FPGA Challenges and Opportunities at 40 nm and Beyond

Vaughn Betz
Senior Director, Software Engineering
Altera
Scaling: Good for FPGAs
Scaling FPGAs: The Opportunity

- ~2X logic, RAM, etc. density
  - More processing on one chip
  - Lower system cost & power
  - Enable higher performance systems

- New features
  - Block RAM, DSP, serial interfaces, …
  - Enable new systems or fewer chips

- (Maybe) Higher speed fabric
  - Higher performance system or
    smaller & cheaper system (narrower datapath)
ASICs & Scaling

- Standard cell ASIC @ 40 nm
  - ~$4 M / mask set * 2 spins = $8 M
  - Test & product engineering ~$7 M
  - Design, verification, software ~$25 M

- Economics
  - $40 M development cost
  - 20% of revenue on R & D → need $200 M revenue
  - 10% market share → Need a $2 B market

- Result
  - Falling ASIC starts
  - Most still in 130 nm and above
  - Structured ASICs
  - ASICs increasing programmability → try to increase market size
Time-to-Market Economics

Favour FPGAs over ASICs
Drive FPGAs to next process early

$99M Max Available Revenue

$62M Max Revenue after Delayed Entry

PRODUCT LIFETIME – 2yr

© 2009 Altera Corporation Altera, Quartus & Stratix are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.
FPGAs: Process Leaders

- **Stratix IV**
  - Shipped in 2008
  - First 40 nm FPGA & one of the first 40 nm chips
    - FPGA designed simultaneously with 40 nm process
  - 3 years & >$200 million to develop hardware + software + IP
  - Process driver: large & regular; contains logic & RAM
  - 40 nm allows integration of new hard ("ASIC") functions

- **Pipelined development**
  - 28 nm underway for two years
Stratix IVGX 230 (Mid-Size Device)

Adaptive Logic Modules

RAM Blocks (M9K & M144K)
Digital Signal Processing

DSP Blocks
General I/O
Serial Interfaces

High Speed Serial Interfaces
## Stratix IV Overview

<table>
<thead>
<tr>
<th>Feature</th>
<th>Stratix III (65 nm)</th>
<th>Stratix IV (40 nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Elements</td>
<td>340k</td>
<td>820k</td>
</tr>
<tr>
<td>RAM bits</td>
<td>16 Mb + 4 Mb</td>
<td>33 Mb + 8.5 Mb</td>
</tr>
<tr>
<td>18x18 multipliers</td>
<td>768</td>
<td>1360</td>
</tr>
<tr>
<td>General I/O</td>
<td>1104</td>
<td>1104</td>
</tr>
<tr>
<td>High-speed serial links</td>
<td>0</td>
<td>48 transmit + 48 receive @ 11.3 Gb/s</td>
</tr>
<tr>
<td>Hard PCIe blocks</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Clock generation</td>
<td>12 PLL(x10)</td>
<td>12 PLL(x10) + 32 serial recovered + 24 serial transmit</td>
</tr>
<tr>
<td>Clock distribution</td>
<td>16 Global + 88 Quadrant + 132 PCLK</td>
<td>16 Global + 88 Quadrant + 132 PCLK</td>
</tr>
</tbody>
</table>
Scaling Challenges
1. Designing & Optimizing the Fabric
FPGA Fabric: Converging Technologies

- Huge space → no one can optimize by instinct
- $250 M + 3 years to implement your ideas
  - Risk of over-conservatism
Architect via Virtual Prototyping

Customer Designs
IP, Reference Designs

Parameterized Quartus
Synthesis

VPR++ Place & Route

Timing, Area, Power Models

VPR++ Analysis:
Speed, Area, Routability, Power
### Major Innovations

<table>
<thead>
<tr>
<th>Family</th>
<th>Major Innovation</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stratix</td>
<td>100% Direct-drive, Optimized segmented routing</td>
<td>+40% Fmax, -40% area vs. APEX (not including process adv.) Redundant and repairable</td>
</tr>
<tr>
<td>Stratix II</td>
<td>Adaptive Logic Module</td>
<td>+26% Fmax, -7% area vs. Stratix (not including process adv.)</td>
</tr>
<tr>
<td>Stratix III</td>
<td>Programmable Power</td>
<td>Full speed of 65 nm process, ½ leakage of 90 nm</td>
</tr>
</tbody>
</table>
Hard Block Evaluation

1. Develop Parameterized Soft IP
2. Create Configurable Hard IP
3. Specific IP in soft fabric
4. Area, Power, Speed
5. Estimate Usage
6. Include routing ports!
7. Net Win?
2. I/O Bandwidth
I/O Bandwidth

- Processing elements scale
  - ~2X more logic, RAM, DSP each generation
  - Stratix IV on-chip RAM bandwidth ~10 TB/s!

- I/O transistors, PCB traces, package balls don’t scale
  - Roughly same number of I/Os per device

- Need: higher speed I/Os to keep datapath fed
  - 8.5 (SIVGX) to 11.3 (SIVGT) Gb/s serial transceivers
  - 1.067 Gb/s (533 MHz) memory interfaces
  - Total: ~150 GB/s bandwidth

- Challenges
  - Circuit speed & timing closure
  - Signal integrity
Stratix IV GX Embedded Transceivers

- Up to 48 receive + 48 transmit transceivers
  - 3 Gb/s to 11.3 Gb/s
  - Clock recovered from data stream
- Very high speed analog
- Many protocols → highly configurable analog & digital logic
Build hard PCIe?

- Soft logic size: tens of thousands of LEs
- Standard cells: smaller
- Hard logic faster: can narrow datapath (smaller)
- Protocol fixed and widely used
- Add muxing and always build all options (Gen1, Gen2, …)
Memory Interfaces (DDR2, DDR3, …)

- Strobe (DQS) sent with several bits of data (DQ)
- Challenge: narrow data-valid window
- Solution:
  - Minimize jitter
  - Harden data capture logic, carefully match delays
  - **Calibrate** by modifying programmable delays for each DQ bit & DQS

Before de-skew -- small valid capture window

<table>
<thead>
<tr>
<th>DQs</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
<th>75</th>
<th>90</th>
<th>105</th>
<th>120</th>
<th>135</th>
<th>150</th>
<th>165</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

De-skew maximizes valid capture window

<table>
<thead>
<tr>
<th>DQs</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
<th>75</th>
<th>90</th>
<th>105</th>
<th>120</th>
<th>135</th>
<th>150</th>
<th>165</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Memory Interface Stack: Hard vs. Soft

- Some PHY circuitry very high speed, needs carefully matched delays
  → Hard circuitry

- PHY calibration: logic frequently changes
  → Soft logic

- Memory controller, multi-port interface
  - Relatively small (low thousands of LEs)
  - Many algorithms and needs
  - Soft logic
3. Device Modeling
Device Modeling Challenges

- Smaller transistors
  - More process variation

- Lower operating voltage, with little $V_{th}$ scaling
  - Increased sensitivity to power supply noise

- 2\textsuperscript{nd}-order transistor effects increasing
  - More timing corners

- Faster clock speeds and edge rates
  - Less ability to guardband
  - Increased importance of jitter & signal integrity models

- Still need fast, easy-to-interpret analysis
ASIC Class Timing Analysis (Timequest)

- Model rise-rise, rise-fall, fall-rise, fall-fall delays
  - Propagate rise and fall delays through circuit
  - *unateness* aware – ignores impossible transitions

- Each delay is a min-max range
  - Covers on-die variation, transistor aging effects

- Analyze *and optimize* at 3 corners
  - A *corner* is a combination of process, temperature, voltage
Interconnect Timing: Leading ASICs

✗ FPGA routing: too many combinations for tables
✓ Quartus circuit simulates each route
  - Non-linear models for transistors
  - Extracted R & C for wires
  - Specialized to FPGA circuitry: 10,000x faster than HSPICE

Full waveform propagated, with all non-linear effects
Calibrated I/O Interfaces

- Calibrated interfaces (e.g. DDR3): timing is not static
- But can’t simply assume calibration works
  - Need to analyze assuming worst-case devices, V, T, & P variation to ensure robust system
- Solution: extend TimeQuest timing analyzer
  - Calibration algorithm modeled
  - Ability to recapture margin explicitly included in timing analysis
Simultaneous switching noise (SSN)

- Noise induced on *victim* I/O due to switching of other *aggressor* I/Os

- Faster edge rates & higher I/O density worsen
- Reduce with FPGA & package design
- But cannot eliminate failures for all designs, on all boards
Quartus II SSN Analyzer

- Models FPGA, package and board
  - Signal paths and power-distribution network

- HSPICE (full design): 1 week
- SSN Analyzer: 30 minutes
- Displays signal margin & problem pins
- Enables analysis of mitigation techniques
Power Integrity

- 0.9 V operating voltage – can only afford tens of mV drop
- Hard & expensive: low-impedence for all freq.
- But design could go from 2 A – 12 A in one cycle!
- Over-engineer or analyze and forbid?
4. Power
Power

- Twice as many transistors
- Naturally more leaky
- But power budget per device fixed
  - About 2 – 20 W for high-end FPGAs

→ Innovate to control power while sacrificing minimum performance
Can We Do Better than the ‘Universal Curve’?

- Optimizing along the curve gives fixed choices
- Choose different values for each transistor
  - Helps for some transistors which are never speed critical (e.g. CRAM)
  - But most transistors will need to compromise between speed & power
Design-Specific Power Optimization

- Only a small fraction of logic is performance critical

Slack Histogram

- Not performance critical
- Performance critical

Number of connections

Slack %

90-100% 80-90% 70-80% 60-70% 50-60% 40-50% 30-40% 20-30% 10-20% 0-10%

0 1,000 2,000 3,000 4,000 5,000 6,000 7,000 8,000

© 2009 Altera Corporation Altera, Quartus & Stratix are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.
Programmable Speed vs. Leakage

Source Body Drain

Gate

0 V

High speed (HS)

V_{\text{Body}} — Automatically controlled by software

Low power (LP)

< 0 V

Note: A simple “model” showing Programmable Power Technology. Actual implementation varies and is patented.
Tradeoff 1: How Much Back-Bias?

- Power minimum at LP ~20% slower than HS
  - Low Power logic has 40% the static power of HS at that point
Tradeoff 2: Control Granularity

- Finer grain control
  - Can set more transistors to low-power state
  - But costs area (well spacing, CRAM, pass gate)

- Critical paths naturally cluster
  - OK for many transistors to be grouped into a tile

- Controlling logic and routing with a single setting
  - Small increase in power, larger area reduction

- Controlling pairs of LABs together: also good

- Stratix IV tile: LAB pair, DSP or RAM block
Most Tiles Are Low Power

All Clocks at Maximum Speed (Worst Case)

- Normal Power Optimization

Average: 11% to 15%
5. Designer Productivity
Designer Productivity

- FPGA density doubles
  - CAD problem size doubles
  - Designers need to create 2X the logic in the same time

- But CPU speed increase $\ll 2X$
  - Faster algorithms
  - Parallel CAD
  - Incremental compile

- Designer typing, debug speed increase $\ll 2X$
  - High quality CAD $\rightarrow$ reduce designer intervention
  - Higher levels of design abstraction
The Compile Time Challenge

Logic Elements (Thousands)

Relative SpecINT

Logic Elements

Relative CPU Speed
Improvements More than Bridge the Gap

Quartus II Compilation Time History
(Relative time for a fixed design, on a fixed CPU)

- Stratix
- Stratix II
- Stratix III (Parallel)
- Stratix IV (Parallel)

QII 9.1 expectation
Incremental Compilation

- Define partitions
  - CAD will not optimize across partitions
  - Can re-synthesize, place and route one partition alone
  - Faster compile time
  - Fewer iterations because other logic unchanged

- Work in progress
  - Incremental compile without the designer identifying partitions?
  - Challenge: global optimizations
Efficiency & Programming Ease

Scaling favours technologies that trade efficiency for simpler programming.

- **Power & System Cost**
- **Development Difficulty & Cost**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Power &amp; System Cost</th>
<th>Development Difficulty &amp; Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>DSP</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>FPGA</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Struct. ASIC</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Std. Cell</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Full Custom</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
The Past (1984): Editing Switches
The Present: HDL Design Flow

Verilog, VHDL

Timing & Other Constraints

Synthesis

Placement and Routing

Timing and Power Analyzer

Timing, Power and Area Optimized Design
The Future: SOPC Builder Switch Fabric

- Focus on your unique functions
- Re-use IP and let SOPC Builder integrate the system
The Future: DSP Builder Design Flow

Matlab/Simulink domain
(System simulation and verification)

HDL/hardware domain
(Hardware implementation/RTL simulation)
Fixed frequency design for high throughput
  - > 400 MHz for huge designs

*Automatic* datapath widen/narrow to match data rate

*Automatic* register insertion & pipeline balancing

*Automatic* time-domain multiplexing of hardware
The Future: High-Level Languages to HW

- Incremental
  - SystemVerilog

- Bigger gains
  - Catapult C, C2H, ImpulseC, AutoPilot
  - RapidMind for FPGAs?
  - Others?

- Co-develop target applications and tool

- Plus novel debug tools
Summary
Summary

- Scaling favours the programmable
  - FPGAs
  - Processors
  - Can ASICs embed enough programmability?

- Challenges
  1. Architecture: fabric and configurable hard blocks
  2. I/O bandwidth
  3. Device modeling
  4. Power & power integrity
  5. Keeping designers productive: compile time, new design & debug tools
Thank You

Stratix IV GX (EP4SGX230)
1 Billion Transistors

GX530: 2 billion transistors