ECE 545
Lecture 9

FPGA Devices
& FPGA Design Flow
Required Reading

Xilinx, Inc.

Spartan-3 FPGA Family

Spartan-3 FPGA Family Data Sheet

Module 1:

• Introduction
• Features
• Architectural Overview
• Package Marking

Module 2:

• CLB Overview
Required Reading

Xilinx, Inc.

Spartan-3 FPGA Family
Spartan-3 Generation FPGA User Guide

Chapter 5 Using Configurable Logic Blocks (CLBs)
Chapter 6 Using Look-Up Tables as Distributed RAM
Chapter 7: Using Look-Up Tables as Shift Registers (SRL16)
Chapter 9: Using Carry and Arithmetic Logic
Required Reading

Xilinx, Inc.

_Virtex-5 FPGA Family_

_Virtex-5 FPGA User Guide_

*Chapter 5: Configurable Logic Blocks (CLBs)*
Required Reading

Altera, Inc.

Stratix III FPGA Family

Stratix III Device Handbook

1. Stratix III Device Family Overview
2. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices
## Two competing implementation approaches

<table>
<thead>
<tr>
<th>ASIC</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Specific Integrated Circuit</strong></td>
<td><strong>Field Programmable Gate Array</strong></td>
</tr>
</tbody>
</table>

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

- No physical layout design; design ends with a **bitstream** used to configure a device
- Bought **off the shelf** and reconfigured by designers themselves
What is an FPGA?

- **Configurable Logic Blocks**
- **I/O Blocks**
- **Block RAMs**
Which Way to Go?

**ASICS**
- High performance
- Low power
- Low cost in high volumes

**FPGAs**
- Off-the-shelf
- Low development cost
- Short time to market
- Reconfigurability

Low power

Low cost in high volumes
Other FPGA Advantages

• Manufacturing cycle for ASIC is very costly, lengthy and engages lots of manpower
  • Mistakes not detected at design time have large impact on development time and cost
  • FPGAs are perfect for rapid prototyping of digital circuits
• Easy upgrades like in case of software
• Unique applications
  • reconfigurable computing
Major FPGA Vendors

SRAM-based FPGAs
• Xilinx, Inc.
• Altera Corp.
• Atmel
• Lattice Semiconductor

Flash & antifuse FPGAs
• Actel Corp.
• Quick Logic Corp.

Share about 90% of the market
Xilinx

- Primary products: FPGAs and the associated CAD software

- Main headquarters in San Jose, CA

- Fabless* Semiconductor and Software Company
  - UMC (Taiwan) {*Xilinx acquired an equity stake in UMC in 1996}
  - Seiko Epson (Japan)
  - TSMC (Taiwan)
  - Samsung (Korea)
Xilinx FPGA Families

- Old families
  - XC3000, XC4000, XC5200
  - Old 0.5µm, 0.35µm and 0.25µm technology. Not recommended for modern designs.

- High-performance families
  - Virtex (220 nm)
  - Virtex-E, Virtex-EM (180 nm)
  - Virtex-II (130 nm)
  - Virtex-II PRO (130 nm)
  - Virtex-4 (90 nm)
  - Virtex-5 (65 nm)
  - Virtex-6 (40 nm)

- Low Cost Family
  - Spartan/XL – derived from XC4000
  - Spartan-II – derived from Virtex
  - Spartan-IIE – derived from Virtex-E
  - Spartan-3 (90 nm)
  - Spartan-3E (90 nm) – logic optimized
  - Spartan-3A (90 nm) – I/O optimized
  - Spartan-3AN (90 nm) – non-volatile
  - Spartan-3A DSP (90 nm) – DSP optimized
  - Spartan-6 (45 nm)
CLB Structure
General structure of an FPGA
Xilinx Spartan 3 CLB

Configurable logic block (CLB)

Slice
Logic cell
Slice
Logic cell
Slice
Logic cell
Slice
Logic cell
Slice
Logic cell
Slice
Logic cell
Spartan 3 CLB Structure

Left-Hand SLICEM
(Logics or Distributed RAM
or Shift Register)

Right-Hand SLICEL
(Logics Only)

Interconnect to Neighbors
Xilinx CLB Slice
CLB Slice Structure

- Each slice contains two sets of the following:
  - Four-input LUT
    - Any 4-input logic function,
    - or 16-bit x 1 sync RAM (SLICEM only)
    - or 16-bit shift register (SLICEM only)
  - Carry & Control
    - Fast arithmetic logic
    - Multiplier logic
    - Multiplexer logic
  - Storage element
    - Latch or flip-flop
    - Set and reset
    - True or inverted inputs
    - Sync. or async. control
LUT (Look-Up Table) Functionality

• Look-Up tables are primary elements for logic implementation

• Each LUT can implement any function of 4 inputs
5-Input Functions implemented using two LUTs

- One CLB Slice can implement any function of 5 inputs
- Logic function is partitioned between two LUTs
- F5 multiplexer selects LUT
5-Input Functions implemented using two LUTs
Xilinx Spartan 3 Multipurpose LUT

- 4-input LUT
- 16 x 1 RAM
- 16-bit SR
Simplified view of a Xilinx Logic Cell
Distributed RAM

- CLB LUT configurable as Distributed RAM
  - A single LUT equals 16x1 RAM
  - Two LUTs Implement Single and Dual-Port RAMs
  - Cascade LUTs to increase RAM size
- Synchronous write
- Synchronous/Asynchronous read
  - Accompanying flip-flops used for synchronous read
Shift Register

- Each LUT can be configured as shift register
  - Serial in, serial out
- Dynamically addressable delay up to 16 cycles
- For programmable pipeline
- Cascade for greater cycle delays
- Use CLB flip-flops to add depth
Shift Register

- Register-rich FPGA
  - Allows for addition of pipeline stages to increase throughput
- Data paths must be balanced to keep desired functionality
Carry & Control Logic
Fast Carry Logic

- Each CLB contains separate logic and routing for the fast generation of sum & carry signals
  - Increases efficiency and performance of adders, subtractors, accumulators, comparators, and counters
- Carry logic is independent of normal logic and routing resources
Full-adder

\[ x + y + c_{in} = (c_{out} \ s)_2 \]
### Full-adder

#### Alternative implementations

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$c_{out}$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$c_{in}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$c_{in}$</td>
<td>$c_{in}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$c_{in}$</td>
<td>$c_{in}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$c_{in}$</td>
</tr>
</tbody>
</table>
Full-adder
Alternative implementations

Implementation used to generate fast carry logic in Xilinx FPGAs

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>c_{out}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>c_{in}</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>c_{in}</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
</tbody>
</table>

\[ p = x \oplus y \]
\[ g = y \]
\[ s = p \oplus c_{in} = x \oplus y \oplus c_{in} \]
Carry & Control Logic in Spartan 3 FPGAs

Hardwired (fast) logic
Carry & Control Logic in Spartan 3 FPGAs

LUT

Function Generator

Hardwired (fast) logic
Simplified View of Spartan-3 FPGA Carry and Arithmetic Logic in One Logic Cell
Simplified View of Carry Logic in One Spartan 3 Slice
\[ T_{\text{ripple-add}} = T_{\text{FA}}(x, y \rightarrow c_{\text{out}}) + (k - 2) \times T_{\text{FA}}(c_{\text{in}} \rightarrow c_{\text{out}}) + T_{\text{FA}}(c_{\text{in}} \rightarrow s) \]

Fig. 5.5 Critical path in a \( k \)-bit ripple-carry adder.
Critical Path for an Adder Implemented Using Xilinx Spartan 3/Spartan 3E FPGAs
## Number and Length of Carry Chains for Spartan 3 FPGAs

<table>
<thead>
<tr>
<th>Device</th>
<th>Number of Carry Chains</th>
<th>Bits per Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S50</td>
<td>24</td>
<td>64</td>
</tr>
<tr>
<td>XC3S200</td>
<td>40</td>
<td>96</td>
</tr>
<tr>
<td>XC3S400</td>
<td>56</td>
<td>128</td>
</tr>
<tr>
<td>XC3S1000</td>
<td>80</td>
<td>192</td>
</tr>
<tr>
<td>XC3S1500</td>
<td>104</td>
<td>256</td>
</tr>
<tr>
<td>XC3S2000</td>
<td>128</td>
<td>320</td>
</tr>
<tr>
<td>XC3S4000</td>
<td>144</td>
<td>384</td>
</tr>
<tr>
<td>XC3S5000</td>
<td>160</td>
<td>416</td>
</tr>
</tbody>
</table>
Bottom Operand Input to Carry Out Delay

$T_{OPCYF}$

0.9 ns for Spartan 3
Carry Propagation Delay $t_{BYP}$

0.2 ns for Spartan 3
Carry Input to Top Sum Combinational Output Delay $T_{CINY}$

1.2 ns for Spartan 3
Critical Path Delays and Maximum Clock Frequencies (into account surrounding registers)

- 8 bits: 3.0 ns or 333 MHz
- 16 bits: 3.8 ns or 263 MHz
- 32 bits: 5.4 ns or 185 MHz
- 64 bits: 8.6 ns or 116 MHz
Accessing Carry Logic

- All major synthesis tools can infer carry logic for arithmetic functions
  - Addition (SUM <= A + B)
  - Subtraction (DIFF <= A - B)
  - Comparators (if A < B then…)
  - Counters (count <= count +1)
Input/Output Blocks (IOBs)
Basic I/O Block Structure

- Three-State
- FF Enable
- Clock
- Set/Reset
- Output
- FF Enable
- Direct Input
- FF Enable
- Registered Input

Three-State Control

Output Path

Input Path
IOB Functionality

- IOB provides interface between the package pins and CLBs
- Each IOB can work as uni- or bi-directional I/O
- Outputs can be forced into High Impedance
- Inputs and outputs can be registered
  - advised for high-performance I/O
- Inputs can be delayed
Other Components of Spartan 3 FPGAs
RAM Blocks and Multipliers in Xilinx FPGAs
Dedicated Multiplier Block
Block RAM

- Most efficient memory implementation
  - Dedicated blocks of memory
- Ideal for most memory requirements
  - 4 to 36 memory blocks in Spartan 3
    - 18 kbits = 18,432 bits per block (16 k without parity bits)
  - Use multiple blocks for larger memories
- Builds both single and true dual-port RAMs
- Synchronous write and read (different from distributed RAM)
Memory Types

Memory
  RAM
  ROM

Memory
  Single port
  Dual port

Memory
  With asynchronous read
  With synchronous read
Memory Types

- Distributed (MLUT-based)
- Block RAM-based (BRAM-based)

- Inferred
- Instantiated
  - Manually
  - Using Core Generator
A simple clock tree
Digital Clock Manager (DCM)

Clock signal from outside world

Special clock pin and pad

Clock Manager

Daughter clocks used to drive internal clock trees or output pins

etc.
Spartan-3 Family Attributes
## Spartan-3 FPGA Family Members

<table>
<thead>
<tr>
<th>Device</th>
<th>System Gates</th>
<th>Equivalent Logic Cells(^1)</th>
<th>CLB Array (One CLB = Four Slices)</th>
<th>Distributed RAM Bits (K=1024)</th>
<th>Block RAM Bits (K=1024)</th>
<th>Dedicated Multipliers</th>
<th>DCMs</th>
<th>Maximum User I/O</th>
<th>Maximum Differential I/O Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S50(^2)</td>
<td>50K</td>
<td>1,728</td>
<td>16</td>
<td>12</td>
<td>192</td>
<td>12K</td>
<td>72K</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC3S200(^2)</td>
<td>200K</td>
<td>4,320</td>
<td>24</td>
<td>20</td>
<td>480</td>
<td>30K</td>
<td>216K</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>XC3S400(^2)</td>
<td>400K</td>
<td>8,064</td>
<td>32</td>
<td>28</td>
<td>896</td>
<td>56K</td>
<td>288K</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>XC3S1000(^2,3)</td>
<td>1M</td>
<td>17,280</td>
<td>48</td>
<td>40</td>
<td>1,920</td>
<td>120K</td>
<td>432K</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>XC3S1500(^3)</td>
<td>1.5M</td>
<td>29,952</td>
<td>64</td>
<td>52</td>
<td>3,328</td>
<td>208K</td>
<td>576K</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>XC3S2000</td>
<td>2M</td>
<td>46,080</td>
<td>80</td>
<td>64</td>
<td>5,120</td>
<td>320K</td>
<td>720K</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>XC3S4000(^3)</td>
<td>4M</td>
<td>62,208</td>
<td>96</td>
<td>72</td>
<td>6,912</td>
<td>432K</td>
<td>1,728K</td>
<td>96</td>
<td>4</td>
</tr>
<tr>
<td>XC3S5000</td>
<td>5M</td>
<td>74,880</td>
<td>104</td>
<td>80</td>
<td>8,320</td>
<td>520K</td>
<td>1,872K</td>
<td>104</td>
<td>4</td>
</tr>
</tbody>
</table>

**Notes:**

1. Logic Cell = 4-input Look-Up Table (LUT) plus a 'D' flip-flop. "Equivalent Logic Cells" equals "Total CLBs" x 8 Logic Cells/CLB x 1.125 effectiveness.
FPGA Nomenclature

Pb-Free Packaging

For additional information on Pb-free packaging, see XAPP427: "Implementation and Solder Reflow Guidelines for Pb-Free Packages".

<table>
<thead>
<tr>
<th>Device</th>
<th>Speed Grade</th>
<th>Package Type / Number of Pins</th>
<th>Temperature Range (T_J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S50</td>
<td>-4</td>
<td>VQ1G100 100-pin Very Thin Quad Flat Pack (VQFP)</td>
<td>C  Commercial (0°C to 85°C)</td>
</tr>
<tr>
<td>XC3S200</td>
<td>-3</td>
<td>CP1G132 132-pin Chip-Scale Package (CSP)</td>
<td>I  Industrial (-40°C to 100°C)</td>
</tr>
<tr>
<td>XC3S400</td>
<td></td>
<td>TQ1G144 144-pin Thin Quad Flat Pack (TQFP)</td>
<td></td>
</tr>
<tr>
<td>XC3S1000</td>
<td></td>
<td>PQ1G208 208-pin Plastic Quad Flat Pack (POQFP)</td>
<td></td>
</tr>
<tr>
<td>XC3S1500</td>
<td></td>
<td>FT1G256 256-ball Fine-Pitch Thin Ball Grid Array (FTBGA)</td>
<td></td>
</tr>
<tr>
<td>XC3S2000</td>
<td></td>
<td>FQ1G320 320-ball Fine-Pitch Ball Grid Array (FBGA)</td>
<td></td>
</tr>
<tr>
<td>XC3S4000</td>
<td></td>
<td>FG1G456 456-ball Fine-Pitch Ball Grid Array (FBGA)</td>
<td></td>
</tr>
<tr>
<td>XC3S5000</td>
<td></td>
<td>FG1G676 676-ball Fine-Pitch Ball Grid Array (FBGA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FG1G900 900-ball Fine-Pitch Ball Grid Array (FBGA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FG1G1156 1156-ball Fine-Pitch Ball Grid Array (FBGA)</td>
<td></td>
</tr>
</tbody>
</table>
FPGA Nomenclature Example

**XC3S1500-4FG320**

- Spartan 3 family
- 1500 k = 1.5 M equivalent logic gates
- Speed grade -4 = standard performance
- 320 pins
- Package type
FPGA Design Flow
Design flow (1)

Design and implement a simple unit permitting to speed up encryption with RC5-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds......

Specification (Lab Experiments)

VHDL description (Your Source Files)

Functional simulation

Synthesis

Post-synthesis simulation
Design flow (2)

- Implementation
- Configuration
- Timing simulation
- On chip testing
Tools used in FPGA Design Flow

Functionally verified VHDL code

Synplicity Synplify Pro

Xilinx ISE

Xilinx XST

Design

VHDL code

Synthesis

Netlist

Implementation

Bitstream
Synthesis
Synthesis Tools

Synplify Pro

Xilinx XST

... and others
architecture MLU_DATAFLOW of MLU is

signal A1:STD_LOGIC;
signal B1:STD_LOGIC;
signal Y1:STD_LOGIC;
signal MUX_0, MUX_1, MUX_2, MUX_3: STD_LOGIC;

begin

    A1<=A when (NEG_A='0') else not A;
    B1<=B when (NEG_B='0') else not B;
    Y<=Y1 when (NEG_Y='0') else not Y1;

    MUX_0<=A1 and B1;
    MUX_1<=A1 or B1;
    MUX_2<=A1 xor B1;
    MUX_3<=A1 xnor B1;

    with (L1 & L0) select
        Y1<=MUX_0 when "00",
        MUX_1 when "01",
        MUX_2 when "10",
        MUX_3 when others;

end MLU_DATAFLOW;
Circuit netlist (RTL view)
Mapping

LUT0

LUT1

LUT2

LUT3

LUT4

LUT5

FF1

FF2
RTL view in Synplify Pro

- General logic structures can be recognized in RTL view

- Comparator
- Incrementer
- MUX
Crossprobing between RTL view and code

- Each port, net or block can be chosen by mouse click from the browser or directly from the RTL View.
- By double-clicking on the element its source code can be seen:

```
-- generate divided clock
IF enable_xmit_clk = '1' OR enable_rcv_clk = '1' THEN
  IF clk_cnt >= unsigned(div msb_lsb) - 1 THEN
    clk_cnt <= others => '0';
  ELSE
    clk_cnt <= unsigned(clk_cnt) + 1;
  END IF;
ELSE
  clk_cnt <= (0=>'1', others=>'0');
END IF;
END PROCESS clk_div;
```

- Reverse crossprobing is also possible: if section of code is marked, appropriate element of RTL View is marked too.
Technology View in Synplify Pro

Technology view is a mapped RTL view. It can be seen by pressing button or by double-click on “.srm” file.

As in case of “RTL View”, buttons can be used here.

Two additional buttons are enabled:

- Show critical path
- Open timing analyst

Pay attention: technology view is usually large and presented on number of sheets.
Viewing critical path

- Critical path can be viewed by pressing on
- Delay values are written near each component of the path
Timing Analyst

- Timing analyst opened by pressing on
- Timing analyst gives a possibility to analyze different paths in the design
- Timing analyst can be opened only from Technology View
Implementation
Implementation

• After synthesis the entire implementation process is performed by FPGA vendor tools
InputFile = c:/Documents and Settings/Milind Parekar/My Documents/ECE_449/ALU/implement/xie0.ini
Executing C:\Xilinx\bin\nt\ngdbuild.exe -p 2S100TQ144-6 -sd "c:\Documents and Settings\Milind Parekar\My Documents\ECE_449\ALU\synthesis" -sd "c:\Documents and Settings\Milind Parekar\My Documents\ECE_449\ALU\compile" -sd "c:\Documents and Settings\Milind Parekar\My Documents\ECE_449\ALU\src" -sd "C:\Program Files\Aldec\Active-HDL 6.2\vlib\SPARTAN2\compile" -uc "alu.ucf" "ALU.edf" "ALU.ngd"

c:\Documents and Settings\Milind Parekar\My Documents\ECE_449\ALU\implement\ver1\rev1>set XILINX=C:\Xilinx

c:\Documents and Settings\Milind Parekar\My Documents\ECE_449\ALU\implement\ver1\rev1>set PATH=C:\Xilinx\bin\nt
Translation

Synthesis

Circuit netlist

Timing Constraints

Electronic Design Interchange Format

Native Constraint File

Constraint Editor or Text Editor

User Constraint File

Translation

Translation

EDIF

NCF

UCF

Translation

Native Generic Database file

NGD
Mapping
Routing

Programmable Connections

FPGA

LUT0

LUT1

LUT2

LUT3

LUT4

LUT5

FF1

FF2
Configuration

• Once a design is implemented, you must create a file that the FPGA can understand
  • This file is called a bit stream: a BIT file (.bit extension)

• The BIT file can be downloaded directly to the FPGA, or can be converted into a PROM file which stores the programming information
Two main stages of the FPGA Design Flow

Synthesis

- Code analysis
- Derivation of main logic constructions
- Technology independent optimization
- Creation of “RTL View”

Map

- Mapping of extracted logic structures to device primitives
- Technology dependent optimization
- Application of “synthesis constraints”
- Netlist generation
- Creation of “Technology View”

Implementation

Technology dependent

Place & Route

- Placement of generated netlist onto the device
- Choosing best interconnect structure for the placed design
- Application of “physical constraints”

Configure

- Bitstream generation
- Burning device
Report files
Map report header

Release 8.1i Map I.24
Xilinx Mapping Report File for Design 'Lab3Demo'

Design Information
---------------------
Command Line : c:\Xilinx\bin\nt\map.exe -p 3S1500FG320-4 -o map.ncd -pr b -k 4
-cm area -c 100 Lab3Demo.ngd Lab3Demo.pcf
Target Device : xc3s1500
Target Package : fg320
Target Speed : -4
Mapper Version : spartan3 -- $Revision: 1.34 $
Mapped Date : Tue Feb 13 17:04:54 2007
Map report

Design Summary

Number of errors: 0
Number of warnings: 0

Logic Utilization:

**Number of Slice Flip Flops:** 30 out of 26,624 1%
Number of 4 input LUTs: 38 out of 26,624 1%

Logic Distribution:

**Number of occupied Slices:** 33 out of 13,312 1%
Number of Slices containing only related logic: 33 out of 33 100%
Number of Slices containing unrelated logic: 0 out of 33 0%

*See NOTES below for an explanation of the effects of unrelated logic

**Total Number 4 input LUTs:** 62 out of 26,624 1%
Number used as logic: 38
Number used as a route-thru: 24
Number of bonded IOBs: 10 out of 221 4%
IOB Flip Flops: 7
Number of GCLKs: 1 out of 8 12%
Asterisk (*) preceding a constraint indicates it was not met.
This may be due to a setup or hold violation.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Requested</th>
<th>Actual</th>
<th>Logic</th>
<th>Absolute</th>
<th>Number of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>* TS_CLOCK = PERIOD TIMEGRP &quot;CLOCK&quot; 5 ns HIGH 50%</td>
<td>5.000ns</td>
<td>5.140ns</td>
<td>4</td>
<td>-0.140ns</td>
<td>5</td>
</tr>
<tr>
<td>TS_gen1Hz_Clock1Hz = PERIOD TIMEGRP &quot;gen1Hz_Clock1Hz&quot; 5 ns HIGH 50%</td>
<td>5.000ns</td>
<td>4.137ns</td>
<td>2</td>
<td>0.863ns</td>
<td>0</td>
</tr>
</tbody>
</table>
# Post layout timing report

Clock to Setup on destination clock CLOCK

<table>
<thead>
<tr>
<th>Source Clock</th>
<th>Dest: Rise</th>
<th>Dest: Rise</th>
<th>Dest: Fall</th>
<th>Dest: Fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK</td>
<td>5.140</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Timing summary:

- Timing errors: 9  Score: 543
- Constraints cover 574 paths, 0 nets, and 187 connections
- Design statistics:
  - Minimum period: 5.140ns  (Maximum frequency: 194.553MHz)
# Xilinx FPGA Devices

<table>
<thead>
<tr>
<th>Technology</th>
<th>Low-cost</th>
<th>High-performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>120/150 nm</td>
<td></td>
<td>Virtex 2, 2 Pro</td>
</tr>
<tr>
<td>90 nm</td>
<td>Spartan 3</td>
<td>Virtex 4</td>
</tr>
<tr>
<td>65 nm</td>
<td></td>
<td>Virtex 5</td>
</tr>
<tr>
<td>45 nm</td>
<td>Spartan 6</td>
<td></td>
</tr>
<tr>
<td>40 nm</td>
<td></td>
<td>Virtex 6</td>
</tr>
</tbody>
</table>
## Altera FPGA Devices

<table>
<thead>
<tr>
<th>Technology</th>
<th>Low-cost</th>
<th>Mid-range</th>
<th>High-performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>130 nm</td>
<td>Cyclone</td>
<td></td>
<td>Stratix</td>
</tr>
<tr>
<td>90 nm</td>
<td>Cyclone II</td>
<td></td>
<td>Stratix II</td>
</tr>
<tr>
<td>65 nm</td>
<td>Cyclone III</td>
<td>Arria I</td>
<td>Stratix III</td>
</tr>
<tr>
<td>40 nm</td>
<td>Cyclone IV</td>
<td>Arria II</td>
<td>Stratix IV</td>
</tr>
</tbody>
</table>
High-Performance Xilinx FPGAs
Virtex 5
Arrangement of Slices within the CLB
Row and Column Relationship between CLBs and Slices
## Major Differences between Xilinx Families

<table>
<thead>
<tr>
<th></th>
<th>Spartan 3 Virtex 4</th>
<th>Virtex 5, Virtex 6, Spartan 6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Look-Up Tables</strong></td>
<td>4-input</td>
<td>6-input</td>
</tr>
<tr>
<td><strong>Number of CLB slices per CLB</strong></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Number of LUTs per CLB slice</strong></td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
## Distributed RAM Configurations

<table>
<thead>
<tr>
<th>RAM</th>
<th>Number of LUTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 x 1S</td>
<td>1</td>
</tr>
<tr>
<td>32 x 1D</td>
<td>2</td>
</tr>
<tr>
<td>32 x 2Q(^2)</td>
<td>4</td>
</tr>
<tr>
<td>32 x 6SDP(^2)</td>
<td>4</td>
</tr>
<tr>
<td>64 x 1S</td>
<td>1</td>
</tr>
<tr>
<td>64 x 1D</td>
<td>2</td>
</tr>
<tr>
<td>64 x 1Q(^3)</td>
<td>4</td>
</tr>
<tr>
<td>64 x 3SDP(^3)</td>
<td>4</td>
</tr>
<tr>
<td>128 x 1S</td>
<td>2</td>
</tr>
<tr>
<td>128 x 1D</td>
<td>4</td>
</tr>
<tr>
<td>256 x 1S</td>
<td>4</td>
</tr>
</tbody>
</table>

### Notes:
1. S = single-port configuration; D = dual-port configuration; Q = quad-port configuration; SDP = simple dual-port configuration.
2. RAM32M is the associated primitive for this configuration.
3. RAM64M is the associated primitive for this configuration.
64 x 1 Single Port
64 x 1 Quad Port
64 x 3
Simple Dual Port
## ROM Configurations

<table>
<thead>
<tr>
<th>ROM</th>
<th>Number of LUTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 x 1</td>
<td>1</td>
</tr>
<tr>
<td>128 x 1</td>
<td>2</td>
</tr>
<tr>
<td>256 x 1</td>
<td>4</td>
</tr>
</tbody>
</table>
32-bit Shift Register, SRL
32-bit Shift Register
Dual 16-bit Shift Register
64-bit Shift Register
96-bit Shift Register

SHIFTIN (D)
A[6:0]
CLK
WE

SRL32

DI1
A[6:2]
CLK
WE

CX (A5)
F7BMUX

SRL32

DI1
A[6:2]
CLK
WE

BX (A6)
F8MUX

SRL32

DI1
A[6:2]
CLK
WE

AX (A5)
F7AMUX

Output (Q)
Registered
Output
(Optional)
Fast Carry Logic Path

* Can be used if unregistered/registered outputs are free.
## Major Differences between Xilinx Families

<table>
<thead>
<tr>
<th></th>
<th>Spartan 3 Virtex 4</th>
<th>Virtex 5, Virtex 6, Spartan 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Single-Port Memory Size per LUT</td>
<td>16 x 1</td>
<td>64 x 1</td>
</tr>
<tr>
<td>Maximum Shift Register Size per LUT</td>
<td>16 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>Number of adder stages per CLB slice</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Low-cost Altera FPGAs
Altera Cyclone III
Logic Element (LE) – Normal Mode
Altera Cyclone III
Logic Element (LE) – Arithmetic Mode
High-Performance Altera FPGAs
High-Level Block Diagram of the Stratix III ALM
Altera Stratix III
Adaptive Logic Modules (ALM) – Normal Mode
$4 \times 2$ Crossbar Switch Example

**4 × 2 Crossbar Switch**

- sel0[1..0]
- inputa
- inputb
- inputc
- inputd

**Implementation in 1 ALM**

- dataf0
- datae0
- dataa
- datab
- datac
- datad
- datae1
- dataf1

- combout0
- combout1

- Six-Input LUT (Function0)
- Six-Input LUT (Function1)
Register Packing

These inputs are available for register packing.
Template for Seven-Input Functions Supported in Extended LUT Mode

This input is available for register packing.
Altera Stratix III, Stratix IV
Adaptive Logic Modules (ALM) – Arithmetic Mode

datae0

dataf0
datac
datab
dataa

dataad
datae1

dataf1

carry_in

4-Input LUT

add0

To general or local routing

To general or local routing

4-Input LUT

4-Input LUT

4-Input LUT

4-Input LUT

4-Input LUT

4-Input LUT

4-Input LUT

4-Input LUT

D Q
reg0

D Q
reg1

carry_out

add1

To general or local routing

To general or local routing
Performing Operation

\[ R = (X < Y) ? Y : X \]
Three Operand Addition
Utilizing Shared Arithmetic Mode

3-Bit Add Example

\[
\begin{align*}
\text{1\textsuperscript{st} stage add is implemented in LUTs.} \\
X3 & \times X2 \times X1 \times X0 \\
Y3 & \times Y2 \times Y1 \times Y0 \\
+ & Z3 \times Z2 \times Z1 \times Z0 \\
\text{2\textsuperscript{nd} stage add is implemented in } s. \\
& S3 \times S2 \times S1 \times S0 \\
& + C3 \times C2 \times C1 \times C0 \\
& R4 \times R3 \times R2 \times R1 \times R0
\end{align*}
\]

<table>
<thead>
<tr>
<th>Binary Add</th>
<th>Decimal Equivalents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 0</td>
<td>14</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>4</td>
</tr>
<tr>
<td>+ 1 1 0 1</td>
<td>+ 13</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>7</td>
</tr>
<tr>
<td>+ 1 1 0 0</td>
<td>+ 2 \times 12</td>
</tr>
<tr>
<td>1 1 1 1 1</td>
<td>31</td>
</tr>
</tbody>
</table>

\[
\text{ALM Implementation}
\]

\[
\begin{align*}
\text{ALM 1} \\
3\text{-Input LUT} & S0 \\
3\text{-Input LUT} & C0 \\
3\text{-Input LUT} & S1 \\
3\text{-Input LUT} & C1 \\
3\text{-Input LUT} & S2 \\
3\text{-Input LUT} & C2 \\
3\text{-Input LUT} & S3 \\
3\text{-Input LUT} & C3
\end{align*}
\]

\[
\text{ALM 2} \\
3\text{-Input LUT} & R0 \\
3\text{-Input LUT} & R1 \\
3\text{-Input LUT} & R2 \\
3\text{-Input LUT} & R3 \\
3\text{-Input LUT} & R4
\]

\[
\text{carrying in } = '0' \\
\text{shared_arith_in } = '0'
\]
LUT-Register Mode
Register Chain
### Example of Resource Utilization Report (1)

<table>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALUTs Used</td>
<td>415 / 38,000 (1%)</td>
</tr>
<tr>
<td>-- Combinational ALUTs</td>
<td>415 / 38,000 (1%)</td>
</tr>
<tr>
<td>-- Memory ALUTs</td>
<td>0 / 19,000 (0%)</td>
</tr>
<tr>
<td>-- LUT_REGs</td>
<td>0 / 38,000 (0%)</td>
</tr>
<tr>
<td>Dedicated logic registers</td>
<td>136 / 38,000 (&lt;1%)</td>
</tr>
<tr>
<td>Combinational ALUT usage by number of inputs</td>
<td></td>
</tr>
<tr>
<td>-- 7 input functions</td>
<td>0</td>
</tr>
<tr>
<td>-- 6 input functions</td>
<td>287</td>
</tr>
<tr>
<td>-- 5 input functions</td>
<td>0</td>
</tr>
<tr>
<td>-- 4 input functions</td>
<td>24</td>
</tr>
<tr>
<td>-- &lt;=3 input functions</td>
<td>104</td>
</tr>
<tr>
<td>Combinational ALUTs by mode</td>
<td></td>
</tr>
<tr>
<td>-- normal mode</td>
<td>335</td>
</tr>
<tr>
<td>-- extended LUT mode</td>
<td>0</td>
</tr>
<tr>
<td>-- arithmetic mode</td>
<td>80</td>
</tr>
<tr>
<td>-- shared arithmetic mode</td>
<td>0</td>
</tr>
</tbody>
</table>
Example of Resource Utilization Report (2)

; Logic utilization ; 701 / 38,000 ( 2 % ) ;
;  -- Difficulty Clustering Design ; Low ;
;  -- Combinational ALUT/register pairs used
; in final Placement ; 476 ;
;  -- Combinational with no register ; 340 ;
;  -- Register only ; 61 ;
;  -- Combinational with a register ; 75 ;
;  -- Estimated pairs recoverable by pairing ALUTs and registers
 as design grows ; -54 ;
;  -- Estimated Combinational ALUT/register pairs
 unavailable ; 279 ;
;  -- Unavailable due to Memory LAB use ; 0 ;
;  -- Unavailable due to unpartnered 7 LUTs ; 0 ;
;  -- Unavailable due to unpartnered 6 LUTs ; 279 ;
;  -- Unavailable due to unpartnered 5 LUTs ; 0 ;
;  -- Unavailable due to LAB-wide signal
 conflicts ; 0 ;
;  -- Unavailable due to LAB input limits ; 0 ;
Example of Resource Utilization Report (3)

; Total registers* ; 136 ;
;   -- Dedicated logic registers ; 136 / 38,000 ( < 1 % ) ;
;   -- I/O registers ; 0 / 2,752 ( 0 % ) ;
;   -- LUT_REGS ; 0 ;
; ALMs: partially or completely used ; 360 / 19,000 ( 2 % ) ;
; Total LABs: partially or completely used ; 42 / 1,900 ( 2 % ) ;
;   -- Logic LABs ; 42 / 42 ( 100 % ) ;
;   -- Memory LABs ; 0 / 42 ( 0 % ) ;
; User inserted logic elements ; 0 ;
; Virtual pins ; 0 ;
; I/O pins ; 20 / 488 ( 4 % ) ;
;   -- Clock pins ; 5 / 16 ( 31 % ) ;
;   -- Dedicated input pins ; 0 / 12 ( 0 % ) ;
; Global signals ; 2 ;
; M9K blocks ; 0 / 108 ( 0 % ) ;
; M144K blocks ; 0 / 6 ( 0 % ) ;
; Total MLAB memory bits ; 0 ;
; Total block memory bits ; 0 / 1,880,064 ( 0 % ) ;
; Total block memory implementation bits ; 0 / 1,880,064 ( 0 % ) ;
; DSP block 18-bit elements ; 0 / 216 ( 0 % ) ;
; PLLs ; 0 / 4 ( 0 % ) ;
; Global clocks ; 2 / 16 ( 13 % ) ;