 EECS150 – Digital Design
Lecture 2 – Synchronous Digital Systems and FPGAs

January 24, 2013

John Wawrzynek
Electrical Engineering and Computer Sciences
University of California, Berkeley

http://www-inst.eecs.berkeley.edu/~cs150

Outline

• Synchronous Systems Introduction
• Field Programmable Gate Arrays (FPGAs) Introduction
• Review of combinational logic
**Integrated Circuit Example**

- PowerPC microprocessor microphotograph
  - Superscalar (3 instructions/cycle)
  - 6 execution units (2 integer and 1 double precision IEEE floating point)
  - 32 KByte Instruction and Data L1 caches
  - Dual Memory Management Units (MMU)
  - External L2 Cache interface with integrated controller and cache tags.

Comprises only transistors and wires.

Connections to outside world (ex. motherboard)
- Memory interface
- Power (Vdd, GND)
- Clock input

**Clock Signal**

A source of regularly occurring pulses used to measure the passage of time.

- Waveform diagram shows evolution of signal value [in voltage] over time.
- Usually comes from an off-chip crystal-controlled oscillator.
- One main clock per chip/system.
- Distributed throughout the chip/system.
- "Heartbeat" of the system. Controls the rate of computation by directly controlling all data transfers.
Data Signals

Random adder circuit at a random point in time:

Observations:
1. Most of the time, signals are in either low- or high-voltage position.
2. When the signals are at the high- or low-voltage positions, they are not all the way to the voltage extremes (or they are past).
3. Changes in the signals correspond to changes in clock signal (but don’t change every cycle).

The facts:
1. Low-voltage represents binary 0 and high-voltage, binary 1.
2. Circuits are designed and built to be “restoring” and deviations from ideal voltages are ignored. Outputs close to ideal.
3. In synchronous systems, all changes follow clock edges.

Bus Signals

Signal wires grouped together often called a bus.

- $X_0$ is called the least significant bit [LSB]
- $X_3$ is called the most significant bit [MSB]
- Capital X represents the entire bus.
  - Here, hexadecimal digits are used to represent the values of all four wires.
  - The waveform for the bus depicts it as being simultaneously high and low. (The hex digits give the bit values). The waveform just shows the timing.
**Circuit Delay**

Digital circuits cannot produce outputs instantaneously.

- In general, the delay through a circuit is called the propagation delay. It measures the time from when inputs arrive until the outputs change.
- The delay amount is a function of many things. Some out of the control of the circuit designer:
  - Processing technology, the particular input values.
- And others under her control:
  - Circuit structure, physical layout parameters.

**Combinational Logic Blocks**

- Example four-input function:

```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>F(0,0,0,0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>F(0,0,0,1)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>F(0,0,1,0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>F(0,0,1,1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>F(0,1,0,0)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>F(0,1,0,1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>F(0,1,1,0)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>F(0,1,1,1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>F(1,0,0,0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>F(1,0,0,1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>F(1,0,1,0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>F(1,0,1,1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>F(1,1,0,0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>F(1,1,0,1)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>F(1,1,1,0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>F(1,1,1,1)</td>
</tr>
</tbody>
</table>
```

- True-table representation of function. Output is explicitly specified for each input combination.
- In general, CL blocks have more than one output signal, in which case, the truth-table will have multiple output columns.
Example CL Block

- 2-bit adder. Takes two 2-bit integers and produces 3-bit result.

<table>
<thead>
<tr>
<th>a1 a0</th>
<th>b1 b0</th>
<th>c2 c1 c0</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>000</td>
</tr>
<tr>
<td>00</td>
<td>01</td>
<td>001</td>
</tr>
<tr>
<td>00</td>
<td>10</td>
<td>010</td>
</tr>
<tr>
<td>00</td>
<td>11</td>
<td>011</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>001</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>010</td>
</tr>
<tr>
<td>01</td>
<td>10</td>
<td>011</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>010</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
<td>011</td>
</tr>
<tr>
<td>11</td>
<td>01</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>110</td>
</tr>
</tbody>
</table>

- Think about true table for 32-bit adder. It’s possible to write out, but it might take a while!

Theorem: Any combinational logic function can be implemented as a networks of logic gates.

Logic “Gates”

- Logic gates are often the primitive elements out of which combinational logic circuits are constructed.
  - In some technologies, there is a one-to-one correspondence between logic gate representations and actual circuits.
  - Other times, we use them just as another abstraction layer (FPGAs have no real logic gates).
- How about these gates with more than 2 inputs?
- Do we need all these types?
Example Logic Circuit

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<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>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- How do we know that these two representations are equivalent?

Logic Gate Implementation

- Logic circuits have been built out of many different technologies. If we have a basic logic gate (AND or OR) and inversion we can build a complete logic family.

CMOS Gate

Hydraulic

Mechanical LEGO logic gates. A clockwise rotation represents a binary “one” while a counter-clockwise rotation represents a binary “zero.”
Restoration

• A necessary property of any suitable technology for logic circuits is "Restoration".

• Circuits need:
  – to ignore noise and other non-idealities at their inputs, and
  – generate "cleaned-up" signals at their output.

• Otherwise, each stage would propagate input noise to their output and eventually noise and other non-idealities would accumulate and signal content would be lost.

Inverter Example of Restoration

Example (look at 1-input gate, to keep it simple):

- Inverter acts like a "non-linear" amplifier
- The non-linearity is critical to restoration
- Other logic gates act similarly with respect to input/output relationship.
Project platform: Xilinx ML505-110

FPGA: Xilinx Virtex-5 XC5VLX110T

Virtex-5 “die photo”

A die is an unpackaged part
**FPGA Overview**

- Basic idea: two-dimensional array of logic blocks and flip-flops with a means for the user to configure (program):
  1. the interconnection between the logic blocks,
  2. the function of each block.

**Simplified version of FPGA internal architecture:**
Why are FPGAs Interesting?

• Technical viewpoint:
  – For hardware/system-designers, like ASICs only better! “Tape-out” new design every few minutes/hours.
  – Does the “reconfigurability” or “reprogrammability” offer other advantages over fixed logic?
  – Dynamic reconfiguration? In-field reprogramming? Self-modifying hardware, evolvable hardware?

Why are FPGAs Interesting?

• Staggering logic capacity growth (10000x):

<table>
<thead>
<tr>
<th>Year Introduced</th>
<th>Device</th>
<th>Logic Cells</th>
<th>&quot;logic gate equivalents&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1985</td>
<td>XC2064</td>
<td>128</td>
<td>1024</td>
</tr>
<tr>
<td>2011</td>
<td>XC7V2000T</td>
<td>1,954,560</td>
<td>15,636,480</td>
</tr>
</tbody>
</table>

– FPGAs have tracked Moore’s Law better than any other programmable device.
**Why are FPGAs Interesting?**

- Logic capacity now only part of the story: on-chip RAM, high-speed I/Os, “hard” function blocks, ...
- Modern FPGAs are “reconfigurable systems”

![Xilinx Virtex-5 LX110T](image)

But, the heterogeneity erodes the “purity” argument. Mapping is more difficult. Introduces uncertainty in efficiency of solution.

---

**FPGAs are in widespread use**

Far more designs are implemented in FPGA than in custom chips.
User Programmability

- Latch-based (Xilinx, Altera, ...)

- Latches are used to:
  1. control a switch to make or break cross-point connections in the interconnect
  2. define the function of the logic blocks
  3. set user options:
     - within the logic blocks
     - in the input/output blocks
     - global reset/clock

- “Configuration bit stream” is loaded under user control

User Programmability

- Latch-based (Xilinx, Altera, ...)

- Latches are used to:
  1. control a switch to make or break cross-point connections in the interconnect
  2. define the function of the logic blocks
  3. set user options:
     - within the logic blocks
     - in the input/output blocks
     - global reset/clock

- “Configuration bit stream” is loaded under user control

Background (review) for upcoming

- A MUX or multiplexor is a combinational logic circuit that chooses between $2^N$ inputs under the control of N control signals.

- A latch is a 1-bit memory (similar to a flip-flop).
Idealized FPGA Logic Block

- 4-input look up table (LUT)
  - implements combinational logic functions
- Register
  - optionally stores output of LUT

4-LUT Implementation

- n-bit LUT is implemented as a $2^n \times 1$ memory:
  - inputs choose one of $2^n$ memory locations.
  - memory locations (latches) are normally loaded with values from user’s configuration bit stream.
  - Inputs to mux control are the CLB inputs.
- Result is a general purpose “logic gate”.
  - n-LUT can implement any function of n inputs!
LUT as general logic gate

- An n-lut as a direct implementation of a function truth-table.
- Each latch location holds the value of the function corresponding to one input combination.

Example: 4-lut

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>AND</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Example: 2-lut

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>AND</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Implements any function of 2 inputs.

How many of these are there?
How many functions of n inputs?

FPGA Generic Design Flow

- Design Entry:
  - Create your design files using:
    - schematic editor or
    - HDL (hardware description languages: Verilog, VHDL)
- Design Implementation:
  - Logic synthesis (in case of using HDL entry) followed by,
  - Partition, place, and route to create configuration bit-stream file
- Design verification:
  - Optionally use simulator to check function,
  - Load design onto FPGA device (cable connects PC to development board), optional “logic scope” on FPGA
    - check operation at full speed in real environment.
Example Partition, Placement, and Route

- Idealized FPGA structure:
- Example Circuit:
  - collection of gates and flip-flops

Circuit combinational logic must be “covered” by 4-input 1-output LUTs.
Flip-flops from circuit must map to FPGA flip-flops.
(Best to preserve “closeness” to CL to minimize wiring.)
Best placement in general attempts to minimize wiring.
Vdd, GND, clock, and global resets are all “prewired”.

Two partitions. Each has single output, no more than 4 inputs, and no more than 1 flip-flop. In this case, inverter goes in both partitions.

Note: the partition can be arbitrarily large as long as it has not more than 4 inputs and 1 output, and no more than 1 flip-flop.
Xilinx FPGAs (interconnect detail)

Colors represent different types of resources:

- Logic
- Block RAM
- DSP (ALUs)
- Clocking
- I/O
- Serial I/O + PCI

A routing fabric runs throughout the chip to wire everything together.
Configurable Logic Blocks (CLBs)

Slices define regular connections to the switching fabric, and to slices in CLBs above and below it on the die.

The LX110T has 17,280 slices.

X-Y naming convention for slices

X0, X2, ... are lower CLB slices.
X1, X3, ... are upper CLB slices.
Y0, Y1, ... are CLB column positions.
**Atoms: 5-input Look Up Tables (LUTs)**

The 5-input LUT computes any 5-input logic function. The timing is independent of function. Latches set during configuration.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>1</td>
</tr>
<tr>
<td>00001</td>
<td>0</td>
</tr>
<tr>
<td>00010</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>11101</td>
<td>0</td>
</tr>
<tr>
<td>11110</td>
<td>0</td>
</tr>
<tr>
<td>11111</td>
<td>1</td>
</tr>
</tbody>
</table>

Spr 2013

**Virtex-5 6-LUTs: Composition of 5-LUTs**

May be used as one 6-input LUT (D6 out) ...

... or as two 5-input LUTS (D6 and D5)

Combinational logic (post configuration)

The LX110T has 69,120 6-LUTs

6-LUT delay is 0.9 ns

Spr 2013
The simplest view of a slice

Four 6-LUTs

Four Flip-Flops

Switching fabric may see combinational and registered outputs.

An actual Virtex-5 slice adds many small features to this simplified diagram. We show them one by one ...

Two 7-LUTs per slice ...

Extra
multiplexers(F7AMUX, F7BMUX)

Extra inputs
(AX and CX)
Or one 8-LUTs per slice ...

Third multiplexer (F8MUX)

Third input (BX)

Configuring the "n" of an n-LUT ...

---

Extra muxes to chose LUT option ...

From eight 5-LUTs ... to one 8-LUT.

Combinational or registered outs.

Flip-flops unused by LUTs can be used standalone.
Virtex 5 Vertical Logic

We can map ripple-carry addition onto carry-chain block.

The carry-chain block also useful for speeding up other adder structures and counters.

Putting it all together ... a SLICEL.

The previous slides explain all SLICEL features.

About 50% of the 17,280 slices in an LX110T are SLICELs.

The other slices are SLICEMs, and have extra features.
Recall: 5-LUT architecture...

32 Latches.
Configured to 1 or 0.

Some parts of a logic design need many state elements.

SLICEMs replace normal 5-LUTs with circuits that can act like 5-LUTs, but can alternatively use the 32 latches as RAM, ROM, shift registers.

Efficient implementation of multiply, add, bit-wise logical. LX110T has 64 in a single column.
Table 1: Virtex-5 FPGA Family Members

<table>
<thead>
<tr>
<th>Device</th>
<th>Configurable Logic Blocks (CLBs)</th>
<th>Virtex-5 Silicon (µm)</th>
<th>Max Distributed RAM (Kb)</th>
<th>DSP48E Blocks</th>
<th>Block RAM Blocks</th>
<th>PowerPC Processor Block</th>
<th>Endpoint Blocks for PCI Express</th>
<th>Emerging MACs</th>
<th>Max Rocket/Falcon Zynq Blocks</th>
<th>Total I/O Buffers</th>
<th>Max User IOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC5VLX30D</td>
<td>80 x 30</td>
<td>4,600</td>
<td>320</td>
<td>32</td>
<td>64</td>
<td>32</td>
<td>1,152</td>
<td>2</td>
<td>N/A</td>
<td>13</td>
<td>400</td>
</tr>
<tr>
<td>XC5VLX60</td>
<td>120 x 30</td>
<td>7,200</td>
<td>480</td>
<td>48</td>
<td>96</td>
<td>48</td>
<td>1,728</td>
<td>6</td>
<td>N/A</td>
<td>17</td>
<td>560</td>
</tr>
<tr>
<td>XC5VLX85</td>
<td>120 x 54</td>
<td>12,060</td>
<td>840</td>
<td>48</td>
<td>192</td>
<td>96</td>
<td>3,456</td>
<td>6</td>
<td>N/A</td>
<td>17</td>
<td>560</td>
</tr>
<tr>
<td>XC5VLX110</td>
<td>160 x 54</td>
<td>17,280</td>
<td>1,120</td>
<td>64</td>
<td>256</td>
<td>128</td>
<td>4,008</td>
<td>6</td>
<td>N/A</td>
<td>23</td>
<td>800</td>
</tr>
<tr>
<td>XC5VLX150</td>
<td>160 x 76</td>
<td>24,320</td>
<td>1,840</td>
<td>128</td>
<td>384</td>
<td>192</td>
<td>6,912</td>
<td>6</td>
<td>N/A</td>
<td>23</td>
<td>800</td>
</tr>
<tr>
<td>XC5VLX200</td>
<td>160 x 108</td>
<td>34,560</td>
<td>2,280</td>
<td>128</td>
<td>384</td>
<td>192</td>
<td>6,912</td>
<td>6</td>
<td>N/A</td>
<td>23</td>
<td>800</td>
</tr>
<tr>
<td>XC5VLX250</td>
<td>240 x 108</td>
<td>51,840</td>
<td>3,420</td>
<td>192</td>
<td>576</td>
<td>268</td>
<td>10,968</td>
<td>6</td>
<td>N/A</td>
<td>33</td>
<td>1,200</td>
</tr>
<tr>
<td>XC5VLX250T</td>
<td>60 x 25</td>
<td>3,120</td>
<td>210</td>
<td>24</td>
<td>52</td>
<td>26</td>
<td>936</td>
<td>1</td>
<td>N/A</td>
<td>7</td>
<td>172</td>
</tr>
<tr>
<td>XC5VLX60T</td>
<td>80 x 30</td>
<td>4,600</td>
<td>320</td>
<td>32</td>
<td>72</td>
<td>35</td>
<td>1,296</td>
<td>2</td>
<td>N/A</td>
<td>12</td>
<td>360</td>
</tr>
<tr>
<td>XC5VLX85T</td>
<td>120 x 30</td>
<td>7,200</td>
<td>480</td>
<td>48</td>
<td>120</td>
<td>50</td>
<td>2,160</td>
<td>4</td>
<td>N/A</td>
<td>15</td>
<td>480</td>
</tr>
<tr>
<td>XC5VLX110T</td>
<td>160 x 54</td>
<td>17,280</td>
<td>1,120</td>
<td>64</td>
<td>296</td>
<td>148</td>
<td>5,328</td>
<td>6</td>
<td>N/A</td>
<td>15</td>
<td>480</td>
</tr>
<tr>
<td>XC5VLX150T</td>
<td>160 x 76</td>
<td>24,320</td>
<td>1,840</td>
<td>128</td>
<td>424</td>
<td>212</td>
<td>7,032</td>
<td>6</td>
<td>N/A</td>
<td>20</td>
<td>660</td>
</tr>
<tr>
<td>XC5VLX200T</td>
<td>160 x 108</td>
<td>34,560</td>
<td>2,280</td>
<td>128</td>
<td>424</td>
<td>212</td>
<td>7,032</td>
<td>6</td>
<td>N/A</td>
<td>20</td>
<td>660</td>
</tr>
<tr>
<td>XC5VLX250T</td>
<td>240 x 108</td>
<td>51,840</td>
<td>3,420</td>
<td>192</td>
<td>648</td>
<td>324</td>
<td>11,664</td>
<td>6</td>
<td>N/A</td>
<td>27</td>
<td>960</td>
</tr>
<tr>
<td>XC5VEX30T</td>
<td>80 x 34</td>
<td>5,440</td>
<td>520</td>
<td>150</td>
<td>166</td>
<td>94</td>
<td>3,024</td>
<td>2</td>
<td>N/A</td>
<td>12</td>
<td>360</td>
</tr>
<tr>
<td>XC5VEX60T</td>
<td>120 x 34</td>
<td>8,160</td>
<td>780</td>
<td>268</td>
<td>264</td>
<td>132</td>
<td>4,752</td>
<td>6</td>
<td>N/A</td>
<td>15</td>
<td>480</td>
</tr>
<tr>
<td>XC5VEX90T</td>
<td>150 x 46</td>
<td>14,720</td>
<td>1,520</td>
<td>640</td>
<td>488</td>
<td>244</td>
<td>8,784</td>
<td>6</td>
<td>N/A</td>
<td>16</td>
<td>960</td>
</tr>
<tr>
<td>XC5VSX240T</td>
<td>240 x 78</td>
<td>37,440</td>
<td>4,200</td>
<td>1,058</td>
<td>1,032</td>
<td>516</td>
<td>18,576</td>
<td>6</td>
<td>N/A</td>
<td>27</td>
<td>960</td>
</tr>
<tr>
<td>XC5VEX150T</td>
<td>200 x 58</td>
<td>23,200</td>
<td>1,500</td>
<td>80</td>
<td>456</td>
<td>228</td>
<td>8,208</td>
<td>6</td>
<td>N/A</td>
<td>40</td>
<td>260</td>
</tr>
<tr>
<td>XC5VSX240T</td>
<td>240 x 78</td>
<td>37,440</td>
<td>4,200</td>
<td>963</td>
<td>648</td>
<td>324</td>
<td>11,664</td>
<td>6</td>
<td>N/A</td>
<td>48</td>
<td>200</td>
</tr>
<tr>
<td>XC5VFX100T</td>
<td>80 x 38</td>
<td>5,120</td>
<td>380</td>
<td>64</td>
<td>136</td>
<td>68</td>
<td>2,448</td>
<td>2</td>
<td>N/A</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>XC5VFX70T</td>
<td>150 x 88</td>
<td>11,200</td>
<td>820</td>
<td>128</td>
<td>296</td>
<td>148</td>
<td>5,328</td>
<td>6</td>
<td>N/A</td>
<td>16</td>
<td>192</td>
</tr>
<tr>
<td>XC5VFX100T</td>
<td>150 x 100</td>
<td>14,000</td>
<td>1,240</td>
<td>256</td>
<td>456</td>
<td>228</td>
<td>8,060</td>
<td>6</td>
<td>N/A</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>XC5VFX130T</td>
<td>200 x 60</td>
<td>20,480</td>
<td>1,580</td>
<td>320</td>
<td>596</td>
<td>298</td>
<td>10,728</td>
<td>6</td>
<td>N/A</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>XC5VFX200T</td>
<td>240 x 68</td>
<td>30,720</td>
<td>2,280</td>
<td>384</td>
<td>912</td>
<td>456</td>
<td>16,340</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>24</td>
</tr>
</tbody>
</table>

To be continued ...

Throughout the semester, we will look at different Virtex-5 features in-depth.

Switch fabric
Block RAM
DSP48 (ALUs)
Clocking
I/O
Serial I/O + PCI