ENG2410
Digital Design: Week #12
“Programmable Logic Technologies”
S. Areibi
School of Engineering
University of Guelph

Week #12 Topics
- The Von Neumann Architecture
- What is Programmable Logic?
- Classification of Programmable Logic
- Field Programmable Gate Arrays
- FPGA CAD
- Applications
- Summary

Resources
Chapter #10, Mano Sections
- 10.3 Programmable Implementations Tech

The Von Neumann Computer
- Principle
  In 1945, the mathematician Von Neumann (VN) demonstrated in study of computation that a computer could have a simple structure, capable of executing any kind of program, given a properly programmed control unit, without the need of hardware modification.

The Von Neumann Computer
- Structure
  - An arithmetic and logic unit (ALU) also called data path for program execution
  - A control unit (control path) featuring a program counter for controlling program execution
  - A memory for storing program and data.
  - The memory consists of the word with the same length

The Von Neumann Computer
- Coding
  A program is coded as a set of instructions to be sequentially executed
- Program execution
  - Instruction Fetch (IF): The next instruction to be executed is fetched from the memory
  - Decode (D): Instruction is decoded (operation?)
  - Read operand (R): Operands read from the memory
  - Execute (EX): Operation is executed on the ALU
  - Write result (W): Results written back to the memory
  - Instruction execution in Cycle (IF, D, R, EX, W)

What is the problem with this computing paradigm?
Bottlenecks in VN Architecture

- Memory: Bottleneck: Interpret instructions, manipulate data
- Central Processing Unit (CPU): Bottleneck: Transport instructions and data
- Bottleneck: Find and access instructions and data

The Von Neumann Computer

- Advantage:
  - Simplicity.
  - Flexibility: any well coded program can be executed
- Drawbacks:
  - Speed efficiency: Not efficient, due to the sequential program execution (temporal resource sharing).
  - Resource efficiency: Only one part of the hardware resources is required for the execution of an instruction. The rest remains idle.
  - Memory access: Memories are about 5 times slower than the processor.
- How to compensate for deficiencies?

Improving Performance of VN (GPPs)

1. Technology Scaling
   - Improve performance (increase clock frequency!)
2. Improving Instruction Set of Processor
3. Application Specific Processors (DSP)
4. Use of Hierarchical Memory System
   - Cache can enhance speed
5. Multiplicity of Functional Units (HW)
   - Adders/Multiples/Dividers (CIC-6600)
6. Pipelining within CPU (HW)
   - A four stage pipeline stage (IF/ID/EX/DB)
7. Overlap CPU & I/O Operations (HW)
   - DMA (Direct Memory Access) can be used to enhance performance
8. Time Sharing (SW)
   - Multi-tasking assigns fixed or variable time slices to multiple programs
9. Parallelism & Multithreading (SW/HW)
   - Compilers/Multi-core systems

Spatial vs. Temporal Computing

Von Neumann Architecture

\[ \begin{align*}
  y & \leftarrow x \\
  z & \leftarrow A \times x \\
  y & \leftarrow 2 \times z \\
  y & \leftarrow 12 \times y \\
  (Ax + B)x + C & \\
\end{align*} \]

Temporal (Processor)

Spatial (ASIC or FPGA)

\[ \begin{align*}
  Ax^2 + Bx + C & \\
\end{align*} \]

ENGG3050

Temporal vs. Spatial Based Computing

Temporal-based execution (software)

- Add
- Sub
- Cmp

Spatial-based execution (reconfigurable computing)

- Add
- Cmp

Ability to extract parallelism (or concurrency) from algorithm descriptions is the key to acceleration using reconfigurable computing

ENGG3380

ENGG4540
Gerald Estrin Fix-Plus Machine

- Attempts to have a flexible hardware structure that can be dynamically modified at run-time to compute a desired function are almost as old as the development of other computing paradigms.
- In 1959, Gerald Estrin, at UCLA, introduced the concept of reconfigurable computing by introducing the Fix-Plus Machine.

Substantial efforts on Reconfiguration

Programmable Logic I

- We learnt in the first part of this course that any combinational logic circuit can be implemented with the sum of min-terms (SOP).
- If we can control the number of AND gates to be used and also control the inputs to the OR gate then we can design a programmable logic circuit.
- Remember when we used a decoder to implement any Boolean function! That was some type of implementing programmable logic!

I. Programmable AND Array

- If we remove fuses $F_A$ and $F_B$ this will disconnect the complementary version of input ‘a’ and the true version of input ‘b’. 
- This leaves the device to perform its new function $y = a \times b'$.
- The process of removing fuses is typically referred to as programming the device (blowing, burning the device).
- Devices based on fusible-link technology are said to be One-time Programmable (OTP).
- Remember: FPGAs are not based on this type of technology.

Decoders: Implementing Logic

- Example: Implement the following boolean functions
  
  $S(A_2, A_1, A_0) = \text{SUM}(m(1,2,4,7))$

  1. Since there are three inputs, we need a 3-to-8 line decoder.
  2. The decoder generates the eight minterms for inputs $A_2A_1A_0$.
  3. An OR GATE forms the logical sum minterms required.

II. Programmable OR Array

Programmable Boolean Functions

Multiplexers can also be used to realize Boolean functions since they consist of an array of AND gates followed by an OR gate.
Classification of PLDs

Programmable Logic Array (PLA)

Complex PLDs (CPLDs)

III. SRAM FPGAs:

Memory units can be used to implement a Boolean function by storing the output of the truth table in the memory and accessing the values by using variables of the truth table as address lines.

Generic FPGA architecture:
There are two main versions of semiconductor RAM devices: Static RAM (SRAM) and Dynamic RAM (DRAM). SRAM based devices can be used to control NMOS transistors to be on/off. This can be very useful to control Multiplexers, Routing, e.t.c.

SRAM based Programmable Cell

An SRAM cell can drive the gate (G) terminal of an NMOS transistor.
If SRAM (M) = 1 then signals passes from S → D
An SRAM cell can be attached to the select line of a MUX to control it.

Look Up Table (LUT)

- The LUT is used to realize any boolean function.
- Assume the function to be realized is \( y = (a \& b) \mid \lnot c \)
- This could be achieved by loading the LUT with the appropriate output values

Basic Logic Element (BLE)

A Basic Logic Element consists of lookup table (LUT), a register that could act as flip flop or a latch, and a multiplexer, along with a few other elements.

Switch Matrix

- Connections between CLBs and IOBs are made using wiring segments in both horizontal and vertical channels lying between the various blocks.
- Four segments meet, on each there is 6 pass transistors.
**Xilinx IOB**

- Three-state TS
- Output Data O
- Input Data 1
- Input Data 2

**CAD for FPGAs**

- Design Entry ➔ Synthesis ➔ Logic Optimization
- Placement ➔ Packing LUTs to CLBs ➔ Mapping to k-LUT
- Routing ➔ Simulation ➔ Configure an FPGA

**CAD for FPGAs: Place & Route**

- Design Entry ➔ Synthesis ➔ Logic Optimization
- Placement ➔ Packing LUTs to CLBs ➔ Mapping to k-LUT
- Routing ➔ Simulation ➔ Configure an FPGA

**Programming an FPGA?**

Technology Mapping
Placement
Routing

**FPGA Placement Problem**

- **Input** – A technology mapped netlist of Configurable Logic Blocks (CLB) realizing a given circuit.
- **Output** – CLB netlist placed in a two-dimensional array of slots such that total wirelength is minimized.

**Global vs. Detailed Routing**

- **Global routing**
- **Detailed routing**
Remember!

Programmable Lookup Tables (LUTs)

Main bottleneck with state-of-the-art fine grain FPGAs is the routing enabled by pass transistors!

Remember!

Programmable Lookup Tables (LUTs)

Look-up-tables are flexible but require lots of configuration and suffer from power dissipation!

Fine Grain FPGAs: Spartan2

- 4K bit RAM blocks
- Large amount of logic
- Program stored in SRAM

Medium Grain: Xilinx Virtex

- Virtex-II FPGA introduced followed by Virtex-II Pro in 2003
  - 444 18x18 Multipliers & 18bit block RAMs introduced
  - Gbit Serial I/O Communications & Power PC Processors Introduced
  - Complex Floating Point Algorithm Implementation now possible

- Virtex-II / Pro
  - 44,000 Logic Slices
  - 444 18Kbits BRAMs
  - 444 18x18 Multipliers
  - 2 PowerPC Processors
  - 20 Gbit I/O
  - 1164 Max User I/O

Zynq - Extensible Processing Platform

Dynamic Partial Reconfiguration

- Partial Reconfiguration is the ability to dynamically modify blocks of logic while the remaining logic continues to operate without interruption.
- Computation sequences are not known at compile time. The system decides, respectively reacts dynamically to application driven reconfiguration requests.
Methods for executing algorithms

Hardware (Application Specific Integrated Circuits)
- Advantages: very high performance and efficient
- Disadvantages: not flexible (can’t be altered after fabrication) * expensive

Reconfigurable computing
- Advantages: fills the gap between hardware and software
- Disadvantages: much higher performance than software

Software-programmed processors
- Advantages: software is very flexible to change
- Disadvantages: performance can suffer if clock is not fast

Advantages:
- fixed instruction set by hardware

Disadvantages:
- very high performance and efficient

Hardware
- Application Specific Integrated Circuits

Software-programmed processors
- processors

Advantages:
- software is very flexible to change

Disadvantages:
- performance can suffer if clock is not fast
- fixed instruction set by hardware

School of Engineering
**FMRI and Real-time Human Body Imaging**
- Technique for determining which parts of the brain are activated by different types of physical sensation or activity – “brain mapping”
- High- and low-resolution scans compared using numerous FFTs
  - Typically post-processed
  - Much error correction needed due to subject movement
- Studying how RC devices can achieve real-time processing

**Biomechanical Kinematics**
- Knee-joint simulation
  - Build a generic model to predict human movement (jumping, walking, etc)
  - Used to study joint replacement stresses without risking patient injury
  - Biomechanical simulations frequently use costly optimization methods
  - Studying how RC-based parallel processing can increase performance

**Image Registration**
- In computer vision, sets of data acquired by sampling the same scene or object at different times, or from different perspectives, will be in different coordinate systems.
- Image registration is the process of transforming the different sets of data into one coordinate system.
- Registration is necessary in order to be able to compare or integrate the data obtained from different measurements.

**Satellite Imaging**
- Satellite imaging used for mapping, environmental studies and defense applications
- High-data rate and low-power demands of space require cutting-edge technology such as RC to provide required processing capabilities
- Including RC devices in the processing chain will eventually enhance performance

**Adaptive Integrated Driver Vehicle Interface**

**Volvo, CRF, PSA, Renault, DaimlerChrysler, Ford, BMW, SEAT, OPEL**

**...Towards a safe use of on board Support Systems and Services: The AIDE Integrated Project**

**ITS Driving Assistance - Information Support**

www.seeingmachines.com
Summary

- Programmable logic comes in different flavors such as PLDs, CPLDs and FPGAs.
- Field Programmable Gate Arrays is a technology introduced in the late 80’s to allow Engineers to implement their design without the need to fabricate the chip as we do in Application Specific Integrated Circuits (ASICs).
- The main components of an FPGA are the CLBs, IOBs and programmable interconnect (Fine Grain FPGAs).
- New technologies of FPGAs include Block Memory, Processors, Multipliers (we start to call these Coarse Grain FPGAs).
- Applications of FPGAs in HPC, Embedded Systems, Cars, Appliances, … (Endless ..)

Programming

- Programming the PLA can be specified in tabular form
- 3 sections,
  1. product terms,
  2. input and AND gates,
  3. Outputs

Programming Table for the PLA in Figure 4-24:

<table>
<thead>
<tr>
<th>Product Terms</th>
<th>Inputs A</th>
<th>Inputs B</th>
<th>Inputs C</th>
<th>Outputs P1</th>
<th>Outputs P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>AC</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BC</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ABC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>