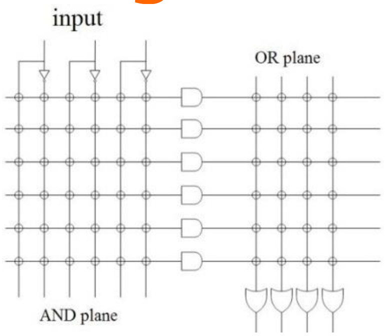
EEE4120F

High Performance Embedded Systems

Lecture 14

Programmable Logics & FPGAs (recap)

FPGA Interns



Lecturer:
Simon Winberg

Outline for today

- Recommended Reading on Verilog HDL
- Programmable Logic Devices
- What is so special about FPGAs
- FPGA interns
- Xilinx Slices

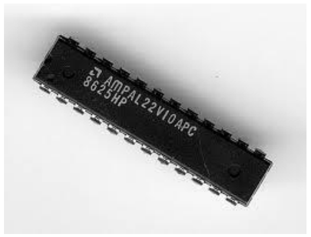
Recommended Reading

- ES2* provided an introduction to HDL and the Verilog HDL, and getting started in Xilinx Vivado and/or the open-source Icarus Verilog Simulator
- If you are not already familiar with Verilog or VHDL, you are encouraged to read Deepak Tala's (founder of ASIC-World) "Verilog Tutorial and Introduction to ASIC-World"
- I recommend ASIC-World.com as a first, go-to place for:
  - Solutions to Common HDL Problems,
  - Additional Tutorials, and
  - Examples of Useful Modules often needed in designs.

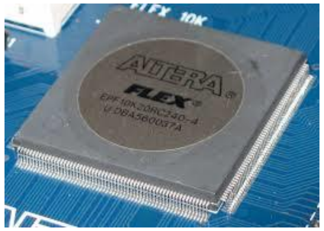
* Embedded System II EEE3096S 3rd year course

Programmable Logic Devices

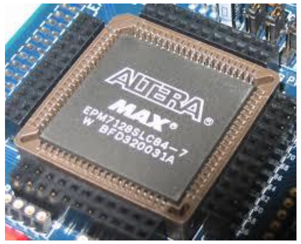
EEE4120F



PLA



PLD + SoC



CPLD

FPGA

Programmable Chips

- In comparison to hard-wired chips, a programmable chip can be reconfigured according to application or user needs
- Provides a means to use the same chip(s) for a variety of different applications.
- Makes programmable chips attractive for use in many products, e.g. prototyping products.
- Further benefits are:
  - Low starting cost (e.g. Web pack+ FPGA dev kit)
  - Risk reduction
  - Quick turnaround time

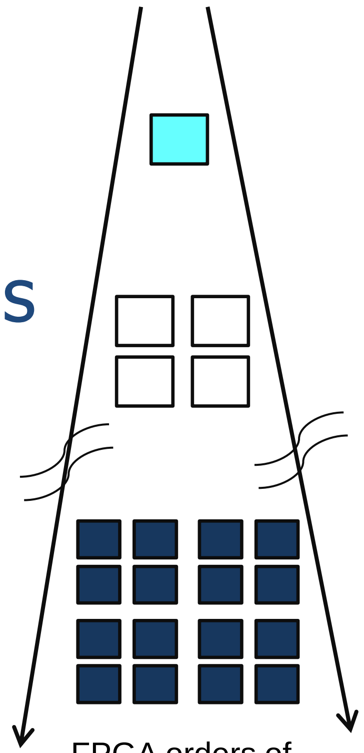
The term PLD refers to “Programmable Logic Device” which could technically be any of the programmable devices (i.e. PLA / CPLD / FPGA), in early work it often referred to a PLA but this is no longer a correct assumption.

ASICs vs. Programmable Chips

- Application Specific Integrated Circuit (or ASICs) have a longer *design cycle* and higher engineering cost than using programmable chips.
- Still a need for ASIC: faster performance and lower cost for high volume
- Generally, programmable chips are suited to low to medium product production. (e.g. product runs needing under 10,000 chips)

PLAs, CPLDs and FPGAs

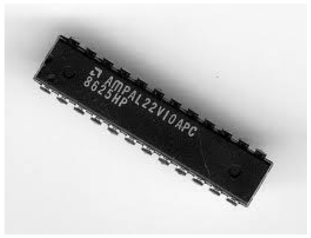
- Programmable logic chips variety in terms
  simple→complex cheap→expensive
- **PLA = Programmable Logic Array**
  - Simple: just AND and OR gates; but *Cheap*
- **CPLA = Complex PLA**
  - Midrange: compose interconnected PLAs
- **FPGA = Field Programmable Gate Array**
  - Complex: programmable logic blocks and programmable interconnects; but *Expensive*



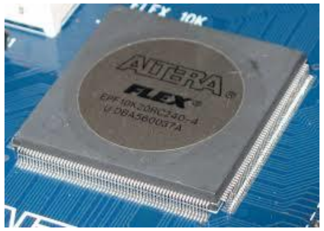
FPGA orders of magnitude larger than CPLD

Some examples of PLDs

PLA: TIBPAL22V10-7C from Texas Instruments is a commonly used



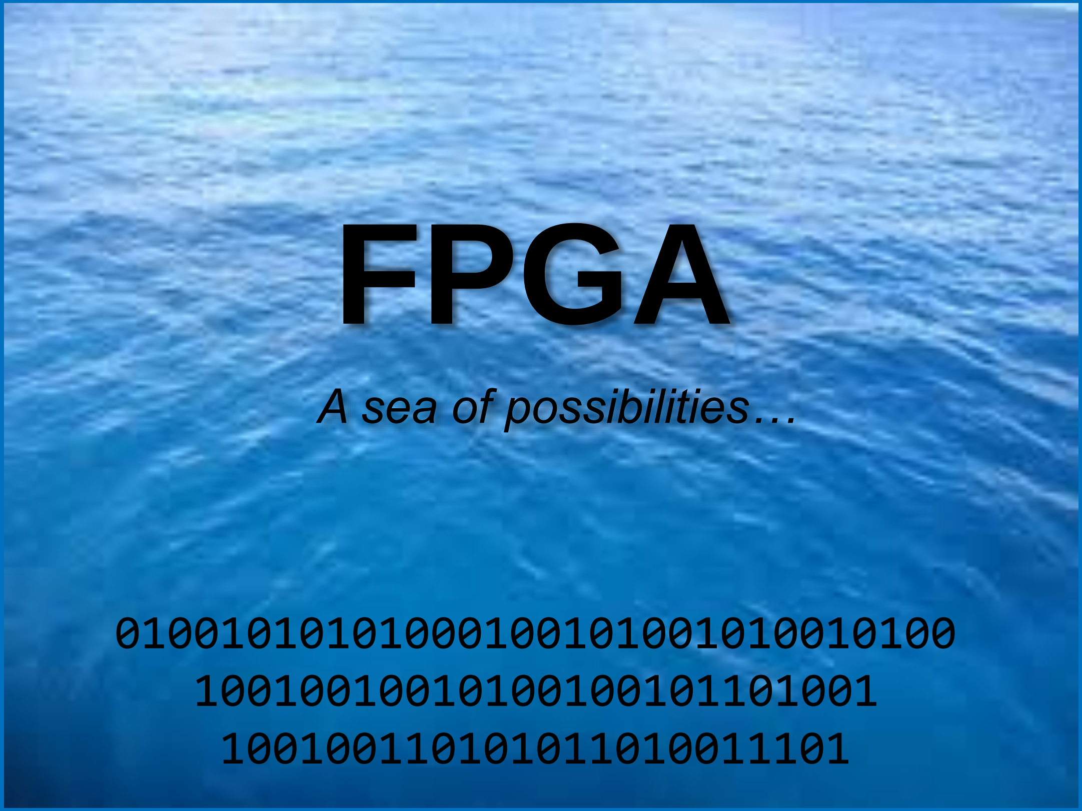
CPLD: The Altera MAXII and arguably the Altera FLEX as well



FPGA: Xilinx Spartan and Virtex range;
Altera Cyclone and
The Xilinx, Altera FPGA are probably the most commonly known manufacturers, others include: Lattice, Microsemi / Actel, Achronix

So what?
What is so special about FPGAs?

So what?
What is so special about FPGAs?



FPGAs – “A sea of possibilities”

- The huge number of logic elements (LEs) within these chips, and their many PIO pins, makes it possible to implement large & complex digital systems in them.
- The ease and speed of programming them provides the ability to rapidly change the hardware (within ms timing) to adapt to application needs.
- Greater potential for testing and tweaking designs before fabricating them as ICs

Any Drawbacks?

Here's just a few drawbacks

- Only does the digital part – still need analogue components, user interface, and circuitry that interacts with the outside world.
- Has a limited number of IO pins that can connect up with external signals.
- Susceptible to EM disturbances, PCB and other components needs to be suitably placed to avoid interfering with functioning of FPGA.
- Typically a slower clock than most fast CPUs nowadays (e.g. 100MHz clock speed).
- Typically has lots of pins that need to be soldered on, needing small track width and multilayer PCBs
- A specialized form of development, combines the challenges of both s/w and h/w
- Often can't achieve full utilization of PLBs
- Limitations of internal interconnects
- Place & route can take a long time to complete

Things can get rather… muddy!

Structure of FPGA

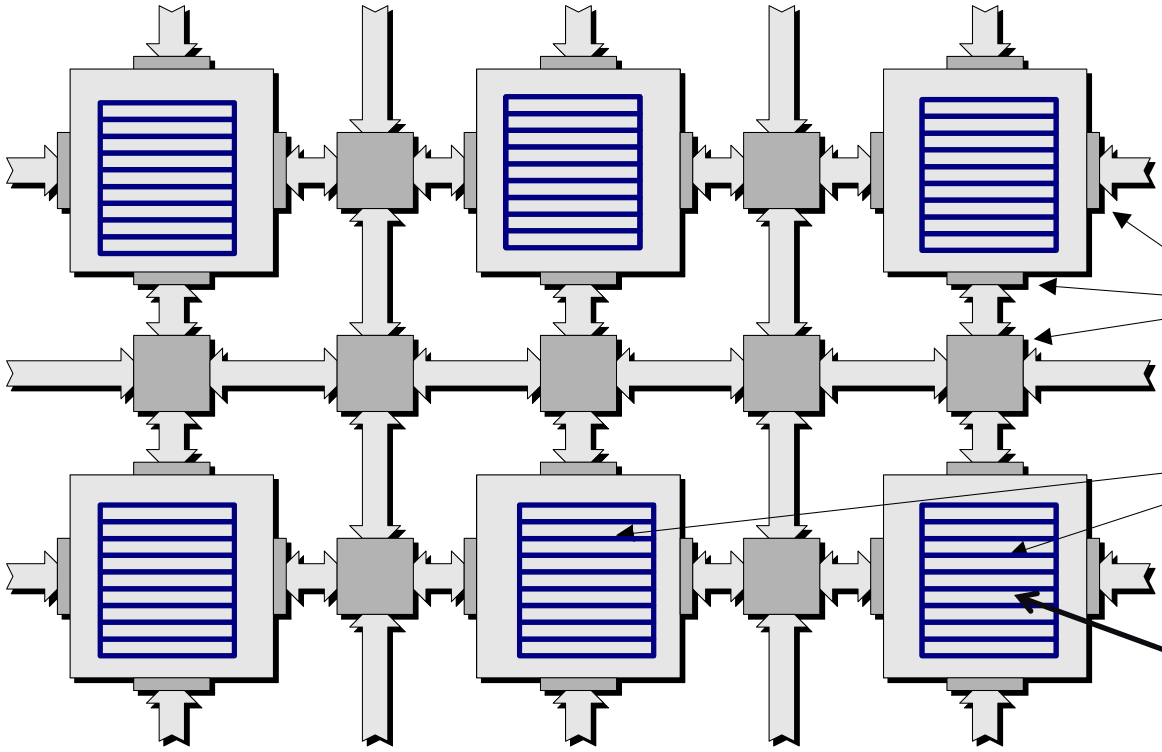
- A completely different architecture for PLAs was introduced in the mid-1980's that uses RAM-based lookup tables instead of AND-OR gates to implement combinational logic
- These devices are called field programmable gate arrays (FPGAs).
- The device consists of an array of configurable logic blocks (CLBs) surrounded by an array of I/O blocks
- FPGAs really don't have AND and OR gates, (they have a few) but rather just RAM look-up tables.

EEE4120F

FPGA Interns (recap)

Somewhat of a recap of ES2, but scan through these slides to ensure you are well versed in these issues.

FPGA internal structure (simplified)



Terminology note:
One often simplifies PLE to LE (i.e. logic element), and PLB to LB (or logic block). So this example has 6x LBs (assuming each block here is a LB not an LE).

Programmable interconnect

Programmable logic blocks

Programmable logic element (PLE) (or FPLE*)

Image adapted from Maxfield (2004)

Note: one programmable logic block (PLB) may contain a complex arrangement of programmable logic elements (PLE). In this example, it suggests the PLEs are LUTs (look-up tables), and there is just one LUT per PLB; but rather assume here there are actually multiple LUTs, let's say 9 of them, per PLB.

The size of a FPGA or programmable logic device (PLD) is measured in the number of LEs (i.e., Logic Elements) that it has.

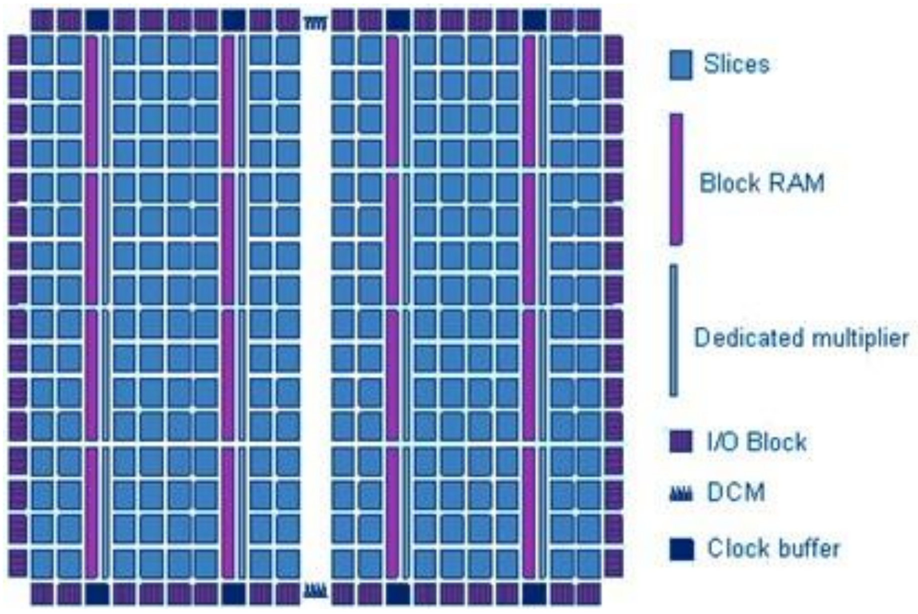
* FPLE = Field Programmable Logic Element

FPGA internal structure (more accurate)

The FPGA array structure (found in more modern FPGAs) comprises:

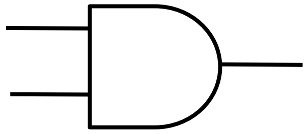
- Slices (which comprise *one or more* PLBs, kind of like a sub-FPGA), composed of:
  - LUTs to implement combinatorial logic, but may have other 'PLEs'
  - Flip-Flops (FF) to implement sequential logic (e.g. x=1; delay; y=1;)
- Routing Network (or 'routing net') to interconnects logic resources / PLBs
- I/O logic to communicate with the outside world (rest of the PCB)
- Clock Management:
  - Phase Locked Loops (PLLs)
  - Digital Clock Managers (DCMs)
- Hard-Macros: (specialized resources)
  - SRAMs blocks
  - Digital Signal Processing (DSP) cells (e.g. multiplier)
  - PCIe interface
  - Gigabit Transceivers
  - etc.

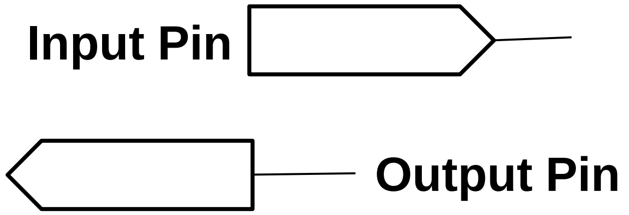
FPGAs may have one or more type of 'slice'. A basic slice comprises Look-Up Tables (LUTs) and flip flops



Logic Elements (LEs) – Remember your logic primitives

- You already know all your logic primitives…
  - The primitive logic gates
    - AND, OR, NOR, NOT, NOR, NAND, XOR
    - AND3, OR4, etc (for multiple inputs).
  - Pins / sources / terminators
    - Ground, VCC
    - Input, output
  - Storage elements
    - JK Flip Flops
    - Latches
  - Others items: delay, mux





Altera Quartus II representations

Disclaimer: In reality, nowadays, FPGAs often don't comprise individually routable LEs. LUTs are a more versatile approach. Often including a number of pre-build 'hard macros' (e.g. adders or multipliers) in PB that are commonly used in combinational logic designs. So, we are effectively considering LEs as primitive gates for education purposes.

Look Up Tables (LUTs)

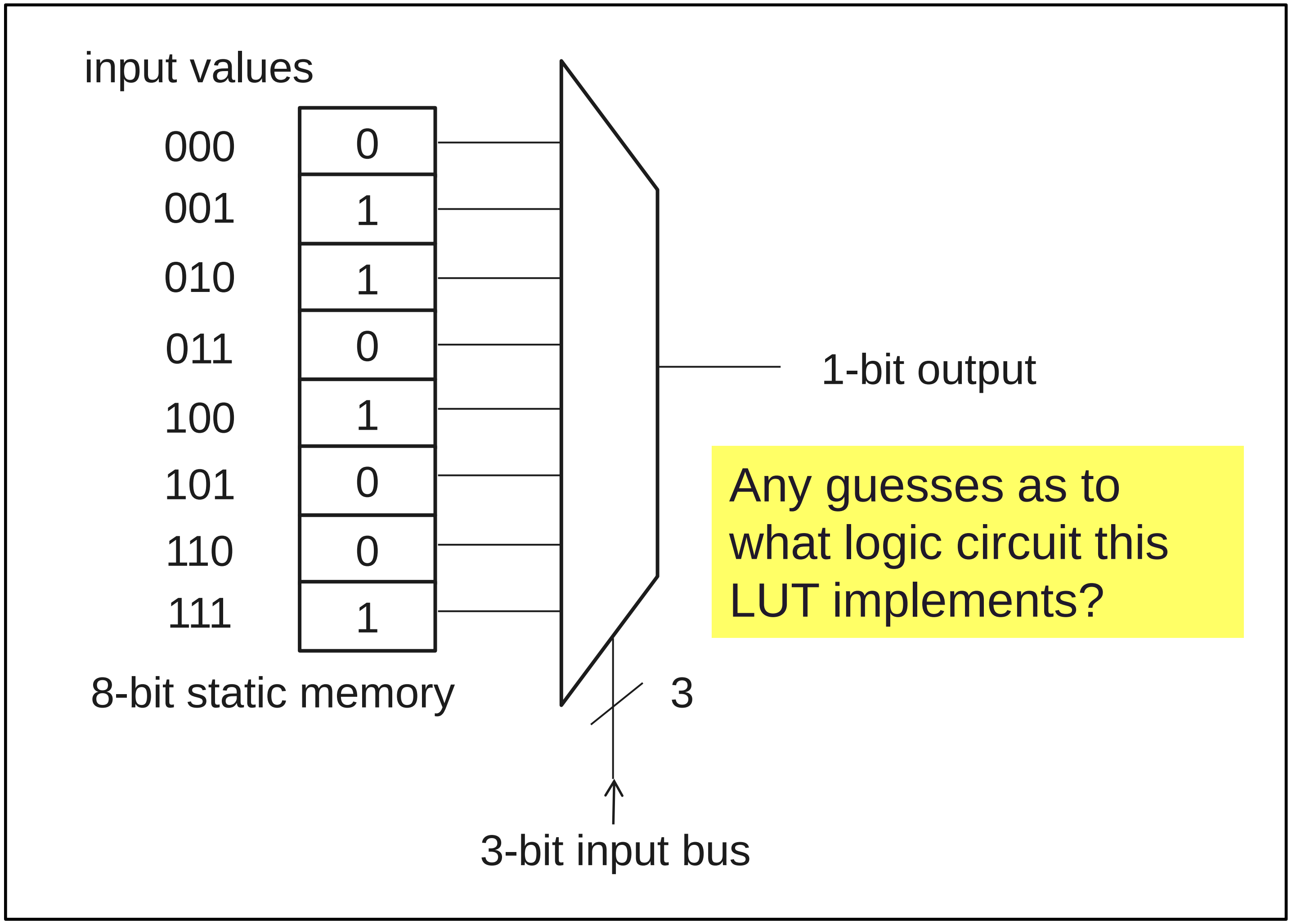
The usual strategy for implementing PLBs

- A simple but powerful approach to FPGA design is to use *lookup tables* for the PLBs. These are usually implemented as a combination of a multiplexer and memory (even just using NOR gates)
- Essentially, this approach is building complex circuits using truth tables (where each LUT enumerates a truth table)

Note: Most FPGAs are not really just a combination of LUTs connected with multiplexers (switching fabric). While an FPGA may well have lots of LUTs for general logic, they usually also have hardened operations / ‘hard-macros’ (usually DSP blocks) that implement efficient adders, comparators etc… see Mainstream PLBs a little later.

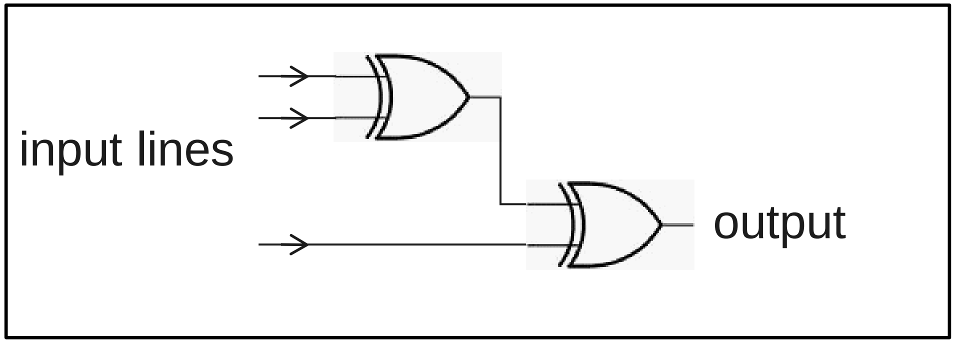
examples follows…

Simple 3-LUT implementation for a PLB



Any guesses as to what logic circuit this LUT implements?

Simple 3-LUT implementation for a PLB

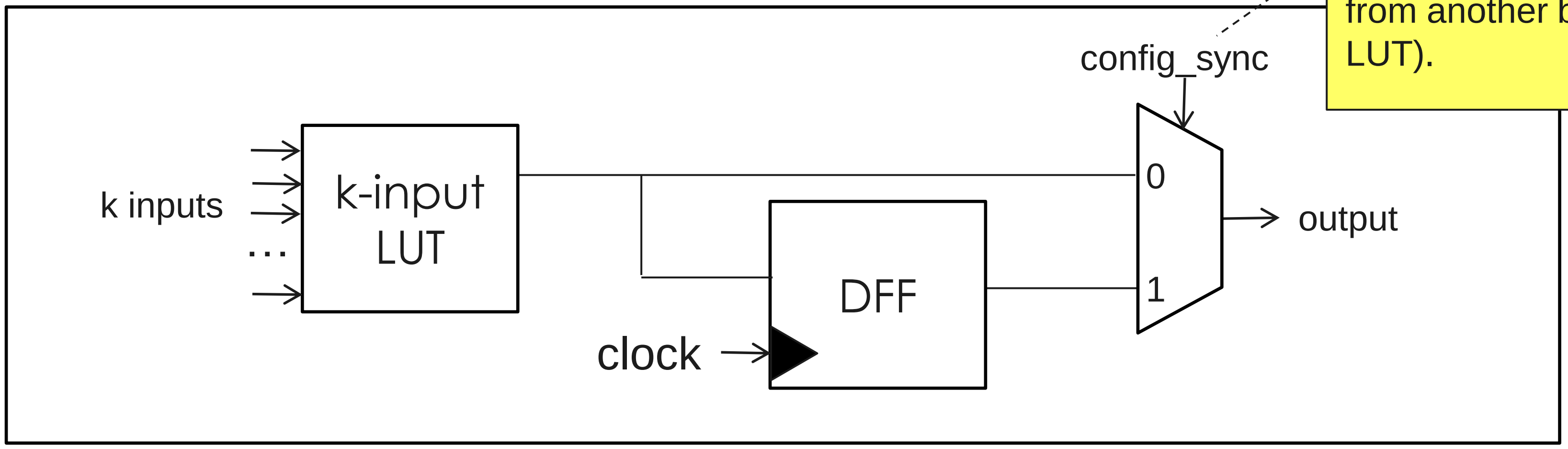


It's an XOR of the 3 input lines!!!

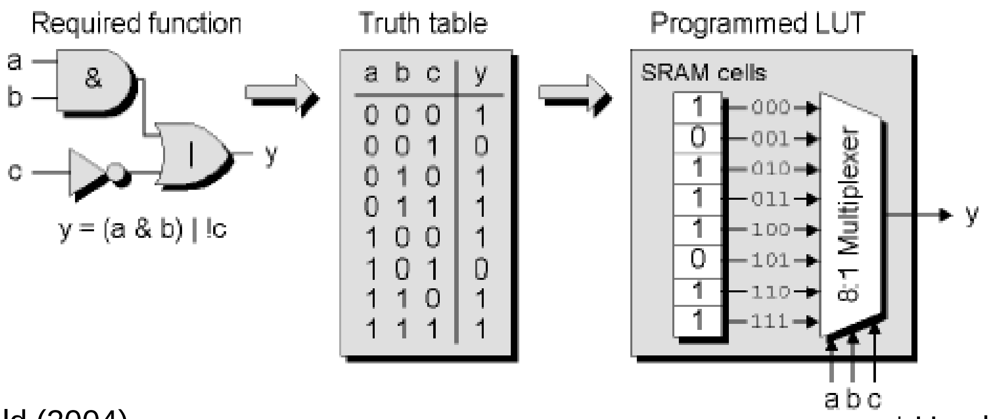
in	out
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

Mainstream* Programmable Logic Block (PLB)

Configure synchronous or asynchronous response (i.e. a line from another big LUT).



Another example for implementing an alternate logic function.



a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

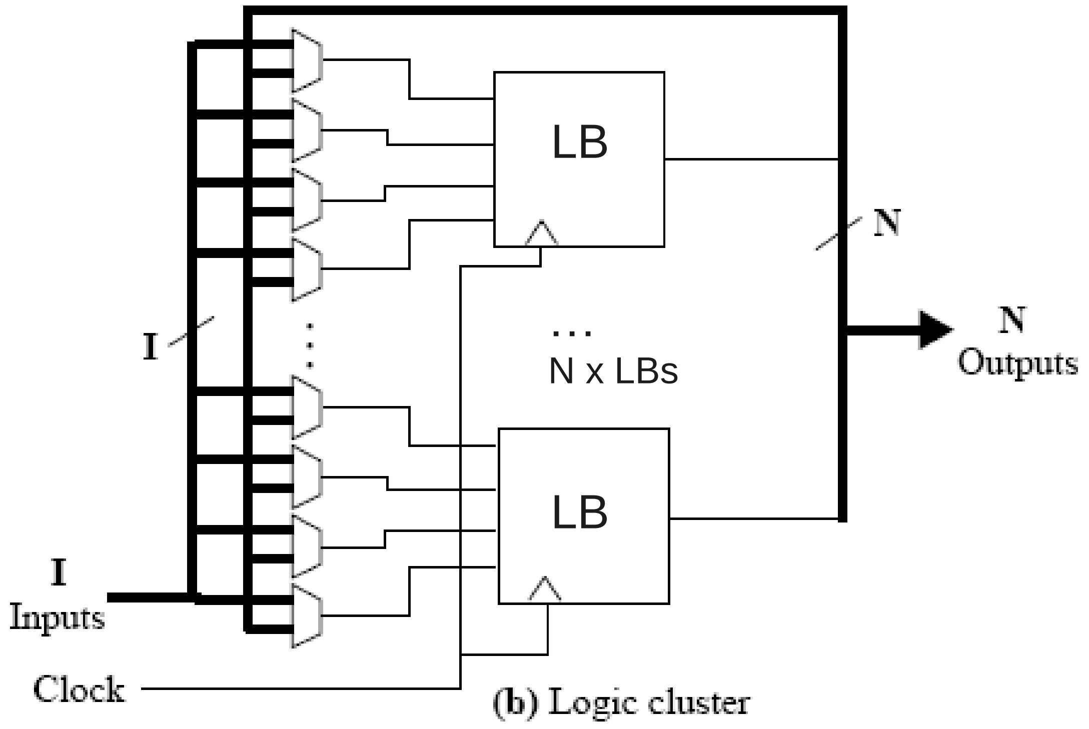
Image adapted from Maxfield (2004)

* Used by manufacturers like Xilinx

Logic block clusters (LBCs) and Configurable logic blocks (CLBs)

- Assume a k-input LUT for each logic block (LB)
- Assume N x LBs per logic cluster
- BLEs in each logic clusters are *fully connected* or *mostly* connected

The diagram shows the same input lines (I) are sent to each LB, in addition to each of the N LBs' output lines. Each LB operates on 4 input lines at a time, and a MUX is used to decide which input to sample. The MUXs may be configured from a separate LUT, or could be controlled by the LB it is connected to.



(b) Logic cluster

Diagram adapted from Sherief Reda (2007), EN2911X Lecture 2 Fall07, Brown University

Xilinx L and M Slices Approach for configurable logic blocks (CLBs)

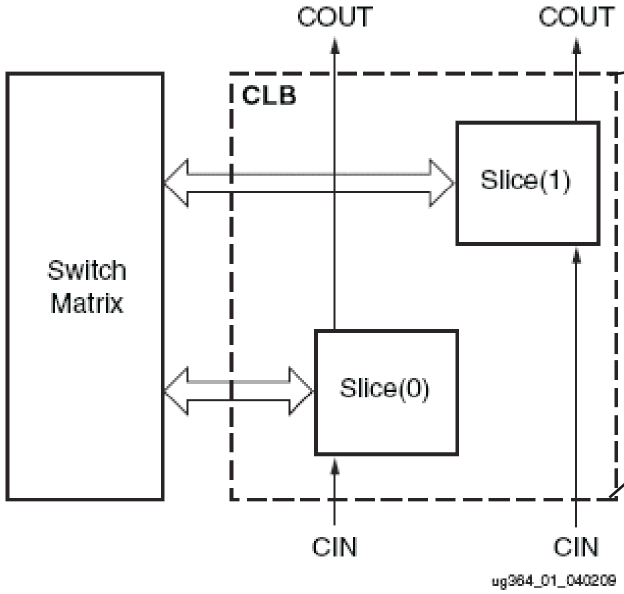


Figure 1: **Arrangement of Slices within the CLB**

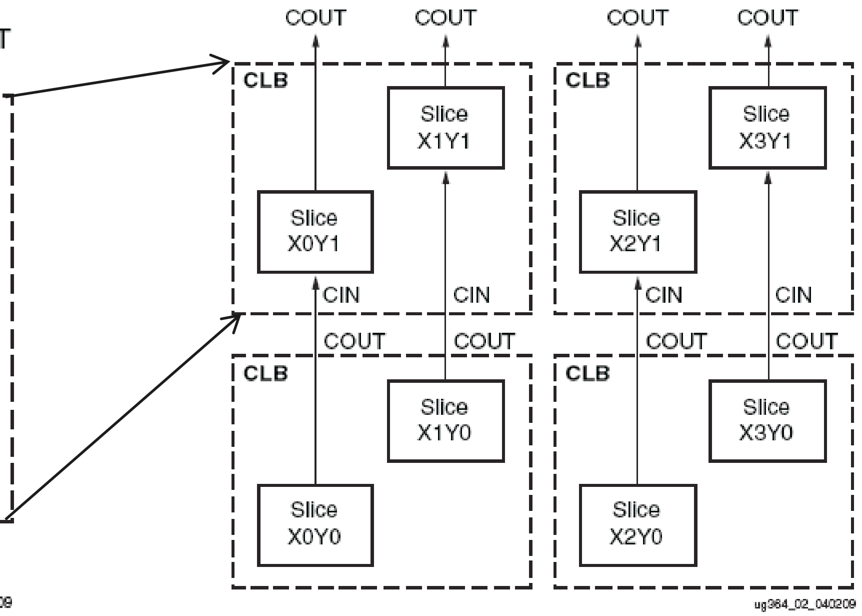


Figure 2: **Row and Column Relationship between CLBs and Slices**

“Every slice contains four logic-function generators (or LUTs), eight storage elements, wide-function multiplexers, and carry logic. These elements are used by all slices to provide logic, arithmetic, and ROM functions. In addition to this, some slices support two additional functions: storing data using distributed RAM and shifting data with 32-bit registers. Slices that support these additional functions are called SLICEM; others are called SLICEL. SLICEM represents a superset of elements and connections found in all slices. Each CLB can contain zero or one SLICEM. Every other CLB column contains a SLICEMs. In addition, the two CLB columns to the left of the DSP48E columns both contain a SLICEL and a SLICEM.”

Source: http://www.xilinx.com/support/documentation/user_guides/ug364.pdf pg 8

EEE4120F

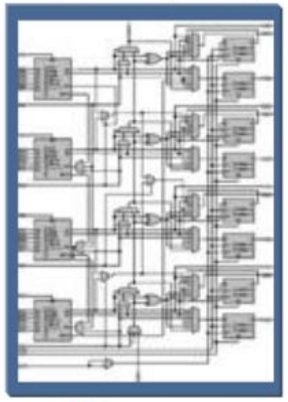
Xilinx Slices

(A more accurate view on
FPGA internal structures)

Spartan-6 CLB Logic Slices

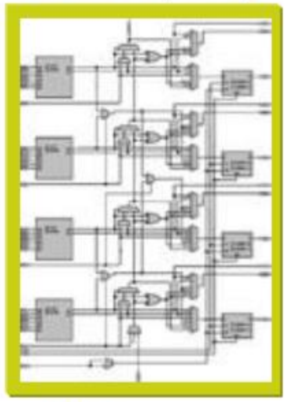
Note: Xilinx tends to use 'CLB' for 'PLB', basically means the same thing (textbooks often use the term PLB).

SliceM (25%)



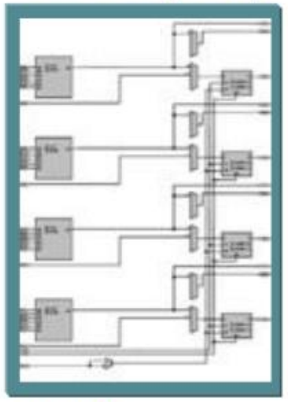
- LUT6
- 8 Registers
- Carry Logic
- Wide Function Muxes
- Distributed RAM / SRL logic

SliceL (25%)



- LUT6
- 8 Registers
- Carry Logic
- Wide Function Muxes

SliceX (50%)



- LUT6
- Optimized for Logic
- 8 Registers

Slice mix chosen for the optimal balance of Cost, Power & Performance

XILINX®

The Xilinx Spartan6 is a fairly good representative of how the logic in a FPGA is structured and connected. The Xilinx FPGAs are based around slides, where a slice (a substantive portion of the FPGA) is structured to support particular design characteristics. For example SliceX provides maximum flexibility for arbitrary logic but not designed around carry logic; whereas the SlideL supports carry logic and bigger muxes (to allow a LB to tap into more wires) but these features might be redundant for less complex designs.

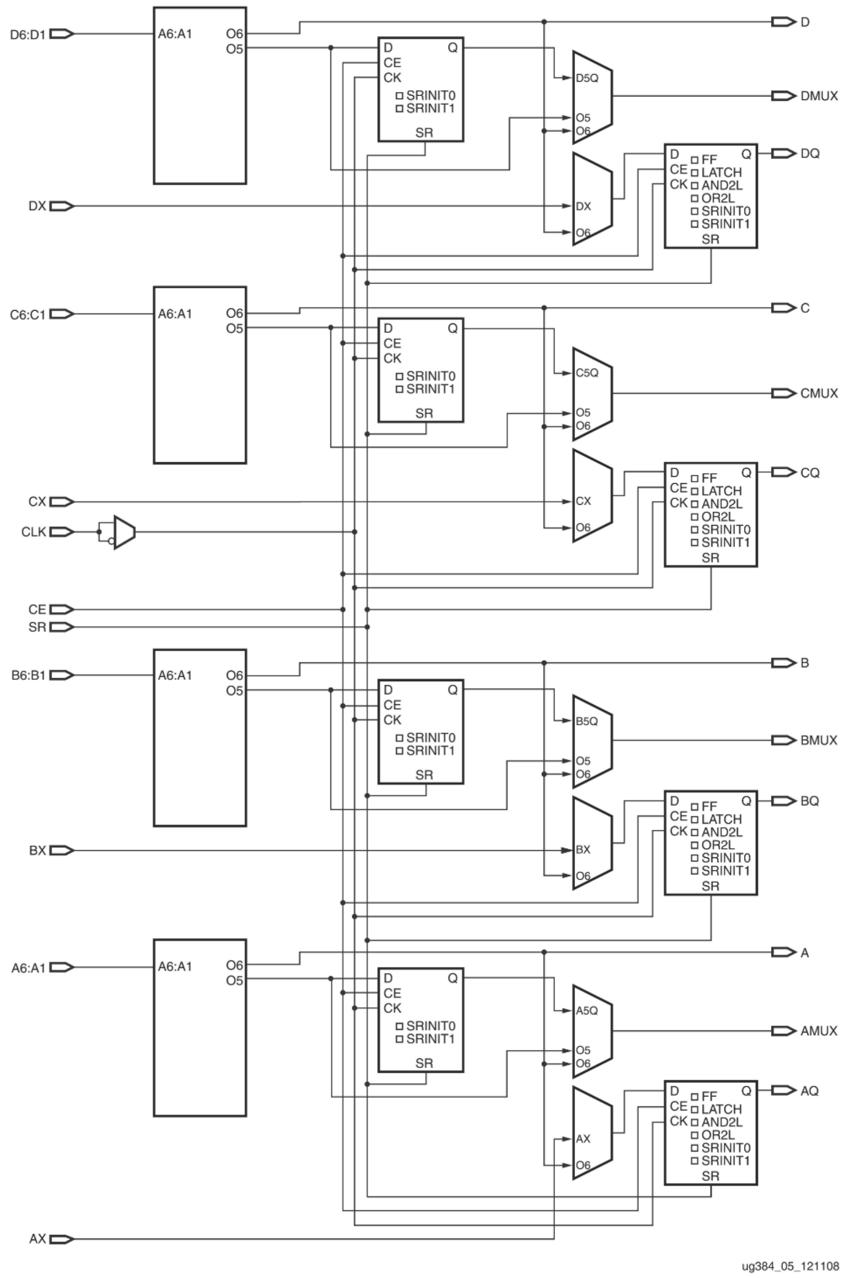


Figure 5: **Diagram of SLICEX**

SLICEX slices are generally the most basic of slices, but also the most flexible. The LEs essentially contain LUTs and flip flops (e.g. to store registers and to configure clocked or un-clocked LE operation)

Source: https://www.xilinx.com/support/documentation/user_guides/ug384.pdf

CLB Overview

XILINX.

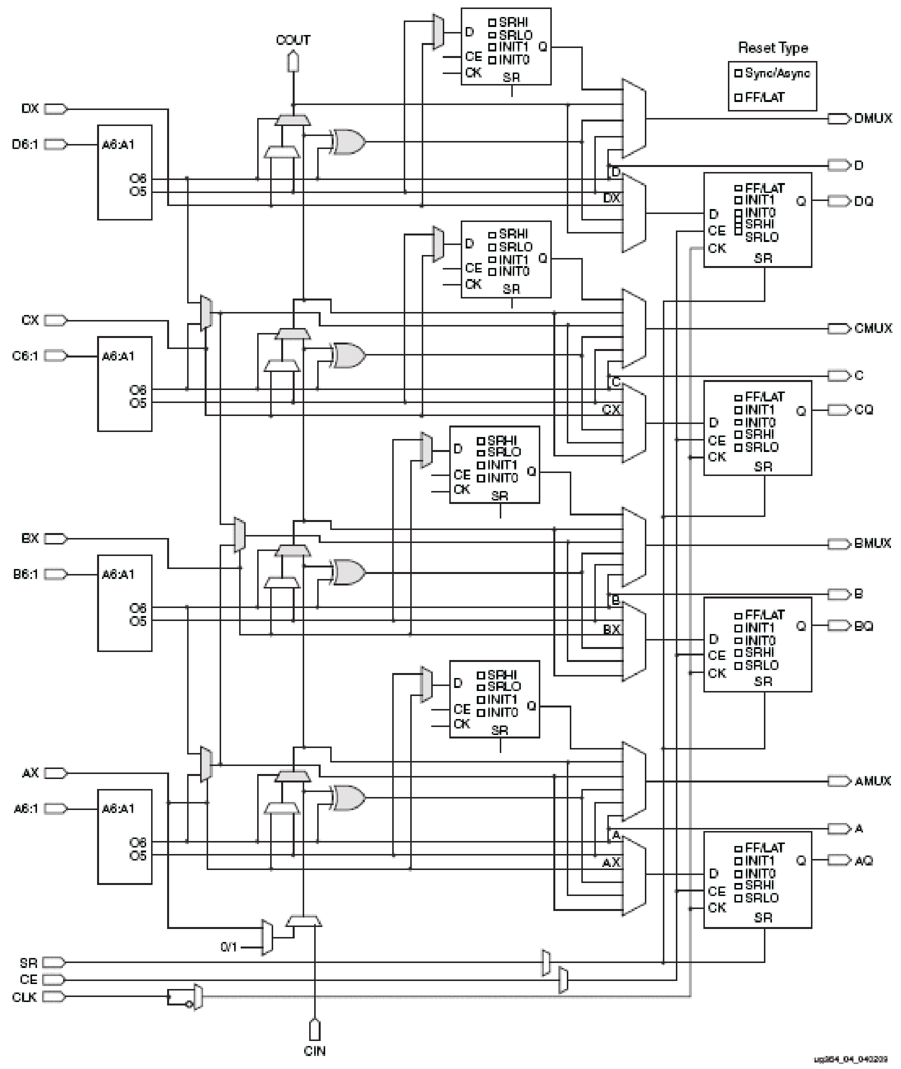


Figure 4: **Diagram of SLICEL**

10 www.xilinx.com **Virtex-6 FPGA CLB User Guide** UG364 (v1.1) September 16, 2009

SLICEL slices contain the standard set of LEs for the particular FPGA concerned. As the diagram shows, it looks a little more complicated than the SLICEX but it is less complicated than the design of a SLICEM (see next slide).

Source: http://www.xilinx.com/support/documentation/user_guides/ug364.pdf pg 10

XILINX. CLB Overview

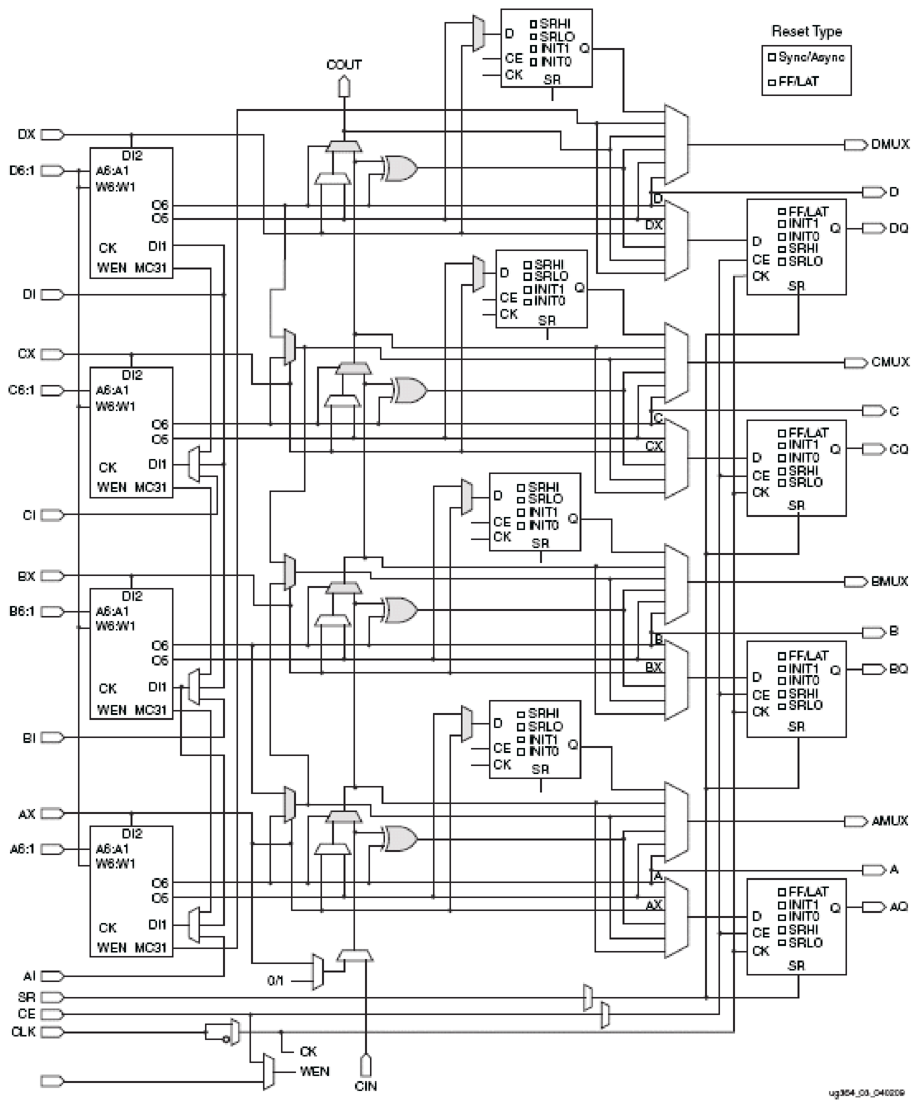
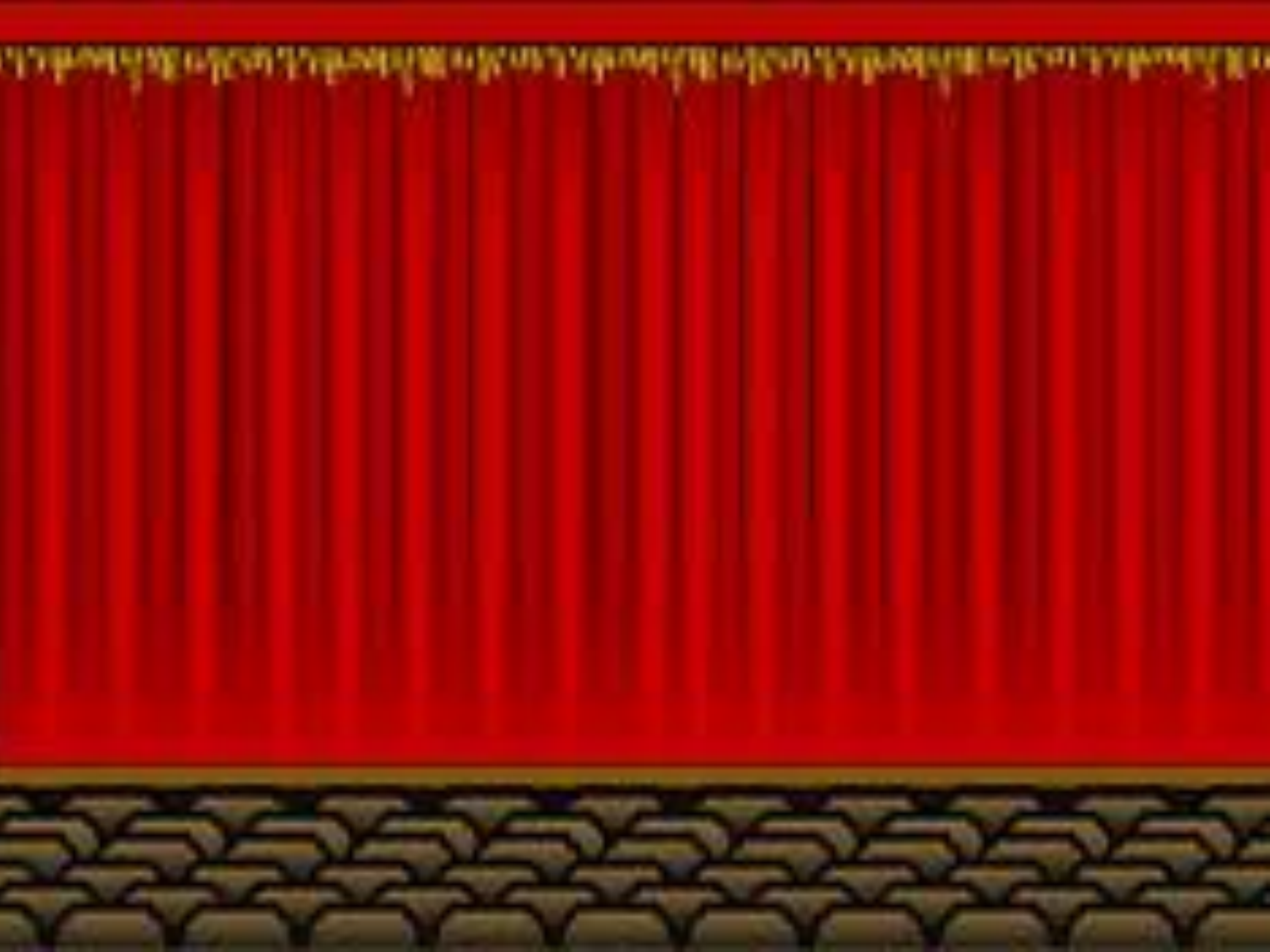


Figure 3: **Diagram of SLICEM**

Virtex-6 FPGA CLB User Guide
UG364 (v1.1) September 16, 2009
www.xilinx.com
9

SLICEM slices support additional functions; they are a *superset* of SLICELs; i.e. the have all the standard LEs plus some additions.

Source: http://www.xilinx.com/support/documentation/user_guides/ug364.pdf pg 9



next episode:
Verilog Coding

Onwards to Xilinx Vivado and [recapping] Verilog programming …

Disclaimers and copyright/licensing details

Image sources:

PLD illustration on title slide - Wikipedia Open Commons
FPGA chip illustrations - Wikipedia Open Commons
Stuck in Mud – fickr (CC2 for free reuse and modification)
Typing ninja – Pixabay http://pixabay.com/ (public domain)