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, 9/11
    
    **Use Piazza and watch TA step-by-step video!!**

  – Handout #3: Term Project Intro

• Readings (see lecture schedule online)
  – skim [Boutros, et al., 2021]
  – 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>Part Number</th>
<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>
<td>12038</td>
</tr>
<tr>
<td>XC2018</td>
<td>1800</td>
<td>100</td>
<td>74</td>
<td>17878</td>
</tr>
</tbody>
</table>

XC2064/XC2018 Logic Cell Arrays: Product Specification

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 1, UltraScale Architecture and Product Datasheet: Overview

what happened is more than Moore
30 Years of Becoming Hardwired
LUT-based Configurable Logic Block (simplified sketch)

- 2 functions ($f$ & $g$) of 3 inputs OR 1 function ($h$) of 4 inputs
- Hardwired FFs (too expensive/slow to fake)
- Just 10s of these in the earliest FPGAs
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 have atrocious setup/hold time
  - 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
Specialized Logic for Fast Carry

- 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?*
Xilinx XC4000 (1990s)

A 16-bit adder requires nine CLBs and has a combinatorial carry delay of 20.5 ns. Compare that to the 30 CLBs and 50 ns, or 41 CLBs and 30 ns in the XC3000 family.
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)
  – Still no hard adders

  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

Where are these hard DSP slices?
How to get to them?
Ultrascale DSP48E2

optional pipeline stages
inferable from RTL and retiming
Aside: Register Retiming

- Local transformations

- Preserves I/O relationships

- Tools use retiming
  - balance critical paths
  - absorb FFs into hard macros

always@(posedge clk) begin
  a1<=a; b1<=b;
  a2<=a1; b2<=b1;
  c<=a2*b2;
end

Pipelined multiply ➔
Stratix/Arria-10 IEEE-754 DSPs

[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 repurposeable 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?
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]
FPGA fabric not true blank slate

• FPGA Macros (especially RAM and DSP)
  – coarse functions and structures
  – some powerful but arbitrarily specific features
  – penalty is too huge to not get it right

• Inferable from RTL but . . . .
  – hard macros only does what it does
  – tools cannot recognize all “functional equivalent” descriptions
  – good idea to check inference report

*Straight out-of-the-box ASIC RTL likely suboptimal, sometimes not-mappable*
Example: Flip-Flops Inference

• Use asynch set or reset
  \[\Rightarrow\] not all FFs have async reset; prevents DSP retiming

• Use both set and reset
  \[\Rightarrow\] no FF has this; emulated externally with LUTs

• Use set and reset operationally
  \[\Rightarrow\] set/reset cannot use special global lines

• Active-low set/reset and enables
  \[\Rightarrow\] need LUTs to turn active-high
How could you know this?

• BRAM cannot be used if combinational read
• Shift registers can be made out of LUTs
  BUT! no set/reset and can’t read middle bits
• Registers will retime into multiplier and DSP (if no asynch reset)
• Use “initial” for power-on reset
• Timing analysis doesn’t do “latches”
• Many, many more like this. . .

Always want to RTMF!
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 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

[Xilinx Vertex II, early 2000]
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 ??from book</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 3\textsuperscript{rd}-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 Vertex Ultrascale Offerings

<table>
<thead>
<tr>
<th>Logic Resources</th>
<th>Device Name</th>
<th>XCVU065</th>
<th>XCVU080</th>
<th>XCVU095</th>
<th>XCVU125</th>
<th>XCVU160</th>
<th>XCVU190</th>
<th>XCVU440</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Logic Cells (K)</td>
<td>783</td>
<td>975</td>
<td>1,176</td>
<td>1,567</td>
<td>2,027</td>
<td>2,350</td>
<td>5,541</td>
<td></td>
</tr>
<tr>
<td>CLB Flip-Flops</td>
<td>716,160</td>
<td>891,424</td>
<td>1,075,200</td>
<td>1,432,320</td>
<td>1,852,800</td>
<td>2,148,480</td>
<td>5,065,920</td>
<td></td>
</tr>
<tr>
<td>CLB LUTs</td>
<td>358,080</td>
<td>445,712</td>
<td>537,600</td>
<td>716,160</td>
<td>926,400</td>
<td>1,074,240</td>
<td>2,532,960</td>
<td></td>
</tr>
<tr>
<td>Memory Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Distributed RAM (Kb)</td>
<td>4,830</td>
<td>3,980</td>
<td>4,800</td>
<td>9,660</td>
<td>12,690</td>
<td>14,490</td>
<td>28,710</td>
<td></td>
</tr>
<tr>
<td>Block RAM/FIFO w/ECC (36Kb each)</td>
<td>1,260</td>
<td>1,421</td>
<td>1,728</td>
<td>2,520</td>
<td>3,276</td>
<td>3,780</td>
<td>2,520</td>
<td></td>
</tr>
<tr>
<td>Block RAM/FIFO (18Kb each)</td>
<td>2,520</td>
<td>2,842</td>
<td>3,456</td>
<td>5,040</td>
<td>6,552</td>
<td>7,560</td>
<td>5,040</td>
<td></td>
</tr>
<tr>
<td>Total Block RAM (Mb)</td>
<td>44.3</td>
<td>50.0</td>
<td>60.8</td>
<td>88.6</td>
<td>115.2</td>
<td>132.9</td>
<td>88.6</td>
<td></td>
</tr>
<tr>
<td>Clock Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMT (1 MMCM, 2 PLLs)</td>
<td>10</td>
<td>16</td>
<td>16</td>
<td>20</td>
<td>28</td>
<td>30</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>I/O DLL</td>
<td>40</td>
<td>64</td>
<td>64</td>
<td>80</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>I/O Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transceiver Fractional PLL</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>13</td>
<td>15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Maximum Single-Ended HP I/Os</td>
<td>468</td>
<td>780</td>
<td>780</td>
<td>780</td>
<td>650</td>
<td>650</td>
<td>1,404</td>
<td></td>
</tr>
<tr>
<td>Maximum Differential HP I/O Pairs</td>
<td>216</td>
<td>360</td>
<td>360</td>
<td>360</td>
<td>300</td>
<td>300</td>
<td>648</td>
<td></td>
</tr>
<tr>
<td>Maximum Single-Ended HR I/Os</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>Maximum Differential HR I/O Pairs</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Integrated IP Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Monitor</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>PCIe® Gen1/2/3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Interlaken</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100G Ethernet</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>GTH 16.3Gb/s Transceivers</td>
<td>20</td>
<td>32</td>
<td>32</td>
<td>40</td>
<td>52</td>
<td>60</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>GTY 30.5Gb/s Transceivers</td>
<td>20</td>
<td>32</td>
<td>32</td>
<td>40</td>
<td>52</td>
<td>60</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Speed Grades</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Extended</td>
<td>-1H-2-3</td>
<td>-1H-2-3</td>
<td>-1H-2-3</td>
<td>-1H-2-3</td>
<td>-1H-2-3</td>
<td>-1H-2-3</td>
<td>-2-3</td>
<td></td>
</tr>
<tr>
<td>Industrial</td>
<td>-1-2</td>
<td>-1-2</td>
<td>-1-2</td>
<td>-1-2</td>
<td>-1-2</td>
<td>-1-2</td>
<td>-1-2</td>
<td></td>
</tr>
</tbody>
</table>

[UltraScale FPGA Product Tables and Product Selection Guide (XMP102)]
## Intel Agilex-10 Offerings

<table>
<thead>
<tr>
<th>PRODUCT LINE</th>
<th>AGF 006</th>
<th>AGF 008</th>
<th>AGF 012</th>
<th>AGF 014</th>
<th>AGF 019</th>
<th>AGF 022</th>
<th>AGF 023</th>
<th>AGF 027</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic elements (LEs)</td>
<td>573,480</td>
<td>764,640</td>
<td>1,178,525</td>
<td>1,437,240</td>
<td>1,918,975</td>
<td>2,208,075</td>
<td>2,308,080</td>
<td>2,692,760</td>
</tr>
<tr>
<td>Adaptive logic modules (ALMs)</td>
<td>194,400</td>
<td>259,200</td>
<td>399,500</td>
<td>487,200</td>
<td>650,500</td>
<td>748,500</td>
<td>782,400</td>
<td>912,800</td>
</tr>
<tr>
<td>ALM registers</td>
<td>777,600</td>
<td>1,036,800</td>
<td>1,598,000</td>
<td>1,948,800</td>
<td>2,602,000</td>
<td>2,994,000</td>
<td>3,129,600</td>
<td>3,651,200</td>
</tr>
<tr>
<td>High-performance crypto</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>blocks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eSRAM memory blocks</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>eSRAM memory size (Mb)</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>M20K memory blocks</td>
<td>2,844</td>
<td>3,792</td>
<td>5,900</td>
<td>7,110</td>
<td>8,500</td>
<td>10,900</td>
<td>10,464</td>
<td>13,272</td>
</tr>
<tr>
<td>M20K memory size (Mb)</td>
<td>56</td>
<td>74</td>
<td>115</td>
<td>139</td>
<td>166</td>
<td>212</td>
<td>204</td>
<td>259</td>
</tr>
<tr>
<td>MLAB memory count</td>
<td>9,720</td>
<td>12,960</td>
<td>19,975</td>
<td>24,360</td>
<td>32,525</td>
<td>37,425</td>
<td>39,120</td>
<td>45,640</td>
</tr>
<tr>
<td>MLAB memory size (Mb)</td>
<td>6</td>
<td>8</td>
<td>12</td>
<td>15</td>
<td>20</td>
<td>23</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>I/O PLL</td>
<td>12</td>
<td>12</td>
<td>16</td>
<td>16</td>
<td>10</td>
<td>16</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>Variable-precision digital</td>
<td>1,640</td>
<td>2,296</td>
<td>3,743</td>
<td>4,510</td>
<td>1,354</td>
<td>6,250</td>
<td>1,640</td>
<td>8,528</td>
</tr>
<tr>
<td>signal processing (DSP) blocks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 x 19 multipliers</td>
<td>3,280</td>
<td>4,592</td>
<td>7,486</td>
<td>9,020</td>
<td>2,708</td>
<td>12,500</td>
<td>3,280</td>
<td>17,056</td>
</tr>
<tr>
<td>Single-precision or half-</td>
<td>2.5 / 5.0</td>
<td>3.5 / 6.9</td>
<td>6.0 / 12.0</td>
<td>6.8 / 13.6</td>
<td>2.0 / 4.0</td>
<td>9.4 / 18.8</td>
<td>2.5 / 5.0</td>
<td>12.8 / 25.6</td>
</tr>
<tr>
<td>precision floating point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operations per second (TFLOPS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum EMIF x72</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

### Resources

- **IEEE 1588 v2 support**
- PMA direct
- Transceiver channel count: Up to 24 channels at 28.9 Gbps (NRZ) / 12 channels at 58 Gbps (PAM4)
- RS & KP FEC
- Networking support:
  - 400Gbe (4 x 100GbE hard IP blocks (10/25 GbE FEC/PCS/MAC))
  - IEEE 1588 v2 support
  - PMA direct
- PCIe hard IP block (4.0 x16) or bifurcable 2x PCIe 4.0 x 8 (EP) or 4x 4.0 x 4 (RP)
- SR-IOV 8PF / 2KVF
- VirtIO support
- Scalable IOV

---

[Intel FPGA Product Catalog]
Today’s Diverging Architectures

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

[Xilinx Versal]
[Achronix Speedster]
[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