CSE140L: Components and Design Techniques for Digital Systems Lab

Memory and Programmable Logic Devices

Tajana Simunic Rosing

Source: Vahid, Katz, Culler
Updates & Outline

• Lab #3
  – Demo due Sunday 10pm, Report due Monday 12pm

• Today: Memory and PLDs
  – Different technologies

• Next: Register Transfer Level Design: Verilog emphasis
Evolution of Programmable Technologies

- Discrete devices: relays, transistors (1940s-50s)
- Discrete logic gates (1950s-60s)
- Integrated circuits (1960s-70s)
  - e.g. TTL packages: Data Book for 100’s of different parts
  - Map your circuit to the Data Book parts
- Gate Arrays (IBM 1970s)
  - “Custom” integrated circuit chips
  - Transistors are already on the chip
  - Place and route software puts the chip together automatically
  - + Large circuits on a chip
  - + Automatic design tools (no tedious custom layout)
  - - Only good if you want 1000’s of parts
Gate Array Technology (IBM - 1970s)

- Simple logic gates
  - Use transistors to implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special blocks at periphery for external connections
- Add wires to make connections
  - Done when chip is fabed
    - “mask-programmable”
    - Construct any circuit
Programmable Logic Technologies

- **Fuse and anti-fuse**
  - Fuse makes or breaks link between two wires
  - Typical connections are 50-300 ohm
  - One-time programmable (testing before programming?)
  - Very high density
- **EPROM and EEPROM**
  - High power consumption
  - Typical connections are 2K-4K ohm
  - Fairly high density
- **RAM-based**
  - Memory bit controls a switch that connects/disconnects two wires
  - Typical connections are .5K-1K ohm
  - Can be programmed and re-programmed *in the circuit*
  - Low density
Comparing RAM Memory

- Register file
  - Fastest
  - But biggest size

- SRAM
  - Fast
  - More compact than register file

- DRAM
  - Slowest
    - And refreshing takes time
  - But very compact
  - Different technology for large caps.

![Size comparison for the same number of bits (not to scale)](image-url)
**ROM Types**

- **Mask-programmed ROM**
  - Programmed at manufacturing time
- **Fuse-Based Programmable ROM**
  - Programming blows fuses
  - **One-Time Programmable ROM**
- **EPROM**
  - Erase with ultraviolet light
- **EEPROM**
  - Erasing one word at a time *electronically*
- **Flash**
  - Erase large blocks of words *simultaneously*
Memory in Verilog

• Modeled as an array of registers

\[
\text{reg}[15:0] \text{ memword}[0:1023]; \quad // \quad 1,024 \text{ registers of 16 bits each}
\]

//Example Memory Block Specification
// Uses enable to control both write and read
//-----------------------------
//Read and write operations of memory.
//Memory size is 64 words of 4 bits each.
module memory (Enable,ReadWrite,Address,DataIn,DataOut);
    input Enable,ReadWrite;
    input [3:0] DataIn;
    input [5:0] Address;
    output [3:0] DataOut;
    reg [3:0] DataOut;
    reg [3:0] Mem [0:63]; \quad //64 x 4 memory
always @ (Enable or ReadWrite)
    if (Enable)
        if (ReadWrite)
            DataOut = Mem[Address]; \quad //Read
        else
            Mem[Address] = DataIn; \quad //Write
    else DataOut = 4'bz; \quad //High impedance state
endmodule

Source: John Wawrzynek
Programmable Logic

• Program a connection
  – Connect two wires
  – Set a bit to 0 or 1

• Regular structures for two-level logic (1960s-70s)
  – All rely on two-level logic minimization
  – PROM connections - permanent
  – EPROM connections - erase with UV light
  – EEPROM connections - erase electrically
  – PROMs
    • Program connections in the _____________ plane
  – PLAs
    • Program the connections in the _____________ plane
  – PALs
    • Program the connections in the _____________ plane
PAL Logic Building Block

- Programmable AND gates
- Fixed OR/NOR gate
- Flipflop/Registered Output
- Feedback to Array
- Tri-state Output
**XOR PALs**

- Useful for comparator logic, arithmetic sums, etc.
  - Use of XOR gates can dramatically reduce the number of AND plane inputs needed to realize certain functions.
- And/Or/XOR Logic
- Feedback
- Registered Outputs
- Tri-State Outputs
PAL: Synchronous vs. Asynchronous Outputs
Programmable Logic Devices (PLD)

- PLDs combine PLA/PAL with memory and other advanced structures

- Types:
  - Antifuse PLDs
  - EPLD & EEPLD
  - FPGAs with RAMs
  - FPGA with processing
    - Digital Signal Processing
    - General purpose CPU

<table>
<thead>
<tr>
<th>Name</th>
<th>Re-programmable</th>
<th>Volatile</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuse</td>
<td>no</td>
<td>no</td>
<td>Bipolar</td>
</tr>
<tr>
<td>EPROM</td>
<td>yes</td>
<td>no</td>
<td>UVCMOS</td>
</tr>
<tr>
<td></td>
<td>out of circuit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EEPROM</td>
<td>yes</td>
<td>no</td>
<td>EECMOS</td>
</tr>
<tr>
<td></td>
<td>in circuit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td>yes</td>
<td>yes</td>
<td>CMOS</td>
</tr>
<tr>
<td></td>
<td>in circuit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Antifuse</td>
<td>no</td>
<td>no</td>
<td>CMOS+</td>
</tr>
</tbody>
</table>
Making Large Programmable Logic Circuits

• Alternative 1: “CPLD”
  – Put a lot of PLDS on a chip
  – Add wires between them whose connections can be programmed
  – Use fuse/EEPROM technology

• Alternative 2: “FPGA”
  – Emulate gate array technology
  – Hence Field Programmable Gate Array
  – You need:
    • A way to implement logic gates
    • A way to connect them together
Field-Programmable Gate Arrays

- PALs, PLAs = 10s – 100s Gate Equivalents
- Field Programmable Gate Arrays = FPGAs
  - Altera MAX Family
  - Actel Programmable Gate Array
  - Xilinx Logical Cell Array
- 1000s - 100000(s) of Gate Equivalents!
**Field-Programmable Gate Arrays**

- **Logic blocks**
  - To implement combinational and sequential logic
- **Interconnect**
  - Wires to connect inputs and outputs to logic blocks
- **I/O blocks**
  - Special logic blocks at periphery of device for external connections
- **Key questions:**
  - How to make logic blocks programmable?
  - How to connect the wires?
  - *After the chip has been manufactured*
Tradeoffs in FPGAs

• Logic block - how are functions implemented: fixed functions (manipulate inputs) or programmable?
  – Support complex functions, need fewer blocks, but they are bigger so less of them on chip
  – Support simple functions, need more blocks, but they are smaller so more of them on chip

• Interconnect
  – How are logic blocks arranged?
  – How many wires will be needed between them?
  – Are wires evenly distributed across chip?
  – Programmability slows wires down – are some wires specialized for long distances?
  – How many inputs/outputs must be routed to/from each logic block?
  – What utilization are we willing to accept? 50%? 20%? 90%?
Antifuse PLDs

• Actel’s Axcelerator Family

• Antifuse:
  – open when not programmed
  – Low resistance when programmed
• Altera’s MAX 7k Block Diagram
• Altera’s MAX 7k Logic Block
SRAM based PLD

- Altera’s Flex 10k Block Diagram
SRAM based PLD

- Altera’s Flex 10k Logic Array Block (LAB)
SRAM based PLD

- Altera’s Flex 10k Logic Element (LE)
Altera’s Stratix II: Block Diagram
FPGA with DSP

- Altera’s Stratix II:
  - DSP Detail
FPGA with General Purpose CPU & Analog

- Actel’s Fusion Family Diagram
  - FPGA with ARM 7 CPU and Analog Components

![Diagram of Actel’s Fusion Family Diagram]

- Flash Memory
- Optional ARM or 8051 Processor
- User Applications
  - 27 FPGAEvolution
- Fusion Applets
- Fusion Smart Backbone
- Analog Smart Peripheral 1
- Analog Smart Peripheral 2
- Analog Smart Peripheral n
- Smart Peripherals in FPGA Fabric (e.g. logic, PLL, FIFO)