CPLD and FPGA

Gaurav Verma
ECE Dept
NIEC
The first PLDs were Programmable Logic Arrays (PLAs).

A PLA is a combinational, 2-level AND-OR device that can be programmed to realise any sum-of-products logic expression.

A PLA is limited by:
- the number of inputs (n)
- the number of outputs (m)
- the number of product terms (p)

We refer to an “n x m PLA with p product terms”. Usually, $p \ll 2^n$.

An n x m PLA with p product terms contains p 2n-input AND gates and m p-input OR gates.
PLD

- Each input is connected to a buffer that produces a true and a complemented version of the signal.

A 4x3 PLA with 6 product terms.

- Potential connections are indicated by Xs.
- The device is programmed by establishing the needed connections.
- The connections are made by fuses.
CPLD
Complex PLDs

• What is the next step in the evolution of programmable logic?
  – More gates!

• How do we get more gates?

• We could put several PALs on one chip and put an interconnection matrix between them!!

  – This is called a **Complex PLD (CPLD)**.
Each logic block is similar to a 22V10.

Programmable interconnect matrix.

Logic block diagram

Figure 1. Ultra37128 Block Diagram
Cypress CPLDs

• Ultra37000 Family
  – 32 to 512 Macrocells
  – Fast ($T_{pd}$ 5 to 10ns depending on number of macrocells)
  – Very good routing resources for a CPLD
Other approaches and Issues

• Another approach to building a “better” PLD is
  • place a lot of primitive gates on a die,
  • and then place programmable interconnect between them:
FPGA
What is an FPGA?

• **Field Programmable Gate Array**

• Fully programmable alternative to a customized chip

• Used to implement functions in hardware

• Also called a Reconfigurable Processing Unit (RPU)
Reasons to use an FPGA

• Hardwired logic is very fast

• Can interface to outside world
  – Custom hardware/peripherals
  – “Glue logic” to custom co-processors

• Can perform bit-level and systolic operations not suited for traditional CPU/MPU
Look Up Tables

- Combinatorial Logic is stored in 16x1 SRAM Look Up Tables (LUTs) in a CLB
- Example:

<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>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</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>

- Capacity is limited by number of inputs, not complexity
- Choose to use each function generator as 4 input logic (LUT) or as high speed sync. dual port RAM

\[ 2^{(2^4)} = 64K \]
The FPGA approach to arrange primitive logic elements (logic cells) arrange in rows/columns with programmable routing between them.

What constitutes a primitive logic element?

Lots of different choices can be made! Primitive element must be classified as a “complete logic family”.

- A primitive gate like a NAND gate
- A 2/1 mux (this happens to be a complete logic family)
- A Lookup table (i.e., 16x1 lookup table can implement any 4 input logic function).

Often combine one of the above with a DFF to form the primitive logic element.
Issues in FPGA Technologies

- **Complexity of Logic Element**
  - How many inputs/outputs for the logic element?
  - Does the basic logic element contain a FF? What type?

- **Interconnect**
  - How fast is it? Does it offer ‘high speed’ paths that cross the chip? How many of these?
  - Can I have on-chip tri-state busses?
  - How routable is the design? If 95% of the logic elements are used, can I route the design?

  - More routing means more routability, but less room for logic elements.
Issues in FPGA Technologies (cont)

- **Macro elements**
  - Are there SRAM blocks? Is the SRAM dual ported?
  - Is there fast adder support (i.e. fast carry chains?)
  - Is there fast logic support (i.e. cascade chains)
  - What other types of macro blocks are available (fast decoders? register files? )

- **Clock support**
  - How many global clocks can I have?
  - Are there any on-chip Phase Logic Loops (PLLs) or Delay Locked Loops (DLLs) for clock synchronization, clock multiplication?
Issues in FPGA Technologies (cont)

• What type of **IO support** do I have?
  – TTL, CMOS are a given
  – Support for mixed 5V, 3.3v IOs?
    • 3.3 v internal, but 5V tolerant inputs?

  – Support for **new low voltage signaling standards**?
    • GTL+, GTL (Gunning Tranceiver Logic) - used on Pentium II
    • HSTL - High Speed Transceiver Logic
    • SSTL - Stub Series-Terminate Logic
    • USB - IO used for Universal Serial Bus (differential signaling)
    • AGP - IO used for Advanced Graphics Port

  – Maximum number of IO? Package types?
    • Ball Grid Array (BGA) for high density IO
Altera FPGA Family

- **Altera Flex10K/10KE**
  - LEs (Logic elements) have 4-input LUTS (look-up tables) +1 FF
  - Fast Carry Chain between LE’s, Cascade chain for logic operations
  - Large blocks of SRAM available as well

- **Altera Max7000/Max7000A**
  - EEPROM based, very fast (Tpd = 7.5 ns)
  - Basically a PLD architecture with programmable interconnect.
  - Max 7000A family is 3.3 v
# Altera Flex 10K FPGA Family

## Table 1. FLEX 10K Device Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>EPF10K10</th>
<th>EPF10K20</th>
<th>EPF10K30</th>
<th>EPF10K40</th>
<th>EPF10K50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical gates (logic and RAM), <em>Note (1)</em></td>
<td>10,000</td>
<td>20,000</td>
<td>30,000</td>
<td>40,000</td>
<td>50,000</td>
</tr>
<tr>
<td>Usable gates</td>
<td>7,000 to 31,000</td>
<td>15,000 to 63,000</td>
<td>22,000 to 69,000</td>
<td>29,000 to 93,000</td>
<td>36,000 to 116,000</td>
</tr>
<tr>
<td>Logic elements (LEs)</td>
<td>576</td>
<td>1,152</td>
<td>1,728</td>
<td>2,304</td>
<td>2,880</td>
</tr>
<tr>
<td>Logic array blocks (LABs)</td>
<td>72</td>
<td>144</td>
<td>216</td>
<td>288</td>
<td>360</td>
</tr>
<tr>
<td>Embedded array blocks (EABs)</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Total RAM bits</td>
<td>6,144</td>
<td>12,288</td>
<td>12,288</td>
<td>16,384</td>
<td>20,480</td>
</tr>
<tr>
<td>Maximum user I/O pins</td>
<td>134</td>
<td>189</td>
<td>246</td>
<td>189</td>
<td>310</td>
</tr>
</tbody>
</table>
## Table 2. FLEX 10K Device Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>EPF10K70</th>
<th>EPF10K100</th>
<th>EPF10K130V</th>
<th>EPF10K250A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical gates (logic and RAM), Note (1)</td>
<td>70,000</td>
<td>100,000</td>
<td>130,000</td>
<td>250,000</td>
</tr>
<tr>
<td>Usable gates</td>
<td>46,000 to 118,000</td>
<td>62,000 to 158,000</td>
<td>82,000 to 211,000</td>
<td>149,000 to 310,000</td>
</tr>
<tr>
<td>LEs</td>
<td>3,744</td>
<td>4,992</td>
<td>6,656</td>
<td>12,160</td>
</tr>
<tr>
<td>LABs</td>
<td>468</td>
<td>624</td>
<td>832</td>
<td>1,520</td>
</tr>
<tr>
<td>EABs</td>
<td>9</td>
<td>12</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>Total RAM bits</td>
<td>18,432</td>
<td>24,576</td>
<td>32,768</td>
<td>40,960</td>
</tr>
<tr>
<td>Maximum user I/O pins</td>
<td>358</td>
<td>406</td>
<td>470</td>
<td>470</td>
</tr>
</tbody>
</table>

**Note to tables:**
(1) For designs that require JTAG boundary-scan testing, the built-in JTAG circuitry contributes up to 31,250 additional gates.
FLEX 10K Device Block Diagram

Dedicated memory
Figure 6. FLEX 10K Logic Element

16 x1 LUT

Look-Up Table (LUT) → Carry Chain

Carry-In → Cascade-In

Cascade Chain

Register Bypass

Programmable Register

to FastTrack Interconnect
to LAB Local Interconnect

Clear/Preset Logic

Chip-Wide Reset

LABCTRL1
LABCTRL2

LABCTRL3
LABCTRL4

Clean/Select

Clock

DFF

Carry-Out

Cascade-Out
Embedded Array Block

- Memory block, Can be configured:
  - 256 x 8, 512 x 4, 1024 x 2, 2048 x 1
Actel FPGA Family

- **MXDS Family**
  - Fine grain Logic Elements that contain Mux logic + DFF
  - Embedded **Dual Port SRAM**
  - One Time Programmable (OTP) - means that no configuration loading on powerup, no external serial ROM
  - **AntiFuse** technology for programming (AntiFuse means that you program the fuse to make the connection).
  - **Fast** (Tpd = 7.5 ns)
  - **Low density** compared to Altera, Xilinx - maximum number of gates is 36,000
Who is Xilinx?

• Provides programmable logic solutions


Programmable Logic Chips

Foundation and Alliance Series Design Software

• Inventor of the Field Programmable Gate Array

• $900M Annual Revenues; 36+% annual growth
Xilinx FPGA Family

• Virtex Family
  – SRAM Based
  – Largest device has 1M gates
  – Configurable Logic Blocks (CLBs) have two 4-input LUTS, 2 DFFs
  – Four onboard Delay Locked Loops (DLLs) for clock synchronization
  – Dedicated RAM blocks (LUTs can also function as RAM).
  – Fast Carry Logic

• XC4000 Family
  – Previous version of Virtex
  – No DLLs, No dedicated RAM blocks
XC4000 Architecture

Programmable Interconnect

Configurable Logic Blocks (CLBs)

I/O Blocks (IOBs)
XC4000E/X Configurable Logic Blocks

- 2 Four-input function generators (Look Up Tables)
  - 16x1 RAM or Logic function
- 2 Registers
  - Each can be configured as Flip Flop or Latch
  - Independent clock polarity
  - Synchronous and asynchronous Set/Reset
ALTERA CYCLONE PROCESSOR

The Cyclone device family offers the following features:
- 2,910 to 20,060 LEs, see Table 1–1
- Up to 294,912 RAM bits (36,864 bytes)
- Supports configuration through low-cost serial configuration device
- Support for LVTTL, LVCMOS, SSTL-2, and SSTL-3 I/O standards
- Support for 66- and 33-MHz, 64- and 32-bit PCI standard
- High-speed (640 Mbps) LVDS I/O support
- Low-speed (311 Mbps) LVDS I/O support
- 311-Mbps RSDS I/O support
- Up to two PLLs per device provide clock multiplication and phase shifting
- Up to eight global clock lines with six clock resources available per logic array block (LAB) row
- Support for external memory, including DDR SDRAM (133 MHz), FCRAM, and single data rate (SDR) SDRAM
- Support for multiple intellectual property (IP) cores, including Altera MegaCore functions and Altera Mega functions Partners Program (AMPPSM) megafuctions.
Cyclone II EP2C20 Device Block Diagram

=> Cyclone® devices contain a two-dimensional row- and column-based architecture to implement custom logic. Column and row interconnects of varying speeds provide signal interconnects between LABs and embedded memory blocks.
LE Operating Modes

The Cyclone II LE operates in one of the following modes:

- Normal mode
- Arithmetic mode
Normal Mode

The normal mode is suitable for general logic applications and combinational functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT. The Quartus II Compiler automatically selects the carry-in or the data3 signal as one of the inputs to the LUT. LEs in normal mode support packed registers and register feedback.
**Arithmetic Mode**

The arithmetic mode is ideal for implementing adders, counters, accumulators, and comparators. An LE in arithmetic mode implements a 2-bit full adder and basic carry chain (see Figure 2–4). LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output. Register feedback and register packing are supported when LEs are used in arithmetic mode.