Outline

- **Field Programmable Gate Arrays**
  - Historical perspective
- **Programming Technologies**
- **Architectures**
  - PALs, PLDs, and CPLDs
  - FPGAs
    - Programmable logic
    - Interconnect network
    - I/O buffers
    - Specialized cores
- **Programming Interfaces**
History

- **Programmable Logic Arrays ~ 1970**
  - Can implement any set of sum-of-products logic equations
  - Incorporated in VLSI devices

- **Programmable Logic Devices ~ 1980**
  - MMI Programmable Array Logic (PAL)
    - 16L8 – combinational logic only
    - 16R8 – sequential logic only
  - AMD 22V10 and Lattice 16V8
  - Complex PLDs – arrays of PLDs with routing network

- **Field Programmable Gate Arrays ~ 1985**
  - Xilinx Logic Cell Array (LCA)

- **CPLD & FPGA architectures became similar ~ 2000**
Programming Technologies

- PLAs were mask programmable
- PALs used fuses for programming
- Early PLDs & CPLDs used floating gate technology
  - Erasable Programmable Read Only Memory (EPROM)
    - Ultra-violet erasable (UVEPROM)
    - Electrically erasable (EEPROM)
    - Flash memory came later and was used for CPLDs
- FPGAs used RAM for programming
- Later trends
  - Fuses were replaced with anti-fuses
    - Better reliability
  - Large CPLDs went to RAM-based programming
Programming Technologies

RAM
- Volatile – must configure after power-up
- In-System Re-programmable (ISR)
- Run-Time Reconfiguration (RTR)
  - dynamic reconfiguration while system is operating

Floating gate technologies
- Non-volatile but re-usable
  - UV EPROM, EEPROM, and flash memory
- In-System Programmable (ISP)
  - EEPROM and flash memory
- In-System Re-programmable (ISR)
  - Flash memory

Fuse/anti-fuse
- Non-volatile but not re-usable
- One Time Programmable (OTP)
PALs

16L8 – combinational logic

- 10 to 16 inputs, each with true and complement signal
- 2 to 8 outputs, each with
  - 7 product terms can AND any of up to 16 inputs or their complements
  - Tri-state control product term for inverting output buffer
- When output in tri-state, I/O pin can be used as input
  - High impedance output with no signal driven
PALs

16R8 – sequential logic

- 8 inputs, each with true & complement
- 8 outputs, each with
  - D flip-flop
    - With feedback for FSMs
  - 8 product terms that can AND any of:
    - 8 inputs or their complements
    - 8 feedbacks or their complements from D flip-flops

- One clock for all FFs
- One tri-state control for all outputs

C. Stroud 8/06
PLDs

22V10 replaced all PALs

- Combinational and/or sequential logic
  - Macrocell program bits C0, C1
- Up to 22 inputs w/complement
- Up to 10 outputs, each with
  - Macrocell
  - 8-16 product terms
  - Tri-state control product term

- Global
  - preset & clear PTs
  - clock
CPLDs

- An array of PLDs
  - Global routing resources for connections
    - PLDs to other PLDs
    - PLDs to/from I/O pins
- Example: Cypress 39K
  - Each Logic Block (LB) similar to a 22V10
  - Each cluster of 8 LBs has two 8K RAMs & one 4K dual-port RAM/FIFO
    - Programmable Interconnect Modules (PIMs) provide interconnections
  - Array of up to 24 clusters with global routing

C. Stroud 8/06
Field Programmable Gate Arrays

- Configuration Memory
- Programmable Logic Blocks (PLBs)
- Programmable Input/Output Cells
- Programmable Interconnect

Typical Complexity = 5 million – 1 billion transistors
Basic FPGA Operation

- Writing configuration memory → defines system function
  - Input/Output Cells
  - Logic in PLBs
  - Connections between PLBs & I/O cells

- Changing configuration memory data → changes system function
  - Can change at anytime
  - Even while system function is in operation
FPGAs

- Configuration memory (42K bits to 80M bits)
- Array of Programmable Logic Blocks (PLBs)
  - 250 to over 25,000 PLBs per FPGA
- Programmable routing network
  - Wire segments (45 to over 400 per PLB)
  - Programmable switches (130 to over 3,600 per PLB)
- Programmable I/O cells around periphery
  - Bi-direction buffer w/ flip-flops/latches (60 to over 1,200 per FPGA)
- Recent trend is to incorporate specialized cores
  - RAMs – single-port, dual-port, FIFOs
    - 128 bits to over 36K bits per RAM
    - 4 to over 575 per FPGA
  - DSPs – 18x18-bit multiplier, 48-bit accumulator, etc.
    - up to 512 per FPGA
  - Microprocessors and/or microcontrollers (up to 2 per FPGA)
## Ranges of Resources

<table>
<thead>
<tr>
<th>FPGA Resource</th>
<th>Small FPGA</th>
<th>Large FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logic</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLBs per FPGA</td>
<td>256</td>
<td>25,920</td>
</tr>
<tr>
<td>LUTs and flip-flops per PLB</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td><strong>Routing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wire segments per PLB</td>
<td>45</td>
<td>406</td>
</tr>
<tr>
<td>PIPs per PLB</td>
<td>139</td>
<td>3,462</td>
</tr>
<tr>
<td><strong>Specialized Cores</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bits per memory core</td>
<td>128</td>
<td>36,864</td>
</tr>
<tr>
<td>Memory cores per FPGA</td>
<td>16</td>
<td>576</td>
</tr>
<tr>
<td>DSP cores</td>
<td>0</td>
<td>512</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input/output cells</td>
<td>62</td>
<td>1,200</td>
</tr>
<tr>
<td>Configuration memory bits</td>
<td>42,104</td>
<td>79,704,832</td>
</tr>
</tbody>
</table>
Basic PLB Architecture

- Look-up Table (LUT) implements truth table
- Memory elements:
  - Flip-flop/latch
  - Some FPGAs - LUTs can also implement small RAMs
- Carry & control logic implements fast adders/subtractors
A Simple PLB

- Two 3-input LUTs
  - Can implement any 4-input combinational logic function

- 1 flip-flop
  - Programmable:
    - Active levels
    - Clock edge
    - Set/reset

- 22 configuration memory bits
  - 8 per LUT
    - C0-7
    - S0-7
  - 6 controls
    - CB0-7

C. Stroud 8/06
Combinational Logic Functions

- Gates are combined to create complex circuits
- Multiplexer example
  - If $S = 0$, $Z = A$
  - If $S = 1$, $Z = B$
  - Very common digital circuit
- Heavily used in FPGAs
  - $S$ input controlled by configuration memory bit
  - We’ll see it again

Truth table

<table>
<thead>
<tr>
<th>S</th>
<th>A</th>
<th>B</th>
<th>Z</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>1</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>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Memory Elements

- **Static memory based on cross-coupled logic gates**
  - Storage obtained by feedback and amplification
  - Flip-flops are static

- **Dynamic memory based on capacitance**
  - No feedback or amplification
  - Voltage decays in time
Flip-Flops

- Use a collection of gates to enable a sample control signal called Clock
- Basic structure is cross-coupled NOR gates
- AND gates provide for sampling control signal - Clock (CK)
- This is a latch – flip-flops are more complicated

<table>
<thead>
<tr>
<th>AND</th>
<th>A B</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOR</th>
<th>A B</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clock</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reset</th>
<th>Set</th>
<th>1 Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CK</th>
<th>R S</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>X X</td>
<td>Q</td>
</tr>
</tbody>
</table>
Look-up Tables

- Recall multiplexer example
- Configuration memory holds outputs for truth table
- Internal signals connect to control signals of multiplexers to select value of truth table for any given input value
Look-up Table Based RAMs

- Normal LUT mode performs read operations
- Address decoder with write enable generates clock signals to latches for write operations
- Small RAMs but can be combined for larger RAMs
Xilinx FPGAs

- **Virtex and Spartan II**
  - Array of 96 to 6,144 PLBs
    - 4 LUTs/RAMs (4-input)
    - 4 FF/latches
  - 4 to 32 4K-bit dual-port RAMs

- **Virtex II, Virtex II Pro, and Spartan 3**
  - Array of 192 to 11,204 PLBs
    - 8 LUTs/RAMs (4-input)
    - 8 FF/latches
  - 4 to 444 18K-bit dual-port RAMs
  - 4 to 444 18×18-bit multipliers
  - 0 to 2 PowerPC processor cores

- **Virtex 4**
  - Array of 1,536 to 22,272 PLBs
    - 4 LUTs/RAMs (4-input)
    - 4 LUTs (4-input)
    - 8 FF/latches
  - 48 to 552 18K-bit dual-port RAMs
    - Also operate as FIFOs
  - 32 to 512 DSP cores 48-bits
  - 0 to 2 PowerPC processor cores
Spartan 3 (XC3S200)

- **PLBs = 24 rows x 20 columns =** 480
  - 4 slices/PLB
    - 2 L slices
      - L = logic
    - 2 M slices
      - M = memory
- **RAMs = 12 18Kbit dual port RAMs**
- **Multipliers = 12 18x18-bit signed**
Spartan 3 PLB Slices

Slice L

Slice M
Specialized Cores

Virtex and Spartan II
Virtex II and Spartan 3

4K-bit RAMs
18K-bit RAMs and 18×18-bit multipliers

RAMs/multipliers
Specialized Cores

C. Stroud 8/06 Overview of FPGAs
Programmable RAMs

- 18 Kbit dual-port RAM
- Each port independently configurable as
  - 512 words x 36 bits
    - 32 data bits + 4 parity bits
  - 1K words x 18 bits
    - 16 data bits + 2 parity bits
  - 2K words x 9 bits
    - 8 data bits + 1 parity bit
  - 4K words x 4 bits (no parity)
  - 8K words x 2 bits (no parity)
  - 16K words x 1 bit (no parity)
- Each port has independently programmable
  - clock edge
  - active levels for write enable, RAM enable, reset
Input/Output Cells

- **Bi-directional buffers**
  - Programmable for input or output
  - Tri-state control for bi-directional operation
  - Flip-flops/latches for improved timing
    - ✔ Set-up and hold times
    - ✔ Clock-to-output delay
  - Pull-up/down resistors

- **Routing resources**
  - Connections to core of array

- **Programmable I/O voltage & current levels**
Spartan 3 I/O Cell
Interconnect Network

- **Wire segments of varying length**
  - $xN = N$ PLBs in length
    - 1, 2, 4, and 6 are most common
  - $xH = \text{half the array in length}$
  - $xL = \text{length of full array}$

- **Programmable Interconnect Points (PIPs)**
  - Also known as Configurable Interconnect Points (CIPs)
  - Transmission gate connects to 2 wire segments
  - Controlled by configuration memory bit
    - $0 = \text{wires disconnected}$
    - $1 = \text{wires connected}$
PIPs

- **Break-point PIP**
  - Connect or isolate 2 wire segments

- **Cross-point PIP**
  - Turn corners

- **Multiplexer PIP**
  - Directional and buffered
  - Select 1-of-N inputs for output
    - Decoded MUX PIP – $N$ config bits select from $2^N$ inputs
    - Non-decoded MUX PIP – 1 config bit per input

- **Compound cross-point PIP**
  - Collection of 6 break-point PIPs
    - Can route to two isolated signal nets
Spartan 3 Routing Resources

PLB consists of 4 slices

- over 2,400 PIPs
- mostly MUX PIPs

- x6 wire segments
- x2 wire segments
- xH & xL wire segments

over 450 total wire segments in PLB
Configuration Interfaces

- **Master** – FPGA retrieves its own configuration from ROM after power-up
  - Serial or Parallel options

- **Slave** – FPGA configured by external source (i.e., a μP)
  - Serial or Parallel options
  - Used for dynamic reconfiguration
  - Can also read configuration memory contents

- **Boundary Scan Interface**
  - 4-wire IEEE standard serial interface for testing
  - Write and read access to configuration memory
    - Not available in all FPGAs
    - Used for dynamic partial reconfiguration
  - Interfaces to FPGA core
    - Not available in all FPGAs
    - Connections between Boundary Scan Interface and internal routing network and PLBs (Xilinx provides 2 of these ports)

- **Other configuration interfaces in some FPGAs**
# Xilinx Configuration Interface Pins

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Driver Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dedicated Pins</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCLK</td>
<td>Input/Output</td>
<td>Active</td>
<td>Configuration clock. Output in Master mode.</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>Input</td>
<td></td>
<td>Asynchronous reset to configuration logic.</td>
</tr>
<tr>
<td>DONE</td>
<td>Input/Output</td>
<td>Active/Open-Drain</td>
<td>Configuration status and start-up control.</td>
</tr>
<tr>
<td>M2, M1, M0</td>
<td>Input</td>
<td></td>
<td>Configuration mode selection.</td>
</tr>
<tr>
<td>TMS</td>
<td>Input</td>
<td></td>
<td>Boundary-scan tap controller.</td>
</tr>
<tr>
<td>TCK</td>
<td>Input</td>
<td></td>
<td>Boundary-scan clock.</td>
</tr>
<tr>
<td>TDI</td>
<td>Input</td>
<td></td>
<td>Boundary-scan data input.</td>
</tr>
<tr>
<td>TDO</td>
<td>Output</td>
<td>Active</td>
<td>Boundary-scan data output.</td>
</tr>
<tr>
<td><strong>Dual Function Pins</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIN (D0)</td>
<td>Input/Output</td>
<td>Active Bidirectional</td>
<td>Serial configuration data input.</td>
</tr>
<tr>
<td>D[0:7]</td>
<td>Input/Output</td>
<td>Active Bidirectional</td>
<td>Slave Parallel configuration data input, readback data output.</td>
</tr>
<tr>
<td>CS</td>
<td>Input</td>
<td></td>
<td>Chip Select (Slave Parallel only).</td>
</tr>
<tr>
<td>WRITE</td>
<td>Input</td>
<td></td>
<td>Active Low write select, read select (Slave Parallel only).</td>
</tr>
<tr>
<td>BUSY/ DOUT</td>
<td>Output</td>
<td>Open-Drain/Active</td>
<td>Busy/Ready status for Slave Parallel (open-drain). Serial configuration data output for serial daisy-chains (active).</td>
</tr>
<tr>
<td>INIT</td>
<td>Input/Output</td>
<td>Open-Drain</td>
<td>Delay configuration, indicate configuration clearing or error.</td>
</tr>
</tbody>
</table>
Master mode

- Configuration sequence during power-up of device
  - Typically from
  - Serial EPROM
  - Master Serial
  - Parallel EPROM
  - Master Parallel
    - 8-bit
    - 32-bit
Daisy Chain Configuration
Slave Configuration

External source can be microprocessor with configuration data stored in memory or generated by processor.
Boundary Scan Interface

User registers can include:
- a) configuration memory
- b) access to FPGA core

TAP controller state diagram
Typical Boundary Scan Cell

<table>
<thead>
<tr>
<th>Operation Mode</th>
<th>Data Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Out &lt;=&gt; In</td>
</tr>
<tr>
<td>Scan</td>
<td>Capture &lt;=&gt; Scan In</td>
</tr>
<tr>
<td>Capture</td>
<td>Capture &lt;=&gt; In</td>
</tr>
<tr>
<td>Update</td>
<td>Update &lt;=&gt; Capture</td>
</tr>
</tbody>
</table>
Typical Boundary Scan Cell

- Primarily designed to test external surface mount connections on printed circuit board
  - **EXTEST instruction**
    - Capture logic values at BSCs
      - Output responses from previous test patterns
    - Shift in test patterns through BSCs
      - As we shift out captured logic values
    - Update test patterns into BSCs
      - New test patterns are applied
  - **Optional: can be used to test inside chip**
    - **INTEST instruction**
      - Same basic procedure as EXTEST but test patterns and responses are to/from device (FPGA) core
Boundary Scan Access to FPGA

- **Configuration memory**
  - Write and read usually separate instructions
  - Other registers for partial reconfiguration
    - Command, frame address, etc.

- **Core access**
  - Ports from TAP to PLB and routing resources
    - Good interface for accessing FPGA core
    - Not supported by all FPGA manufacturers
  - Includes TCK, TDI, TDO access to core with active enable when core access instruction is in IR
FPGA Configuration Memories

- **PLB addressable**
  - Good for partial reconfiguration
  - X-Y coordinates of PLB location to be written
    - Requires tag to identify which resources will be configured

- **Frame addressable**
  - Vertical or horizontal frame
  - Access to all PLBs in frame
    - Only portion of logic and routing resources accessible in a given frame
    - Many frames to configure PLBs
      - Major address for column, minor address for frame