Latvijas Universitāte
Datorikas fakultāte

FPGA

Kurss "Ievads digitālajā projektēšanā"

Lekcija 17.12.2010

Autors: **Artis Mednis**

Rīga 2010

What is FPGA?

- **F**ield-**P**rogrammable **G**ate **A**rray
- Semiconductor device
- Can be configured by the customer or designer after manufacturing
- To program an FPGA you specify how you want the chip to work:
  - logic circuit diagram
  - source code in a hardware description language (HDL)

Primary using of FPGA's

- To implement any logical function
- But... similar tasks could be solved using **A**pplication-**S**pecific **I**ntegrated **C**ircuit (ASIC). Why we need FPGA?
- Advantage of FPGA:
  - ability to update the functionality after shipping

Inside FPGA

- Programmable logic components called "logic blocks"
- Hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"
- Logic blocks can be:
  - configured to perform complex combinational functions
  - simple logic gates like AND and XOR
  - memory elements:
    - simple flip-flops
    - complete blocks of memory

FPGA – pro & contra

- Advantages:
  - shorter time to market
  - ability to re-program in the field to fix bugs
  - lower non-recurring (one-time) engineering costs
- Disadvantages:
  - usually slower than their fixed ASIC counterparts
  - draw more power
  - achieve less functionality using a given amount of circuit complexity

Optimal solution?

- First developing hardware on ordinary FPGA's...
- ... then manufacture final version so it can no longer be modified after the design has been committed ☺
- **Discussion – is it OK or not?**

A little bit of history

- Roots of FPGAs - **C**omplex **P**rogrammable **L**ogic **D**evices (CPLD's) – 1980s
- First FPGA – 1984 – Ross Freeman, Xilinx co-founder
- Scalability:
  - CPLD - from several thousand to tens of thousands of logic gates
  - FPGA - from tens of thousands to several million of logic gates

Differences between CPLD & FPGA

- CPLD:
  - restrictive structure:
    - one or more programmable logic arrays
    - relatively small number of clocked registers
  - less flexibility
  - more predictable timing delays
  - higher logic-to-interconnect ratio
- FPGA:
  - dominated by interconnect
  - more flexible
  - more complex to design

More about differences

- FPGA:
  - higher-level embedded functions:
    - adders
    - multipliers
  - embedded memories
  - logic blocks implement decoders
  - logic blocks implement mathematical functions
  - partial re-configuration (one portion of the device is re-programmed while other portions continue running)

Modern development

- Coarse-grained architectural approach:
  - logic blocks and interconnects of traditional FPGA’s
  - embedded microprocessors and related peripherals
  - complete "system on a programmable chip“
- Alternate approach:
  - "soft" processor cores that are implemented within the FPGA logic

More about modern development

- Ability to be reprogrammed at "run time”:
  - idea of reconfigurable computing or reconfigurable systems
  - CPU’s that reconfigure themselves to suit the task at hand
- Software-configurable microprocessors:
  - hybrid approach
  - array of processor cores
  - FPGA-like programmable cores
  - all this on the same chip

Soft processors

- Hard (embedded) CPU core will outperform a soft-core CPU

- Xilinx:
  - MicroBlaze – 32 bit RISC architecture
  - PicoBlaze – 8 bit RISC architecture

Applications of FPGA's

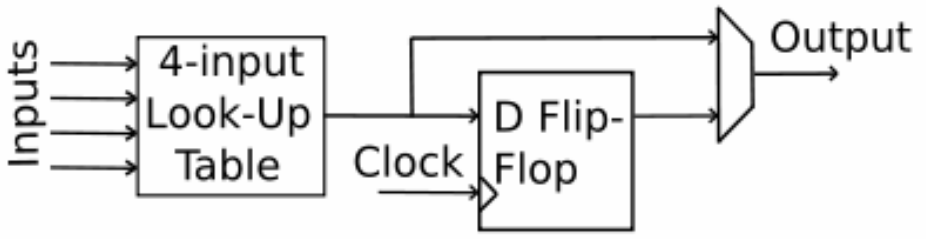
- Digital signal processing
- ASIC prototyping
- Computer vision
- Cryptography
- Computer hardware emulation
- ...

- Best suited for parallel computing
- High performance computing limited by the extremely long turn-around times of current design tools (4-8 hours for even minor changes of source code)

- Pārtraukums 10 minūtes

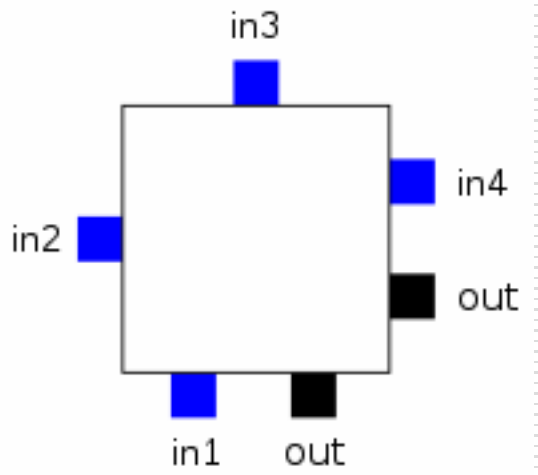
Architecture I

- Typical:
  - array of configurable logic blocks (CLB's)
  - routing channels
  - multiple I/O pads may fit into the height of one row or the width of one column in the array
  - all the routing channels have the same width (number of wires)
- Logic block:
  - 4-input lookup table (LUT) (now 6)
  - flip-flop



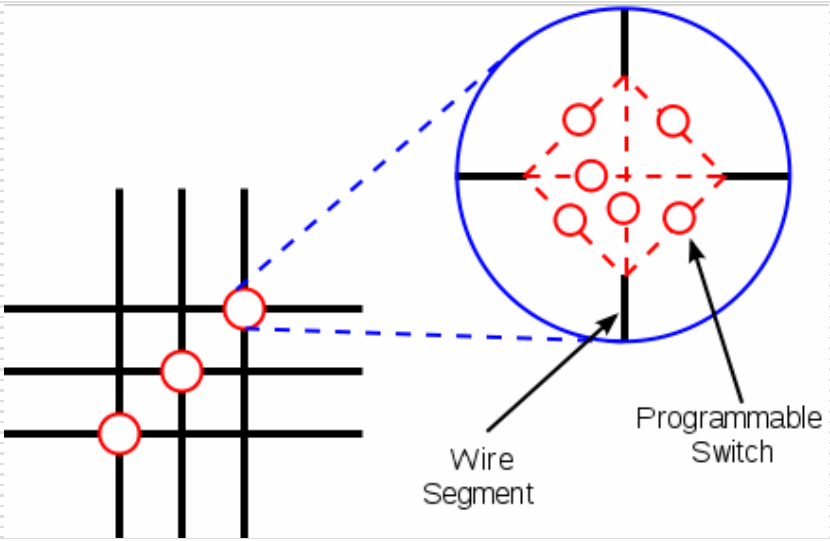
Architecture II

- Clock signals are routed via special-purpose routing networks
- Each input is accessible from one side of the logic block
- Output pin can connect to routing wires in both the channel to the right and the channel below the logic block
- Each logic block output pin can connect to any of the wiring segments in the channels adjacent to it



Architecture III

- Each wiring segment spans only one logic block before it terminates in a switch box
- When a wire enters a switch box, there are three programmable switches that allow it to connect to three other wires in adjacent channel segments
- The topology of switches is planar or domain-based



Design & programming I

- Hardware description language (HDL):
  - most common – VHDL & Verilog
  - easier to work with when handling large structures
- Schematic design:
  - easier visualisation of a design
- **Discussion – what is your favorite design approach?**

Design & programming II

- Process:
  - generation of netlist
  - place-and-route
  - timing analysis
  - simulation
  - generation of binary file
  - transfer:
    - direct to FPGA (via JTAG)
    - to external memory device (EEPROM)

Design & programming III

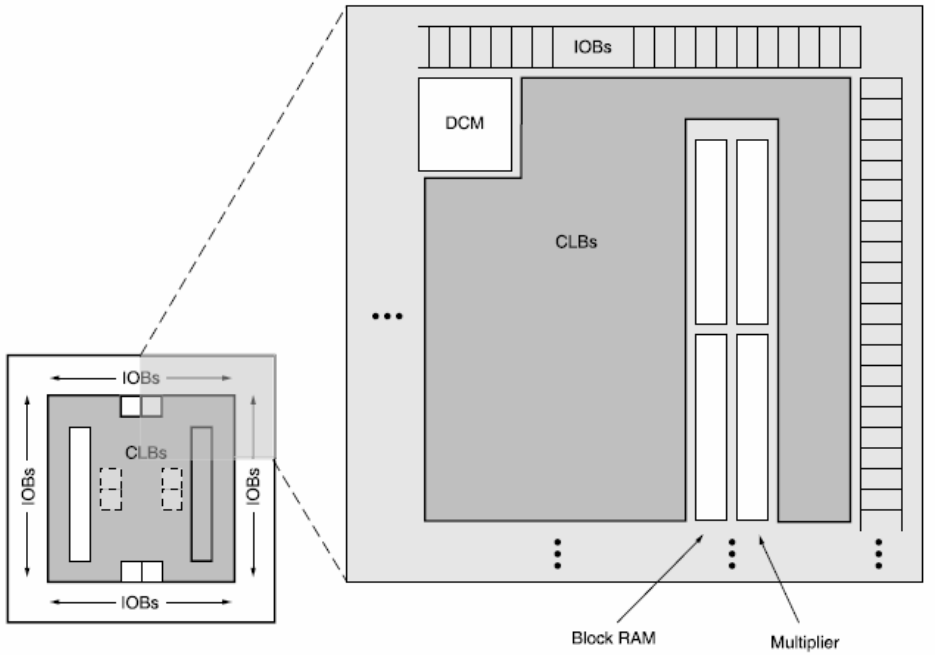
- Simplifying of design:
  - libraries of predefined complex functions
  - circuits that have been tested and optimized:
    - IP Cores
    - OpenCores
- Multiple simulations:
  - creating test benches to simulate the system and observe results
  - repeated to confirm the synthesis proceeded without errors
  - propagation delays can be added and the simulation run again

Manufacturers

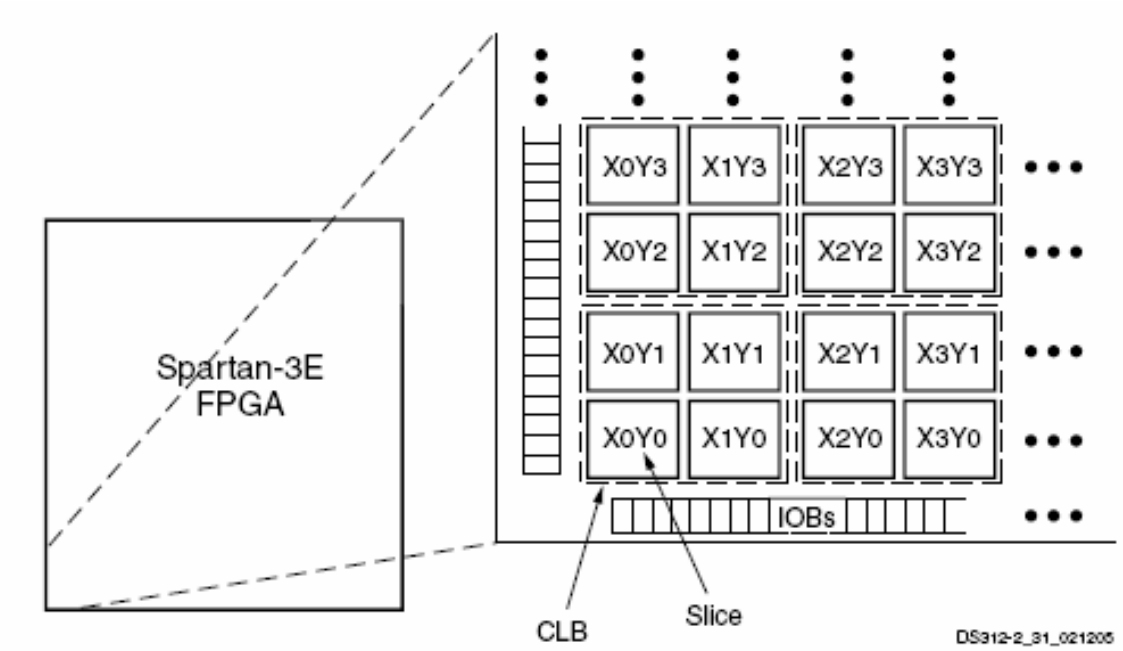
- Leaders:
  - Altera
  - Xilinx
- Others:
  - Lattice Semiconductor
  - Actel
  - Atmel
  - QuickLogic

XC3S500E (Spartan 3E) I

- Architecture:
  - Configurable Logic Blocks (CLBs)
  - Input/Output Blocks (IOBs)
  - Block RAM
  - Multiplier Blocks
  - Digital Clock Manager (DCM) Blocks

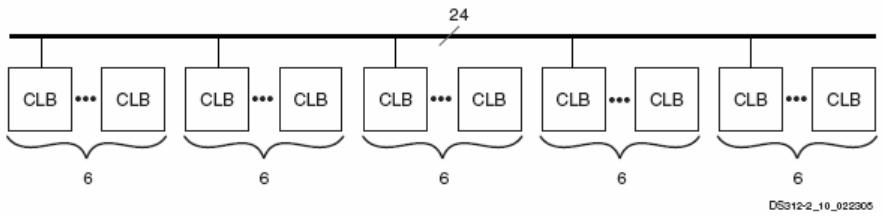
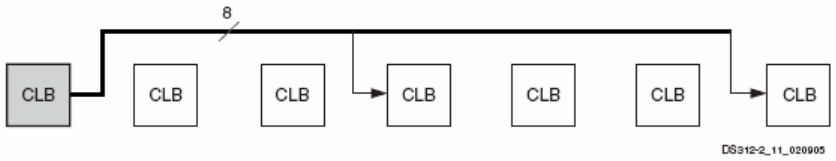
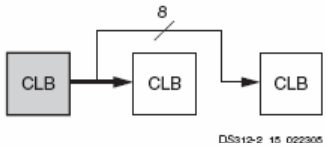
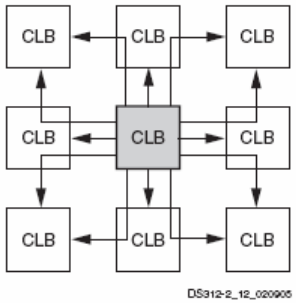


XC3S500E (Spartan 3E) II

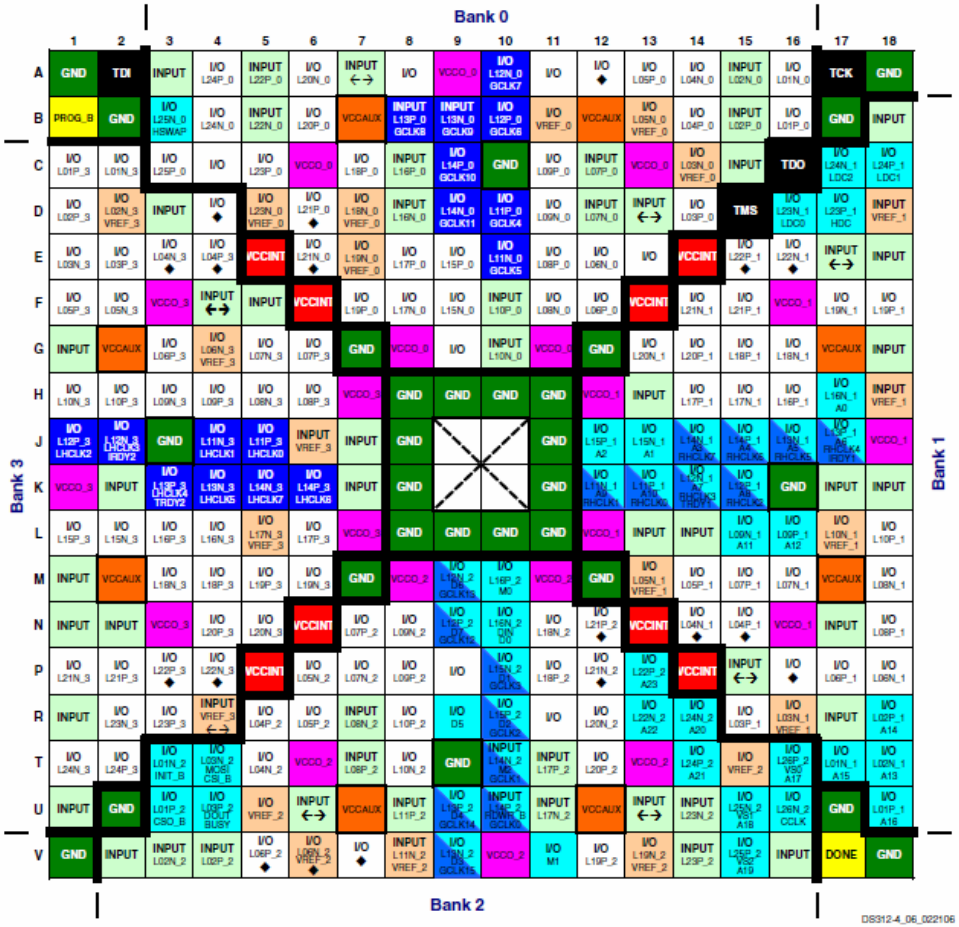


Device	CLB Rows	CLB Columns	CLB Total(1)	Slices	LUTs / Flip-Flops	Equivalent Logic Cells	RAM16 / SRL16	Distributed RAM Bits
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496

XC3S500E (Spartan 3E) III

Horizontal and Vertical Long Lines (horizontal channel shown as an example)	
Horizontal and Vertical Hex Lines (horizontal channel shown as an example)	
Horizontal and Vertical Double Lines (horizontal channel shown as an example)	
Direct Connections	

XC3S500E (Spartan 3E) IV



102-120	**I/O:** Unrestricted, general-purpose user I/O
47-48	**INPUT:** Unrestricted, general-purpose input pin
2	**CONFIG:** Dedicated configuration pins
18	**N.C.:** Not connected. Only the XC3S500E has these pins (◆).
46	**DUAL:** Configuration pin, then possible user-I/O
16	**CLK:** User I/O, input, or global buffer input
4	**JTAG:** Dedicated JTAG port pins
28	**GND:** Ground
20-21	**VREF:** User I/O or input voltage reference for bank
20	**VCCO:** Output voltage supply for bank
8	**VCCINT:** Internal core supply voltage (+1.2V)
8	**VCCAUX:** Auxiliary supply voltage (+2.5V)

LUT6 “price/performance”

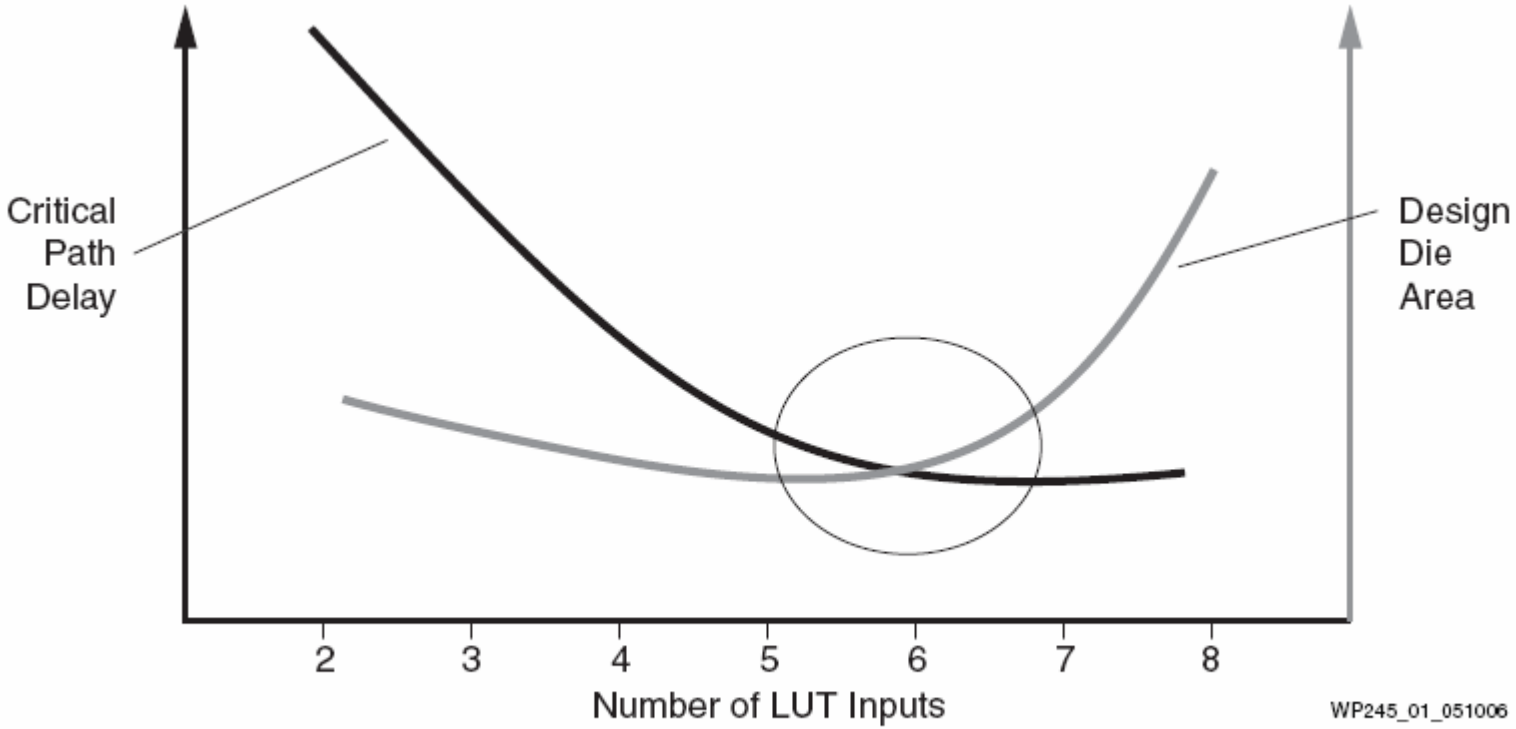


Figure 1: **Making the Right Trade-off for LUT Input Architecture at 65 nm**

LUT6 MUX

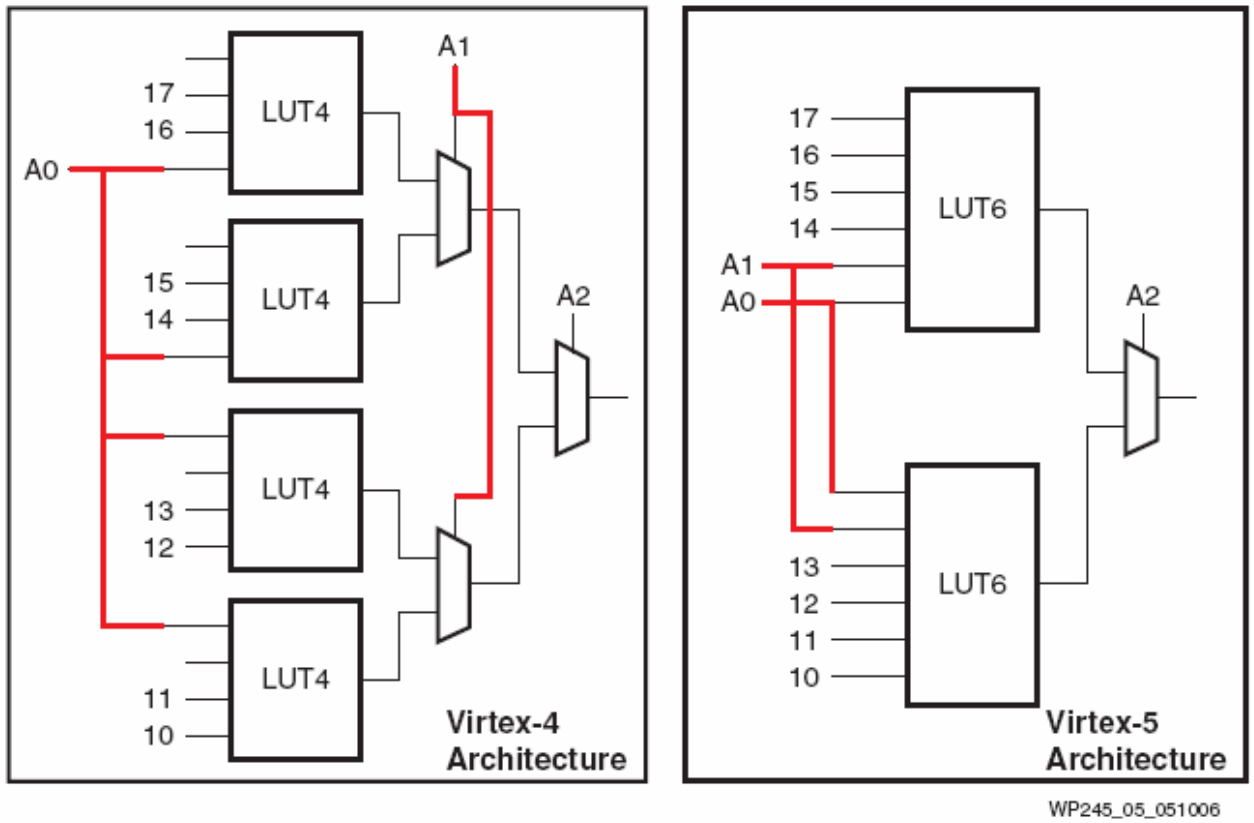


Figure 5: **8:1 MUX Implementation Using Virtex-4 and Virtex-5 Architectures**

LUT6 RAM

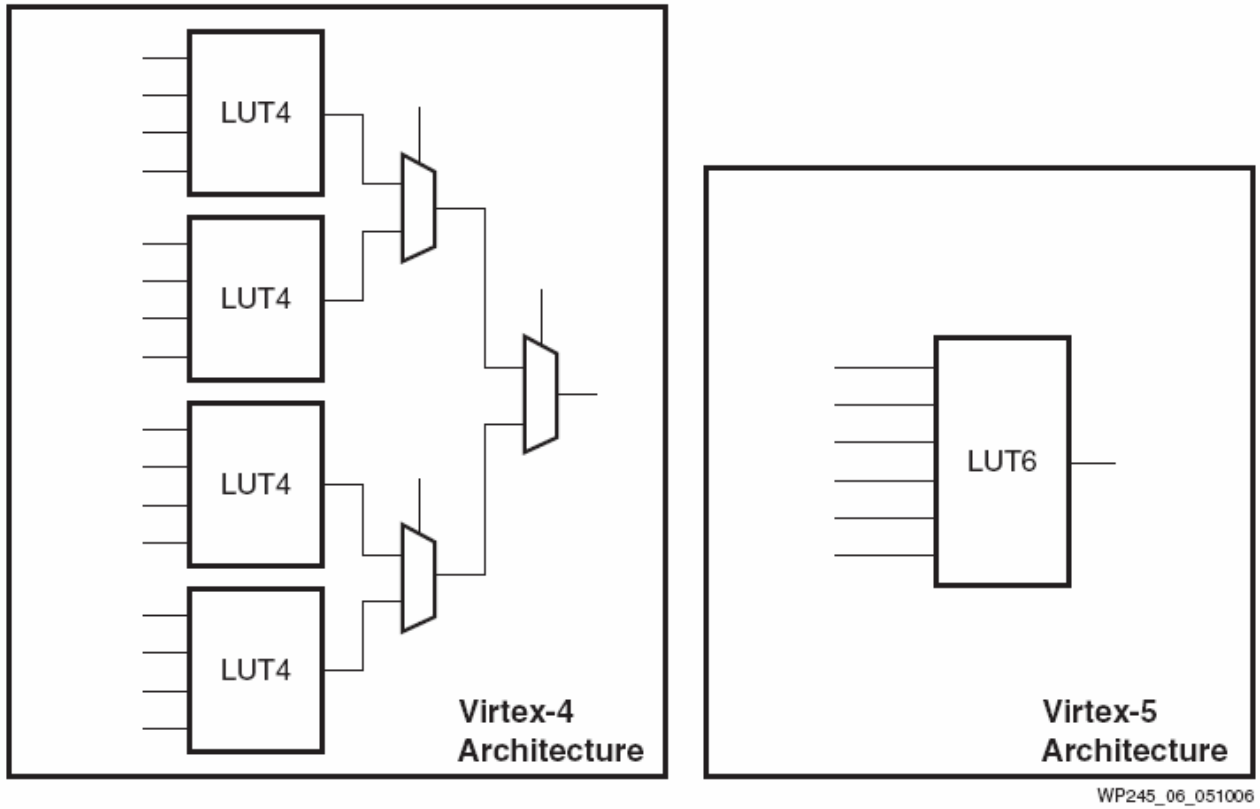


Figure 6: **64-Bit LUT RAM Implementation in Virtex-4 and Virtex-5 FPGAs**

Praktiskie darbi

- Strādājam pie kursa projekta KP3

Pateicos par uzmanību!

Jautājumi?