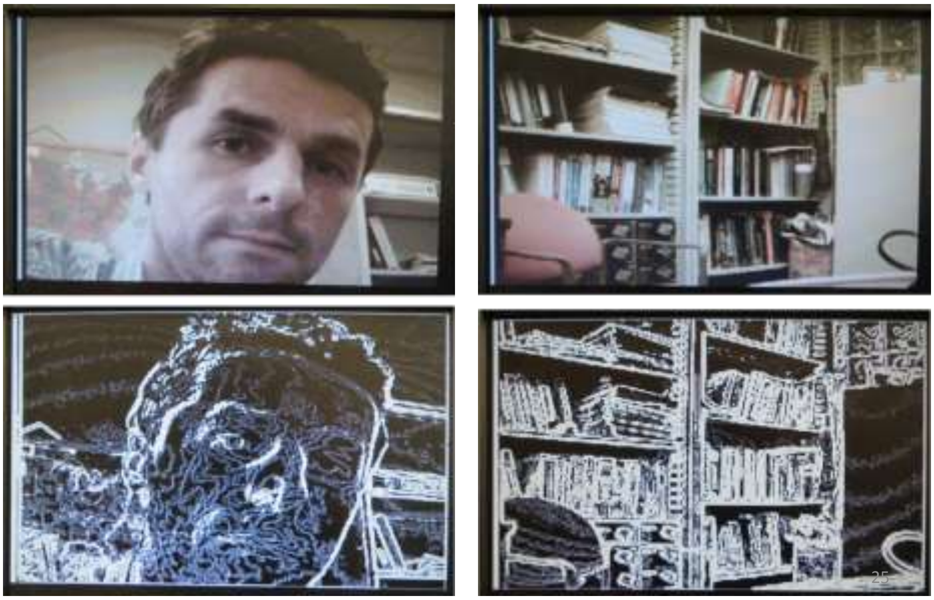
Hardware and Software

- Hardware (HW):
  - DE2-115 FPGA development board (made by Terasic); USD 300
  - Cyclone IV E FPGA (located on board already, made by Altera); USD 12
  - OV7670 camera module (CMOS sensor made by OmniVision); USD 10
- Software (SW):
  - Altera’s Quartus II integrated design environment; Free
  - Programming in VHDL language

Block Diagram



Results



Summary

- FPGAs are more and more prevalent!
- They are here to stay!
- They offer a flexible platform for increasingly complex systems!
- Design automation tools (i.e., CAD tools) take care of the entire design process from VHDL/Verilog → configuration Bitstream file
- See Cris' course: *EECE-4740/5740 Advanced VHDL Design and FPGAs*:
  - http://dejazzer.com/eece4740/index.html