FROM LAST TIME...

Computer Systems (Combinational Logic)

- Sinks and Sources
- Fan In and Fan Out
- Decoupling Capacitors
UNIT 3:
COMBINATIONAL LOGIC

PART E:
PROGRAMMABLE DEVICES
A NEAT BREADBOARD CAN BE A THING OF BEAUTY...
... BUT SPAGHETTI WIRING IS A PROBLEM
PROGRAMMABLE LOGIC DEVICES (PLDS) CAN FIX THIS MESS

Image: http://dangerousprototypes.com/2012/05/04/diy-ir-toy-and-cpld-breakout-boards/
THEORETICAL PLD

- Create an array of gates, or logic blocks, and only connect up those that are needed.
- Assume unused inputs do not influence output...
THEORETICAL PLD

Location of connection array varies by device!
GENERAL PLD STRUCTURE

- Inverters
- Product Terms
- Sum-of-Product
- Input Matrix
- AND Matrix
- OR Matrix
- Sequential Logic (optional)
- Inverters
- Output Matrix
PROGRAMMABLE LOGIC DEVICE (PLD)

Customization is accomplished by appropriate connections (or disconnections) between components.

One-time programmability:
- Disconnections caused by intentionally blowing fuses (bipolar)
- Connections created using anti-fuse technology (CMOS)

Repeated programmability (erasable):
- Erasable Programmable ROM (EPROM) – Erased with UV light
- Electrically Erasable PROM (EEPROM) – physically larger than EPROM
- Flash EPROM – benefits of EEPROM, but smaller size
- Static random-access memory (SRAM) – volatile (contents lost at power-down)
ERASABLE PROGRAMMABLE READ-ONLY MEMORY (EPROM)

PROGRAMMABLE LOGIC DEVICE (PLD)

PLDs can be categorized by their complexity and architecture:

- SPLDs (Simple Programmable Logic Devices)
  - ROM (Read-Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)
  - GAL (Generic Array Logic)

- Complex Programmable Logic Device (CPLD)

- Field Programmable Gate Array (FPGA)
SIMPLE PLDs

Differ in which gate arrays are configurable

<table>
<thead>
<tr>
<th>Type</th>
<th>AND Array</th>
<th>OR Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>Fixed</td>
<td>Programmed</td>
</tr>
<tr>
<td>PLA</td>
<td>Programmed</td>
<td>Programmed</td>
</tr>
<tr>
<td>PAL/GAL</td>
<td>Programmed</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
READ-ONLY MEMORY (ROM) AS A LOGIC DEVICE

A $2N \times b$ ROM is a combinational circuit with $N$ inputs and $b$ outputs.

Two ways of interpreting the ROM:

- A place to store information, with each address holding a specific Boolean value.
- A combinational circuit generating a **sum-of-products** function with no minimization.

Does not require specialized hardware or software, but ROM is slow, expensive, and power inefficient. Cannot be used for sequential logic.
**PROGRAMMABLE ARRAY LOGIC (PAL)**

Field programmable devices implementing *sum-of-product* logic functions

- Programmable AND plane
- Fixed OR plane

\[
Z_0 = X_1 \cdot X_3 + X_0 \cdot X_2
\]

\[
Z_1 = X_1 \cdot X_2 \cdot X_3 + X_0 \cdot X_1 \cdot X_2 \cdot X_3 + X_0 \cdot X_3
\]

PAL is a trade name of Lattice Semiconductor.
PROGRAMMABLE ARRAY LOGIC (PAL)

PROGRAMMABLE ARRAY LOGIC (PAL)
PROGRAMMABLE LOGIC ARRAY (PLA)

Add logic efficiency at the expense of internal complexity

- Both AND plane and OR plane are programmable
- More pins and more functions per chip.

\[ Z_0 = X_1 \cdot X_3 + X_0 \cdot X_2 \]

\[ Z_1 = \overline{X_1} \cdot X_2 \cdot X_3 + \overline{X_0} \cdot X_1 \cdot \overline{X_2} \cdot X_3 + X_0 \cdot X_3 \]

\[ Z_2 = X_1 \cdot X_3 + \overline{X_1} \cdot X_2 \cdot X_3 \]
COMPLEX PLDs (CPLD)

Instead of hundreds of gates (as in SPLDs), complex PLDs contain thousands to tens of thousands of logic gates.

Combines programmable AND/OR arrays with "macrocells" that can perform combinatorial or sequential logic

- Many gate arrays and macrocells available inside a CPLD, with selectable functionality
- Makes efficient use of the chip for lots of small equations
- Retains certain PLA functions, but with less complexity
- Logic usually formed using \textit{sum-of-products}

Non-volatile memory; can function on system start-up
Less expensive than FPGA, with more predictable propagation delays
COMPLEX PLDs (CPLD)

Image: https://commons.wikimedia.org/wiki/File:Altera_MAX_7128_2500_gate_CPLD.jpg
COMPLEX PLDs (CPLD)

LAB = Logic Array Block (uses PALs)
PIA = Programmable Interconnect Array

Altera MAX 3000A Architecture
COMPLEX PLDs (CPLD)

CPLDs are ideal for high-bandwidth applications where propagation delay must be minimized, or where non-volatile memory is needed. Such applications include:

- Fast combinational logic
- Counters
- Decoders
- "Glue" logic
- Bus protocol translation
- I/O decoding
- Sequencing of device power-up
PROGRAMMABLE LOGIC DEVICE (PLD)

ASIC (Application Specific IC)
- Highly integrated logic device that uses programmable internal interconnections to allow arbitrary (application specific) logic
- Can be made for high volume use
- Expensive! Non-recurring engineering (NRE) charge needs to be amortized

FPGA (Field Programmable Gate Array)
- Uses lookup tables (LUT) instead of logic gates
- Used in National Instruments cRIO
- Xilinx and Altera control over 80% of the FPGA market
FIELD PROGRAMMABLE GATE ARRAYS (FPGAs)

Image from: FPGAs!? Now What? By Dave Vendenbout, XESS Corporation
Available at: http://www.xess.com/static/media/appnotes/FpgasNowWhatBook.pdf
FPGA ARCHITECTURE

I/O = Input/Output Block
SB = Switching Block
CLB = Control Logic Block
WHAT DOES A CONTROL LOGIC BLOCK DO?

- The control logic block (CLB) architecture varies by device family.
- Each CLB takes between 3 and 10 binary inputs, and generates one or two outputs in accordance with a Boolean logic function specified by the user.
- Rather than rendering logic through physical gates (as is done with CPLDs), a FPGA's control logic is implemented using small look-up tables (built as RAM).
- In most FPGA devices, combinatorial outputs can be stored for later use.
FIELD PROGRAMMABLE GATE ARRAYS (FPGAs)

FPGAs are ideal for implementing complex code that requires memory and sequential logic. Applications include:

- Data signal processing
- Parallel process control
- Data encryption
- Pattern recognition
- Software-defined radio
- Medical imaging
# FPGAs COMPARED TO CPLDs

<table>
<thead>
<tr>
<th>FPGAs</th>
<th>CPLDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on look-up table (LUT), resulting in higher density.</td>
<td>Based on programmable AND array and fixed OR array.</td>
</tr>
<tr>
<td>More expensive (&gt; $15)</td>
<td>Cheaper (&gt; $10)</td>
</tr>
<tr>
<td>Volatile SRAM technology.</td>
<td>Non-volatile EEPROM technology.</td>
</tr>
<tr>
<td>Up to 250,000 logic elements, suitable for more complex applications.</td>
<td>Up to 500 logic elements, often used for simpler logic applications.</td>
</tr>
<tr>
<td>Programmed using either schematic entry or text entry.</td>
<td></td>
</tr>
</tbody>
</table>
PLD PROGRAMMING TOOLS

CAD tools for designing digital systems should cover the following process phases:

- **Description (logic specification)**
- **Design (logic synthesis)** – includes various optimization steps to reduce cost and improve performance while generating a "netlist"
- **Simulation (logic verification)** – verify design with respect to its specification
PLD PROGRAMMING TOOLS

Approach 1: Develop schematic, compile, and program device.

Image: http://saaubi.people.wm.edu/TeachingWebPages/Physics351_Fall2009/Week0/Physics351_Fall2009_ad.html
Approach 2: Write code, compile, and program device.
PLD PROGRAMMING TOOLS

Hardware Description Languages (HDL)
- Similar to modern structured programming languages but supports computations that can occur in parallel.
- Works with both CPLDs and FPGAs

Open-standard HDLs
- VHDL (IEEE 1076)
- Verilog (IEEE 1364)

Proprietary HDLs
- AHDL (Altera HDL)
- ABEL (Advanced Boolean Expression Language, owned by Xilinx)
- CUPL
library IEEE;
use IEEE.std_logic_1164.all;

entity ANDGATE is
  port (
    I1 : in std_logic;
    I2 : in std_logic;
    O  : out std_logic);
end entity ANDGATE;

architecture RTL of ANDGATE is
begin
  O <= I1 and I2;
end architecture RTL;
VERILOG

- Syntax similar to C programming language
- Uses hierarchy of modules to implement design

// andgate.v
module andgate(A, B, Y);
  input A, B;
  output Y;
  assign Y = A & B;
endmodule
SYSTEM ON A CHIP

FPGA technology allows users to embed entire digital signal processing (DSP) and microprocessors (uP) onto a single chip. Faster, less expensive, and more reusable than application specific integrated circuits (ASICs)
COMING UP...

Computer Systems
  - Sequential logic
  - Finite state machines