Antifuse-Based FPGAs: Actel & QuickLogic

CSET 4650
Field Programmable Logic Devices

Dan Solarek
FPGA Design Flow

- HDL-based FPGA design flow, as shown at right, increases productivity by allowing you to work at higher levels of abstraction — the register-transfer level instead of the Boolean logic (gate) level.

- Central to HDL-based design and the increased size of FPGAs are two strategically important tools:
  - simulation for design verification and
  - synthesis for automatic implementation of the RTL design to the gate-level (FPGA place and route level).
FPGA Design Flow

- This flowchart shows a variation of the FPGA design flow
- ISE software, as well as similar software from other vendors, may offer several tools at each step – not all of which are essential to use for each design
- Learn the generalized flow as a guide to your design efforts
Antifuse FPGAs

- One-time programmable devices
- Primary vendors
  - Actel
  - QuickLogic
- No longer producing antifuse devices
  - Xilinx
  - Cypress

Finish Actel and talk about QuickLogic devices
Actel’s Current Antifuse Devices

- **Axcelerator**
  - High-speed antifuse FPGAs with gate densities of up to 2 million equivalent gates
- **SX-A / SX**
  - Antifuse devices 8k to 72k gates
- **eX**
  - Antifuse devices 3k to 12k gates
- **MX**
  - Antifuse devices 3k to 54k gates
Actel Axcelerator Family

- Actel’s newest FPGA family

### Axcelerator Family Selector Guide

<table>
<thead>
<tr>
<th></th>
<th>AX125</th>
<th>AX250</th>
<th>AX500</th>
<th>AX1000</th>
<th>AX2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equiv. System Gates</td>
<td>125,000</td>
<td>250,000</td>
<td>500,000</td>
<td>1,000,000</td>
<td>2,000,000</td>
</tr>
<tr>
<td>Typical Gates</td>
<td>82,000</td>
<td>154,000</td>
<td>288,000</td>
<td>612,000</td>
<td>1,060,000</td>
</tr>
<tr>
<td>Total RAM Bits</td>
<td>18,432</td>
<td>55,296</td>
<td>73,728</td>
<td>165,888</td>
<td>294,912</td>
</tr>
<tr>
<td>Max Registers</td>
<td>1,344</td>
<td>2,816</td>
<td>5,376</td>
<td>12,096</td>
<td>21,504</td>
</tr>
<tr>
<td>Total Modules</td>
<td>2,016</td>
<td>4,224</td>
<td>8,064</td>
<td>18,144</td>
<td>32,266</td>
</tr>
<tr>
<td>Dedicated Registers</td>
<td>672</td>
<td>1,408</td>
<td>2,688</td>
<td>6,048</td>
<td>10,752</td>
</tr>
<tr>
<td>RAM Blocks</td>
<td>4</td>
<td>12</td>
<td>16</td>
<td>36</td>
<td>64</td>
</tr>
<tr>
<td>Max No. of LVDS Pairs</td>
<td>84</td>
<td>124</td>
<td>168</td>
<td>258</td>
<td>342</td>
</tr>
<tr>
<td>PLLs</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>User I/Os</td>
<td>168</td>
<td>248</td>
<td>336</td>
<td>516</td>
<td>684</td>
</tr>
<tr>
<td>I/O Registers</td>
<td>504</td>
<td>744</td>
<td>1,008</td>
<td>1,548</td>
<td>2,052</td>
</tr>
<tr>
<td>Packages</td>
<td>CS180, FG256, FG324</td>
<td>CQ208, CQ352, FG256, FG484, PQ208</td>
<td>CQ208, CQ352, FG484, FG676, PQ208</td>
<td>BG729, CQ352, CG624, FG484, FG676, FG896</td>
<td>CQ352, CG624, FG896, FG1152</td>
</tr>
</tbody>
</table>

AX Die
Actel Axcelerator Family

- Axcelerator family naming convention
- five devices in the Axcelerator family, vary in number of equivalent gates
- four speed grades
- a variety of package options and operating temperature ranges
- same convention used for all Actel FPGAs
Actel Axcelerator Family

Axcelerator Family uses the Sea-of-Modules architecture

Typical of Xilinx FPGAs
Actel Axcelerator Family

- Axcelerator family interconnect elements
- Uses a metal-to-metal antifuse programmable interconnect element that resides between the upper two layers of metal
- Eliminates the channels of routing and interconnect resources between logic modules
Actel Axcelerator Family

- Two types of logic modules:
  - register cell (R-cell)
  - combinatorial cell (C-cell)
- C-Cell can implement more than 4,000 combinational functions of up to five inputs
- R-Cell contains a flip-flop featuring asynchronous clear, asynchronous preset, and active-low enable control signals

AX C-Cell and R-Cell
Actel Axcelerator Family

- Cluster:
  - two C-cells
  - a single R-cell
  - two Transmit (TX) and two Receive (RX) routing buffers

- Two Clusters form a SuperCluster
If one or more of the logic modules in a SuperCluster are used by a particular signal path, the other logic modules are still available for use by other paths.
Actel Axcelerator Family

The C–C–R pattern of the AX Cluster enables efficient implementation (minimum delay) of two-bit carry logic for improved arithmetic performance.
Actel Axcelerator Family

- AX device architecture (AX 1000 example)
Actel Axcelerator Family

- At the chip level, SuperClusters are organized into Core Tiles, which are arrayed to build up the full chip.
- This table shows the number of Core Tiles per device.
- Each core tile consists of an array of 336 SuperClusters and four SRAM blocks.
  - 176 SuperClusters and three SRAM blocks for the AX250.

<table>
<thead>
<tr>
<th>Device</th>
<th>Number of Core Tiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX125</td>
<td>1 regular tile</td>
</tr>
<tr>
<td>AX250</td>
<td>4 smaller tiles</td>
</tr>
<tr>
<td>AX500</td>
<td>4 regular tiles</td>
</tr>
<tr>
<td>AX1000</td>
<td>9 regular tiles</td>
</tr>
<tr>
<td>AX2000</td>
<td>16 regular tiles</td>
</tr>
</tbody>
</table>
Actel Axcelerator Family

- Surrounding the array of core tiles are blocks of I/O Clusters and the I/O bank ring
- The SRAM blocks are arranged in a column on the left side of the core tile
Actel Axcelerator Family

- The AX hierarchical routing structure ties the logic modules, the embedded memory blocks, and the I/O modules together.
Actel Axcelerator Family

At the lowest level, in and between SuperClusters, there are three local routing structures:
- FastConnect
- DirectConnect
- CarryConnect

DirectConnects provide the highest performance routing inside the SuperClusters by connecting a C-Cell to the adjacent R-Cell.

DirectConnects do not require an antifuse to make the connection and achieve a signal propagation time of less than 0.1 ns.
FastConnects provide high-performance, horizontal routing inside the SuperCluster and vertical routing to the SuperCluster immediately below it.

Only one programmable connection is used in a FastConnect path, delivering a maximum routing delay of 0.4 ns.
Actel Axcelerator Family

- CarryConnects are used for routing carry logic between adjacent SuperClusters.
- They connect the FCO output of one two-bit, C-Cell carry logic to the FCI input of the two bit, C-Cell carry logic of the SuperCluster below it.
- CarryConnects do not require an antifuse to make the connection and achieve a signal propagation time of less than 0.1 ns.
Actel Axcelerator Family

- The next level contains the core tile routing.
- Over the SuperClusters within a core tile, both vertical and horizontal tracks run across rows or columns, respectively.
At the chip level, vertical and horizontal tracks extend across the full length of the device, both north-to-south and east-to-west. These tracks are composed of highway routing that extend the entire length of the device (segmented at core tile boundaries) as well as segmented routing of varying lengths.
QuickLogic Devices
Metal-Metal Antifuse: QuickLogic

Metal-metal antifuses allow direct connections to the metal layers, and consume less area (reduced capacitance) in contrast to poly-diffusion antifuse (ONO antifuse).

Metal–metal antifuse. (a) An idealized (but to scale) cross section of a QuickLogic metal–metal antifuse in a two-level metal process. (b) A metal–metal antifuse in a three-level metal process that uses contact plugs. The conductive link usually forms at the corner of the via where the electric field is highest during programming.
Metal-Metal Antifuse: QuickLogic

- Cross-section of the ViaLink antifuse
QuickLogic Antifuse FPGAs

- Eclipse II
- Eclipse
- EclipsePlus
- QuickRAM
- pASIC 3
# Eclipse II Family

## Eclipse II Product Family Members

<table>
<thead>
<tr>
<th>Max Gates</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>47,052</td>
<td>63,840</td>
<td>188,946</td>
<td>248,160</td>
<td>320,640</td>
<td></td>
</tr>
<tr>
<td>Logic Array</td>
<td>16 x 8</td>
<td>16 x 16</td>
<td>32 x 20</td>
<td>40 x 24</td>
<td>48 x 32</td>
</tr>
<tr>
<td>Logic Cells</td>
<td>128</td>
<td>256</td>
<td>640</td>
<td>960</td>
<td>1,536</td>
</tr>
<tr>
<td>Max Flip-Flops</td>
<td>532</td>
<td>884</td>
<td>1,709</td>
<td>2,670</td>
<td>4,002</td>
</tr>
<tr>
<td>Max I/O</td>
<td>92</td>
<td>124</td>
<td>143</td>
<td>250</td>
<td>310</td>
</tr>
<tr>
<td>RAM Modules</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>RAM Bits</td>
<td>9,216</td>
<td>9,216</td>
<td>36,864</td>
<td>46,100</td>
<td>55,300</td>
</tr>
<tr>
<td>PLLs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ECUs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

### Packages

<table>
<thead>
<tr>
<th>VQFP</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TQFP</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>144</td>
<td>144</td>
<td>144</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TFBGA (0.8 mm)</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>196</td>
<td>196</td>
<td>196</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PQFP</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>208</td>
<td>208</td>
<td>208</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LFBGA (0.8 mm)</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>280</td>
<td>280</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BGA (1.0 mm)</th>
<th>QL8025</th>
<th>QL8050</th>
<th>QL8150</th>
<th>QL8250</th>
<th>QL8325</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>484</td>
<td>484</td>
</tr>
</tbody>
</table>
Eclipse II Family

- Eclipse II Block Diagram
Eclipse II Family

- The Eclipse II logic cell is a dual register, multiplexer-based logic cell.
- It is designed for wide fan-in and multiple, simultaneous output functions.
- Both registers share CLK, SET, and RESET inputs.
- The second register has a two-to-one multiplexer controlling its input.
- The register can be loaded from the NZ output or directly from a dedicated input.
Eclipse II Family

- The complete logic cell consists of two six-input AND gates, four two-input AND gates, seven two-to-one multiplexers, and two D flip-flops with asynchronous SET and RESET controls.
- The cell has a fan-in of 30 (including register control lines), fits a wide range of functions with up to 17 simultaneous inputs, and has six outputs (four combinational and two registered).
- The high logic capacity and fan-in of the logic cell accommodates many user functions with a single level of logic delay while other architectures require two or more levels of delay.
Eclipse II Family

- Shown at right, the Eclipse II 2,304-bit RAM Module
- The Eclipse II Product Family includes up to 24 dual-port 2,304-bit RAM modules for implementing RAM, ROM, and FIFO functions.
Eclipse II Family

- Cascaded RAM Modules
- Each module is user-configurable into two different block organizations and can be cascaded horizontally to increase their effective width, or vertically to increase their effective depth
Eclipse II Family

- The number of RAM modules varies from 4 to 24 blocks for a total of 9.2 K to 55.3 K bits of RAM.
- Using the two “mode” pins, designers can configure each module into 128 x 18 and 256 x 9.
- The blocks are also easily cascaddable to increase their effective width and/or depth.
Eclipse II Family

- Embedded Computational Unit (ECU)
- By embedding a dynamically reconfigurable computational unit, the Eclipse II device can address various arithmetic functions efficiently.
- ECU blocks are placed next to the SRAM circuitry for efficient memory/instruction fetch and addressing for DSP algorithmic implementations.

ECU Block Diagram
Eclipse II Family

- Up to twelve 8-bit MAC functions can be implemented per cycle for a total of 1 billion MACs/s when clocked at 100 MHz.
- Additional multiply-accumulate functions can be implemented in the programmable logic.

<table>
<thead>
<tr>
<th>Device</th>
<th>ECUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL8325</td>
<td>12</td>
</tr>
<tr>
<td>QL8250</td>
<td>10</td>
</tr>
<tr>
<td>QL8150</td>
<td>0</td>
</tr>
<tr>
<td>QL8050</td>
<td>0</td>
</tr>
<tr>
<td>QL8025</td>
<td>0</td>
</tr>
</tbody>
</table>

ECU Blocks
Eclipse II Family

- Phase Locked Loop (PLL)
- The QuickLogic built-in PLLs support a wider range of frequencies than many other PLLs.
- These PLLs also have the ability to support different ranges of frequency multiplications or divisions, driving the device at a faster or slower rate than the incoming clock frequency.

PLL Block Diagram
Eclipse II Family

- Eclipse II I/O Cell
- Eclipse II offers banks of programmable I/Os that address many of the bus standards that are popular today.
- Each bi-directional I/O pin is associated with an I/O cell which features an input register, an input buffer, an output register, a three-state output buffer, an output enable register, and 2 two-to-one output multiplexers.
Assignment

- Read over the datasheets for:
  - Eclipse
  - EclipsePlus
  - QuickRAM
  - pASIC 3
- Compare the architecture of these to the Eclipse II family
- Come to class on Monday prepared to discuss your findings