CHAPTER 3

Programmable Logic Technology

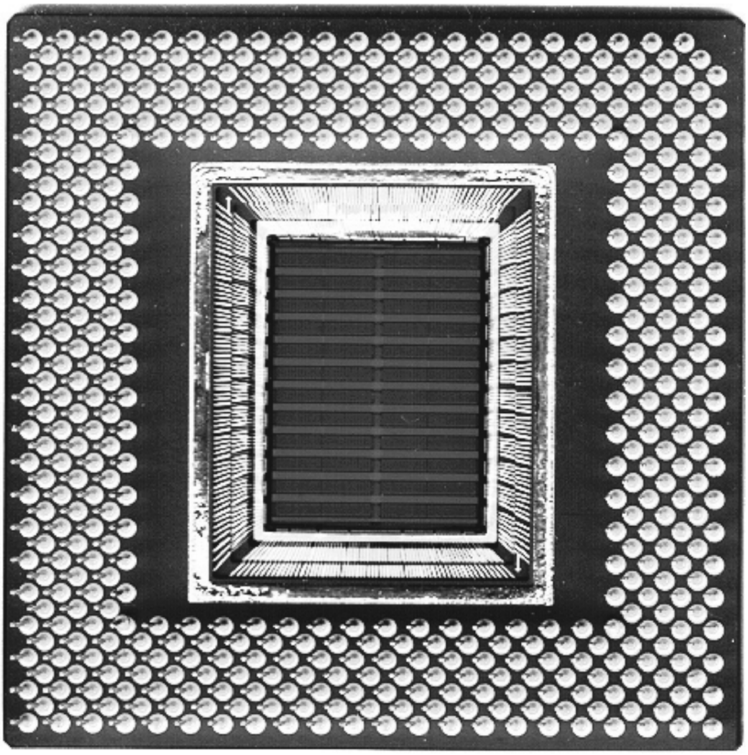


Photo: An Altera Flex 10K100 FPGA containing 10,000,000 Transistors and 100,000 gates. The FPGA is in a pin grid array (PGA) package. The cover has been removed so that the chip die is visible in the center of the package.

3 Programmable Logic Technology

A wide spectrum of devices is available for the implementation of digital logic designs as shown in Figure 3.1. Older traditional off-the-shelf integrated circuit chips, such as SSI and MSI TTL, performed a fixed operation defined by the device manufacturer. A user must connect a number of different chip types to build even a simple logic circuit with this older technology. A large number of chips will also be required as each chip contains only a few basic logic gates. Application specific integrated circuits (ASICs), complex programmable logic devices (CPLDs), and field programmable gate arrays (FPGAs) are integrated circuits whose internal functional operation is defined by the user. ASICs require a final customized manufacturing step for the user-defined function. A CPLD or FPGA requires user programming to perform the desired operation.

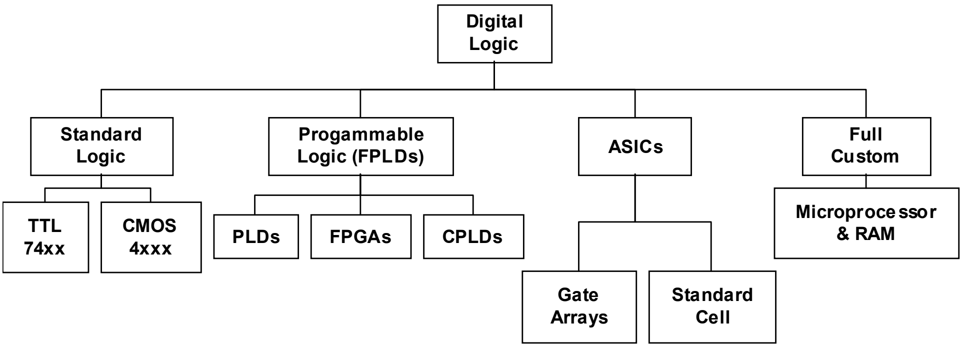


Figure 3.1 Digital logic technologies.

The design tradeoffs of the different technologies are seen in Figure 3.2. Full custom VLSI development of a design at the transistor level can require several years of engineering effort for design and testing. Such an expensive development effort is warranted only for the highest volume devices. This approach can generate the highest performance devices. Examples of full custom devices include the microprocessor and RAM chips used in PCs.

ASICs can be divided into three categories, Gate Arrays, Standard Cell and Structured. Gate Arrays are built from arrays of pre-manufactured logic cells. A single logic cell can implement a few gates or a flip-flop. A final manufacturing step is required to interconnect the sea of logic cells on a gate array. This interconnection pattern is created by the user to implement a particular design. Standard Cell devices contain no fixed internal structure. For standard cell devices, the manufacturer creates a custom photographic mask to build the chip based on the user's selection of devices, such as controllers, ALUs, RAM, ROM, and microprocessors from the manufacturer's standard cell library. New Structured ASICs are similar to gate arrays but each array element contains more logic. They offer tradeoffs somewhere between other ASICs and FPGAs.

Since ASICs require custom manufacturing, additional time and development costs are involved. Several months are normally required and substantial setup fees are charged. ASIC setup fees can be as high as a few million dollars. Additional effort in testing must be performed by the user since chips are tested after the final custom-manufacturing step. Any design error in the chip will lead to additional manufacturing delays and costs. For products with long lifetimes and large volumes, this approach has a lower cost per unit than CPLDs or FPGAs. Economic and performance tradeoffs between ASICs, CPLDs, and FPGAs are changing with each new generation of devices and design tools.

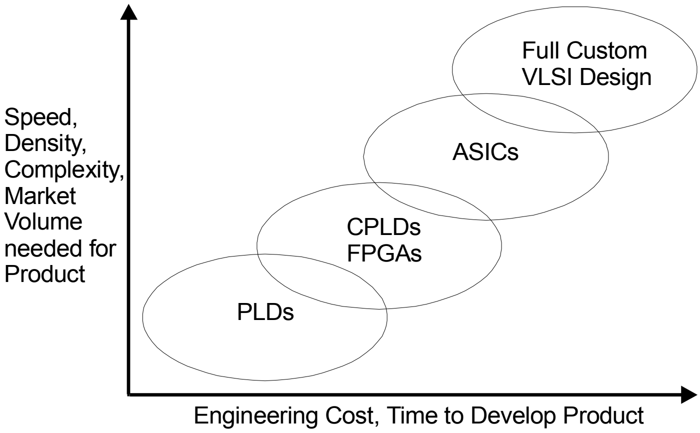

Full Custom
VLSI Design
ASICs
CPLDs
FPGAs
PLDs
Speed, Density, Complexity, Market Volume needed for Product
Engineering Cost, Time to Develop Product

Figure 3.2 Digital logic technology tradeoffs.

Simple programmable logic devices (PLDs), such as programmable array logic (PALs), and programmable logic arrays (PLAs), have been in use for over thirty years. An example of a small PLA is shown in Figure 3.3. First, the logic equation is minimized and placed in sum of products (SOP) form. The PLA has four inputs, A, B, C, and D shown in the upper left corner of Figure 3.3. Every input connects to an inverter, making the inverted values of A, B, C, and D available for use. Each product term is implemented using an AND gate with several inputs. Outputs from the two product term's AND gates then feed into an OR gate.

A special shorthand notation is used in PLAs and PALs to represent the large number of inputs present in the AND and OR gate arrays. A gate input is present at each point where the vertical and horizontal signal lines cross in Figure 3.3. Note that this means that the two AND gates actually have eight inputs and the OR gate has two inputs in the PLA. Every input signal and its complement is available as an input to the AND gates. Each gate input in the PLA is controlled by a fuse. Initially all fuses are intact. By blowing selected fuses, or programming the PLA, the desired SOP equation is produced. The top AND gate in Figure 3.3 has fuses intact to the A and B inputs, so it produces the AB product term. The lower AND gate has fuses set to produce $C\overline{D}$. The OR gate has both fuses intact, so it ORs both product terms from the AND gates to produce the final output, F = $AB + C\overline{D}$.

Small PLDs can replace several older fixed function TTL-style parts in a design. Most PLDs contain a PLA-like structure in which a series of AND gates with selectable or programmable inputs, feed into an OR gate. In PALs, the OR gate has a fixed number of inputs and is not programmable. The AND gates and OR gate are programmed to directly implement a sum-of-products Boolean equation. On many PLDs, the output of the OR gate is connected to a flip-flop whose output can then be feed back as an input into the AND gate array. This provides PLDs with the capability to implement simple state machines. A PLD can contain several of these AND/OR networks.

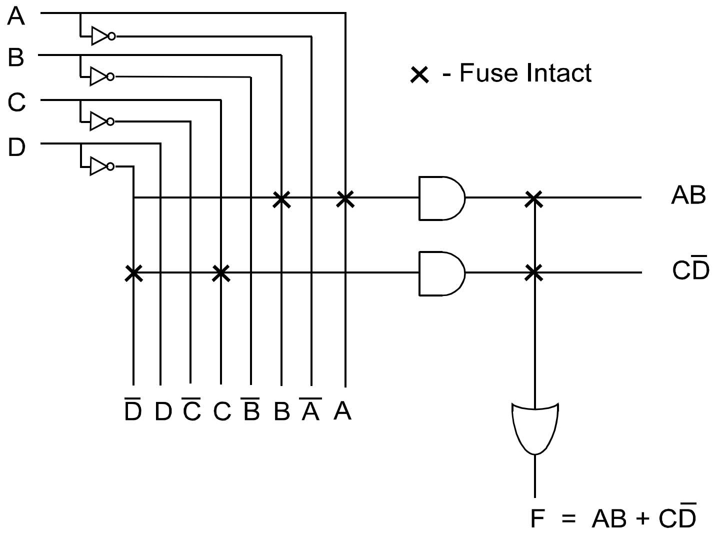

A
B
C
D
× - Fuse Intact
AB
$C\overline{D}$
$\overline{D}$ D $\overline{C}$ C $\overline{B}$ B $\overline{A}$ A
F = AB + $C\overline{D}$

Figure 3.3 Using a PLA to implement a Sum of Products equation.

In more recent times, higher densities, higher speed, and cost advantages have enabled the use of programmable logic devices in a wider variety of designs. CPLDs and FPGAs are the highest density and most advanced programmable logic devices. Designs using a CPLD or FPGA typically require several weeks of engineering effort instead of months. These devices are also sometimes collectively called field programmable logic devices (FPLDs).

ASICs and full custom designs provide faster clock times than CPLDs or FPGAs since they are hardwired and do not have programmable interconnect delays. Since ASICs and full custom designs do not require programmable interconnect circuitry they use less chip area, less power, and have a lower per unit manufacturing cost in large volumes. Initial engineering and setup costs for ASICs and full custom designs are much higher.

For all but the most time critical design applications, CPLDs and FPGAs have adequate speed with maximum clock rates typically in the range of 50-400MHz; however, clock rates up to 1GHz have been achieved on new generation FPGAs and many have a few high-speed 1-10 GHz output pins.

3.1 CPLDs and FPGAs

Internally, CPLDs and FPGAs typically contain multiple copies of a basic programmable logic element (LE) or cell. The logic element can implement a network of several logic gates that then feed into 1 or 2 flip-flops. Logic elements are arranged in a column or matrix on the chip. To perform more complex operations, logic elements can be automatically connected to other logic elements on the chip using a programmable interconnection network. The interconnection network is also contained in the CPLD or FPGA. The interconnection network used to connect the logic elements contains row and/or column chip-wide interconnects. In addition, the interconnection network often contains shorter and faster programmable interconnects limited only to neighboring logic elements.

When a design approaches the device size limits, it is possible to run out of either gate, interconnect, or pin resources when using a CPLD or FPGA. CPLDs tend to have faster and more predictable timing properties while FPGAs offer the highest gate densities and more features.

Clock signals in large FPGAs normally use special low-skew global clock buffer lines. These are dedicated pins connected to an internal high-speed bus. This special bus is used to distribute the clock signal to all flip-flops in the device at the same time to minimize clock skew. If the global clock buffer line is not used, the clock is routed through the chip just like a normal signal. The clock signal could arrive at flip-flops at widely different times since interconnect delays will vary in different parts of the chip. This delay time can violate flip-flop setup and hold times and can cause metastability or unpredictable operation in flip-flops. Most large designs with a common clock that is used throughout the FPGA will require the use of the global clock buffer.

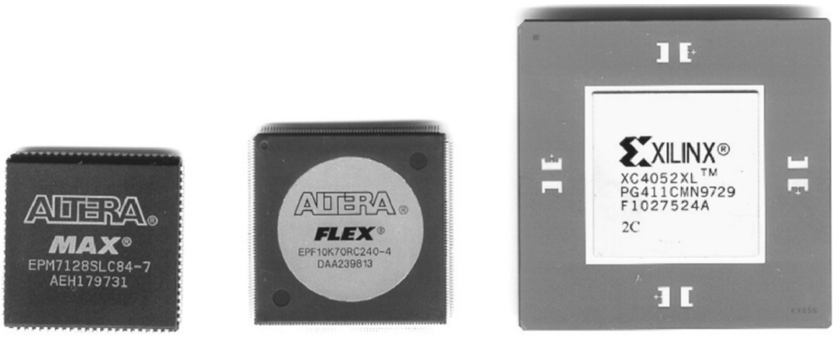


Figure 3.4 Examples of FPGAs and advanced high pin count package types.

The size of CPLDs and FPGAs is typically described in terms of useable or equivalent gates. This refers to the maximum number of two input NAND gates available in the device. This should be viewed as a rough estimate of size only.

The internal architecture of three examples of CPLD and FPGA device technologies, the Altera MAX 7000, the Altera Cyclone, and the Xilinx 4000 family will now be examined. An example of each of these devices is shown in Figure 3.4. From left to right the chips are an Altera MAX 7128S CPLD in a Plastic J-Lead Chip Carrier (PLCC), an Altera Cyclone 10K70 FPGA in a Plastic Quad Flat Pack (PQFP), and a Xilinx XC4052 FPGA in a ceramic Pin Grid Array Package (PGA). The PGA package has pins on .1" centers while the PQFP has pins on .05" centers at the edges of the package. Both Altera and Xilinx devices are available in a variety of packages.

Packaging can represent a significant portion of the FPGA chip cost. The number of I/O pins on the FPGA package often limits designs. Larger ceramic packages such as a PGA with more pins are more expensive than plastic.

3.2 Altera MAX 7000S Architecture – A Product Term CPLD Device

The multiple array matrix (MAX) 7000S is a CPLD device family with 600 to 20,000 gates. This device is configured by programming an internal electrically erasable programmable read only memory (EEPROM). Since an EEPROM is used for programming, the configuration is retained when power is removed. This device also allows in-circuit reprogrammability.

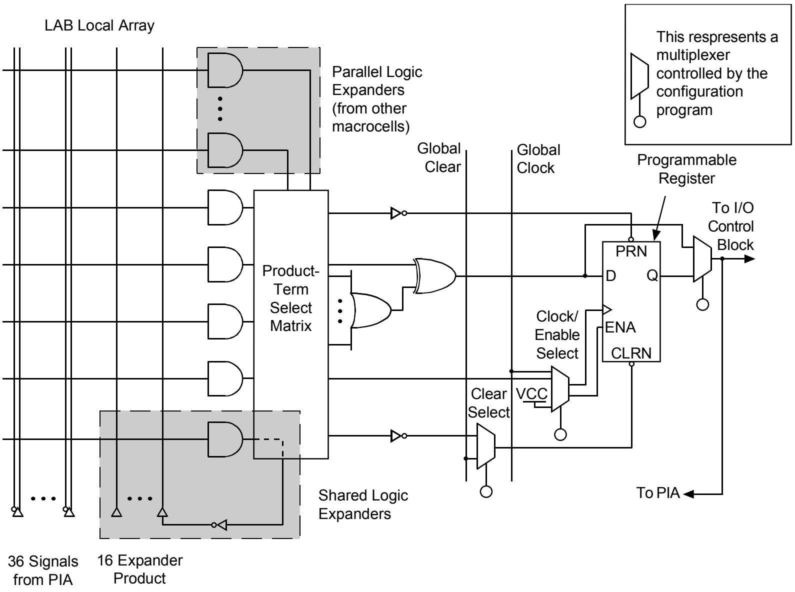


Figure 3.5 MAX 7000 macrocell.

The 7000 device family contains from 32 to 256 macrocells of the type seen in Figure 3.5. Similar to the early PALs, an individual macrocell contains five programmable AND gates with wide inputs that feed into an OR gate with a programmable inversion at the output. Just like a PAL, the AND/OR network is designed to implement Boolean equations expressed in sum-of-products form. Inputs to the wide AND gate are available in both normal and inverted forms. Parallel expanders are included that are used to borrow extra product terms from adjacent macrocells for logic functions needing more than five product terms.

The output from the AND/OR network can then be fed into a programmable flip-flop. Inputs to the AND gates include product terms from other macrocells in the same local block or signals from the chip-wide programmable interconnect array (PIA). The flip-flop contains Bypass, Enable, Clear and Preset functions and can be programmed to act as a D flip-flop, Toggle flip-flop, JK flip-flop, or SR latch.

Macrocells are combined into groups of 16 and called logic array blocks (LABs), for the overall device architecture as shown in Figure 3.6. The PIA can be used to route data to or from other LABs or external pins on the device. Each I/O pin contains a programmable tri-state output buffer. An FPGA's I/O pin can thus be programmed as input, output, output with a tri-state driver, or even tri-state bi-directional.

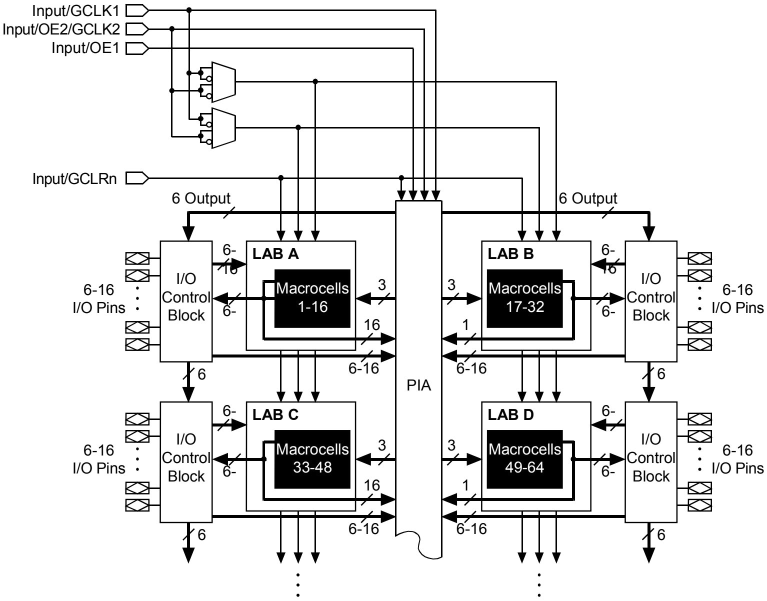


Figure 3.6 MAX 7000 CPLD architecture.

3.3 Altera Cyclone Architecture – A Look-Up Table FPGA Device

The Cyclone device is configured by loading internal static random access memory (SRAM). Since SRAM is used in FPGAs, the configuration will be lost whenever power is removed. In actual systems, a small external low-cost serial flash memory or programmable read only memory (PROM) is normally used to automatically load the FPGA's programming information when the device powers up.

FPGAs contain a two-dimensional row and column-based architecture to implement user logic. A column and row interconnection network provides signal connections between Logic Array Blocks (LABs) and embedded memory blocks. Interconnect delay times are on the same order of magnitude as logic delays.

The Cyclone FPGA's logic array consists of LABs, with 10 Logic Elements (LEs) in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device. Cyclone devices range from 2,910 to 20,060 LEs.

M4K RAM embedded memory blocks are dual-port memory blocks with 4K bits of memory plus parity (4,608 bits). These blocks provide dual-port or single-port memory from 1 to 36-bits wide at up to 200 MHz. These blocks are grouped into columns across the device in between certain LABs. The Cyclone EP1C6 and EP1C12 contain 92K and 239K bits of embedded RAM respectively.

Each of the Cyclone device's I/O pins is fed by an I/O element (IOE) located at the ends of LAB rows and columns around the periphery of the device. I/O pins support various single-ended and differential I/O standards. Each IOE contains a bidirectional I/O buffer and three registers for registering input, output, and output-enable signals.

Cyclone devices also provide a global low-skew clock network and up to two Phase Locked Loops (PLLs). The global clock network consists of eight global clock lines that drive throughout the entire device. The global clock network can provide clocks for all resources within the device, such as IOEs, LEs, and memory blocks. Cyclone PLLs provide general-purpose clocking with clock multiplication/division and phase shifting as well as external outputs for high-speed differential I/O support.

Figure 3.7 shows a Cyclone logic element. Logic gates are implemented using a look-up table (LUT), which is a high-speed 16 by 1 SRAM. Four inputs are used to address the LUT's memory. The truth table for the desired gate network is loaded into the LUT's SRAM during programming. A single LUT can therefore model any network of gates with four inputs and one output. The multiplexers seen in Figure 3.7 are all controlled by bits in the FPGA's SRAM configuration memory.

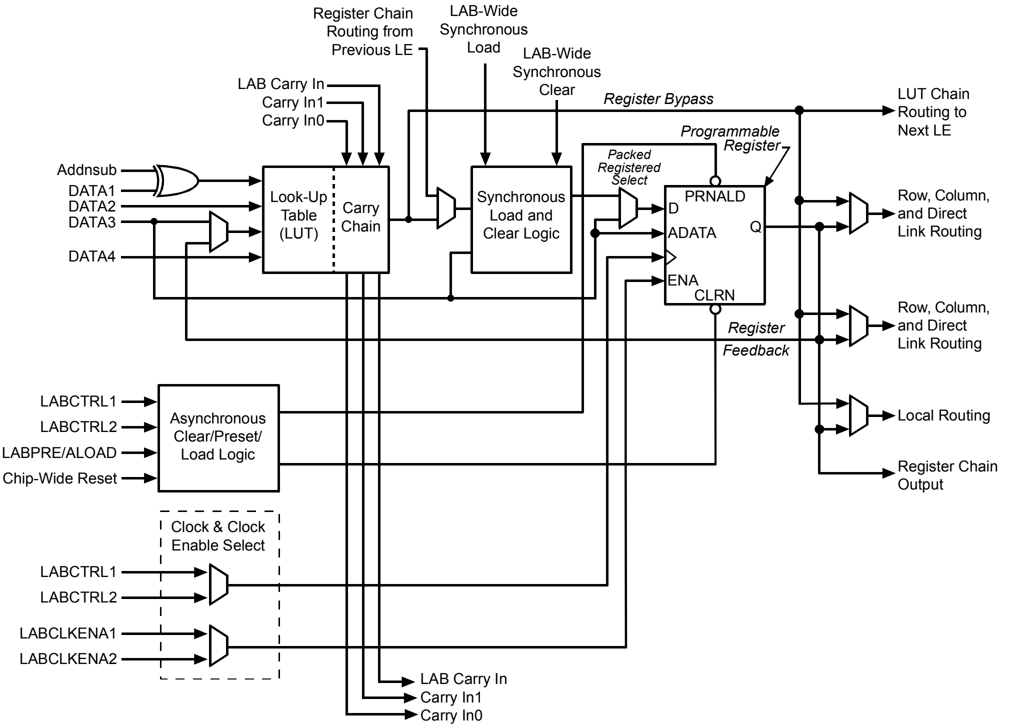
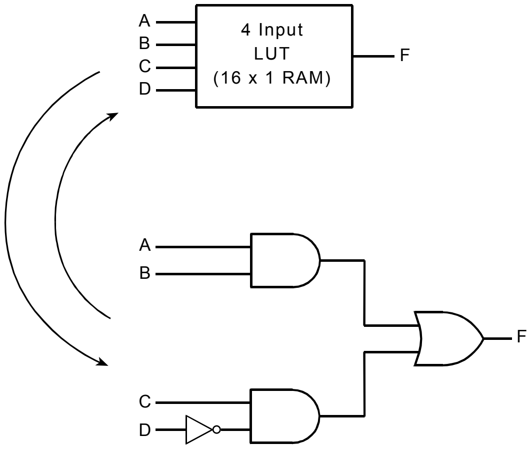


Figure 3.7 Cyclone Logic Element (LE).

An example showing how a LUT can model a gate network is shown in Figure 3.8. First, the gate network is converted into a truth table. Since there are four inputs and one output, a truth table with 16 rows and one output is needed. The truth table is then loaded into the LUT's 16 by 1 high-speed SRAM when the FPGA is programmed.

Note that the four gate inputs, A, B, C, and D, are used as address lines for the RAM and that F, the output of the truth table, is the data that is stored in the LUT's RAM. In this manner, the LUT's RAM implements the gate network by performing a RAM based table lookup instead of using actual logic gates.



RAM Contents				
Address				Data
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Figure 3.8 Using a look-up table (LUT) to model a gate network.

More complex gate networks require interconnections with additional neighboring logic elements. The output of the LUT can be fed into a D flip-flop and then to the interconnection network. The clock, Clear, and Preset can be driven by internal logic or an external I/O pin. The flip-flop can be programmed to act as a D flip-flop, T flip-flop, JK flip-flop, or SR latch. Carry and Cascade chains connect to all LEs in the same row.

Figure 3.9 shows a Logic Array Block (LAB). A logic array block is composed of ten logic elements (LEs). Both programmable local LAB and chip-wide row and column interconnects are available. Carry chains are also provided to support faster addition operations.

Input-output elements (IOEs) are located at each of the device’s I/O pins. IOEs contain a programmable tri-state driver and an optional 1-bit flip-flop register. Each I/O pin can be programmed as input, output, output with a tri-state driver, or even tri-state bi-directional with or without a register. Four clock I/O pins connect to the eight low-skew global clock buffer lines that are provided in the device.

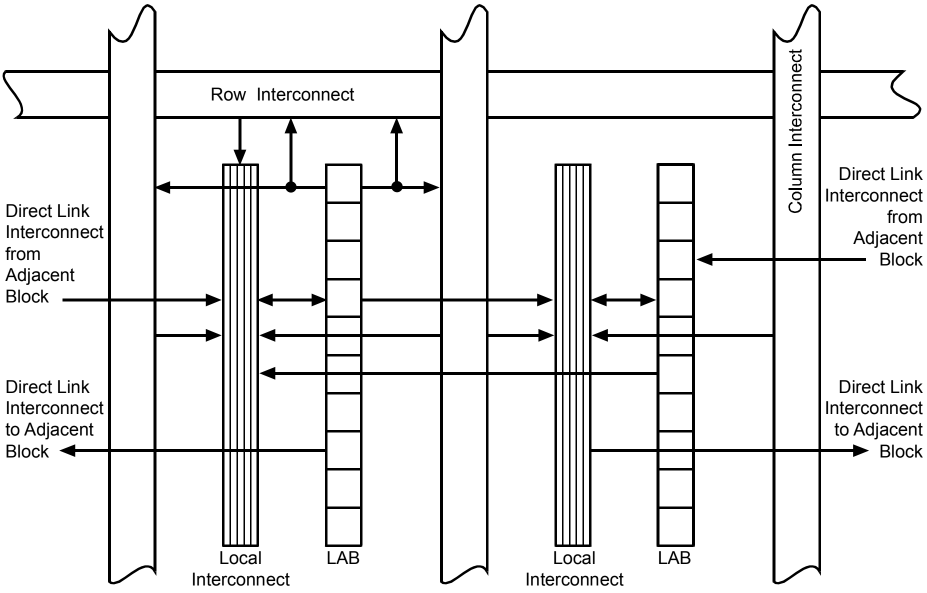

Row Interconnect
Column Interconnect
Direct Link Interconnect from Adjacent Block
Direct Link Interconnect to Adjacent Block
Direct Link Interconnect from Adjacent Block
Direct Link Interconnect to Adjacent Block
Local Interconnect
LAB
Local Interconnect
LAB

Figure 3.9 Cyclone Logic Array Blocks (LAB) and Interconnects.

3.4 Xilinx 4000 Architecture – A Look-Up Table FPGA Device

The Xilinx 4000 Family was a popular first generation FPGA device family with 2,000 to 180,000 usable gates. It is configured by programming internal SRAM. Figure 3.10 is a photograph of a six-inch silicon wafer containing several XC4010E 10,000 gate FPGA chip dice. Figure 3.11 is a contrast-enhanced view of a single XC4010E die. If you look closely, you can see the 20 by 20 array of logic elements and the surrounding interconnect lines. Die that pass wafer-level inspection and testing are sliced from the wafer and packaged in a chip. FPGA yields are typically 90% or higher after the first few production runs.

As seen in Figure 3.12, this device contains a more complex logic element called a configurable logic block (CLB). Each CLB contains three SRAM-based lookup tables. Outputs from the LUTs can be fed into two flip-flops and routed to other CLBs. A CLB's lookup tables can also be configured to act as a 16 by 2 RAM or a dual-port 16 by 1 RAM. High-speed carry logic is provided between adjacent CLBs.

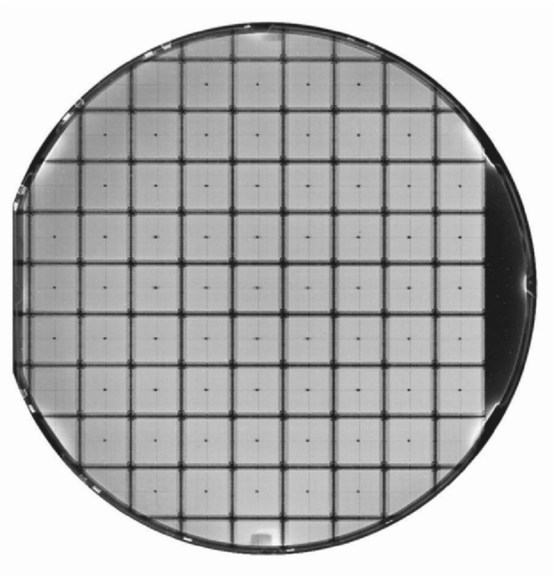


Figure 3.10 Silicon wafer containing XC4010E 10,000 gate FPGAs.

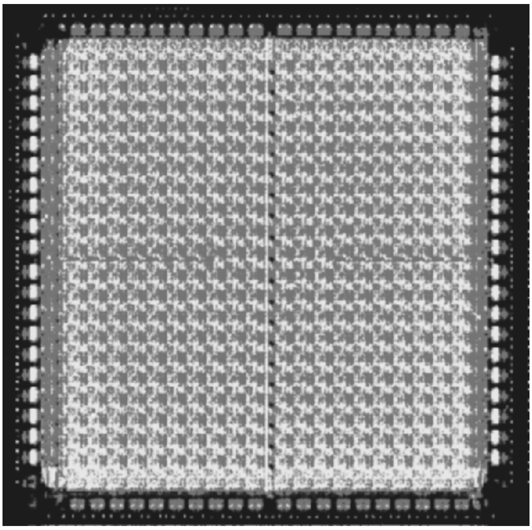


Figure 3.11 Single XC4010E FPGA die showing 20 by 20 array of logic elements and interconnect.

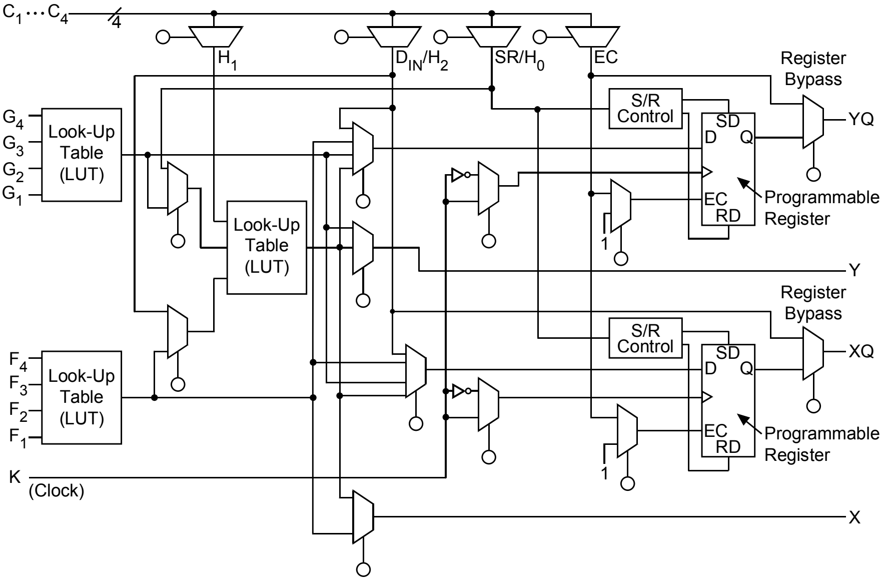


Figure 3.12 Xilinx 4000 Family Configurable Logic Block (CLB).

CLBs are arranged in a square matrix with a programmable hierarchical interconnection network. Devices in the family contain from 100 to 3,136 CLBs. The multiplexers seen in Figure 3.12 are all controlled by bits in the FPGA's SRAM configuration memory.

The complex hierarchical interconnection network contains varying length row, column, and neighboring CLB interconnect structures. Eight low-skew global clock buffers are also provided. Input-output blocks (IOBs), contain programmable tri-state drivers and optional registers. Each I/O pin can be programmed as input, output, output with a tri-state driver, or tri-state bi-directional with or without a register. In the more recent Xilinx Virtex 4 FPGAs, each CLB now contains four circuits similar to the earlier 4000 CLBs.

3.5 Computer Aided Design Tools for Programmable Logic

Increasing design complexity and higher gate densities are forcing digital designs to undergo a paradigm shift. Old technology, low-density logic families, such as the TTL 7400 or simple PLD families are rarely if ever used in new designs. With logic capacities of an individual FPGA chip approaching 10,000,000 gates, manual design at the gate level is no longer a viable option in complex systems. Rapid prototyping using hardware description languages (HDLs), IP cores, and logic synthesis tools has all but replaced traditional gate-level design with schematic capture entry. These new HDL-based logic synthesis tools can be used for both ASIC and FPGA-based designs. The two most widely used HDLs at the present time are VHDL and Verilog.

The typical FPGA CAD tool design flow is shown in Figure 3.13. After design entry using an HDL or schematic, the design is automatically translated, optimized, synthesized, and saved as a netlist. (A netlist is a text-based representation of a logic diagram.) A functional simulation step is often added prior to the synthesis step to speed up simulations of large designs.

An automatic tool then fits the design onto the device by converting the design to use the FPGA's logic elements, first by placing the design in specific logic element locations in the FPGA and then by selecting the interconnection network routing paths. The place and route process can be quite involved and can take several minutes to compute on large designs. On large devices, combinatorial explosion (exponential growth) will prevent the tool from examining all possible place and route combinations. When designs require critical timing, some tools support timing constraints that can be placed on critical signal lines. These optional constraints are added to aid the place and route tool in finding a design placement with improved performance.

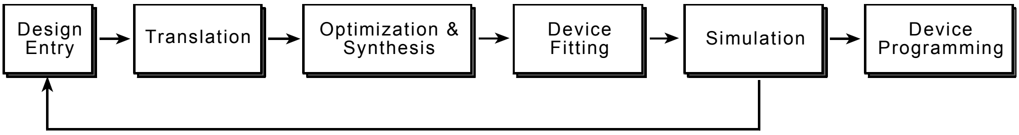


Figure 3.13 CAD tool design flow for FPGAs.

After place and route, simulation can be performed using actual gate and interconnect time delays from a detailed timing model of the device. Although errors can occur at any step, the most common path is to find errors during an exhaustive simulation. The final step is device programming and hardware verification on the FPGA.

3.6 Next Generation FPGA CAD tools

A few HDL synthesis tools now support behavioral synthesis. Unlike the more widely used register transfer level (RTL) models contained in this book, behavioral synthesis models do not specify the exact states and sequence of register transfers. A separate constraint file specifies the number of clocks needed to obtain selected signals and the tool automatically generates the state machines, logic, and register transfers needed.

Although not currently in widespread use for current designs, newer FPGA CAD tools are also appearing based on other languages such as C and Java. Some of these system-level tools output VHDL or Verilog models as an intermediate step. New HDLs such as SystemVerilog (www.systemverilog.org) and SystemC (www.systemC.org) provide enhanced support for verification.

Tools that automatically generate an FPGA design from other engineering tools such as MATLAB-Simulink or LabVIEW have also been introduced. These graphical based tools are primarily aimed at DSP application development for FPGAs using a library of specialized DSP blocks.

3.7 Applications of FPGAs

The last decade has seen ever increasing application areas for FPGAs. A recent market study found over twelve times as many new FPGA-based designs as ASIC-based designs, and ASIC setup costs continue to increase. New generation FPGAs can have nearly ten million gates with clock rates approaching 1GHz. Example application areas include single chip replacements for old multichip technology designs, Digital Signal Processing (DSP), image processing, multimedia applications, high-speed communications and networking equipment such as routers and switches, the implementation of bus protocols such as peripheral component interconnect (PCI), microprocessor glue logic, co-processors, and microperipheral controllers.

Several large FPGAs with an interconnection network are used to build hardware emulators. Hardware emulators are specially designed commercial devices used to prototype and test complex hardware designs that will later be implemented on gate arrays or custom VLSI devices. Hardware emulators are commonly used to build a prototype quickly during the development and testing of microprocessors. Several of the recent Intel and AMD processors used in PCs were tested on FPGA-based hardware emulators before the full custom VLSI processor chip was produced.

A newer application area is reconfigurable computing. In reconfigurable computing, FPGAs are quickly reprogrammed or reconfigured multiple times during normal operation to enable them to perform different computations at different times for a particular application.

3.8 Features of New Generation FPGAs

Each new generation of FPGAs increases in size and performance. In addition to more logic elements, embedded memory blocks, and interconnects, other new features are appearing. Some FPGAs contain a mix of both product term and lookup tables to implement logic. Such product term structures typically require less chip area to implement the complex gating logic present in large state machines and address decoders. Many FPGAs include several phase-locked loops (PLLs). These PLLs are used to multiply, divide, and adjust high-speed clock signals. Similar to microprocessors used in PCs, many new FPGAs use a lower 1.5 to 3 Volt internal core power supply. To easily interface to external processor and memory chips, new FPGAs feature selectable I/O standards on I/O pins.

High-speed hardware multipliers and multiply accumulators (MACs) are also available in FPGA families targeted for multiply intensive DSP and graphics applications. Several FPGAs from Altera and Xilinx are available with commercial internal RISC microprocessor intellectual property (IP) cores. These include the Nios, ARM, Microblaze, and PowerPC. The Nios and Microblaze processors are an HDL model that is synthesized using the FPGA's standard logic elements. The ARM, and PowerPC are commercial IP cores with custom VLSI layouts. These new devices are a hybrid that contains both ASIC and FPGA features. Several processors can be implemented in a single FPGA.

These FPGAs come with additional software tools for the processor, including C/C++ compilers. Some processor cores are available with a small operating system kernel. These new large FPGAs with a microprocessor IP core are targeted for System on-a-Chip (SOC) applications. When an FPGA is used for SOC applications it is also called System on-a-Programmable Chip (SOPC).

On many of the largest FPGAs, redundant rows of logic elements are included to increase yields. As any VLSI device gets larger the probability of a manufacturing defect increases. If a defective logic element is found during initial testing, the entire row is mapped out and replaced with a spare row of logic elements. This operation is transparent to the user.

3.9 For additional information

This short overview of programmable logic technology has provided a brief introduction to FPGA architectures. Altera and Xilinx have the largest market share of current FPGA vendors. Additional CPLD and FPGA manufacturers include Lattice, Actel, Atmel, Quicklogic, and Cypress. Actel, Quicklogic, and Cypress have one-time programmable FPGA devices. These devices utilize antifuse programming technology. Antifuses are open circuits that short circuit or have low impedance only after programming. Trade publications such as *Electronic Design News* periodically have a comparison of the available devices and manufacturers.

The March 2007 issue of *IEEE Computer* contains several articles on recent developments in FPGA tools and reconfigurable computing using FPGAs.

Altera MAX 7000, Cyclone, Cyclone II, and Stratix II family data manuals with a more in-depth explanation of device hardware details are available free online at Altera’s website, http://www.altera.com.

For other examples of FPGA architectures, details on the Xilinx Spartan 3E and Virtex II Pro families can be found at http://www.xilinx.com.

An introduction to the mathematics and algorithms used internally by digital logic CAD tools can be found in *Synthesis and Optimization of Digital Circuits* by Giovanni De Micheli, McGraw-Hill, 1994 and *Logic Synthesis and Verification Algorithms* by Hactel and Somenzi, Springer Publishers, 1996. *The Design Warrior’s Guide to FPGAs* by Clive Maxfield, Elsevier, 2004 contains an overview of commercial FPGA devices and commercial EDA tool flows for FPGA design.

3.10 Laboratory Exercises

1. Show how the logic equation (A AND NOT(B)) OR (C AND NOT(D)) can be implemented using the following:

   A. The PLA in Figure 3.3
   B. The LUT in Figure 3.9

   Be sure to include the PLA fuse pattern and contents of the LUT.

2. Examine the compiler report file and use the chip editor to explain how the OR-gate design in the tutorial in Chapter 1 was mapped into the Cyclone device.

3. Retarget the design from Chapter 1 to a MAX 7000S device. Examine the compiler report file and use the chip editor to explain how the OR-gate design in the tutorial in Chapter 1 was mapped into the MAX device.

4. Show how the logic equation (A AND NOT(B)) OR (C AND NOT(D)) can be implemented in the following:

   A. A MAX Logic Element
   B. A Cyclone Logic Element
   C. An XC4000 CLB

   Be sure to include the contents of any LUTs required and describe the required mux settings.

5. Using data sheets available on the web, compare and contrast the features of newer generation FPGAs such as Altera’s Cyclone III and Stratix III, and Xilinx’s Virtex II and Virtex 4 families.