ENGIN 112

Intro to Electrical and Computer Engineering

Lecture 38

Programmable Logic
Overview

- Programmable logic offers designers opportunity to customize chips
- Programmable logic devices have a fixed logic structure
- Programmable array logic contain AND-OR circuits
  - First introduced in early 1980’s
- Field programmable gate arrays (FPGAs) contain small blocks that implement truth tables
  - First introduced in 1985 (Xilinx Corporation)
- Software used to convert user designs to programming information
Design Implementation

- Chip creation is a long and difficult process
- Millions of dollars required to create custom silicon
  - Simulation, synthesis, fabrication (lots of jobs for engineers)

Fig. 8-1  Process of HDL Simulation and Synthesis
Programmable Logic Design

° “Generic” chip created and then customized by designer

° Programming information used
  • Like a ROM

° Analogy – sign making
  • Custom sign – more expensive, customized by manufacturer, difficult to change
  • Sign built by consumer from individual letters – less expensive, not quite as nice, easier to change (remix letters)
Programmable Array Logic

- Implements sum-of-products expressions
- Four external inputs (and complements)
- Feedback path from output $F_1$
- Product term connections made via switches

Fig. 7-16 PAL with Four Inputs, Four Outputs, and Three-Wide AND-OR Structure
Programmable Array Logic

• Consider implementing the following expression

\[ I_1 I_2 I_3 + I_2' I_3' I_4 + I_1 I_4 = F_1 \]

• Note that only functions of up to three product terms can be implemented

  - Larger functions need to be chained together via the feedback path

Fig. 7-16 PAL with Four Inputs, Four Outputs, and Three-Wide AND-OR Structure
Reconfigurable Hardware

- Each logic element operates on four one-bit inputs.
- Output is one data bit.
- Can perform any Boolean function of four inputs.

\[2^{2^4} = 64K\text{ functions!}\]
Field-Programmable Gate Array

- Each *logic element* outputs one data bit.
- Interconnect programmable between elements.
- Interconnect *tracks* grouped into channels.
FPGA Architecture Issues

- Need to explore architectural issues.
- How much functionality should go in a logic element?
- How many routing tracks per channel?
- Switch “population”? 
Translating a Design to an FPGA

C program

\[ C = A + B \]

- CAD to translate circuit from text description to physical implementation well understood.
- CAD to translate from C program to circuit not well understood.
- Very difficult for application designers to successfully write high-performance applications

Need for design automation!
Circuit Compilation

1. Technology Mapping

Assign a logical LUT to a physical location.

2. Placement

Select wire segments and switches for interconnection.

3. Routing
Two Bit Adder

Made of Full Adders

Logic synthesis tool reduces circuit to SOP form

\[ S = ABC_i + A'B'C_i + AB'C_i' + A'BC_i' \]

\[ C_o = ABC_i + A'BC_i + AB'C_i + ABC_i' \]
Dynamic Reconfiguration

• What if I want to exchange part of the design in the device with another piece?
• Need to create architectures and software to incrementally change designs.
• Effectively a “configuration cache”
Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)
Xilinx XC4000 Routing

Small boxes represent switches
Summary

- Programmable logic allows for designers to easily create custom designs
- Programmable array logic contains AND-OR structures to implement SOP equations
- FPGAs contain small memories and numerous wires for routing
- Designers create designs in Verilog
  - Design translated to the chip via software
- Hands on experience in ECE353, ECE354