



EECS 151/251A

Spring 2019

**Digital Design and Integrated
Circuits**

Instructor:

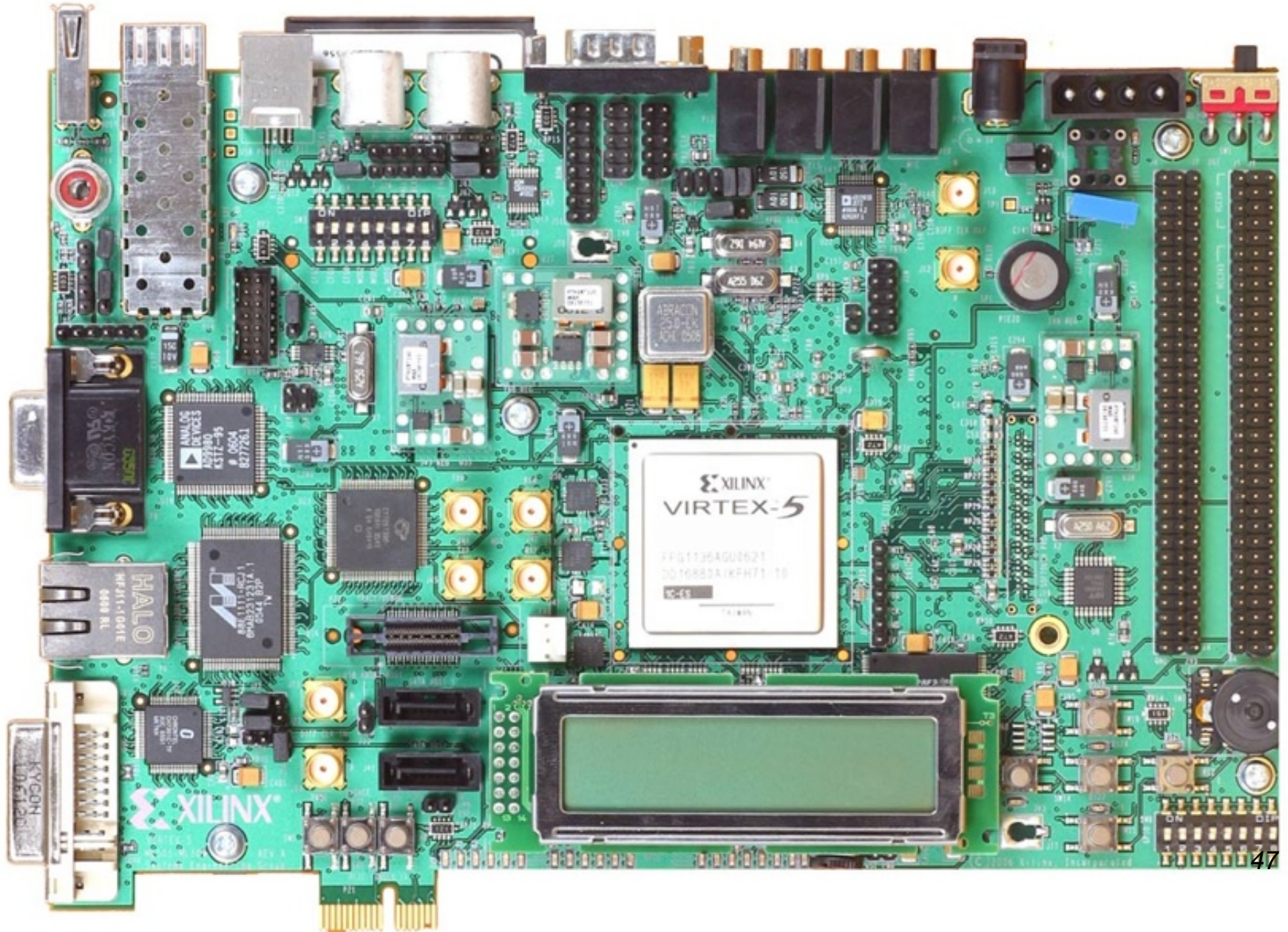
J. Wawrzynek

Lecture 5

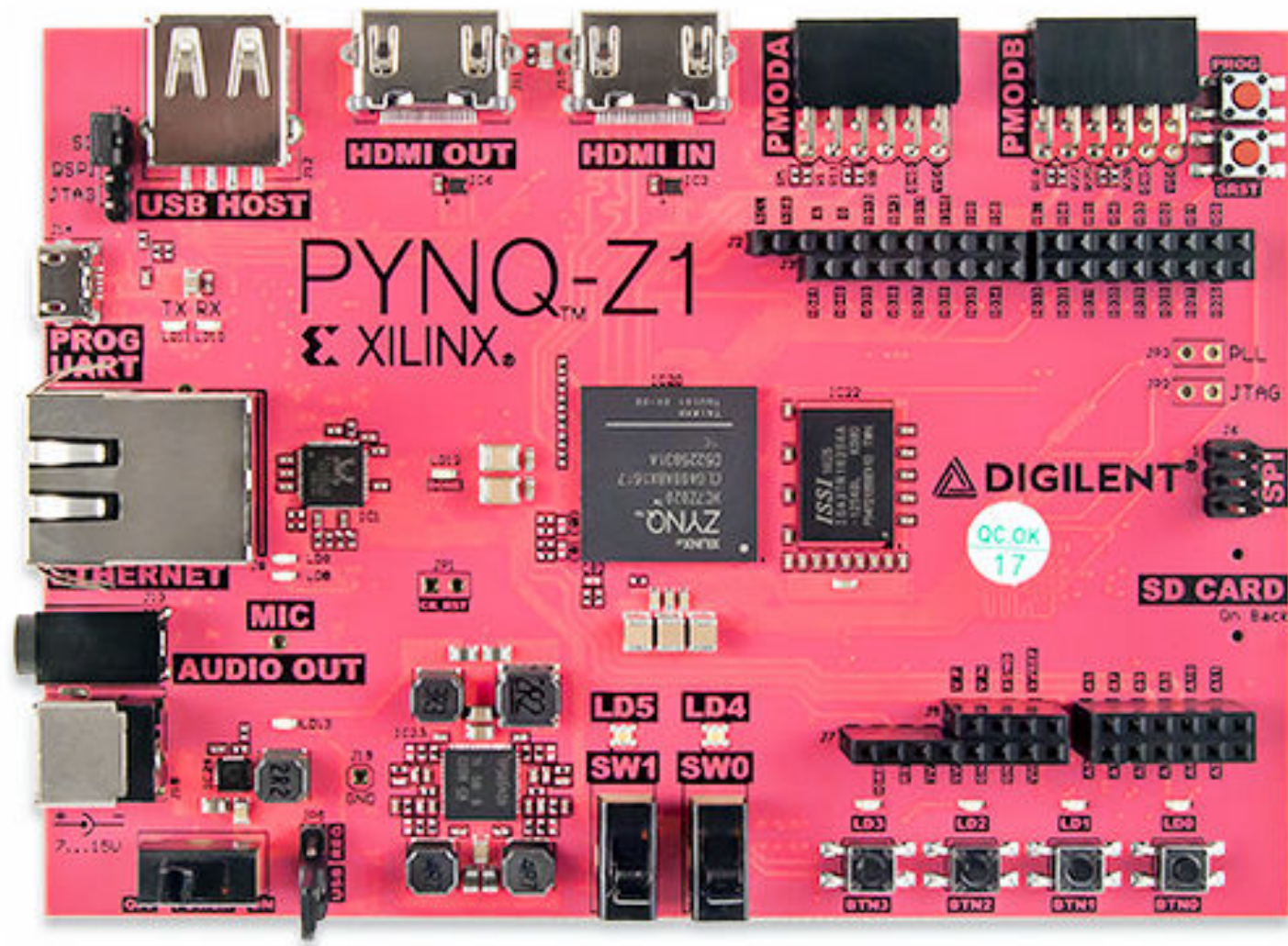


FPGAs

CS150 Project platform: Xilinx ML505-110



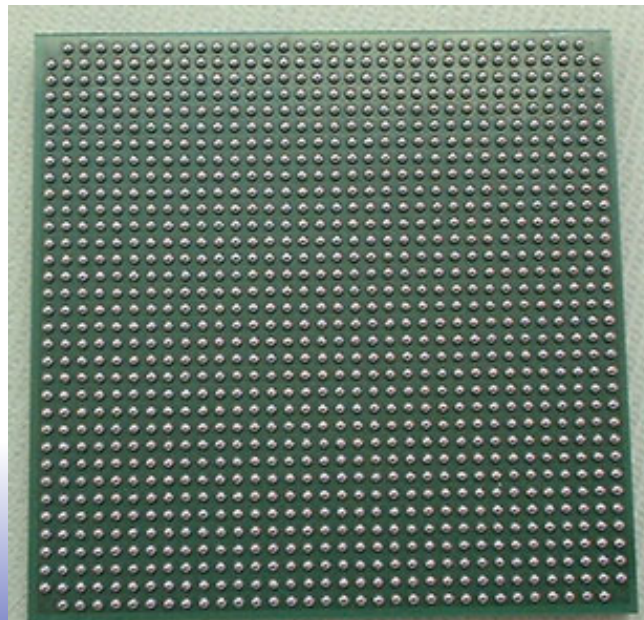
EECS151 FPGA Lab Board



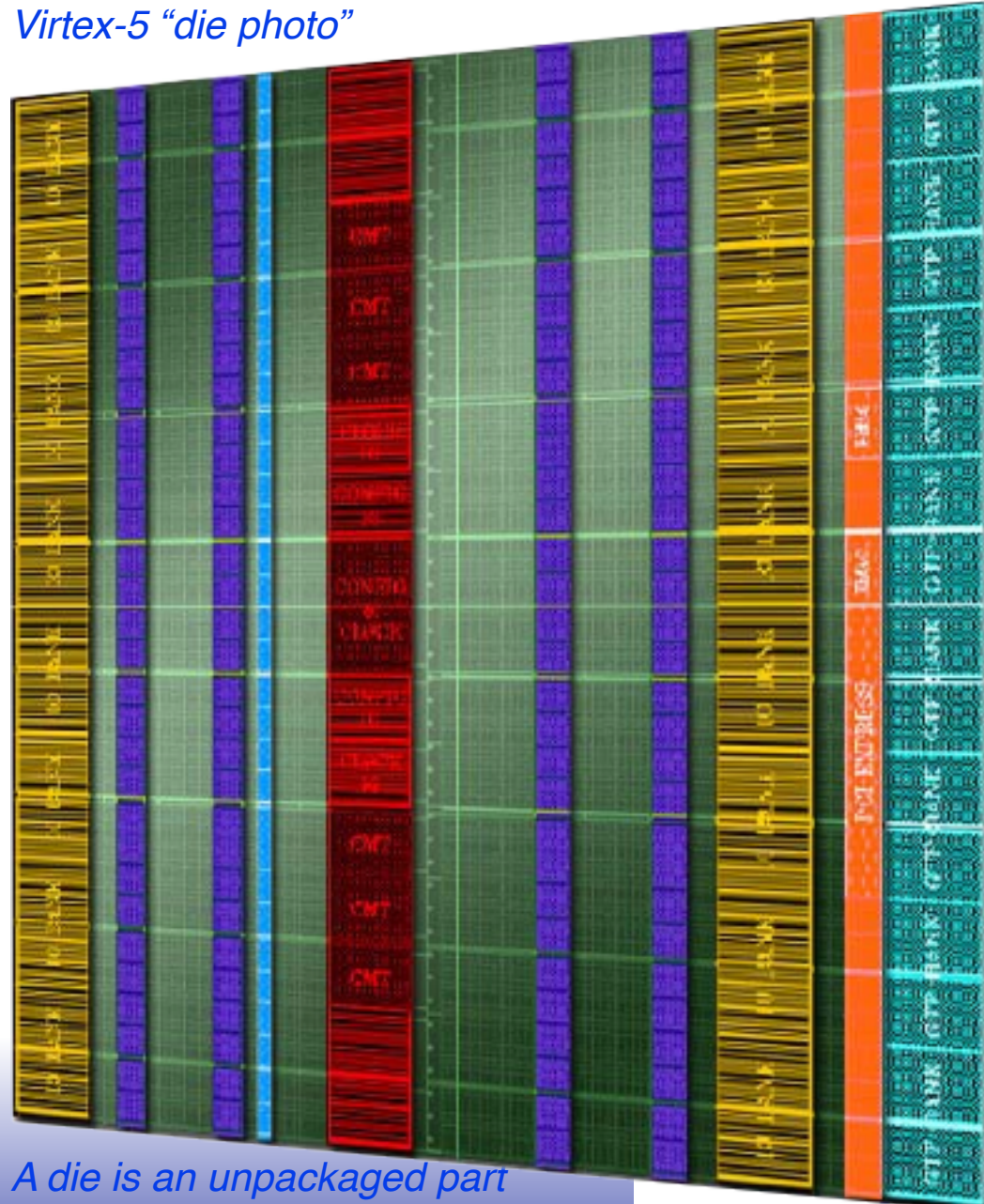
FPGA: Xilinx Virtex-5 XC5VLX110T



Virtex-5 "die photo"

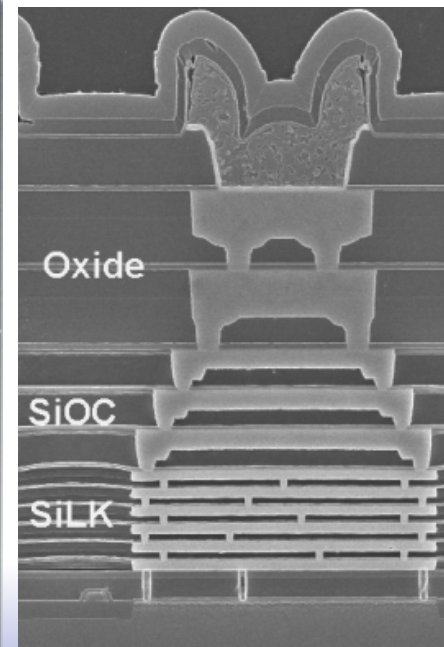
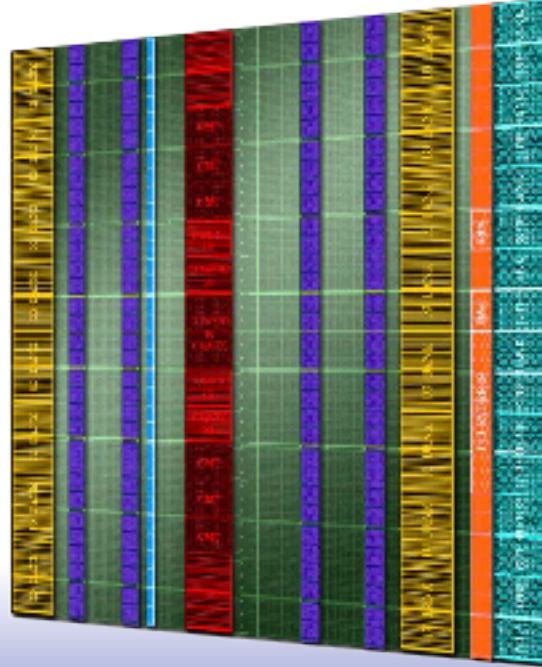
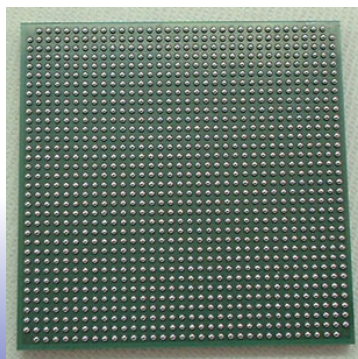
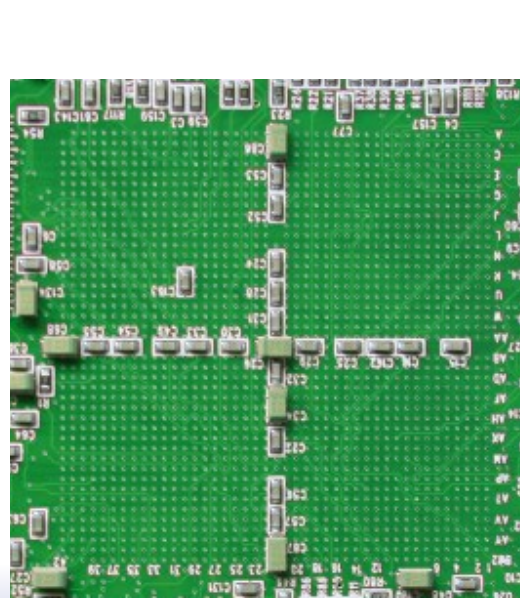
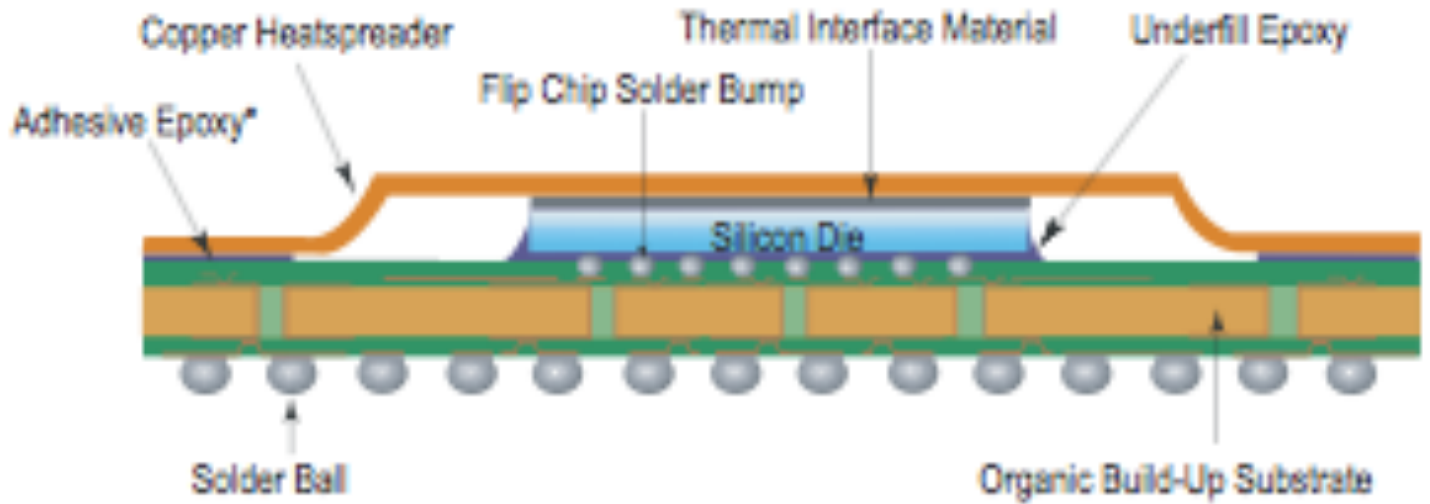


A die is an unpackaged part



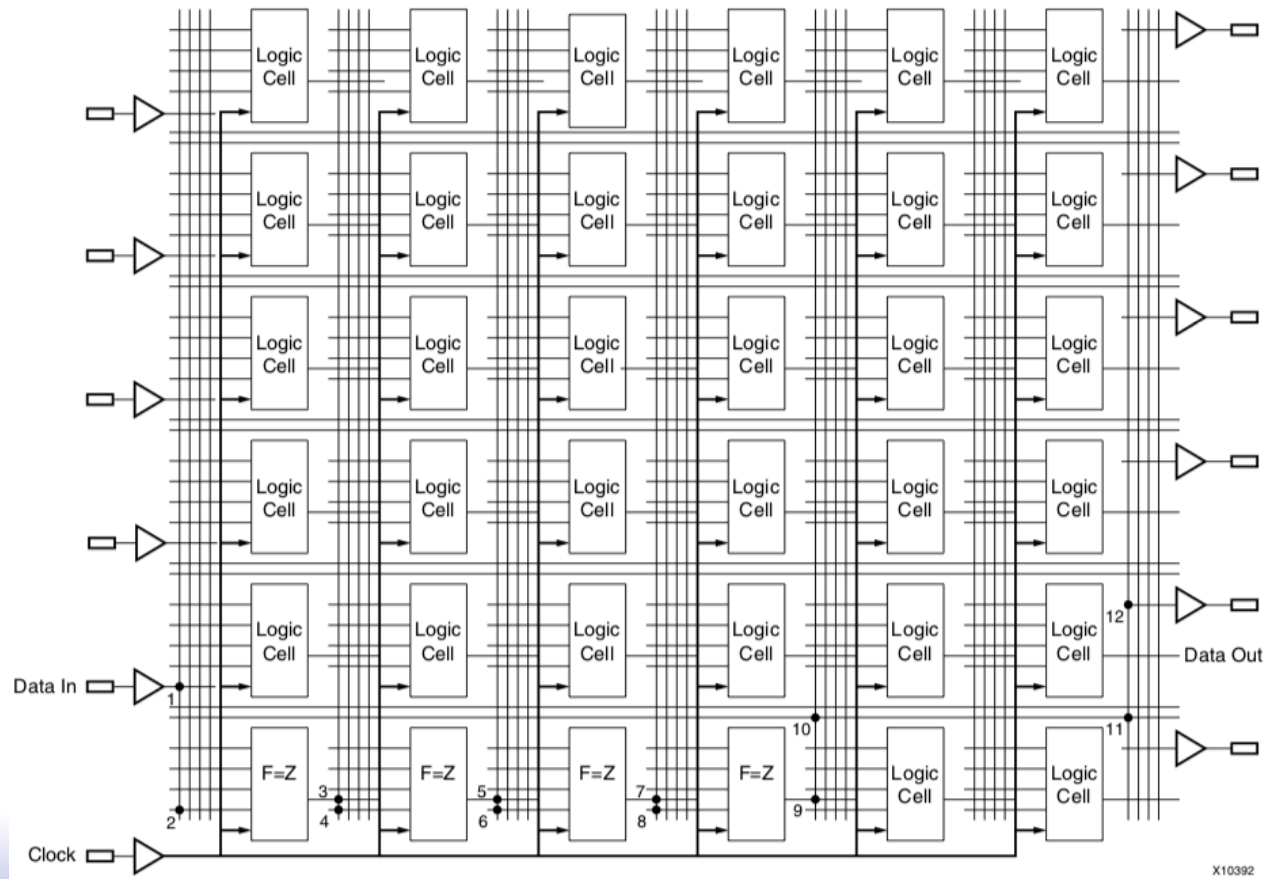
From die to PC board ...

Ball Grid
Array (BGA)
Flip-Chip
Package



FPGA Overview

- Basic idea: two-dimensional array of logic blocks and flip-flops with a means for the user to configure (program):
 - the interconnection between the logic blocks,
 - the function of each block.



Simplified version of FPGA internal architecture

Why are FPGAs Interesting?

□ Technical viewpoint:

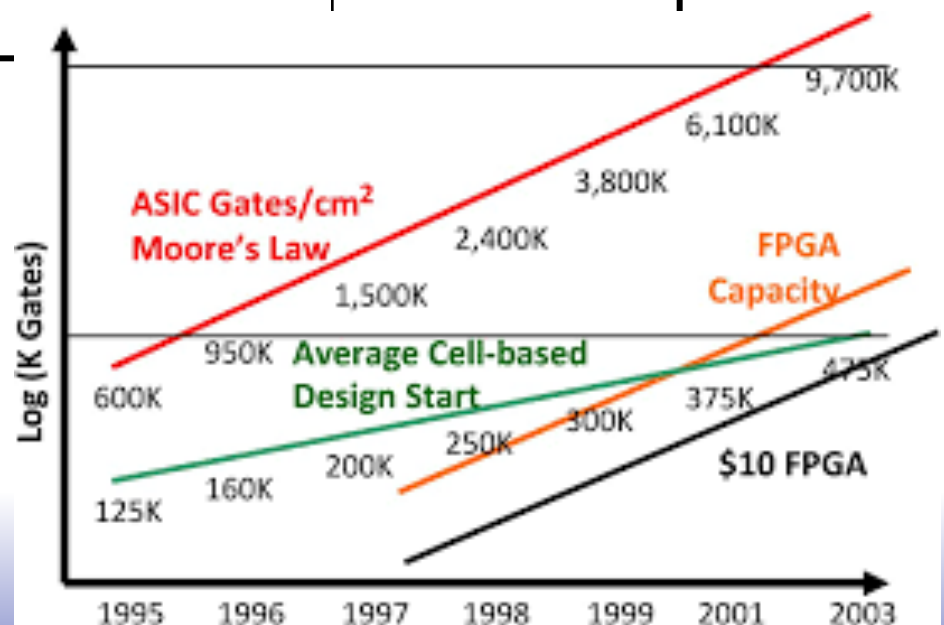
- For hardware/system-designers, like ASICs - only better: “Tape-out” new design every few minutes/hours.
- “reconfigurability” or “reprogrammability” may offer other advantages over fixed logic?
 - In-field reprogramming? Dynamic reconfiguration?
Self-modifying hardware, evolvable hardware?

Why are *FPGAs* Interesting?

- Staggering logic capacity growth (10000x):

| Year Introduced | Device | Logic Cells | “logic gate equivalents” |
|-----------------|-----------|-------------|--------------------------|
| 1985 | XC2064 | 128 | 1024 |
| 2011 | XC7V2000T | 1,954,560 | 15,636,480 |

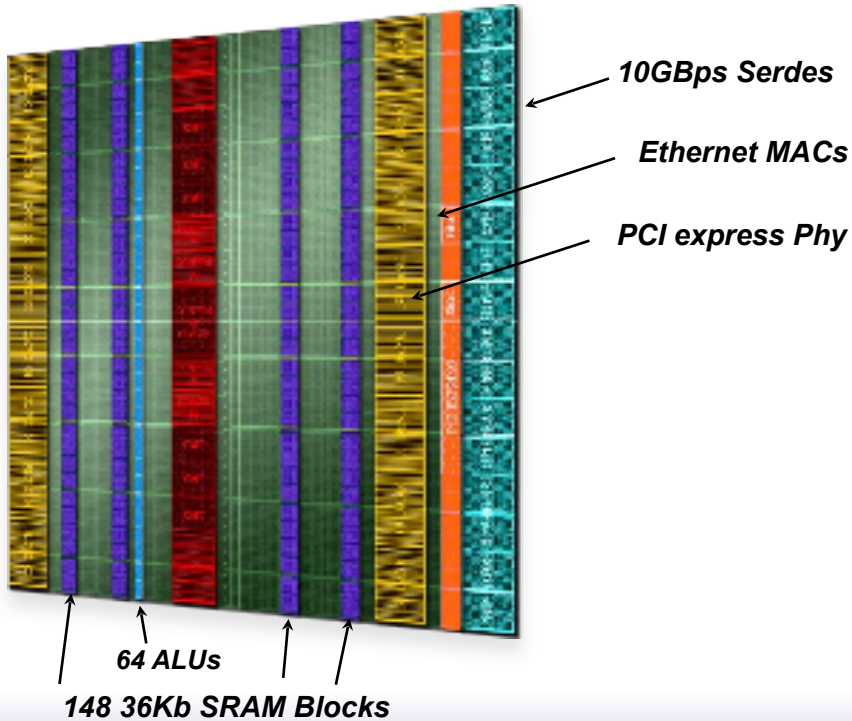
- FPGAs have tracked Moore’s Law better than any other programmable device.



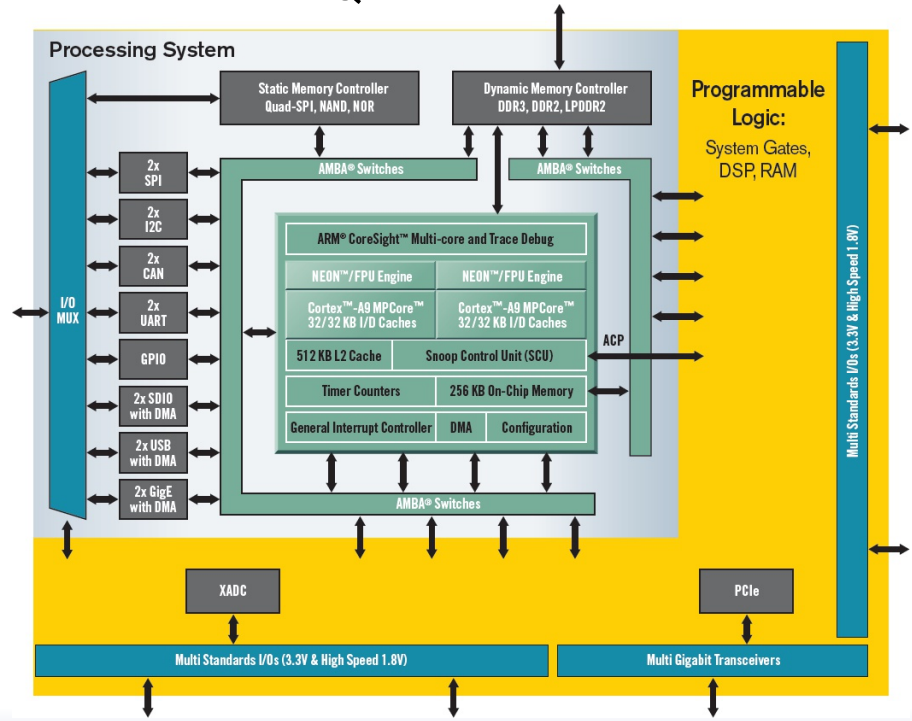
Why are FPGAs Interesting?

- Logic capacity now only part of the story: on-chip RAM, high-speed I/Os, “hard” function blocks, ...
- Modern FPGAs are “reconfigurable systems”

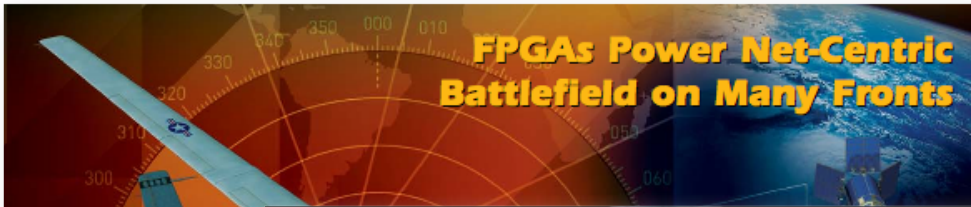
Xilinx Virtex-5 LX110T



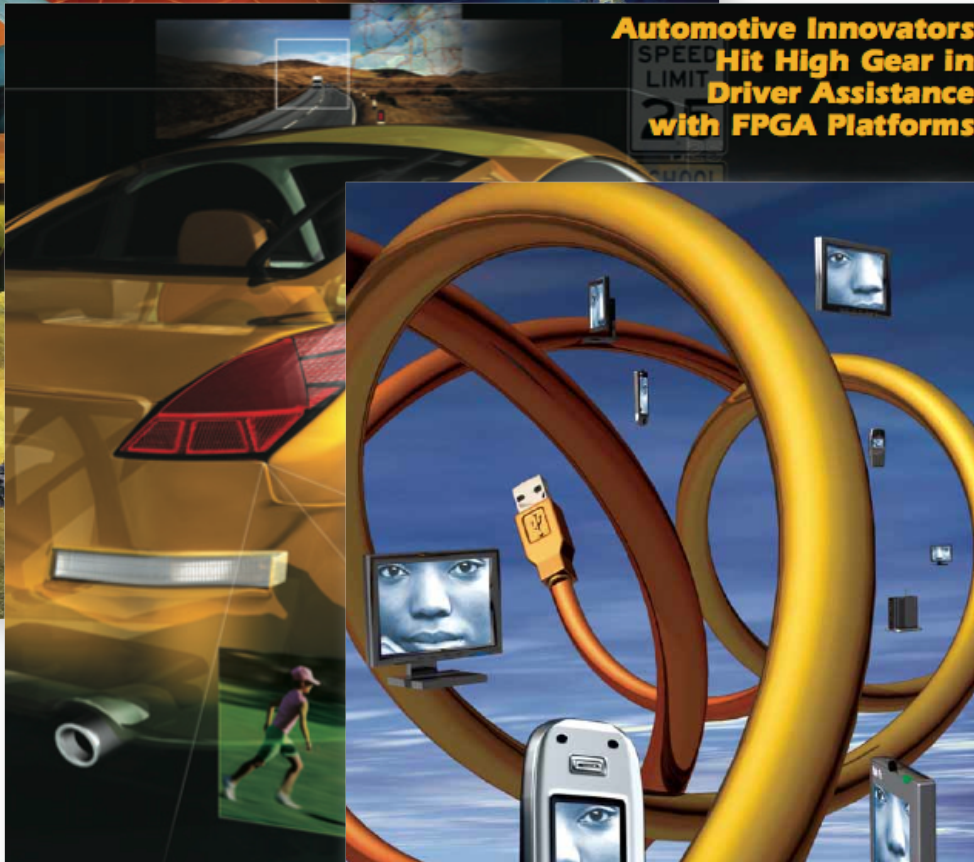
Xilinx ZYNQ



FPGAs are in widespread use

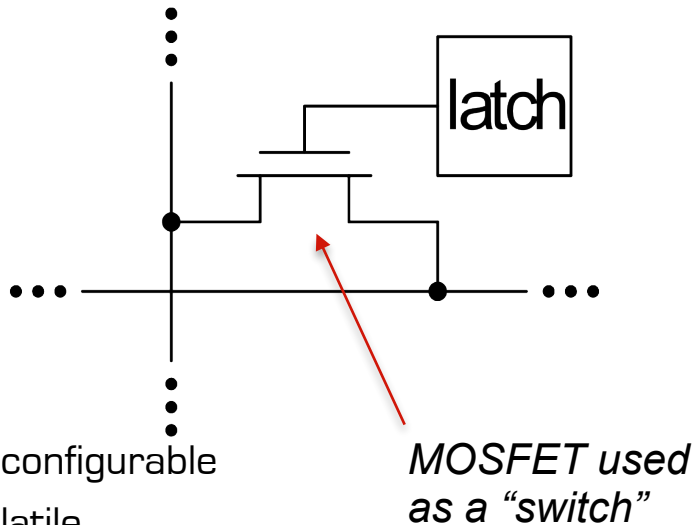


Far more different designs are implemented in FPGA than in custom chips.



User Programmability

- Latch-based (Xilinx, Intel/Altera, ...)

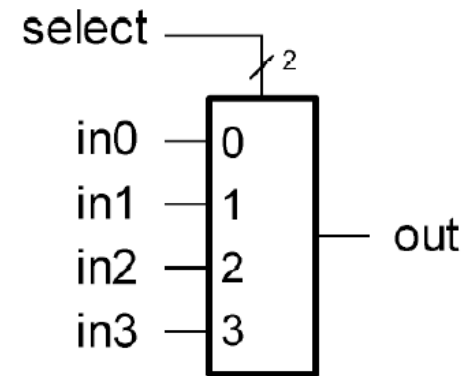


- + reconfigurable
- volatile
- relatively large.

- Latches are used to:
 1. control a switch to make or break cross-point connections in the interconnect
 2. define the function of the logic blocks
 3. set user options:
 - within the logic blocks
 - in the input/output blocks
 - global reset/clock
- “Configuration bit stream” is loaded under user control

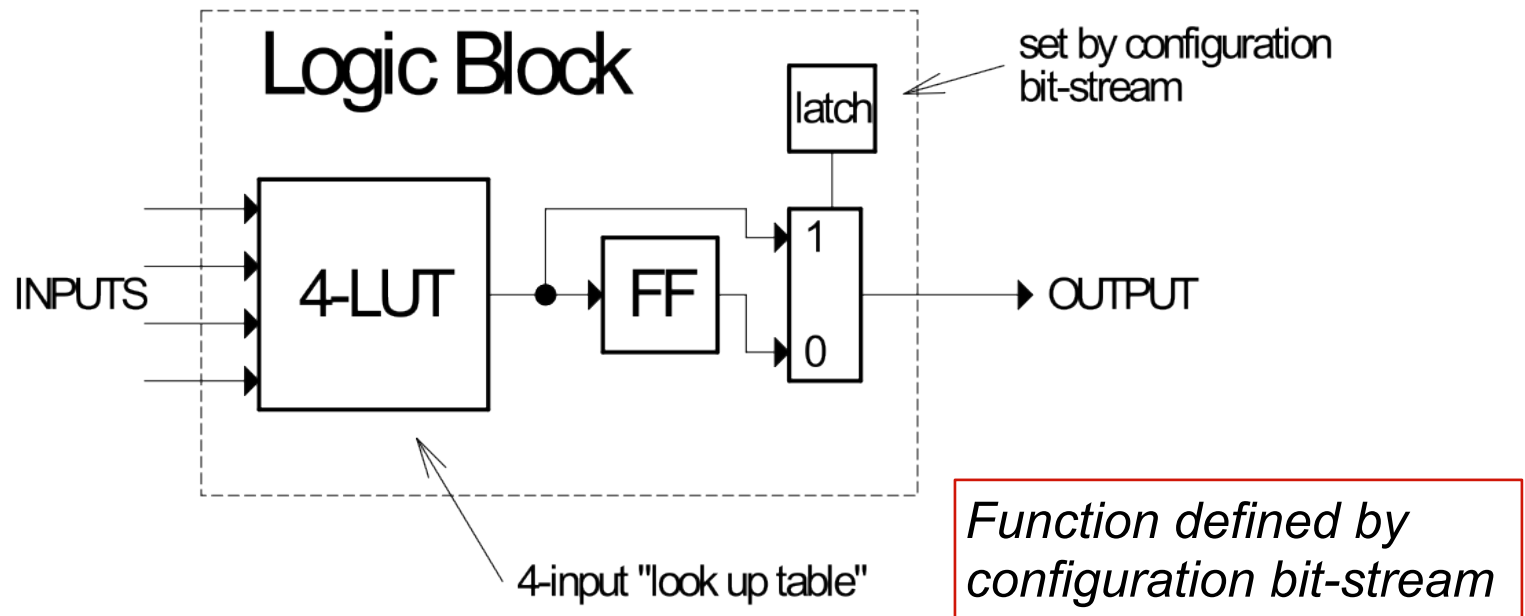
Background (review) for upcoming

- A MUX or multiplexor is a combinational logic circuit that chooses between 2^N inputs under the control of N control signals.



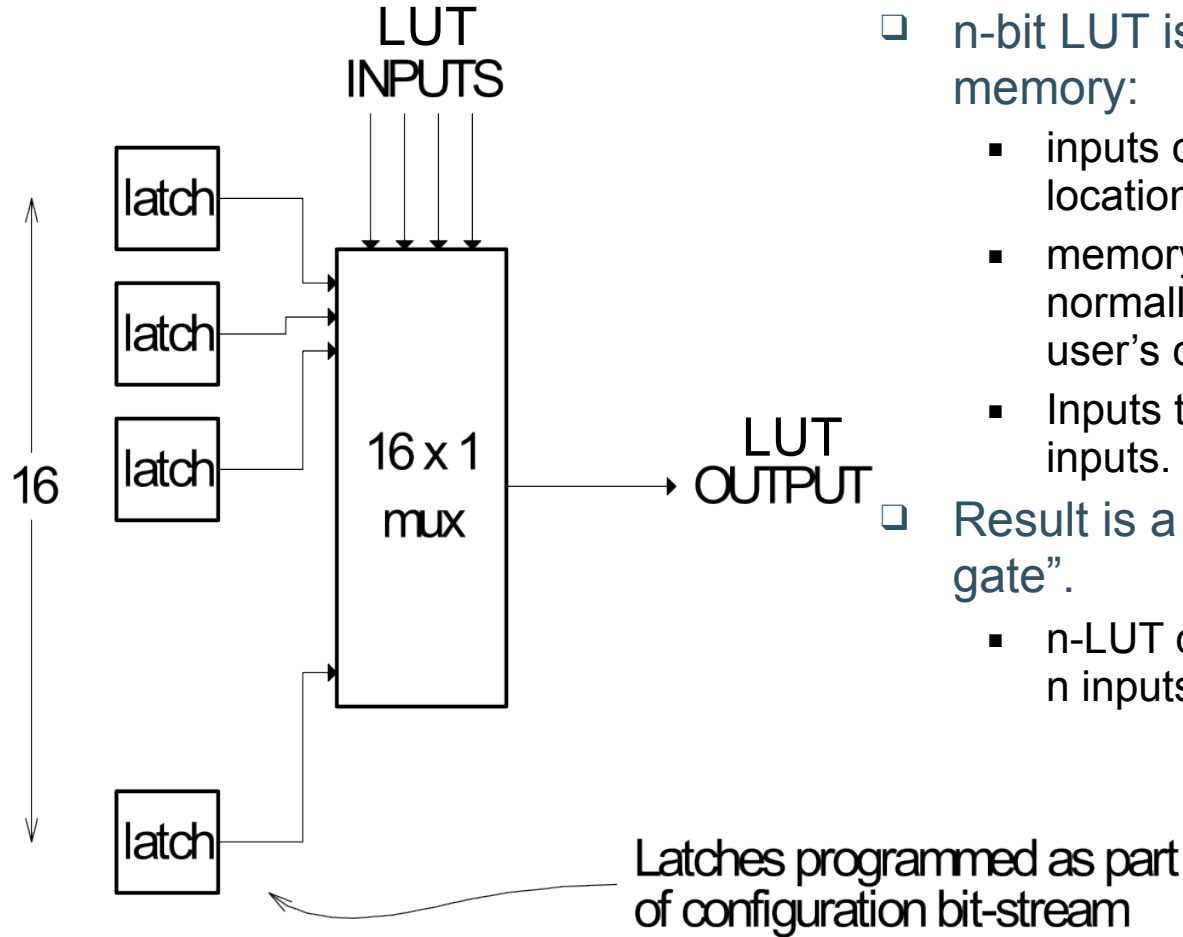
- A latch is a 1-bit memory (similar to a flip-flop).

Idealized FPGA Logic Block



- ❑ 4-input look up table (LUT)
 - implements combinational logic functions
- ❑ Register (Flip-flop)
 - optionally stores output of LUT

4-LUT Implementation



- n-bit LUT is implemented as a $2^n \times 1$ 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!

LUT as general logic gate

- An n-lut is a direct implementation of a function truth-table.
- Each latch location holds the value of the function corresponding to one input combination.

Example: 2-input functions

| INPUTS | AND | OR | | | |
|--------|-----|----|---|---|---|
| 00 | 0 | 0 | | | |
| 01 | 0 | 1 | | | |
| 10 | 0 | 1 | • | • | • |
| 11 | 1 | 1 | | | |

A 2-lut Implements any function of 2 inputs.

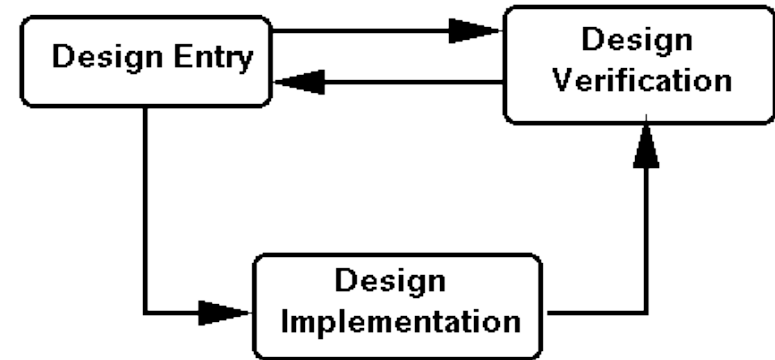
How many of these are there?

How many functions of n inputs?

Example: 4-lut

| INPUTS | | |
|--------|------------|----------------------|
| 0000 | F(0,0,0,0) | ← store in 1st latch |
| 0001 | F(0,0,0,1) | ← store in 2nd latch |
| 0010 | F(0,0,1,0) | ← |
| 0011 | F(0,0,1,1) | ← |
| 0011 | | |
| 0100 | • | |
| 0101 | • | |
| 0110 | • | |
| 0111 | | |
| 1000 | | |
| 1001 | | |
| 1010 | | |
| 1011 | | |
| 1100 | | |
| 1101 | | |
| 1110 | | |
| 1111 | | |

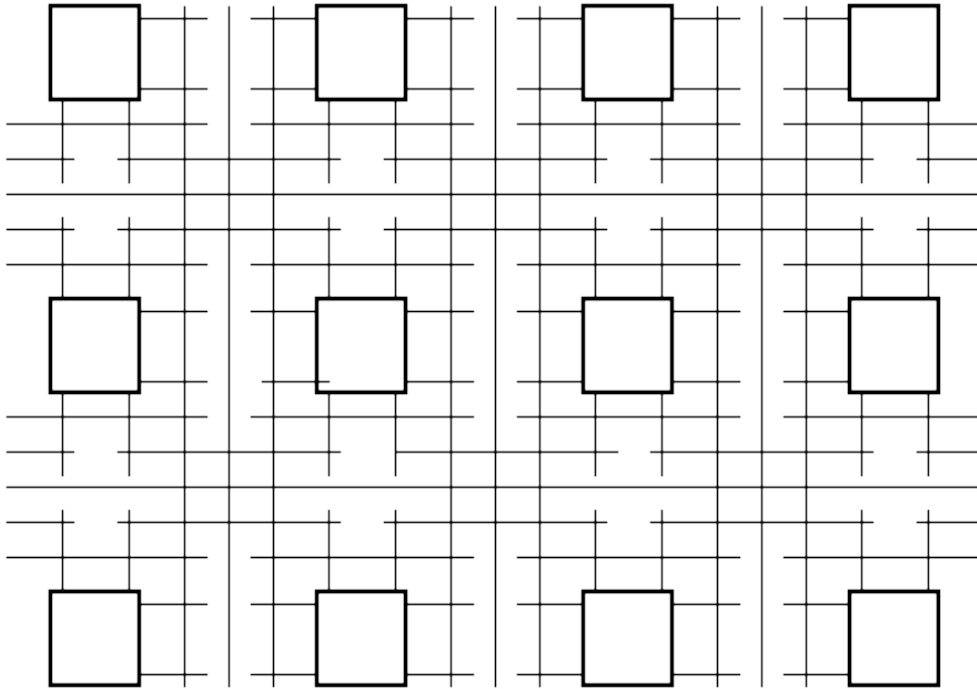
FPGA Generic Design Flow



- Design Entry:
 - Create your design files using:
 - schematic editor or
 - HDL (hardware description languages: Verilog, VHDL)
- Design Implementation:
 - Logic synthesis (in case of using HDL entry) followed by,
 - Partition, place, and route to create configuration bit-stream file
- Design verification:
 - Optionally use simulator to check function,
 - Load design onto FPGA device (cable connects PC to development board), optional “logic scope” on FPGA
 - check operation at full speed in real environment.

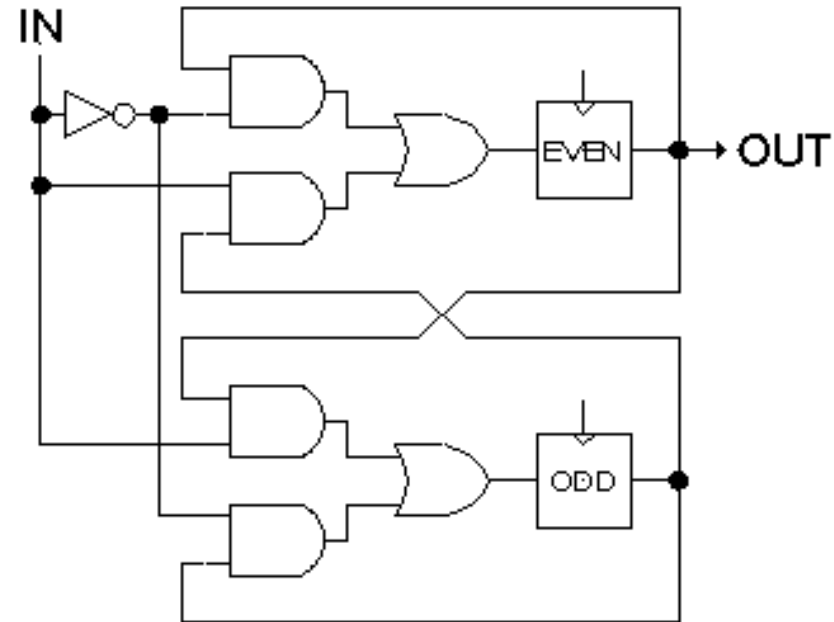
Example Partition, Placement, and Route

- Idealized FPGA structure:



- Example Circuit:

- collection of gates and flip-flops



Circuit combinational logic must be “covered” by 4-input 1-output LUTs.

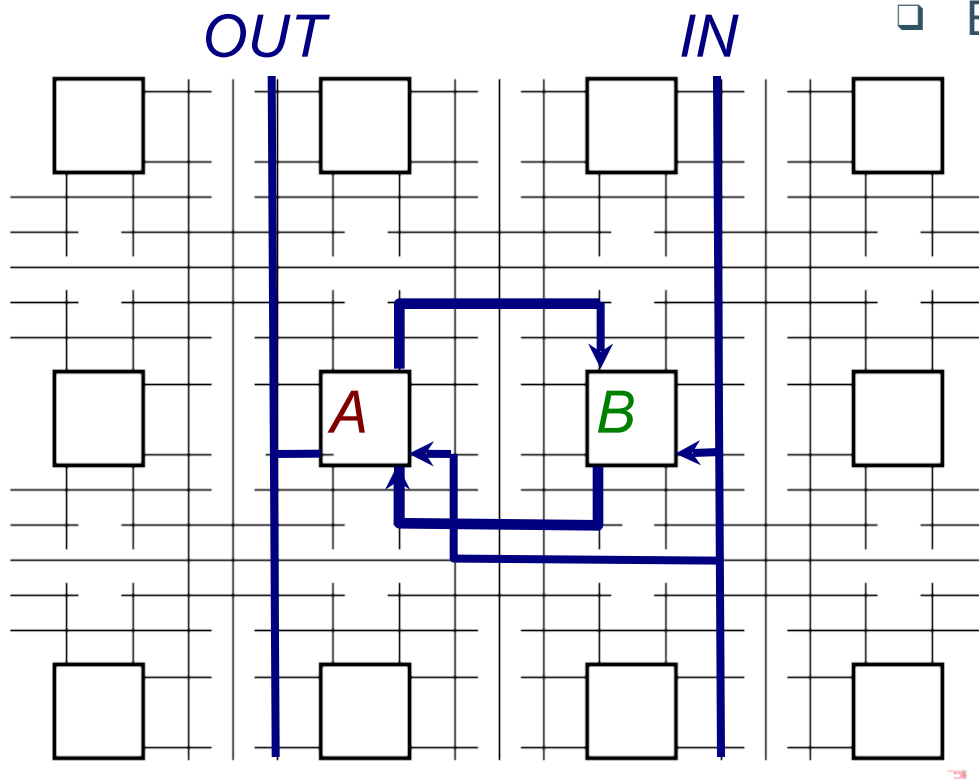
Flip-flops from circuit must map to FPGA flip-flops.

(Best to preserve “closeness” to CL to minimize wiring.)

Best placement in general attempts to minimize wiring.

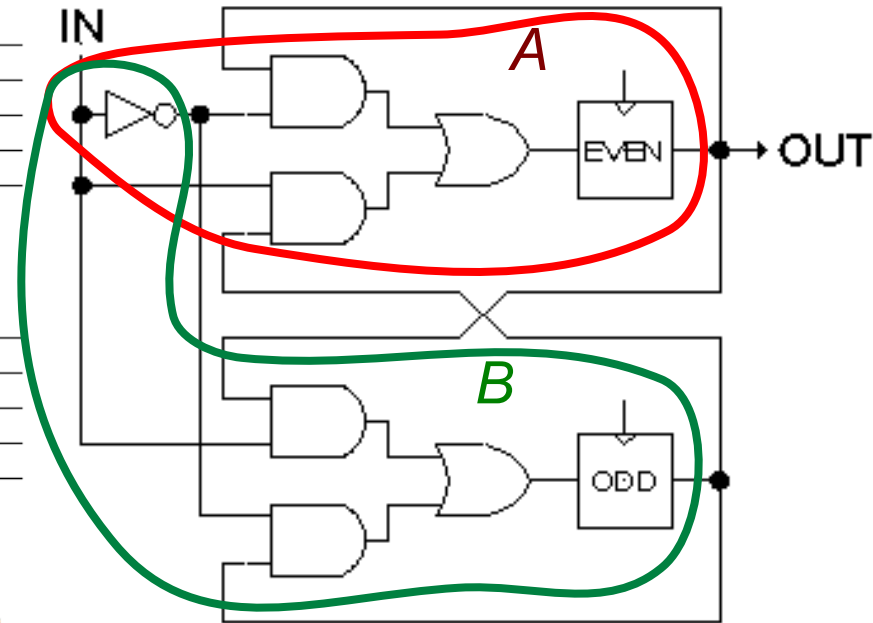
V_{dd}, GND, clock, and global resets are all “prewired”.

Example Partition, Placement, and Route



Example Circuit:

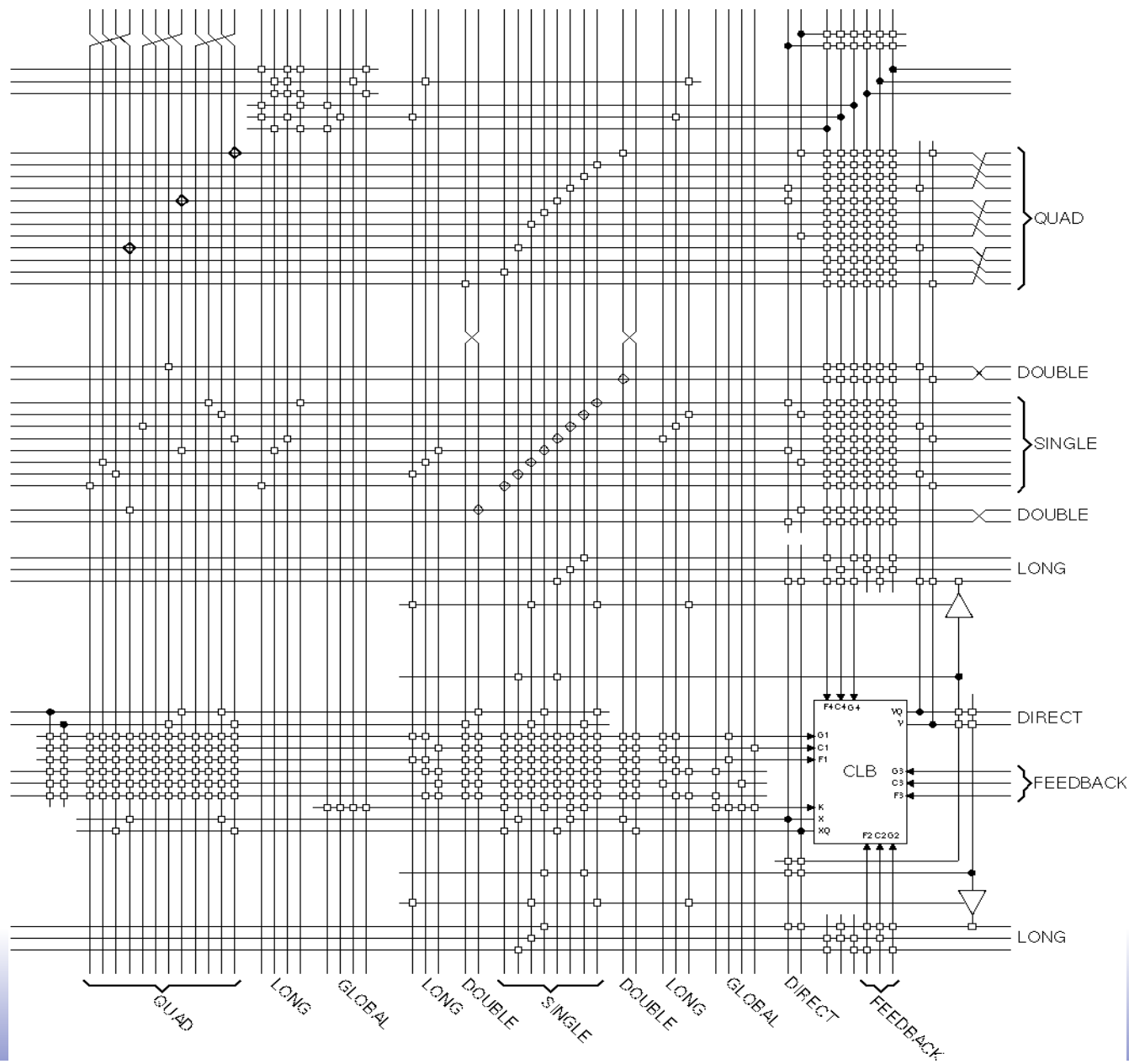
- collection of gates and flip-flops



Two partitions. Each has single output, no more than 4 inputs, and no more than 1 flip-flop. In this case, inverter goes in both partitions.

Note: the partition can be arbitrarily large as long as it has not more than 4 inputs and 1 output, and no more than 1 flip-flop.

Xilinx FPGAs (interconnect detail)



Colors represent different types of resources:

Logic

Block RAM

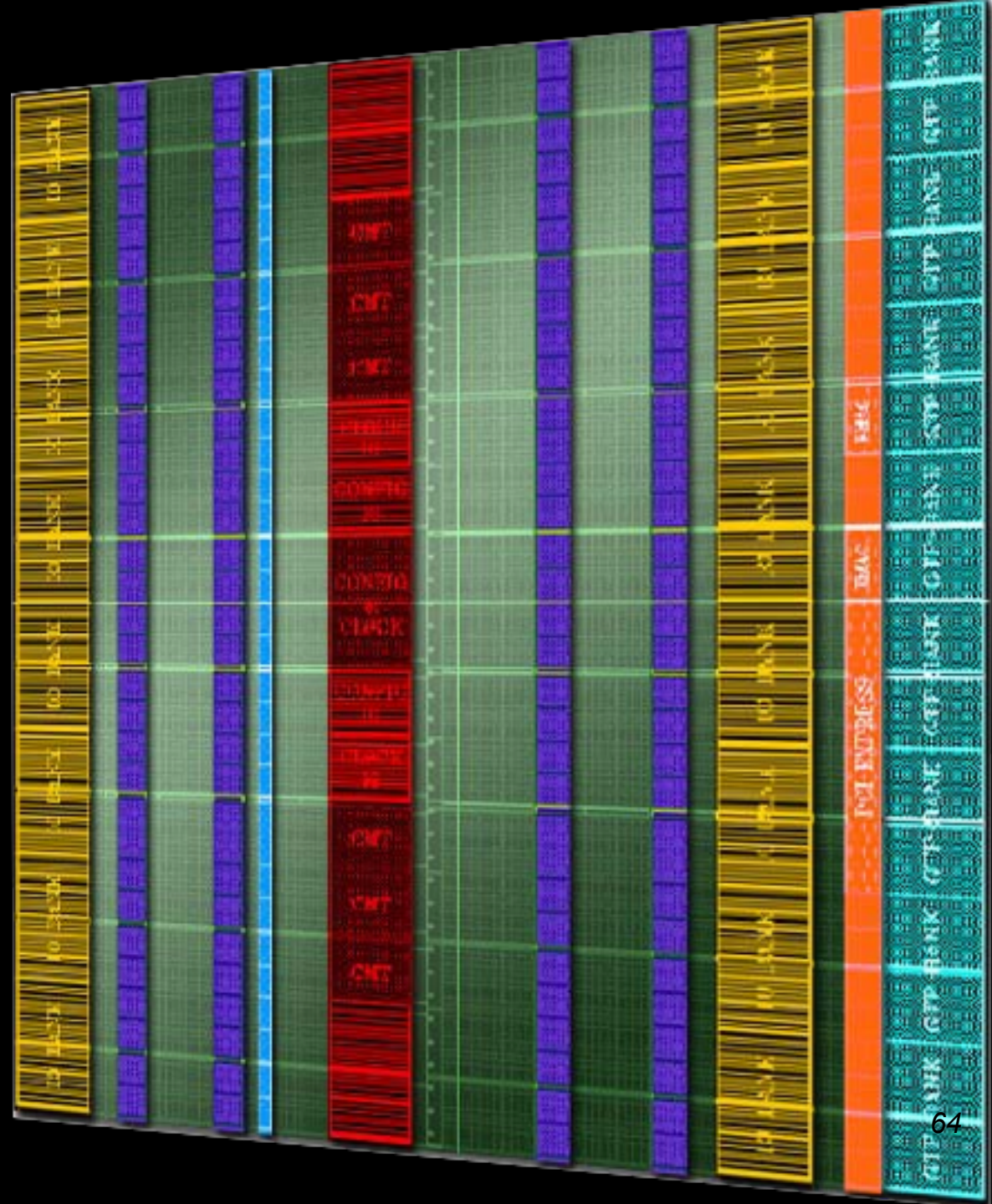
DSP (ALUs)

Clocking

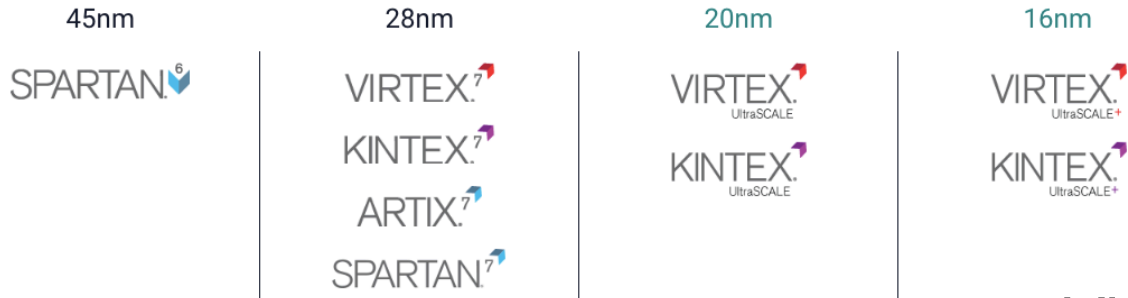
I/O

Serial I/O + PCI

A routing fabric runs throughout the chip to wire everything together.



State-of-the-Art - Xilinx FPGAs

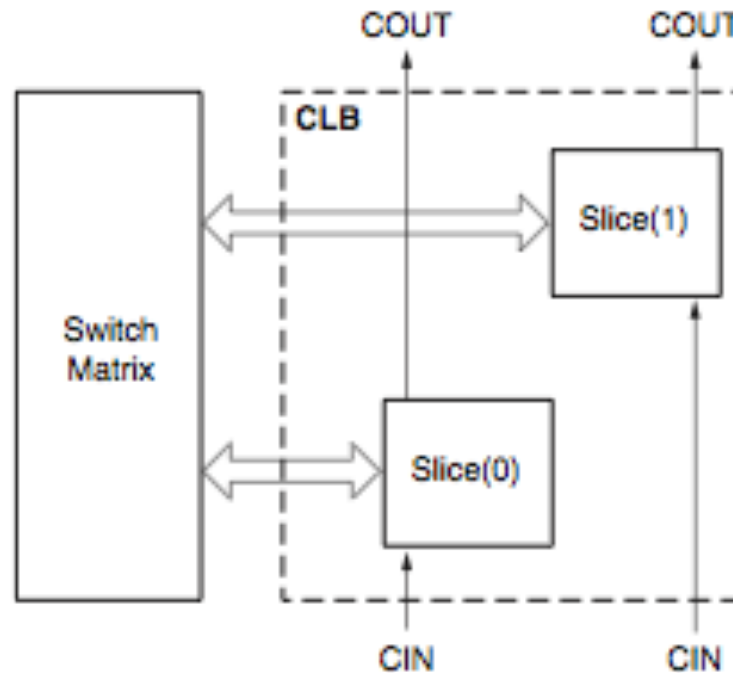


Virtex Ultra-scale

| Device Name | VU3P | VU5P | VU7P | VU9P | VU11P | VU13P | VU27P | VU29P | VU31P | VU33P | VU35P | VU37P | |
|---|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| System Logic Cells (K) | 862 | 1,314 | 1,724 | 2,586 | 2,835 | 3,780 | 2,835 | 3,780 | 962 | 962 | 1,907 | 2,852 | |
| CLB Flip-Flops (K) | 788 | 1,201 | 1,576 | 2,364 | 2,592 | 3,456 | 2,592 | 3,456 | 879 | 879 | 1,743 | 2,607 | |
| CLB LUTs (K) | 394 | 601 | 788 | 1,182 | 1,296 | 1,728 | 1,296 | 1,728 | 440 | 440 | 872 | 1,304 | |
| Max. Dist. RAM (Mb) | 12.0 | 18.3 | 24.1 | 36.1 | 36.2 | 48.3 | 36.2 | 48.3 | 12.5 | 12.5 | 24.6 | 36.7 | |
| Total Block RAM (Mb) | 25.3 | 36.0 | 50.6 | 75.9 | 70.9 | 94.5 | 70.9 | 94.5 | 23.6 | 23.6 | 47.3 | 70.9 | |
| UltraRAM (Mb) | 90.0 | 132.2 | 180.0 | 270.0 | 270.0 | 360.0 | 270.0 | 360.0 | 90.0 | 90.0 | 180.0 | 270.0 | |
| HBM DRAM (GB) | – | – | – | – | – | – | – | – | 4 | 8 | 8 | 8 | |
| HBM AXI Interfaces | – | – | – | – | – | – | – | – | 32 | 32 | 32 | 32 | |
| Clock Mgmt Tiles (CMTs) | 10 | 20 | 20 | 30 | 12 | 16 | 16 | 16 | 4 | 4 | 8 | 12 | |
| DSP Slices | 2,280 | 3,474 | 4,560 | 6,840 | 9,216 | 12,288 | 9,216 | 12,288 | 2,880 | 2,880 | 5,952 | 9,024 | |
| Peak INT8 DSP (TOP/s) | 7.1 | 10.8 | 14.2 | 21.3 | 28.7 | 38.3 | 28.7 | 38.3 | 8.9 | 8.9 | 18.6 | 28.1 | |
| PCIe® Gen3 x16 | 2 | 4 | 4 | 6 | 3 | 4 | 1 | 1 | 0 | 0 | 1 | 2 | |
| PCIe Gen3 x16/Gen4 x8 / CCIX ⁽¹⁾ | – | – | – | – | – | – | – | – | 4 | 4 | 4 | 4 | |
| 150G Interlaken | 3 | 4 | 6 | 9 | 6 | 8 | 6 | 8 | 0 | 0 | 2 | 4 | |
| 100G Ethernet w/ KR4 RS-FEC | 3 | 4 | 6 | 9 | 9 | 12 | 11 | 15 | 2 | 2 | 5 | 8 | |
| Max. Single-Ended HP I/Os | 520 | 832 | 832 | 832 | 624 | 832 | 520 | 676 | 208 | 208 | 416 | 624 | |
| GTY 32.75Gb/s Transceivers | 40 | 80 | 80 | 120 | 96 | 128 | 32 | 32 | 32 | 32 | 64 | 96 | |
| GTM 58Gb/s PAM4 Transceivers | | | | | | | 32 | 48 | | | | | |
| 100G / 50G KP4 FEC | | | | | | | 16 / 32 | 24 / 48 | | | | | |
| Extended ⁽²⁾ | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 | -1 -2 -2L -3 |
| Industrial | -1 -2 | -1 -2 | -1 -2 | -1 -2 | -1 -2 | -1 -2 | -1 -2 | -1 -2 | – | – | – | – | |

Configurable Logic Blocks (CLBs)

Slices define regular connections to the switching fabric, and to slices in CLBs above and below it on the die.

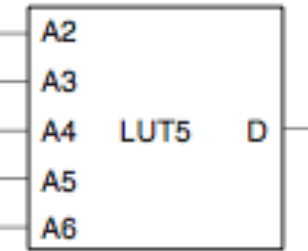


Atoms: 5-input Look Up Tables (LUTs)

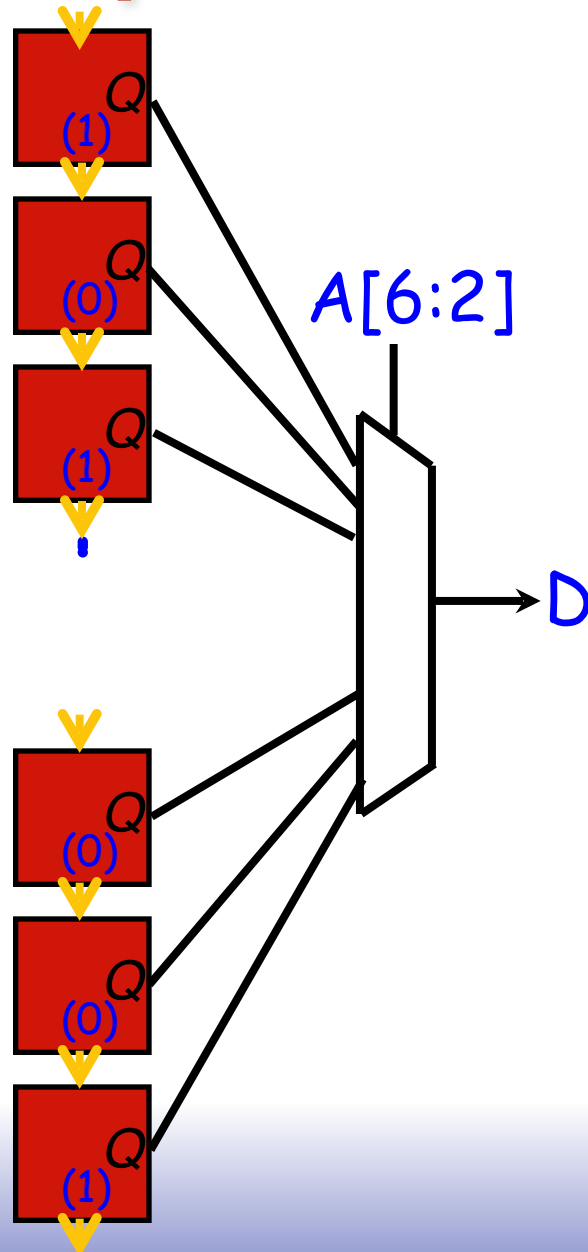
Computes any 5-input logic function.

Timing is independent of function.

Latches set during configuration.



| $A[6:2]$ | D |
|----------|---|
| 00000 | 1 |
| 00001 | 0 |
| 00010 | 1 |
| . | . |
| 11101 | 0 |
| 11110 | 0 |
| 11111 | 1 |



Virtex 6-LUTs: Composition of 5-LUTs

*May be used
as one
6-input LUT
(D6 out) ...*

*... or as two
5-input LUTS
(D6 and D5)*

*Combinational
logic*

(post configuration)

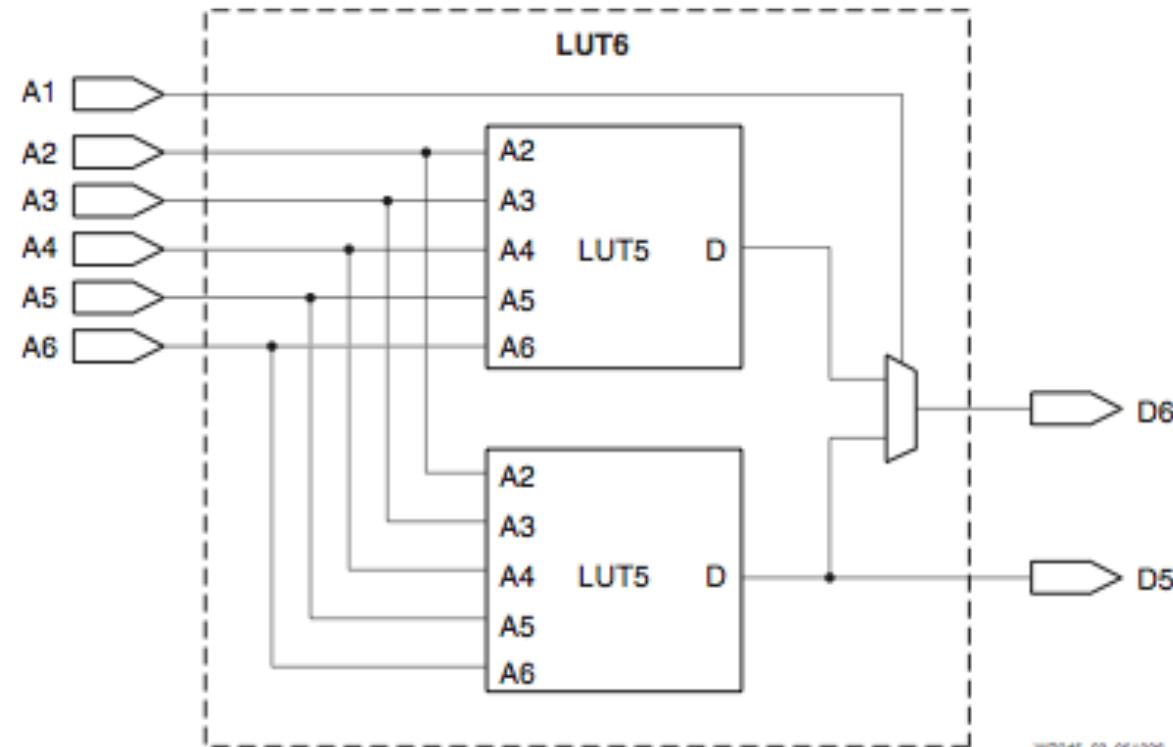
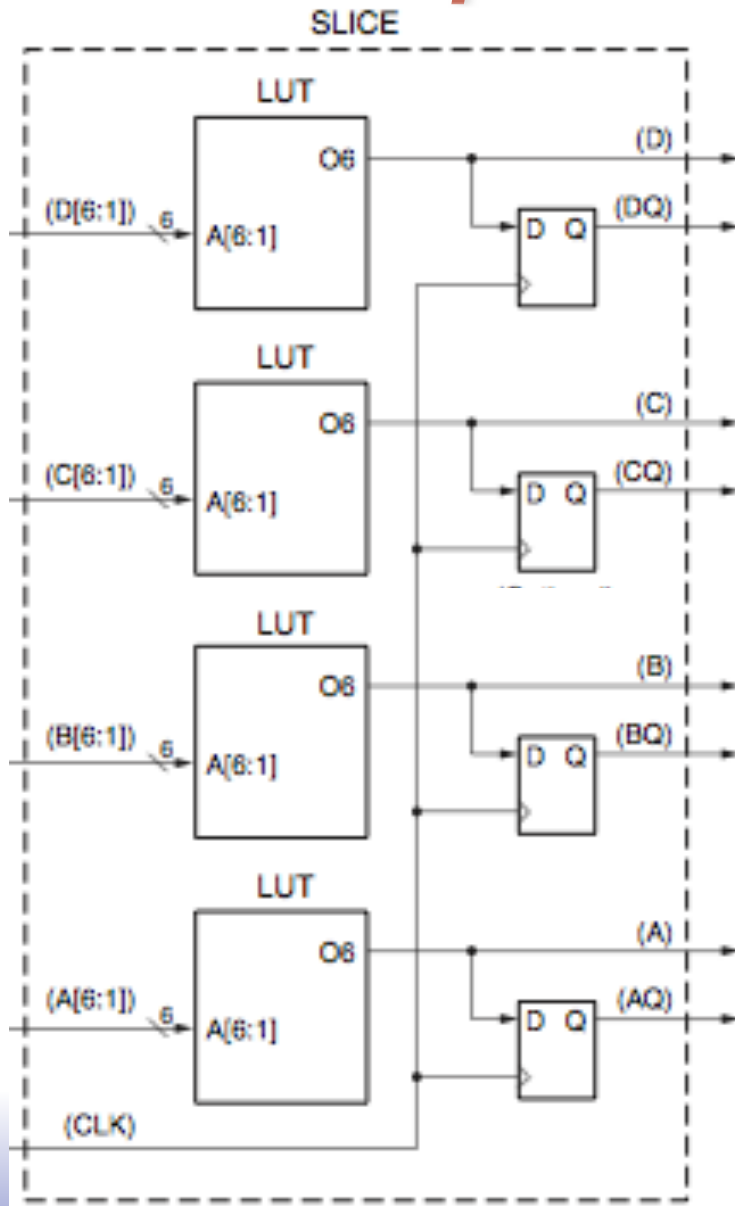


Figure 3: Block Diagram of a Virtex-5 6-Input LUT

The simplest view of a slice



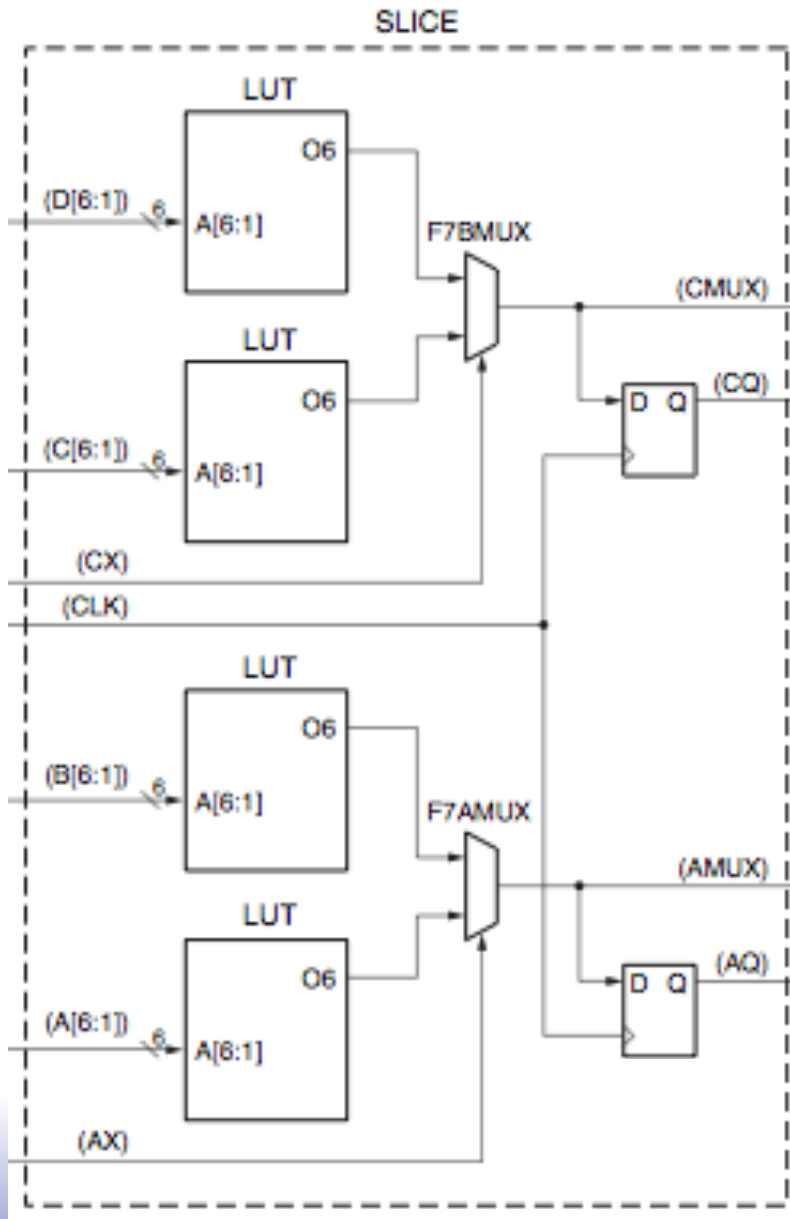
Four 6-LUTs

Four Flip-Flops

*Switching fabric may see
combinational and registered
outputs.*

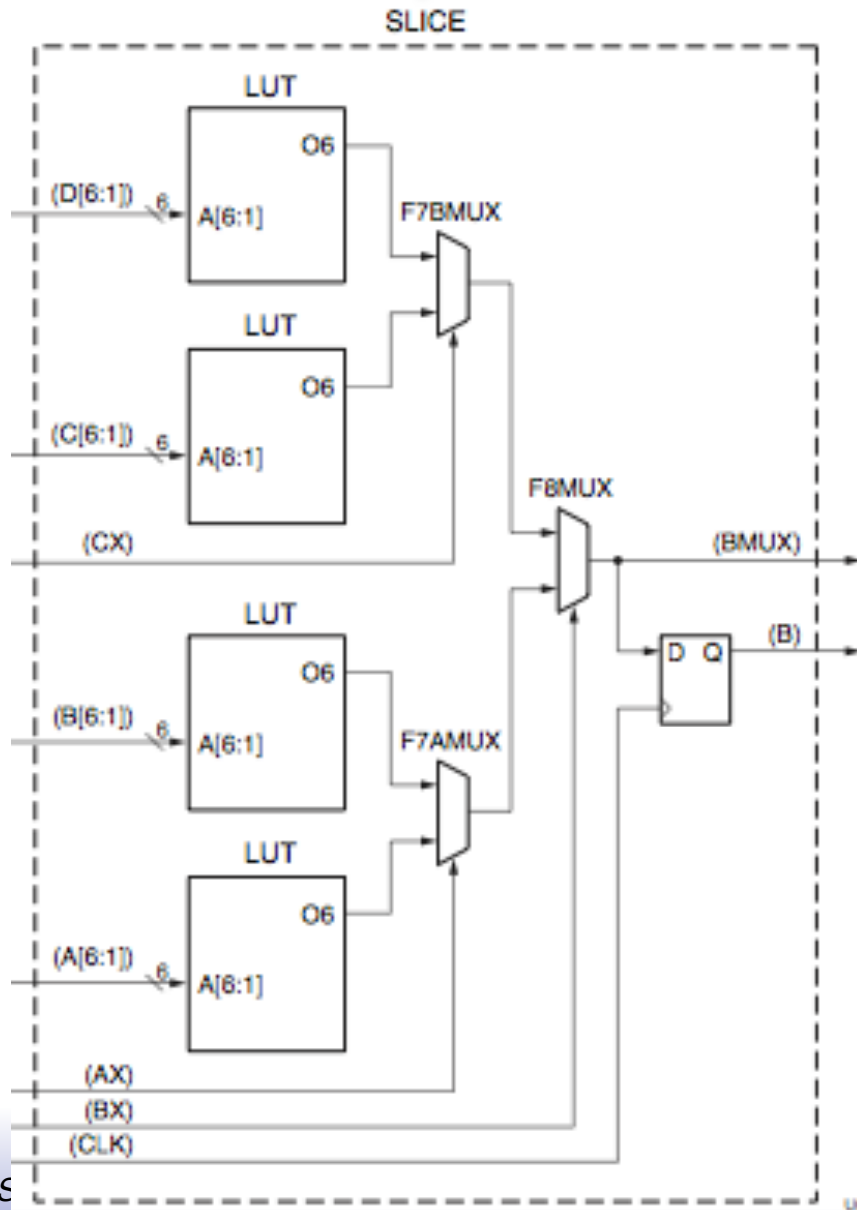
An actual Virtex slice adds many small features to this simplified diagram. We show them one by one ...

Two 7-LUTs per slice ...



*Extra
multiplexers (F7AMUX,
F7BMUX)
Extra inputs
(AX and CX)*

Or one 8-LUTs per slice ...

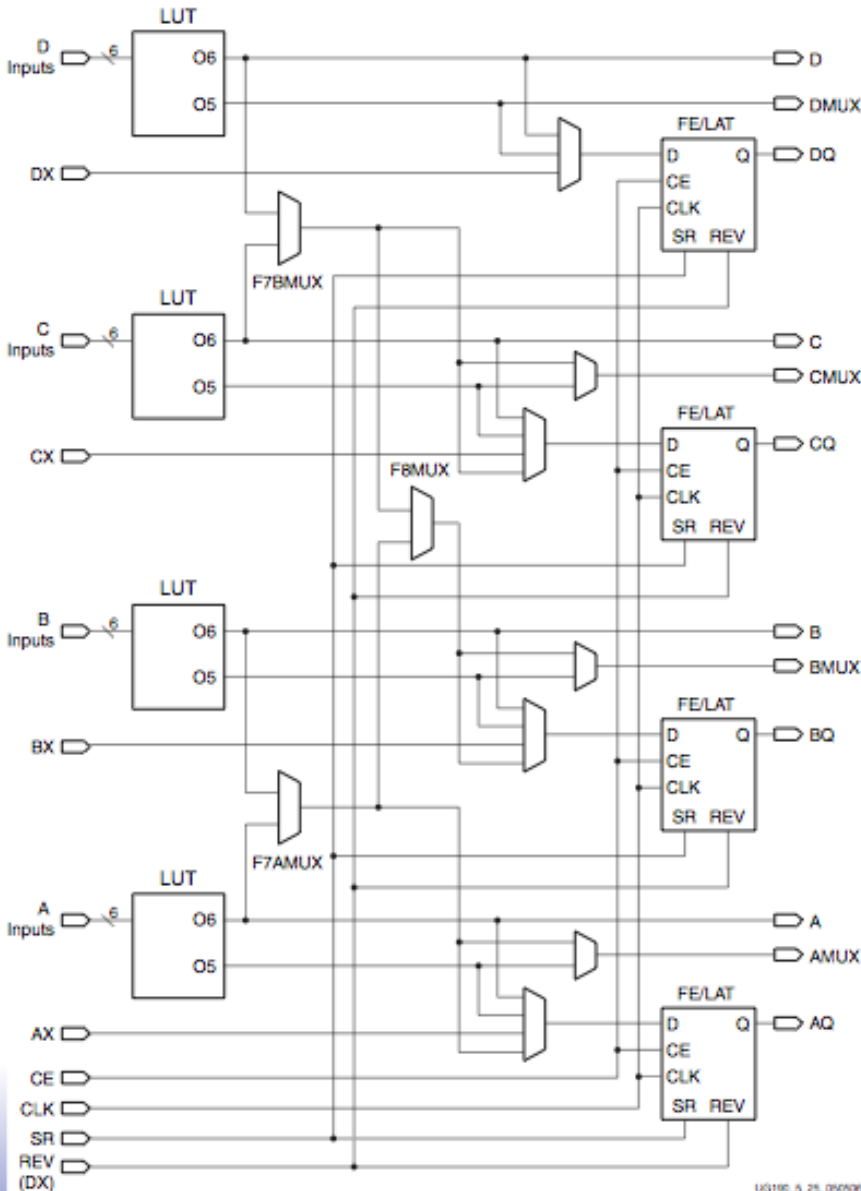


Third
multiplexer($F8MUX$)

Third input
(BX)

Configuring the
"n" of an n-LUT ...

Extra muxes to chose LUT option ...



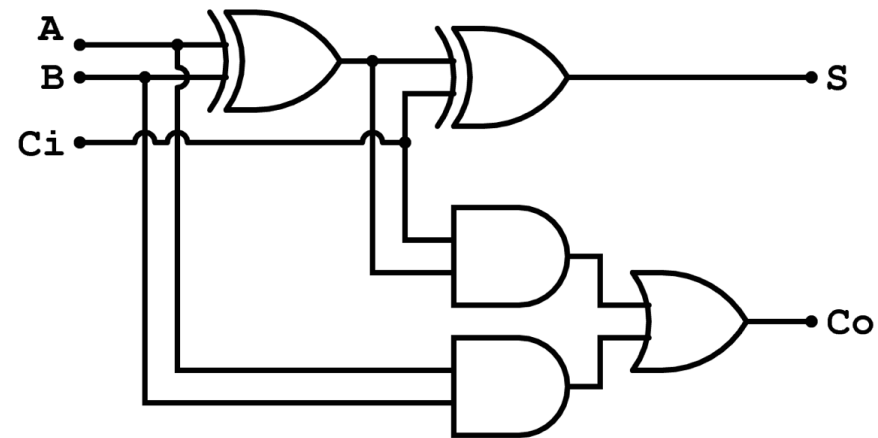
*From eight 5-LUTs
... to one 8-LUT.*

*Combinational
or registered outs.*

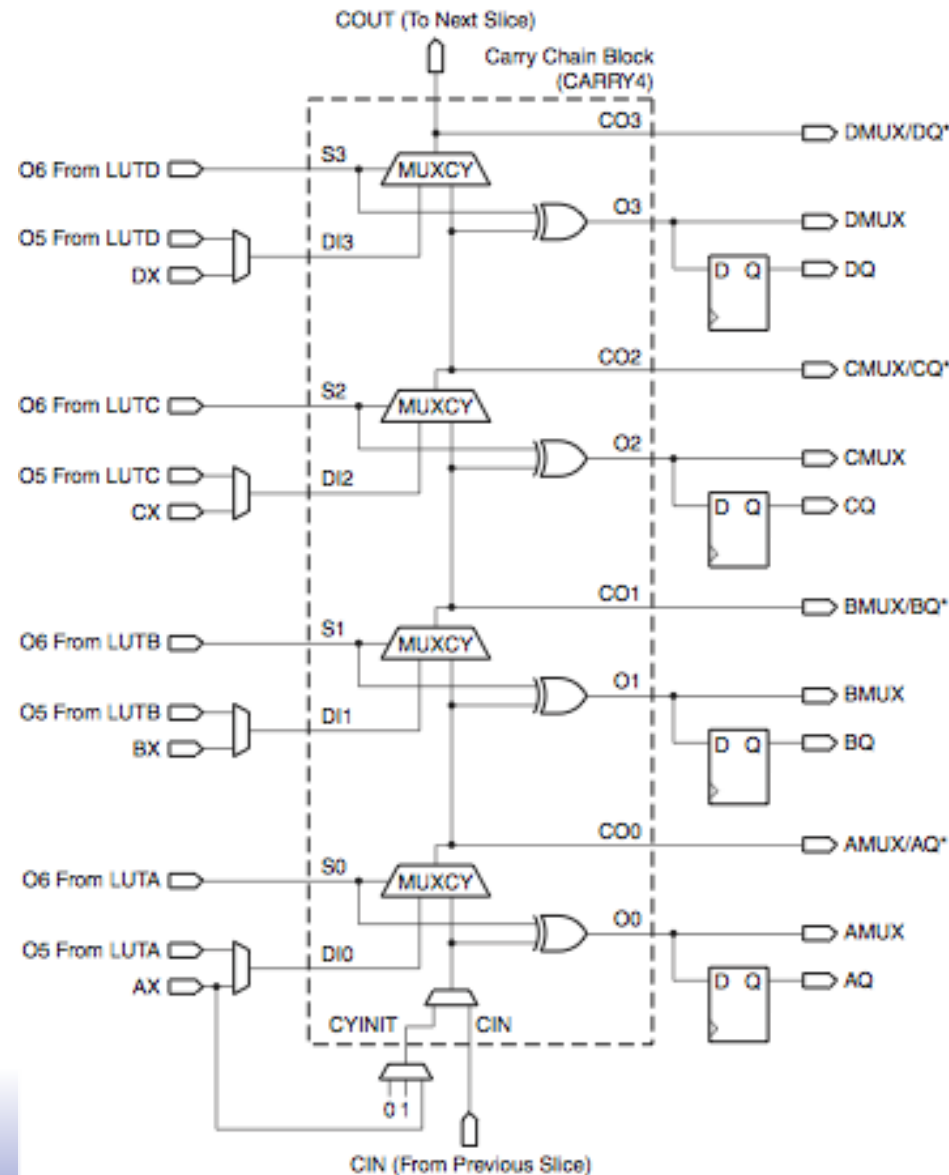
*Flip-flops unused by
LUTs can be used
standalone.*

Virtex Vertical Logic

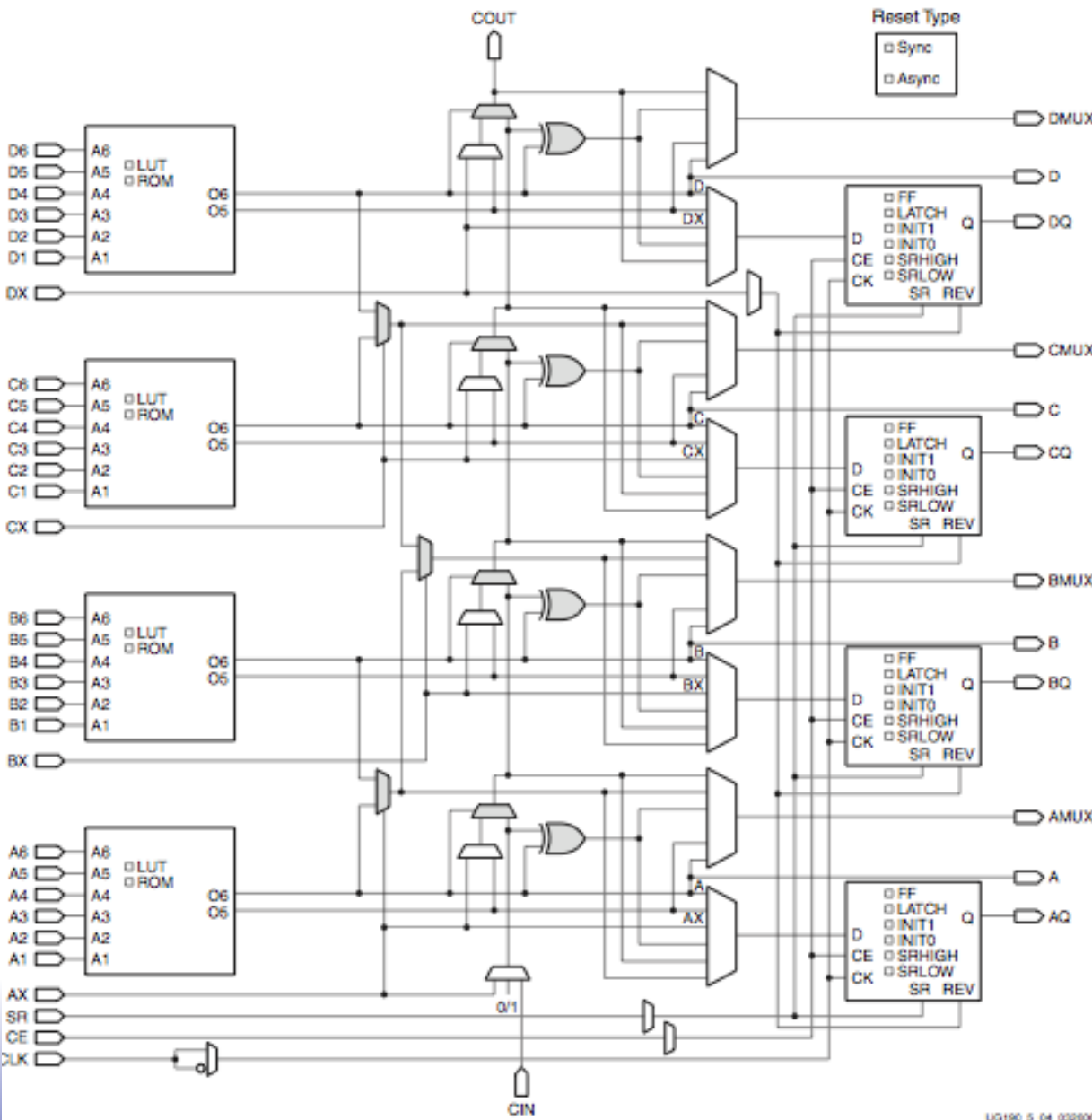
We can map ripple-carry addition onto carry-chain block.



The carry-chain block also useful for speeding up other adder structures and counters.



Putting it all together ... a SLICEL.

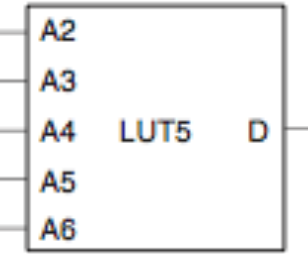


The previous slides explain all SLICEL features.

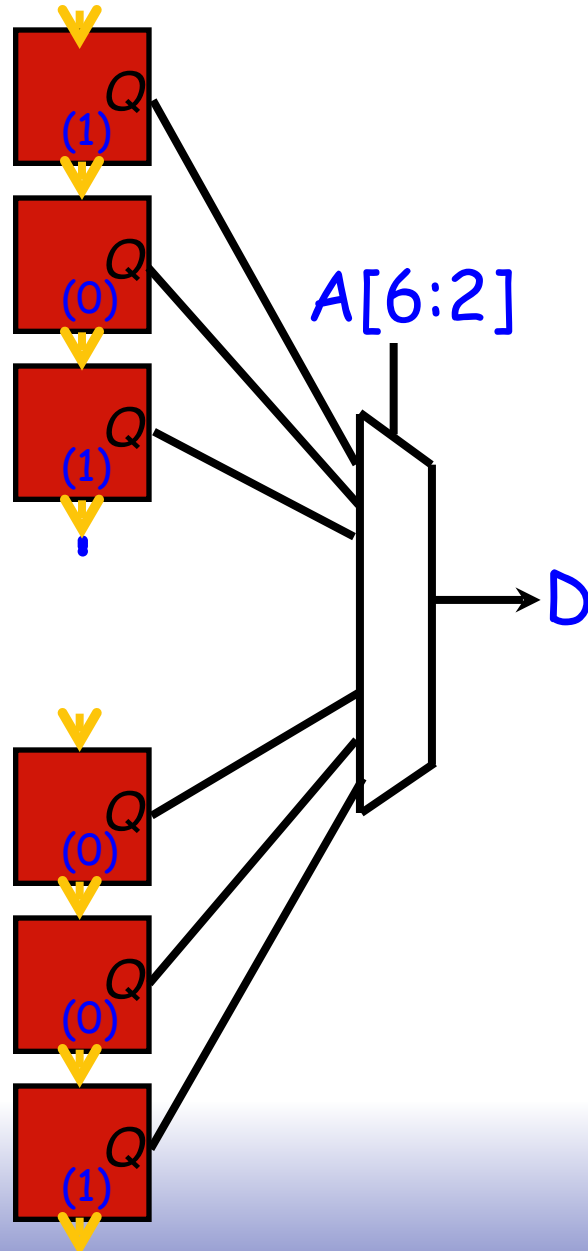
About 50% of the are SLICELs.

The other slices are SLICEMs, and have extra features.

Recall: 5-LUT architecture ...



| $A[6:2]$ | D |
|----------|-----|
| 00000 | 1 |
| 00001 | 0 |
| 00010 | 1 |
| ... | ... |
| 11101 | 0 |
| 11110 | 0 |
| 11111 | 1 |

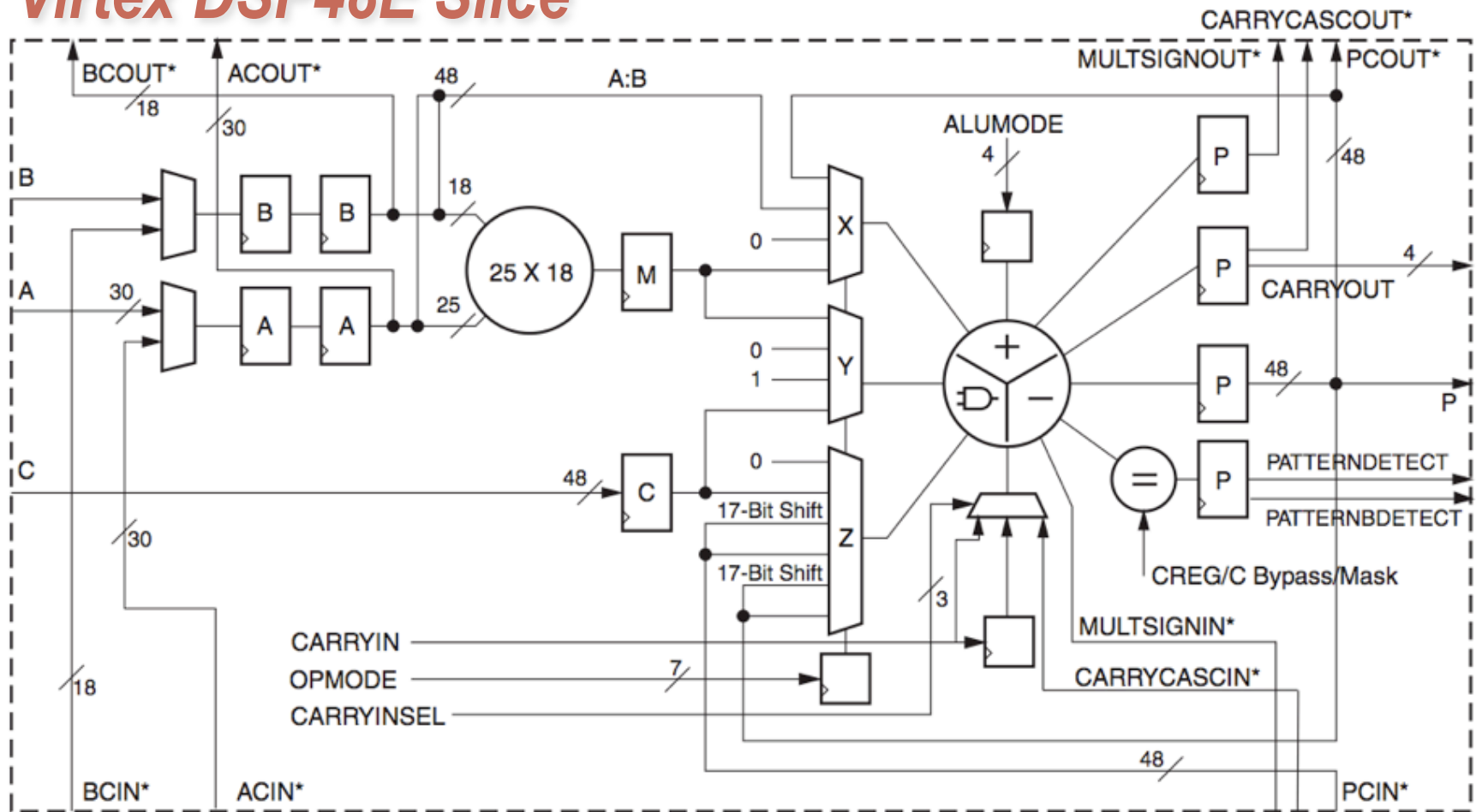


- 32 Latches.
- Configured to 1 or 0.

Some parts of a logic design need many state elements.

SLICEMs replace normal 5-LUTs with circuits that can act like 5-LUTs, but can alternatively use the 32 latches as RAM, ROM, shift registers.

Virtex DSP48E Slice



*These signals are dedicated routing paths internal to the DSP48E column. They are not accessible via fabric routing resources.

UG193_ct_01_032806

Efficient implementation of multiply, add, bit-wise logical.

To be continued ...

Throughout the semester, we will look at different FPGA features in-depth.

- Switch fabric
- Block RAM
- DSP48 (ALUs)
- Clocking
- I/O
- Serial I/O + PCI

