LSN 6
Programmable Logic Devices

ECT 224 Digital Computer Fundamentals
LSN 6 – What Are PLDs?

- Functionless devices in base form
  - Require programming to operate
  - The logic function of the device is programmed by the user
- Replaces fixed function ICs and associated hard wiring
- High density of logic circuits
LSN 6 – Why Use PLDs?

- Total system costs
- Time to market
- Programmability
- Reliability
- Electromagnetic interference
- Design security capabilities
Why Use PLDs?

- Total system costs
  - The unit costs of entry level PLDs have been driven down to the point where they are equal to or even below those of discrete logic devices
LSN 6 – Why Use PLDs?

- System Cost Example
LSN 6 – Why Use PLDs?

• Time to market / Programmability
  – PLDs cannot only get to market faster, they stay in the market longer
  – PLDs provide remote bug fixes and feature upgrades that avoid costly hardware changes
LSN 6 – Why Use PLDs?

• Reliability
  – By employing a lower number of devices over the discrete TTL equivalent circuits, PLDs provide a significantly improved FIT rate
  – PLD-based systems require fewer components and layers which
    • Reduces PC board layout density
    • Lowers heat dissipation
    • Reduces EMI levels
LSN 6 – Why Use PLDs?

• Electromagnetic interference
  – EMI originates from the switching of digital circuits
  – EMI compliance carries a cost and high risk, as it can delay the product introduction
  – PLDs significantly reduce EMI through fewer external components, and other “free” features including:
    • Programmable I/O slew rate, programmable ground
    • Programmable I/O signaling, and phase-locked loops
LSN 6 – Why Use PLDs?

- Design security capabilities
  - Unlike discrete logic devices which are extremely susceptible to reverse engineering which is as simple as reading the part number directly from the device
  - PLDs inherently require a user-defined bit stream which easily prevents customer readback
LSN 6 – PLD Organization

- Programmable array
- Matrix of conductors that form rows and columns with a programmable link at each point
- One time programmable (OTP) devices
  - Interconnections are fused or open anti-fuses
    - Blown to create an open circuit
    - Melted to create connections
- Re-programmable devices
  - Interconnections are electrically erasable CMOS (E2CMOS) cells
    - Programmed ON or OFF
LSN 6 – PLD Organization

- Program volatility
  - Typically, volatile PLDs provide higher density, more features and lower cost compared to non-volatile PLDs
  - Volatile PLDs lose configuration when powered off
    - Utilize SRAM data storage technology
    - External memory is required to store the configuration, which creates security risks
  - Non-volatile PLDs retain programming data when the power is off
    - Utilize EPROM, EEPROM, or FLASH data storage technologies
    - Do not need an external memory device
LSN 6 – PLD Types

- **SPLD** (simple programmable logic devices)
  - Replaces several fixed function logic ICs
- **CPLD** (complex programmable logic devices)
  - Replaces 2-64 SPLDs
- **FPGA** (field programmable gate arrays)
  - Different internal architecture than SPLD/CPLD
  - Highest logic capacity with arrays from 64 to thousands of logic gate groups
    - Gate groups called “blocks”
LSN 6 – SPLD Classifications

- **PROM** (programmable read only memory)
  - Set of no-programmable (fixed) AND gates that act as a decoder and a programmable OR array

- **PLA** (programmable logic array)
  - Contains both a programmable OR and AND array

- **PAL** (programmable array logic)
  - Contains a programmable AND array with a fixed OR array with output logic

- **GAL** (generic array logic)
  - Reprogrammable AND array with a fixed OR array and programmable output logic
LSN 6 – Programmable Array Logic

Input lines

Fixed connection
Single line with slash represents multiple AND gate inputs. (In this case, 2 inputs)
Each variable is connected to a different line.

Product term lines

Fuse blown (no connection)
Fuse intact (connection)

X = AB + A̅B + A̅B̅
Example:

- Show a programmed PAL representation for the SOP expression

\[ X = \overline{A}BC + A\overline{B}C + AB \]
LSN 6 – Programmable Array Logic

• PAL output combinational logic
  – Several common output logic configurations can be used together in a single PAL device
  – Combinational output
    • The output is used for an SOP function
    • Typically available as active-HIGH, active-LOW, or as a high-impedance disconnect state
LSN 6 – Programmable Array Logic

- Combinational input/output
  - The output function must feedback to be an input to the array or to be used to make the I/O pin an input only.

![Diagram of Programmable Array Logic](attachment:diagram.png)
LSN 6 – Programmable Array Logic

Programmable AND array

Fixed OR array

Output logic

Output logic

Output 1

Output 2

Output m

Programmable AND array

I/O 1

I/O 2

I/O 3

I/O 4

I/O 5

I/O 6

I/O 7

I/O 8

I/O 9

I/O 10

I/O 11

I/O 12

I/O 13

I/O 14

I/O 15

I/O 16

I/O 17

I/O 18

I/O 19

I/O 20

I/O 21

I/O 22

I/O 23

I/O 24

I/O 25

I/O 26

I/O 27

I/O 28

I/O 29

I/O 30

I/O 31

I/O 32

I/O 33

I/O 34

I/O 35

I/O 36

I/O 37

I/O 38

I/O 39

I/O 40

I/O 41

I/O 42

I/O 43

I/O 44

I/O 45

I/O 46

I/O 47

I/O 48

I/O 49

I/O 50

I/O 51

I/O 52

I/O 53

I/O 54

I/O 55

I/O 56

I/O 57

I/O 58

I/O 59

I/O 60

I/O 61

I/O 62

I/O 63

I/O 64

I/O 65

I/O 66

I/O 67

I/O 68

I/O 69

I/O 70

I/O 71

I/O 72

I/O 73

I/O 74

I/O 75

I/O 76

I/O 77

I/O 78

I/O 79

I/O 80

I/O 81

I/O 82

I/O 83

I/O 84

I/O 85

I/O 86

I/O 87

I/O 88

I/O 89

I/O 90

I/O 91

I/O 92

I/O 93

I/O 94

I/O 95

I/O 96

I/O 97

I/O 98

I/O 99

I/O 100

I/O 101

I/O 102

I/O 103

I/O 104

I/O 105

I/O 106

I/O 107

I/O 108

I/O 109

I/O 110

I/O 111

I/O 112

I/O 113

I/O 114

I/O 115

I/O 116

I/O 117

I/O 118

I/O 119

I/O 120

I/O 121

I/O 122

I/O 123

I/O 124

I/O 125

I/O 126

I/O 127

I/O 128

I/O 129

I/O 130

I/O 131

I/O 132

I/O 133

I/O 134

I/O 135

I/O 136

I/O 137

I/O 138

I/O 139

I/O 140

I/O 141

I/O 142

I/O 143

I/O 144

I/O 145

I/O 146

I/O 147

I/O 148

I/O 149

I/O 150

I/O 151

I/O 152

I/O 153

I/O 154

I/O 155

I/O 156

I/O 157

I/O 158

I/O 159

I/O 160

I/O 161

I/O 162

I/O 163

I/O 164

I/O 165

I/O 166

I/O 167

I/O 168

I/O 169

I/O 170

I/O 171

I/O 172

I/O 173

I/O 174

I/O 175

I/O 176

I/O 177

I/O 178

I/O 179

I/O 180

I/O 181

I/O 182

I/O 183

I/O 184

I/O 185

I/O 186

I/O 187

I/O 188

I/O 189

I/O 190

I/O 191

I/O 192

I/O 193

I/O 194

I/O 195

I/O 196

I/O 197

I/O 198

I/O 199

I/O 200

I/O 201

I/O 202

I/O 203

I/O 204

I/O 205

I/O 206

I/O 207

I/O 208

I/O 209

I/O 210

I/O 211

I/O 212

I/O 213

I/O 214

I/O 215

I/O 216

I/O 217

I/O 218

I/O 219

I/O 220

I/O 221

I/O 222

I/O 223

I/O 224

I/O 225

I/O 226

I/O 227

I/O 228

I/O 229

I/O 230

I/O 231

I/O 232

I/O 233

I/O 234

I/O 235

I/O 236

I/O 237

I/O 238

I/O 239

I/O 240

I/O 241

I/O 242

I/O 243

I/O 244

I/O 245

I/O 246

I/O 247

I/O 248

I/O 249

I/O 250

I/O 251

I/O 252

I/O 253

I/O 254

I/O 255
LSN 6 – Generic Array Logic

- **GAL**
  - Type of PAL with a reprogrammable array of AND gates
  - SOP form
LSN 6 – CPLD

- Contains multiple groups of PAL/GAL-like arrays with programmable interconnections
- Each PAL/GAL-like group is called a logic-array-block (LAB), function-block, or similar term
  - Each group contains several PAL/GAL-like arrays called macrocells
LSN 6 – CPLD
LSN 6 – FPGA

• Contains an array of logic blocks with programmable row and column interconnected channels surrounded by programmable I/O blocks

• Most FPGA architectures is based on a type of memory called a look-up-table (LUT) instead of AND/OR gate arrays

• Each logic block contains several logic elements (LE)
LSN 6 – FPGA
LSN 6 – FPGA

- Lookup table
  - Also called Function Generators (FGs)
  - Capacity is limited by the number of inputs, not by the complexity
  - Delay through the LUT is constant

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<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>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<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>
<td>1</td>
</tr>
</tbody>
</table>
LSN 6 – FPGA

- Programmable interconnections
  - FPGAs typically use SRAM or antifuse to produce interconnection links between logic blocks
    - Antifuse
      - Melts to produce connection (NO)
      - One-time programmable
LSN 6 – FPGA

- **SRAM**
  - On-chip SRAM cell controls the state of transistor connections between interconnect lines
  - Reprogrammable by changing the program stored in the SRAM device
LSN 6 – PLD Design Flow

• Start with an idea or specification that needs to be implemented
• Next begin the circuit design
  – Schematics
  – Hardware description languages
  – Other methods (equations, etc.)
• Implementation of design
  – Schematic capture
  – Coding HDL
  – Developing state diagrams
LSN 6 – PLD Design Flow

• Synthesis of logic design
  – Develop a model of design for target PLD
• Verification of logic design through simulation
  – Simulation with test vectors and schematic simulators
  – Formal verification using HDL simulators
• Various types of verification
  – Logical correctness
  – Static timing analysis
  – In system verification
LSN 6 – PLD Design Flow

• Program the internal circuitry of the PLD to implement the logic operations
  – Download design to target device
    • The programmed device has the same operation as its associated fixed-function logic circuit
• Programmed devices must be tested to ensure they operate as specified
  – Verify on an expensive tester / in system (production)
  – Verify in the lab with debug equipment (prototype)
LSN 6 – PLD Programming

• Schematic entry
  – User draws the logic design using graphical representations of standard logic components and their interconnections

• Text-based entry
  – User enters a logic design using a Hardware Description Language (HDL)
    • Verilog
    • VHDL

• State-flow entry
LSN 6 – Homework

• Reading
  – Chapter 11.1 and 11.5

• Assignment – HW8
  – Chapter 11, problems 1, 4, 5, 16, and 17

• References
  – http://www.netrino.com/Articles/ProgrammableLogic/index.html
  – Xilinx white paper “The Advantages of Migrating from Discrete 7400 Logic Devices to CPLDs”
  – FPGA & Structured ASIC Journal article “All is Not SRAM”