Two competing implementation approaches

**ASIC**
- Application Specific Integrated Circuit
  - designed all the way from behavioral description to physical layout
  - designs must be sent for expensive and time consuming fabrication in a semiconductor foundry

**FPGA**
- Field Programmable Gate Array
  - 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?

<table>
<thead>
<tr>
<th>ASICs</th>
<th>FPGAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>High performance</td>
<td>Off-the-shelf</td>
</tr>
<tr>
<td>Low power</td>
<td>Low development cost</td>
</tr>
<tr>
<td>Low cost in high volumes</td>
<td>Short time to market</td>
</tr>
<tr>
<td></td>
<td>Reconfigurability</td>
</tr>
</tbody>
</table>

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. Share about 90% of the market
- Altera Corp.
- Atmel
- Lattice Semiconductor

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

Xilinx

- Primary products: FPGAs and the associated CAD software
- Fabless Semiconductor and Software Company
- Main headquarters in San Jose, CA
- 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
  - 0.5µm, 0.35µm and 0.25µm technology. Not recommended for modern designs.
- High-performance families
  - Virtex (220 nm)
  - Virtex-E, Virtex-4M (180 nm)
  - Virtex-6 (280 nm)
  - Virtex-5 (40 nm)
  - Virtex-6 (40 nm)
- Low Cost Family
  - Spartan/XL – derived from XC4000
  - Spartan-II – derived from Virtex
  - Spartan-II – derived from Virtex-2
  - Spartan-3 (90 nm)
  - Spartan-3E (90 nm) – logic optimized
  - Spartan-3A (90 nm) – I/O optimized
  - Spartan-3A (90 nm) – non-volatile.
  - Spartan-3A DSP (90 nm) – DSP optimized
  - Spartan-3A (65 nm)
General structure of an FPGA

Xilinx Spartan 3 CLB

Spartan 3 CLB Structure

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

Xilinx Spartan 3 Multipurpose LUT

- 16-bit SR
- 16 x 1 RAM
- 4-input LUT

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
- 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
\[
x + y + c_{in} = (c_{out} \oplus s)
\]

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

Alternative implementations

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

Implementation used to generate fast carry logic in Xilinx FPGAs

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

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**
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>XC3S500</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>394</td>
</tr>
<tr>
<td>XC3S7000</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

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
**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**

- Distributed (MLUT-based)
- Block RAM-based (BRAM-based)
- Inferred
- Instantiated
- Manually
- Using Core Generator

**Dedicated Multiplier Block**

**A simple clock tree**
**Digital Clock Manager (DCM)**

Clock signal from outside world

Special clock pin and pad

Clock Manager

Daughter clock node used to drive internal clock trees or output pins

---

**Spartan-3 Family Attributes**

---

**Spartan-3 FPGA Family Members**

<table>
<thead>
<tr>
<th>Device</th>
<th>Spartan family</th>
<th>Equivalences</th>
<th>LUTs</th>
<th>TTQs</th>
<th>Configurable Logic Blocks (CLBs)</th>
<th>Block RAMs (BRAMs)</th>
<th>I/O Banks (IOB)</th>
<th>I/O Functions</th>
<th>Package Type</th>
<th>Minimum Speed (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S1000-3FG0676</td>
<td>5000</td>
<td>4224</td>
<td>15</td>
<td>12</td>
<td>4000</td>
<td>4</td>
<td>20</td>
<td>15</td>
<td>4800</td>
<td>20</td>
</tr>
<tr>
<td>XC3S1000-4FG900</td>
<td>5000</td>
<td>4224</td>
<td>15</td>
<td>12</td>
<td>4000</td>
<td>4</td>
<td>20</td>
<td>15</td>
<td>4800</td>
<td>20</td>
</tr>
<tr>
<td>XC3S1000-5FG1106</td>
<td>5000</td>
<td>4224</td>
<td>15</td>
<td>12</td>
<td>4000</td>
<td>4</td>
<td>20</td>
<td>15</td>
<td>4800</td>
<td>20</td>
</tr>
<tr>
<td>XC3S1000-7FG1156</td>
<td>5000</td>
<td>4224</td>
<td>15</td>
<td>12</td>
<td>4000</td>
<td>4</td>
<td>20</td>
<td>15</td>
<td>4800</td>
<td>20</td>
</tr>
<tr>
<td>XC3S1000-8FG1256</td>
<td>5000</td>
<td>4224</td>
<td>15</td>
<td>12</td>
<td>4000</td>
<td>4</td>
<td>20</td>
<td>15</td>
<td>4800</td>
<td>20</td>
</tr>
</tbody>
</table>

---

**FPGA Nomenclature Example**

**XC3S1500-4FG320**

- Spartan 3 family
- 1500 k equivalent logic gates
- speed grade 4 = standard performance
- 320 pins

---

**FPGA Nomenclature**

---

**FPGA Design Flow**
Design flow (1)

Specification (Lab Experiments)
VHDL description (Your Source Files)
Functional simulation
Synthesis
Post-synthesis simulation

Design flow (2)

Implementation
Timing simulation
Configuration
On chip testing

Tools used in FPGA Design Flow

Synplify Pro
Xilinx XST
Syntegrity
Xilinx XST

Synthesis Tools

Synplify Pro
Xilinx XST

... and others

Logic Synthesis

VHDL description
Circuit netlist
Circuit netlist (RTL view)

Mapping

RTL view in Synplify Pro
- General logic structures can be recognized in RTL view

Crossprobing between RTL view and code
- Each port, net, or clock can be chosen by mouse click from the browser or directly from the RTL view
- Ify double clicking on the element its source code can be seen.

Technology View in Synplify Pro
- Technology view is mapped RTL view. It can be seen by pressing the button or by double clicking on "view" file.
- An example of a technology view is shown.
- Two additional windows are enabled - view critical path - view timing analysis

Viewing critical path
- Critical path can be viewed by pressing on
- Delay values are written on each component of the path
Implementation

- After synthesis the entire implementation process is performed by FPGA vendor tools

Translation

- Synthesis
- Circuit netlist
- Timing Constraints
- Native Constraint File
- User Constraint File
- Translation
- EDIF
- NCF
- UCF
- Native Generic Database file

Mapping
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
**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%)

**Place & route report**

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 Level</th>
<th>Absolute Slack</th>
<th>Number of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS_CLOCK = PERIOD TIMEGRP &quot;CLOCK&quot; 5 ns</td>
<td>5.000ns</td>
<td>5.140ns</td>
<td>4</td>
<td>-0.140ns</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS_gen1Hz_Clock1Hz = PERIOD TIMEGRP &quot;gen1 Hz_Clock1Hz&quot; 5 ns HIGH 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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:Fall</th>
<th>Dest:Rise</th>
<th>Dest:Fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK</td>
<td>5.140ns</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>Virtex 2, 2 Pro</td>
<td></td>
</tr>
<tr>
<td>90 nm</td>
<td>Spartan 3</td>
<td>Virtex 4</td>
</tr>
<tr>
<td>65 nm</td>
<td>Spartan 6</td>
<td>Virtex 5</td>
</tr>
<tr>
<td>40 nm</td>
<td>Spartan 6</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>Arria I</td>
<td>Stratix</td>
</tr>
<tr>
<td>90 nm</td>
<td>Cyclone II</td>
<td>Arria I</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>Look-Up Tables</th>
<th>Spartan 3</th>
<th>Virtex 4</th>
<th>Virtex 5, Virtex 6, Spartan 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of LUTs per CLB slice</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4-input</td>
<td>6-input</td>
<td></td>
<td></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 2QD(1)</td>
<td>4</td>
</tr>
<tr>
<td>32 x 64QD(2)</td>
<td>4</td>
</tr>
<tr>
<td>64 x 1S</td>
<td>2</td>
</tr>
<tr>
<td>64 x 1D</td>
<td>2</td>
</tr>
<tr>
<td>64 x 2QD</td>
<td>4</td>
</tr>
<tr>
<td>64 x 32QD(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; 2Q = single dual-port configuration.
2. RAM8DM is the associated primitive for this configuration.
3. RAM4BM is the associated primitive for this configuration.

64 x 1 Single Port

64 x 1 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 Diagram](image)

### Dual 16-bit Shift Register

![Dual 16-bit Shift Register Diagram](image)
### 64-bit Shift Register

### 96-bit Shift Register

### Fast Carry Logic Path

### Major Differences between Xilinx Families

<table>
<thead>
<tr>
<th></th>
<th>Spartan 3</th>
<th>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>
<td></td>
</tr>
<tr>
<td>Maximum Shift Register Size per LUT</td>
<td>16 bits</td>
<td>32 bits</td>
<td></td>
</tr>
<tr>
<td>Number of adder stages per CLB slice</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

### Altera Cyclone III

Logic Element (LE) – Normal Mode

### Low-cost Altera FPGAs
Altera Cyclone III Logic Element (LE) – Arithmetic Mode

High-Level Block Diagram of the Stratix III ALM

4 × 2 Crossbar Switch Example

Register Packing
Template for Seven-Input Functions
Supported in Extended LUT Mode

Performing Operation

R = (X < Y) ? Y : X

LUT-Register Mode

Three Operand Addition
Utilizing Shared Arithmetic Mode

Three Operand Addition
Utilizing Shared Arithmetic 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 Registers</td>
<td>136 / 38,000 (&lt;1 %)</td>
</tr>
</tbody>
</table>

#### Combinational ALUT usage by number of inputs

- 7 input functions: 0
- 6 input functions: 287
- 5 input functions: 0
- <=3 input functions: 104

#### Combinational ALUTs by mode

- normal mode: 335
- extended LUT mode: 80
- arithmetic mode: 0

### Example of Resource Utilization Report (2)

<table>
<thead>
<tr>
<th>Logic utilization</th>
<th>701 / 38,000 (2 %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- Difficulty Clustering design</td>
<td>Low</td>
</tr>
<tr>
<td>-- Combinational ALUT/register pairs used in Final Placement</td>
<td>474</td>
</tr>
<tr>
<td>-- Combinational with no register</td>
<td>340</td>
</tr>
<tr>
<td>-- Register only</td>
<td>41</td>
</tr>
<tr>
<td>-- Combinational with a register</td>
<td>75</td>
</tr>
<tr>
<td>-- Estimated pairs recoverable by pairing ALUTS and registers as design grows</td>
<td>-54</td>
</tr>
<tr>
<td>-- Estimated Combinational ALUT/register pairs unavailable</td>
<td>279</td>
</tr>
<tr>
<td>-- Unavailable due to Memory LAB use</td>
<td>8</td>
</tr>
<tr>
<td>-- Unavailable due to unpartnered 5 LUTs</td>
<td>0</td>
</tr>
<tr>
<td>-- Unavailable due to unpartnered 6 LUTs</td>
<td>279</td>
</tr>
<tr>
<td>-- Unavailable due to unpartnered 7 LUTs</td>
<td>0</td>
</tr>
<tr>
<td>-- Unavailable due to LAB-wide signal conflicts</td>
<td>0</td>
</tr>
<tr>
<td>-- Unavailable due to LAB input limits</td>
<td>0</td>
</tr>
</tbody>
</table>

### Example of Resource Utilization Report (3)

<table>
<thead>
<tr>
<th>Total registers*</th>
<th>136</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- Dedicated Logic Registers</td>
<td>136 / 38,000 (&lt;1 %)</td>
</tr>
<tr>
<td>-- I/O Registers</td>
<td>0 / 2,752 (0 %)</td>
</tr>
<tr>
<td>-- LUT Registers</td>
<td>0</td>
</tr>
<tr>
<td>ALMs: partially or completely used</td>
<td>40 / 19,000 (2 %)</td>
</tr>
<tr>
<td>-- Logic LABs</td>
<td>40 / 19,000 (2 %)</td>
</tr>
<tr>
<td>-- Memory LABs</td>
<td>0 / 42 (0 %)</td>
</tr>
<tr>
<td>User inserted logic elements</td>
<td>0</td>
</tr>
<tr>
<td>Virtual pins</td>
<td>0</td>
</tr>
<tr>
<td>-- Clock pins</td>
<td>5 / 488 (1 %)</td>
</tr>
<tr>
<td>-- Dedicated input pins</td>
<td>0 / 12 (1 %)</td>
</tr>
<tr>
<td>Global signals</td>
<td>2</td>
</tr>
<tr>
<td>M144k Blocks</td>
<td>0 / 4 (0 %)</td>
</tr>
<tr>
<td>DSP Block 18-bit elements</td>
<td>0 / 216 (0 %)</td>
</tr>
<tr>
<td>PLLs</td>
<td>0 / 4 (0 %)</td>
</tr>
<tr>
<td>Global clocks</td>
<td>2 / 14 (1 %)</td>
</tr>
</tbody>
</table>