**Field Programmable Gate Arrays (FPGAs)**

**Introduction**

Simulations and prototyping have been a very important part of the electronics industry since a very long time now. Before heading in for the actual fabrication of a dedicated hardware, everyone would want to be sure that what they are making will work the way they want it to. Over all these years while electronics companies offered dedicated hardware in their products, it was not possible for the end user to reconfigure them to his own needs. This need led to the growth of a new market segment of customer configurable Field Programmable integrated circuits called Field Programmable Gate Arrays or FPGAs.

**History**

The FPGA share a common history with most Programmable Logic Devices. The first of this kind of devices was the Programmable Read Only Memory. Further driven by need of specifically implementing logic circuits, Philips invented the Field-Programmable Logic Array (FPLA) in the 1970s. This consisted of two planes, a programmable wired AND-plane and the other as wired OR. It could implement functions in the Sum of Products form.

To overcome difficulties of cost and speed, Programmable Array Logics were developed which had only one programmable ‘AND’ plane fed into fixed OR gates. PALs and PLAs along with other variants are grouped as Simple Programmable Logic Devices (SPLDs). In order to cater to growing technological demands, SPLDs were integrated onto a single chip and interconnects were provided to programmably connect the SPLD blocks. These were called Complex PLDs and were first pioneered by Altera, the first in the family being Classic EPLDs and then, MAX series.

Then another class of Electronic devices, Mask-Programmable Gate Arrays consisting of transistor arrays which could be connected using custom wires motivated the design of the FPGAs. Transistors gave way to Logic Blocks and the customization could now be performed by the user on the field and not in the manufacturing lab. The credit to develop the first commercially viable FPGA goes to Xilinx co-founders Ross Freeman and Bernard Vonderschmitt. The XC2064 was invented in 1985 consisting of 64 Configurable Logic Blocks with 3 Look Up Tables. It was in late 1980s when Steve Casselman’s proposed experiment to implement a computer with 6,00,000 reprogrammable gates found sponsors in US Naval Surface Warfare department and later a patent in 1992.

By the end of 1990, a lot of competition sprung up in manufacturing FPGAs when Xilinx’s market share started to decline. Players like Actel, Altera, Lattice, QuickLogic, Cypress, Lucent and SiliconBlue started entering this field and carving their niche in the world FPGA Market along with Xilinx, as FPGA started gaining acceptance in applications like Digital Signal Processing and Telecommunications. In 1997, Adrian Thompson succeeded in merging a genetic algorithm technology with FPGA and started a new age of Evolvable hardware.

FPGA/mohd uzir kamaluddin/october 2017
FPGA’s can be considered to be building bricks which allow desired customization of the hardware. These are a special form of PLDs with higher densities and with increased capability of implementing functionality in a shorter time span using CAD. The FPGA’s are available in various flavours based on the programming technology used. These may be programmed using:

1. Antifuse Technology, which can be programmed only once. Devices manufactured by QuickLogic are examples of this type. Configuration is done by burning a set of fuses. These act as replacements for Application Specific ICs (ASIC) and used in places where protection of intellectual property is top priority.

2. Flash Technology based Programming, like devices from Actel. The FPGA may be reprogrammed several thousand times, taking a few minutes in the field itself for reprogramming and has non-volatile memory.

3. SRAM Technology based FPGAs, the currently dominating technology offering unlimited reprogramming and very fast reconfiguration and even partial reconfiguration during operation itself with little additional circuitry. Most companies like Altera, Actel, Atmel and Xilinx manufacture such devices.
Defining the Parts of an FPGA

Every FPGA chip is made up of a finite number of predefined resources with programmable interconnects to implement a reconfigurable digital circuit and I/O blocks to allow the circuit to access the outside world.

The Different Parts of an FPGA

FPGA resource specifications often include the number of configurable logic blocks, number of fixed function logic blocks such as multipliers, and size of memory resources like embedded block RAM. Of the many FPGA chip parts, these are typically the most important when selecting and comparing FPGAs for a particular application.

The configurable logic blocks (CLBs) are the basic logic unit of an FPGA. Sometimes referred to as slices or logic cells, CLBs are made up of two basic components: flip-flops and lookup tables (LUTs). Various FPGA families differ in the way flip-flops and LUTs are packaged together, so it is important to understand flip-flops and LUTs.

Flip-Flops

Flip-flops are binary shift registers used to synchronize logic and save logical states between clock cycles within an FPGA circuit. On every clock edge, a flip-flop latches the 1 or 0 (TRUE or FALSE) value on its input and holds that value constant until the next clock edge.

FPGA/mohd uzir kamaluddin/october 2017
Lookup Tables (LUTs)

Much of the logic in a CLB is implemented using very small amounts of RAM in the form of LUTs. It is easy to assume that the number of system gates in an FPGA refers to the number of NAND gates and NOR gates in a particular chip. But, in reality, all combinatorial logic (ANDs, ORs, NANDs, XORs, and so on) is implemented as truth tables within LUT memory. A truth table is a predefined list of outputs for every combination of inputs.

Irrespective of the different manufacturers and slightly different architectures and feature sets, most of the FPGA’s have a common generic approach. The main component blocks of any FPGA are a flexible programmable ‘Configurable Logic Block’ (CLB), surrounded by programmable ‘Input/Output Blocks’ with a hierarchy of routing channels interconnecting various blocks on the board. Additionally, these may consist of Clock DLLs for clock distribution and control and Dedicated Block RAM memories.

Configurable Logic Block

The basic building block of a Configurable Logic Block is the logic cell. A logic cell may consist of an input function generator, carry logic and a storage element. The function generators are implemented as Look Up Tables depending on the input. For example, a Xilinx Spartan II has 4 inputs LUT where each LUT can provide a 16X1 bit Synchronous RAM which can be further multiplexed using multiplexers. An LUT may also be used as a Shift register which is used to capture burst-mode data. The storage elements may be

FPGA/mohd uzir kamaluddin/october 2017
used as edge sensitive flip-flops or level sensitive latches. The arithmetic logic includes an XOR gate for full adder operation along with dedicated carry logic lines. The figure below shows an FPGA slice:

Input/Output Block

This block features inputs and outputs supporting a wide range of signaling standards and interfaces. A basic Input/Output block is shown below:
The buffers in the Input and output paths route the input and output signals to the internal logic and the output pads either directly or via a flip-flop. The buffer can be set to conform to various supported signaling standards which might even be user defined and externally set.

Routing Matrix

In any assembly line it is often the slowest segment which sets the overall production rate. Much in the same way, it is the route that takes the longest delay that eventually determines the performance of the entire electronic system. Thus routing algorithms are brought into place for the design of the most efficient paths to deliver optimum performance. Routing is on various levels like Local, between LUTs, flip-flops and the General Routing Matrix, General Purpose Routing between various CLBs, I/O Routing between I/O Blocks and CLBs, Dedicated Routing for a certain classes of signals for maximizing performance and Global Routing for distributing clocks and other signals with very high fanout.

The Spartan 3E starter kit

The key features of the Spartan-3E starter kit board are:

- Xilinx XC3S500E Spartan-3E FPGA
  1. Up to 232 user I/O pins
  2. 320 pin FBGA package
  3. Over 10,000 logic cells