18-643 Lecture 3: FPGA on Moore’s Law

James C. Hoe
Department of ECE
Carnegie Mellon University
Housekeeping

• Your goal today: get caught up on 3 decades of progress (upto 2010’ish)

• Notices
  – Complete survey on Canvas, past due
  – Handout #2: lab 0, due noon, 9/13
    **Be sure to watch Shashank’s very helpful video!!**
  – Handout #3: Term Project Intro

• Readings (see lecture schedule online)
  – for next time: skim [Ahmed, et al., 2016] and [Chromczak, et al., 2020]
Where we stopped last time: FPGA as Universal Fabric

I/O pins

programmable lookup tables (LUT) and flip-flops (FF)
aka “soft logic” or “fabric”

programmable routing
Fast-forward through Moore’s Law

<table>
<thead>
<tr>
<th>Logic Capacity (gates)</th>
<th>Configurable Logic Blocks</th>
<th>User I/Os</th>
<th>Configuration Program (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC2064</td>
<td>1200</td>
<td>64</td>
<td>58</td>
</tr>
<tr>
<td>XC2018</td>
<td>1800</td>
<td>100</td>
<td>74</td>
</tr>
</tbody>
</table>

XC2064/XC2018 Logic Cell Arrays: Product Specification

<table>
<thead>
<tr>
<th>MPSoC Processing System</th>
<th>Kintex UltraScale</th>
<th>Kintex UltraScale+</th>
<th>Virtex UltraScale</th>
<th>Virtex UltraScale+</th>
<th>Zynq UltraScale+</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Logic Cells (K)</td>
<td>318–1,451</td>
<td>356–1,143</td>
<td>783–5,541</td>
<td>862–3,780</td>
<td>103–1,143</td>
</tr>
<tr>
<td>Block Memory (Mb)</td>
<td>12.7–75.9</td>
<td>12.7–34.6</td>
<td>44.3–132.9</td>
<td>23.6–94.5</td>
<td>4.5–34.6</td>
</tr>
<tr>
<td>UltraRAM (Mb)</td>
<td>0–36</td>
<td>90–360</td>
<td>90–360</td>
<td>0–36</td>
<td></td>
</tr>
<tr>
<td>HBM DRAM (GB)</td>
<td>0–8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP (Slices)</td>
<td>768–5,520</td>
<td>1,368–3,528</td>
<td>600–2,880</td>
<td>2,280–12,288</td>
<td>240–3,528</td>
</tr>
<tr>
<td>DSP Performance (GMAC/s)</td>
<td>8,180</td>
<td>6,287</td>
<td>4,268</td>
<td>21,897</td>
<td>6,287</td>
</tr>
<tr>
<td>Transceivers</td>
<td>12–64</td>
<td>16–76</td>
<td>36–120</td>
<td>32–128</td>
<td>0–72</td>
</tr>
<tr>
<td>Max. Transceiver Speed (Gb/s)</td>
<td>16.3</td>
<td>32.75</td>
<td>30.5</td>
<td>32.75</td>
<td>32.75</td>
</tr>
<tr>
<td>Max. Serial Bandwidth (full duplex) (Gb/s)</td>
<td>2,086</td>
<td>3,268</td>
<td>5,616</td>
<td>8,384</td>
<td>3,268</td>
</tr>
<tr>
<td>Integrated Blocks for PCIe®</td>
<td>1–6</td>
<td>0–5</td>
<td>2–6</td>
<td>2–6</td>
<td>0–5</td>
</tr>
<tr>
<td>Memory Interface Performance (Mb/s)</td>
<td>2,400</td>
<td>2,666</td>
<td>2,400</td>
<td>2,666</td>
<td>2,666</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>312–832</td>
<td>280–668</td>
<td>338–1,456</td>
<td>208–832</td>
<td>82–668</td>
</tr>
<tr>
<td>I/O Voltage (V)</td>
<td>1.0–3.3</td>
<td>1.0–3.3</td>
<td>1.0–3.3</td>
<td>1.0–1.8</td>
<td>1.0–3.3</td>
</tr>
</tbody>
</table>

[Table 1, UltraScale Architecture and Product Datasheet: Overview]
30 Years of Becoming Hardwired
Why Hardwired Logic

• LUTs already can do everything (digital)
• Revisit: why hardwired flip-flop in CLB?
  – would take 4 LUTs to make 1 M-S flip-flop
  – LUT-built FF would have poor timing
  – almost all designs affected in cost and speed
• Makes sense to hardwire a functionality
  – needed by everyone (or by the big customers)
  – expected benefit outweigh displaced LUT area, i.e.,
    • much more expensive/slow in LUTs
    • easy/cheap to ignore when not in use

Hardwiring is a great thing if it is usable and is used
E.g., Special Support for Addition

- A full-adder fits perfectly in 1 CLB with 2x3LUTs
- But carry propagation slow---flow through several configurable connections and two switch blocks
- Addition is pretty important to most designs
Fast Carry Logic (1990s)

- Cost = 1 (real) wire and 1 mux
- Huge win in adder performance (32-bit@33MHz)

If arithmetic is so important, why not put in real adders? How about multipliers?
Hard Multipliers (2000s)

• Motivating forces
  – DSP became an important domain
  – very expensive and slow to multiply in LUTs
  – dies large enough to spare some area

• Virtex-II hardwired multiplier “macro” blocks
  – 18-bit inputs, full 36-bit product
  – explicit instantiation or inferable from RTL
  – relatively cheap (since native implementation)
  – but no hard adders, why?

Adders came later as a part of MAC in DSP slices

In the meanwhile, multiply faster/cheaper than add!!
An Early Multiplier Blocks
Xilinx Virtex-II, circa 2000

[Figure 54: Multiplier Block]

Where are these hard DSP slices?
How to get to them?
MACROs: a disturbance in the force . . .

Too much vs not enough?
Benefit of using macro outweigh cost of getting to one?

[Figure 48: Virtex-II Platform FPGAs: Complete Data Sheet]
optional pipeline stages
inferable from RTL and retiming
Stratix/Arria-10 IEEE-754 DSPs

[Intel Stratix-10 FPGA Features]
Stratix-10 NX AI Tensor Block

Optimized for small datatypes INT4/8 and Block FP12/16 used in ML

[Intel Stratix-10 FPGA Features]
Memory

• Flip-flops relatively scarce (only 1-bit per CLB)
• Need more storage when applications moved beyond FSM controllers and glue logic
• Option A: LUTs repurposable as 16x1-bit SRAMs
• Option B: 4Kb (now 32Kb) 2-ported SRAM blocks
  – very compact, very fast because native in silicon
  – explicit instantiation or inferable from RTL
    (tool can even decide which SRAM option to use)
  – configurable and combinable to a wide range of sizes and aspect ratios

Where are they? How to connect up to them?
Processor Cores

• Not everything needs to be in hardware; not everything improves when made into hardware
• Augment fabric with simple embedded CPUs
  – provide universality of functionality
  – easy handling of irregular, sequential operations
  – easy handling anything that doesn’t need to be fast
• Interests developed in early 2000s when FPGA applications grew to whole systems with DRAM, video, and Ethernets, etc.

*Hard or soft core?*
Hardcore vs Softcore

• First came PowerPC hardcores on Virtex-II
  – you got 2 whether you needed it or not
  – new tool promote IP-based system building
  – entirely soft-logic built surroundings: busses and IPs (DRAM controller, Ethernet, video, . . . .)

• Microblaze softcores took over in later rounds
  – Xilinx proprietary ISA (runs OS, gcc and all that)
  – configurable for cost-performance tradeoff
  – available in RTL to some folks
  – by this time, softcore footprint and performance was acceptable

Several 3rd-party softcores existed in that era, e.g., LEON SPARC
Embedding PowerPC in Fabric

- everything else is soft
- two hierarchies of soft-logic busses (slow and slower)
- special on-chip memory (OCM) port allows ld/st directly into fabric
- CoreGen Library of IPs to hang off the busses
Hardcores Return in Virtex7 (~2010)

- This time in a complete, full-speed, fully-capable, two-core Cortex-A9 system
- Latest Ultrascale uses 64-bit ARMv8 Cortex-A53 + ARM R5 + Mali GPU
- Why ARMs?

[Figure 3-1, Zynq-7000 All Programmable SoC Technical Reference Manual]
Hardcore vs Softcore

- **Table 4.2: The Zynq Book**

<table>
<thead>
<tr>
<th>Processor</th>
<th>Configuration</th>
<th>DMIPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroBlaze 900LUT/700FF/2BRAM to 3800LUT/3200FF/6DSP/21BRAM</td>
<td>area optimized (3-stage)</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>perf. optimized (5-stage) with branch optimizations</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td>perf. optimized (5-stage) without branch optimizations</td>
<td>259</td>
</tr>
<tr>
<td>ARM Cortex-A9</td>
<td>1GHz; both cores combined</td>
<td>5000</td>
</tr>
</tbody>
</table>

- **Table 4.3: The Zynq Book**

<table>
<thead>
<tr>
<th>Processor</th>
<th>Configuration</th>
<th>CoreMark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroBlaze</td>
<td>125MHz; 5-stage (Virtex-5)</td>
<td>238</td>
</tr>
<tr>
<td>ARM Cortex-A9</td>
<td>1GHz; both cores combined</td>
<td>5927</td>
</tr>
<tr>
<td>ARM Cortex-A9</td>
<td>800MHz; both cores combined</td>
<td>4737</td>
</tr>
</tbody>
</table>

PPC405 about 1/5 of ARM in Figure 4.3 of The Zynq Book
Die Area “Return on Investment”

Soft-logic logic dominates die area, but compute/storage concentrated in DSP and BRAM—consider what if 100% soft or 100% hard.
Xilinx ASMBL Architecture
(Application Specific Modular Block Arch.)

- Xilinx fabric assembled from composable tall-and-thin strip types, CLB, BRAM, DSP, I/O, etc.
- Derivative products at the cost of just new masks
  - vary capacity by composing more or less strips
  - domain-specialization by varying ratios of strips e.g., {DSP+IP} vs logic for DSP vs ASIC replacement market
  - variations handled by parameterization in design tool algorithms
Stacked Silicon Interconnect (SSI)

- 2.5D stacking: multiple dies on passive interposer
  - lower latency, higher bandwidth, lower power than crossing package
  - much better yield than equivalent capacity monolithic device
  - mix dies for domain-specialization
  - possible to insert customer proprietary dies?

[Figure 1, Stacked & Loaded: Xilinx SSI, 28-Gbps I/O Yield Amazing FPGAs, Xcell, Q1 2011]
Intel’s take on 2.5D with EMIB

- monolithic fabric
- displace noisy, hot analog IPs
- connect same-package HBMs
- connect 3rd-party chiplets?

[Figure 8, Enabling Next-Generation Platforms Using Altera’s 3D System-in-Package Technology]
Reviewing Hard IPs Added Over Time

- **1990s**
  - fast carry
  - LUT RAM
  - block RAM

- **2000s**
  - programmable clock generator
  - PowerPC core
  - gigabit transceiver
  - multiplier and DSP splices
  - Ethernet and PCI-E

- **2010s**
  - system monitor
  - ADC
  - power management
  - ARM cores and GPU
  - DRAM controller
  - floating point arithmetic
  - “UltraRAM” hierarchy (up to 500Mbits)
  - HBM controllers

- **2020s** . . . . . . next lecture
Chicken or Egg First?

- **1990s**: glue logic, embedded cntrl, interface logic
  - reduce chip-count, increase reliability
  - rapid roll-out of “new” products

- **2000s**: DSP and HPC
  - strong need for performance
  - abundant parallelism and regularity
  - low-volume, high-valued

- **2010s**: communications and networking
  - throughput performance
  - fast-changing designs and standards
  - price insensitive
  - $value in field updates and upgrades
SoC with reconfigurable fabric (2010s)

## Xilinx Ultrascale Offerings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Kintex UltraScale</th>
<th>Kintex UltraScale+</th>
<th>Virtex UltraScale</th>
<th>Virtex UltraScale+</th>
<th>Zynq UltraScale+</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPSoC Processing System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Logic Cells (K)</td>
<td>318-1,451</td>
<td>356-1,143</td>
<td>783-5,541</td>
<td>862-3,780</td>
<td>103-1,143</td>
</tr>
<tr>
<td>Block Memory (Mb)</td>
<td>12.7-75.9</td>
<td>12.7-34.6</td>
<td>44.3-132.9</td>
<td>23.6-94.5</td>
<td>4.5-34.6</td>
</tr>
<tr>
<td>UltraRAM (Mb)</td>
<td>0-36</td>
<td></td>
<td>90-360</td>
<td>0-36</td>
<td></td>
</tr>
<tr>
<td>HBM DRAM (GB)</td>
<td></td>
<td></td>
<td>0-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP (Slices)</td>
<td>768-5,520</td>
<td>1,368-3,528</td>
<td>600-2,880</td>
<td>2,280-12,288</td>
<td>240-3,528</td>
</tr>
<tr>
<td>DSP Performance (GMAC/s)</td>
<td>8,180</td>
<td>6,287</td>
<td>4,268</td>
<td>21,897</td>
<td>6,287</td>
</tr>
<tr>
<td>Transceivers</td>
<td>12-64</td>
<td>16-76</td>
<td>36-120</td>
<td>32-128</td>
<td>0-72</td>
</tr>
<tr>
<td>Max. Transceiver Speed (Gb/s)</td>
<td>16.3</td>
<td>32.75</td>
<td>30.5</td>
<td>32.75</td>
<td>32.75</td>
</tr>
<tr>
<td>Max. Serial Bandwidth (full duplex) (Gb/s)</td>
<td>2,086</td>
<td>3,268</td>
<td>5,616</td>
<td>8,384</td>
<td>3,268</td>
</tr>
<tr>
<td>Integrated Blocks for PCIe®</td>
<td>1-6</td>
<td>0-5</td>
<td>2-6</td>
<td>2-6</td>
<td>0-5</td>
</tr>
<tr>
<td>Memory Interface Performance (Mb/s)</td>
<td>2,400</td>
<td>2,666</td>
<td>2,400</td>
<td>2,666</td>
<td>2,666</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>312-832</td>
<td>280-668</td>
<td>338-1,456</td>
<td>208-832</td>
<td>82-668</td>
</tr>
<tr>
<td>I/O Voltage (V)</td>
<td>1.0-3.3</td>
<td>1.0-3.3</td>
<td>1.0-3.3</td>
<td>1.0-1.8</td>
<td>1.0-3.3</td>
</tr>
</tbody>
</table>

[Table 1, UltraScale Architecture and Product Datasheet: Overview]
# Intel Stratix-10 Offerings

## Product Line
<table>
<thead>
<tr>
<th>PRODUCT LINE</th>
<th>GX 400 SX 400</th>
<th>GX 650 SX 650</th>
<th>GX 850 SX 850</th>
<th>GX 1100 SX 1100</th>
<th>GX 1650 SX 1650</th>
<th>GX 2100 SX 2100</th>
<th>GX 2500 SX 2500</th>
<th>GX 2800 SX 2800</th>
<th>GX 4500 SX 4500</th>
<th>GX 5500 SX 5500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic elements (LEs)</td>
<td>378,000</td>
<td>612,000</td>
<td>841,000</td>
<td>1,092,000</td>
<td>1,624,000</td>
<td>2,056,000</td>
<td>2,422,000</td>
<td>2,753,000</td>
<td>4,463,000</td>
<td>8,100,000</td>
</tr>
<tr>
<td>Adaptive logic modules (ALMs)</td>
<td>128,160</td>
<td>207,360</td>
<td>284,960</td>
<td>370,080</td>
<td>560,640</td>
<td>679,680</td>
<td>821,160</td>
<td>933,120</td>
<td>1,512,820</td>
<td>1,867,680</td>
</tr>
<tr>
<td>ALM registers</td>
<td>512,040</td>
<td>829,440</td>
<td>1,139,840</td>
<td>1,480,320</td>
<td>2,202,160</td>
<td>2,718,720</td>
<td>3,284,000</td>
<td>3,732,480</td>
<td>6,051,280</td>
<td>7,470,720</td>
</tr>
<tr>
<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>
### Hyper-Registers from Intel® HyperFlex™ FPGA architecture

| Millions of Hyper-Registers distributed throughout the monolithic FPGA fabric |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| M20K memory blocks | 1,537 | 2,489 | 3,477 | 4,401 | 5,851 | 6,501 | 9,963 | 11,721 | 7,033 | 7,033 |
| M20K memory size (Mb) | 30 | 49 | 68 | 86 | 114 | 127 | 195 | 229 | 137 | 137 |
| M2K memory size (Mb) | 2 | 3 | 4 | 6 | 8 | 11 | 13 | 15 | 23 | 29 |
| MLAB memory size (Mb) | 648 | 1,152 | 2,016 | 2,620 | 3,145 | 3,744 | 4,911 | 5,760 | 1,880 | 1,880 |
| MAB memory size (Mb) | 1,296 | 2,304 | 4,032 | 5,040 | 6,290 | 7,488 | 10,022 | 11,520 | 3,000 | 3,000 |
| MAB memory size (Mb) | 2,6 | 4,6 | 8,1 | 10,1 | 12,6 | 15,0 | 20,0 | 23,0 | 7,9 | 7,9 |
| Peak floating-point performance (TFLOPS) | 1.0 | 1.8 | 3.2 | 4.0 | 5.0 | 6.0 | 8.0 | 9.2 | 3.2 | 3.2 |
| Secure device manager | AES-256/SHA-256 bitstream encryption/authentication, physically unclonable function (PUF), ECDSA 256/384 boot code authentication, side channel attack protection |
| Hard processor system | Quad-core 64-bit ARM® Cortex®-A53 up to 1.5 GHz with 32 KB I/O cache, NEON® coprocessor, 1 MB L2 cache, direct memory access (DMA), system memory management unit, cache coherency unit, hard memory controllers, USB 2.0 x2, 1G EMAC x3, UART x2, SPI x4, I²C x5, general-purpose timers x7, watchdog timer x4 |
| Maximum user I/O pins | 392 | 400 | 736 | 736 | 736 | 736 | 736 | 736 | 816 | 816 |
| Total LVDS pairs 1.6 Gbps (RX or TX) | 192 | 192 | 360 | 360 | 336 | 336 | 336 | 576 | 576 | 816 |
| GXT full duplex transceiver count | 24 | 48 | 48 | 48 | 96 | 96 | 96 | 96 | 24 | 24 |
| GX full duplex transceiver count (up to 30 Gbps) | 16 | 32 | 32 | 32 | 64 | 64 | 64 | 64 | 16 | 16 |
| GX full duplex transceiver count (up to 17.4 Gbps) | 8 | 16 | 16 | 16 | 32 | 32 | 32 | 32 | 8 | 8 |
| PCI Express® (PCIe®) hard intellectual property (IP) blocks (Gen3 x16) | 1 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 1 | 1 |
| Memory devices supported | DDR4, DDR3, DDR2, DDR, QDR II, QDR II+, RLDRAM II, RLDRAM III, HMC, MSys |
Today’s Diverging Architectures

Are they FPGAs?
• spatial data/compute
• highly concurrent
• finely controllable
• reprogrammable

[Achronix Speedster MLP]
[Xilinx Versal]
[Intel Agilex]
Parting Thoughts

- FPGAs steadily moved away from universal fabric
  - efficiency of hardwired logic (driven by application demands) complements flexibility of reconfig. logic
  - architected deliberately to play up this advantage
- Retain a high degree of regularity to ease design and manufacturing
  - fastest way to use up transistors from Moore’s Law
  - power and performance advantage by just being first on new process
- Architectural evolution both push-and-pull with applications