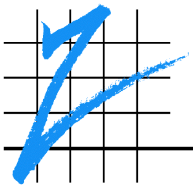

Introduction to FPGA Design

By Bob Zeidman
President
Zeidman Consulting
bob@ZeidmanConsulting.com
www.ZeidmanConsulting.com

Embedded Systems Conference Europe
1999
Classes 304, 314

1. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are becoming a critical part of every system design. Many vendors offer many different architectures and processes. Which one is right for your design? How do you design one of these so that it works correctly and functions as you expect in your entire system? These are the questions that this paper sets out to answer.

The first sections of this paper deals with the internal architecture and characteristics of these devices. Programmable logic devices are described in an overview, leading up to a detailed description of the Field Programmable Gate Array. The various architectures of these devices are examined in detail along with their tradeoffs, which allow you to decide which particular device is right for your design.

The next sections of this paper is about the design flow for an FPGA-based project. This section describes the phases of the design that need to be planned. This allows a designer or project manager to allocate resources and create a schedule.

The final sections of this paper discuss in detail, the design, simulation, and testing issues that arise when designing an FPGA. Understanding these issues will allow you to design a chip that functions correctly in your system and will be reliable throughout the lifetime of your product.

2. THE MASKED GATE ARRAY ASIC

An Application Specific Integrated Circuit, or ASIC, is a chip that can be designed by an engineer with no particular knowledge of semiconductor physics or semiconductor processes. The ASIC vendor has created a library of cells and functions that the designer can use without needing to know precisely how these functions are implemented in silicon. The ASIC vendor also typically supports software tools that automate such processes as synthesis and circuit layout. The ASIC vendor may even supply application engineers to assist the ASIC design engineer with the task. The vendor then lays out the chip, creates the masks, and manufactures the ASICs.

The gate array is an ASIC with a particular architecture that consists of rows and columns of regular transistor structures. Each basic cell, or gate, consists of the same small number of transistors which are not connected. In fact, none of the transistors on the gate array are initially

connected at all. The reason for this is that the connection is determined completely by the design that you implement. Once you have your design, the layout software figures out which transistors to connect. First, your low level functions are connected together. For example, six transistors could be connected to create a D flip-flop. These six transistors would be located physically very close to each other. After your low level functions have been routed, these would in turn be connected together. The software would continue this process until the entire design is complete. This row and column structure is illustrated in Figure 1.

The ASIC vendor manufactures many unrouted die which contain the arrays of gates and which it can use for any gate array customer. An integrated circuit consists of many layers of materials including semiconductor material (e.g., silicon), insulators (e.g., oxides), and conductors (e.g., metal). An unrouted die is processed with all of the layers except for the final metal layers that connects the gates together. Once your design is complete, the vendor simply needs to add the last metal layers to the die to create your chip, using photomasks for each metal layer. For this reason, it is sometimes referred to as a Masked Gate Array to differentiate it from a Field Programmable Gate Array.

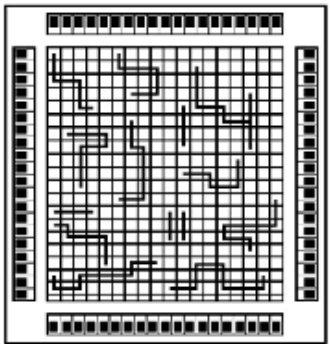

Figure 1 Masked Gate Array Architecture

3. THE EVOLUTION OF PROGRAMMABLE DEVICES

Programmable devices have gone through a long evolution to reach the complexity that they have today. The following sections give an approximately chronological discussion of these devices from least complex to most complex.

3.1 Programmable Read Only Memories (PROMs)

Programmable Read Only Memories, or PROMs, are simply memories that can be inexpensively programmed by the user to contain

a specific pattern. This pattern can be used to represent a microprocessor program, a simple algorithm, or a state machine. Some PROMs can be programmed once only. Other PROMs, such as EPROMs or EEPROMs can be erased and programmed multiple times.

PROMs are excellent for implementing any kind of combinatorial logic with a limited number of inputs and outputs. For sequential logic, external clocked devices such as flip-flops or microprocessors must be added. Also, PROMs tend to be extremely slow, so they are not useful for applications where speed is an issue.

3.2 Programmable Logic Arrays (PLAs)

Programmable Logic Arrays (PLAs) were a solution to the speed and input limitations of PROMs. PLAs consist of a large number of inputs connected to an AND plane, where different combinations of signals can be logically ANDed together according to how the part is programmed. The outputs of the AND plane go into an OR plane, where the terms are ORed together in different combinations and finally outputs are produced. At the inputs and outputs there are typically inverters so that logical NOTs can be obtained. These devices can implement a large number of combinatorial functions, though not all possible combinations like a PROM can. However, they generally have many more inputs and are much faster.

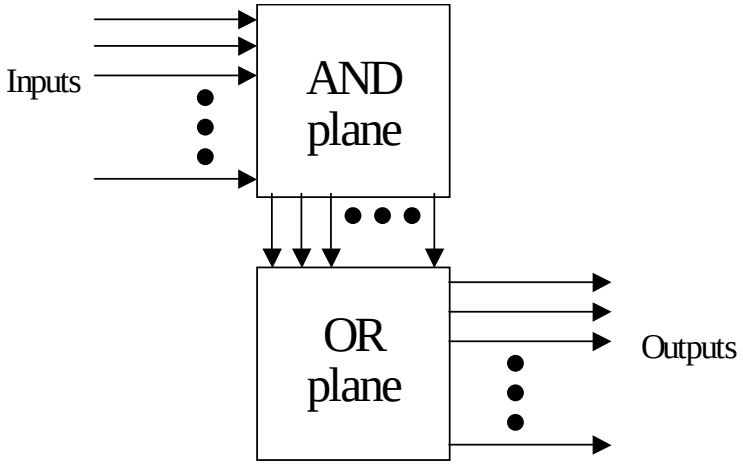

Figure 2 PLA Architecture

3.3 Programmable Array Logic (PALs)

The Programmable Array Logic (PAL) is a variation of the PLA. Like the PLA, it has a wide, programmable AND plane for ANDing inputs together. However, the OR plane is fixed, limiting the number of terms that

can be ORed together. Other basic logic devices, such as multiplexers, exclusive ORs, and latches are added to the inputs and outputs. Most importantly, clocked elements, typically flip-flops, are included. These devices are now able to implement a large number of logic functions including clocked sequential logic need for state machines. This was an important development that allowed PALs to replace much of the standard logic in many designs. PALs are also extremely fast.

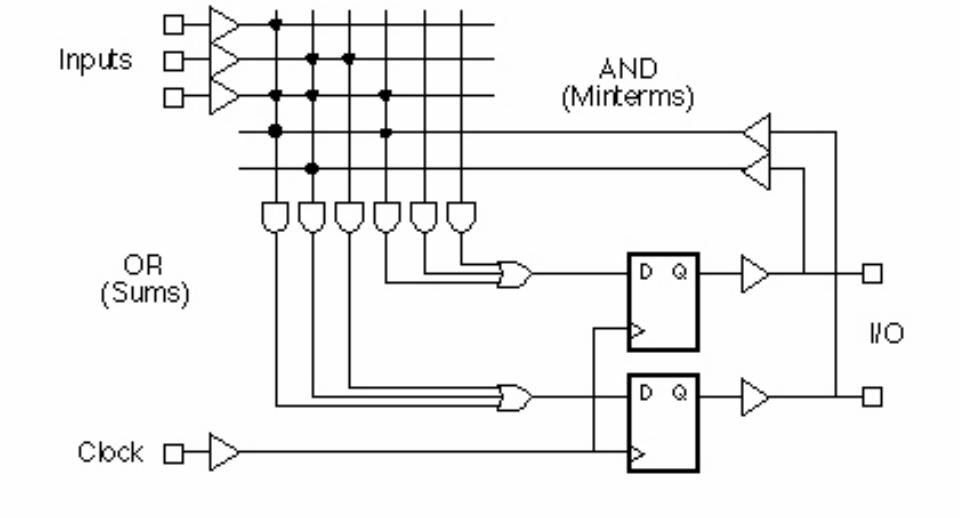

Figure 3 PAL Architecture

3.4 CPLDs and FPGAs

Ideally, though, the hardware designer wanted something that gave him or her the flexibility and complexity of an ASIC but with the shorter turn-around time of a programmable device. The solution came in the form of two new devices - the Complex Programmable Logic Device (CPLD) and the Field Programmable Gate Array. As can be seen in Figure 4, CPLDs and FPGAs bridge the gap between PALs and Gate Arrays. CPLDs are as fast as PALs but more complex. FPGAs approach the complexity of Gate Arrays but are still programmable.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.