Journal on Intelligent Electronic Systems, Vol.1, No.1, November 2007

Application Of Programmable Logic Devices In Safety Systems

S. Ilango Sambasivan & M. K. Misra
Electronics & Instrumentation Division, Indira Gandhi Centre for Atomic Research,
Kalpakkam-603 102, Tamil Nadu, INDIA.
E-mail : sis@igcar.gov.in

Abstract

The evolution of Microelectronics has made the application of Programmable Logic Devices (PLDs) inevitable in modern digital circuit designs and when they are deployed in Safety Systems, their Reliability and Safety need to be proved beyond doubt. The design will have to meet the stringent requirements. This paper provides a review of commercially available PLDs and the design methodologies to be adopted while deploying them in safety systems.

Key words : PLD, error detection, correction codes

I. INTRODUCTION

Safety Systems are those systems whose failure or failure to operate may cause injuries to system, personnel and the environment. e.g. an aircraft flight control or Nuclear reactor control system. Such systems should be highly reliable and safe. Hence, the selection and application of Programmable Logic Devices (PLDs) for Safety Critical Applications have to be made judiciously. A Programmable Logic Device or PLD is a digital device but unlike a conventional IC which has a fixed function, a PLD has an undefined function at the time of manufacture. Before it can be used in a circuit it must be programmed. The Umbrella term "PLDs" subdivides into several categories: SPLDs/PALs, CPLDs and FPGAs. Refer Fig. 1. All are Programmable Logic Devices, though their internal architecture implementations differ. The PLD market consists of low and high capacity devices. Low capacity devices, called simple PLDs, typically contain fewer than 600 usable gates and include products such as PALs and GALs. Simple PLDs are manufactured using CMOS technology offering EPROM, EEPROM, and FLASH memory elements. High Capacity Programmable Logic Device (HCPLD) typically contains more than 600 usable gates, and includes both CPLDs and FPGAs. HCPLDs are manufactured using CMOS technology with EPROM, EEPROM, FLASH, SRAM, and antifuse options. HCPLDs can be differentiated by their interconnect structure: CPLDs use continuous interconnect structures, while FPGAs use segmented interconnect structures (Refer Fig. 8).

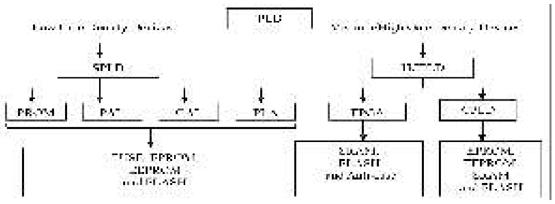

Fig. 1. Programmable logic devices classification

II. PROGRAMMABLE LOGIC DEVICES (PLDs)

Programmable Logic Devices (PLDs) are standard ICs with configurable logic and flip-flops linked together with programmable interconnects. Memory cells control and define the function that the logic performs and how the various logic functions are interconnected. Though various devices use different architectures, all are based on this fundamental principle. User through programming defines the functionality of the PLD.

Simple programmable logic devices (SPLDs)

SPLDs are the smallest and consequently the least-expensive form of programmable logic. An SPLD is typically comprised of 4 to 22 macrocells and can typically replace a few 7400-series TTL devices. One macrocell is typically equivalent to about 30 gates. Each of the macrocells is an array of AND and OR gates and typically fully connected to the others in the device. Refer Figure 2. Most SPLDs use either fuses or non-volatile memory cells such as EPROM, EEPROM, or FLASH to define the functionality. Programmable Read Only Memory (PROM), Programmable Array Logic (PAL), Generic Array Logic (GAL) & Programmable Logic Array (PLA) are SPLDs.

PROM has a fixed AND plane and a programmable OR plane; PAL has a programmable AND plane and a fixed OR plane; PLA has both programmable AND and OR planes; GAL devices have re-programmable AND array and fixed OR array and programmable O/P logic.

Complex programmable logic devices (CPLDs)

CPLDs provide a range of high-density programmable logic devices. It's an array of SPLDs. The architecture is based on a number of logic blocks that are connected by a Programmable Interconnect Matrix (PIM). Refer Fig. 3. The PIM distributes signals from the logic block outputs and all input pins to the logic block inputs. A typical CPLD is equivalent to 2 to 64 SPLDs. A CPLD typically contains from tens to a few hundred macrocells. A group of eight to 16 macrocells forms logic block. The macrocells within a logic block are usually fully connected. If a device contains multiple logic blocks, then the logic blocks are further interconnected by PIM. CPLDs are manufactured using one of four-process technologies EPROM, EEPROM, SRAM or FLASH. EPROM-based CPLDs are usually one-time programmable (OTP) unless they are in an UV-erasable windowed package.

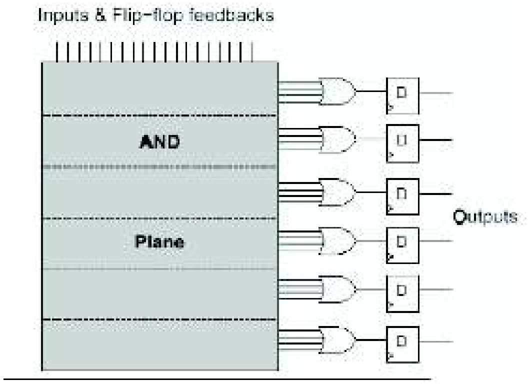

Fig. 2. Structure of a PAL

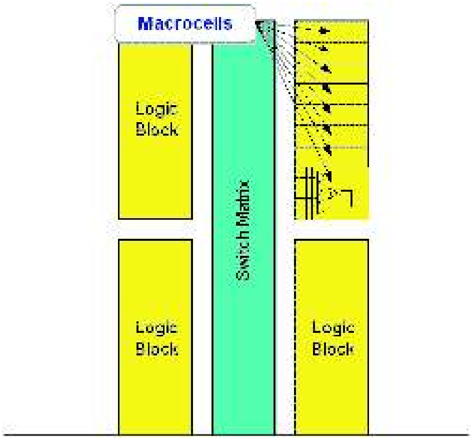

Fig. 3. Sturcture of a CPLD

Field programmable gate arrays (FPGAs)

FPGAs are distinct from SPLDs and CPLDs and typically offer the highest logic capacity. FPGA is usually just larger and more complex than SPLD & CPLD. A generic description of an FPGA is a programmable logic device with an internal array of logic blocks, surrounded by a ring of programmable input/output blocks, connected together via programmable interconnect Refer Figure 4. There are three basic categories of FPGAs in the market today SRAM based FPGAs, FLASH based FPGAs and antifuse based FPGAs. SRAM, FLASH based FPGAs are inherently re-programmable, even in-system. In contrast, anti-fuse devices are one-time programmable (OTP). Once programmed, they cannot be modified.

III. PROGRAMMING TECHNOLOGIES

EPROM and EEPROM technology

Applying a programming high voltage V_{PP} (usually greater than 12V) to the drain of the n-channel EPROM transistor programs the EPROM cell. A high electric field causes electrons flowing toward the drain to move so fast they "jump" across the insulating gate oxide where they are trapped on the bottom, floating gate. This phenomenon is called hot-electron injection or avalanche injection. Electrons trapped on the floating gate raise the threshold voltage of the n-channel EPROM transistor. Once programmed, an n-channel EPROM device remains off even V_{DD} applied to the top gate. Refer Fig. 5.

SRAM technology

The configuration cell drives the gates of other transistor on the chip-either turning pass transistor or transmission gates on to make the connection or off to break a connection. Refer Fig. 6.

Antifuse technology

An antifuse is normally an open circuit until a programming current (about 5mA) is passed through it. In a poly-diffusion antifuse the high current density causes a large power dissipation in a small area, which melts a thin insulating dielectric between polysilicon and diffusion electrodes and forms a thin (about 20 nm in diameter), permanent, and resistive silicon link. Refer Fig. 7.

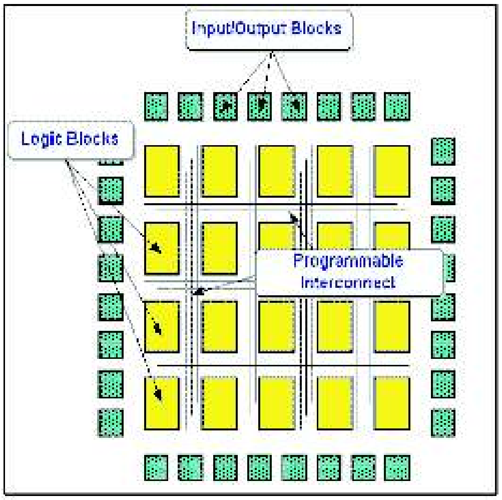

Fig. 4. Structure of a FPGA

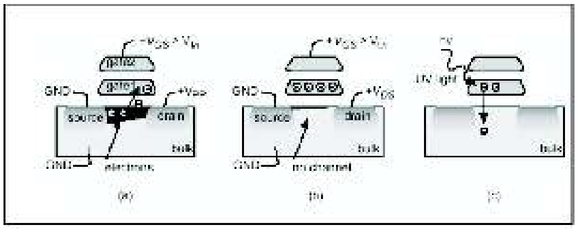

Fig. 5. An EPROM Transistor

a. With High Voltage (>12V) Programming Volatge
b. Electrons Stuck on gate 1
c. Ultra Violet (UV) light provides enough energy for electrons stuck on gate 1 to “jump” to the bulk

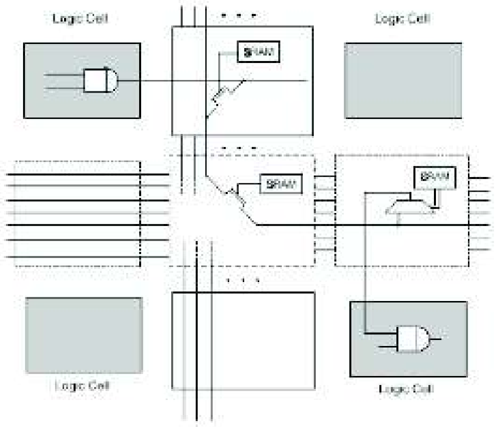

Fig. 6. SRAM Controlled programmable switch

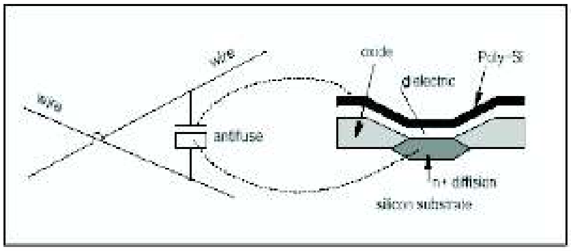

Fig. 7. Actel antifuse structure

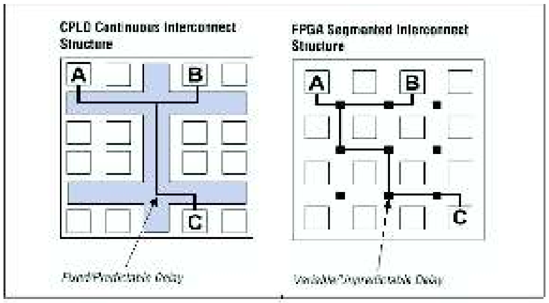
CPLD Continuous Interconnect Structure
A
B
C
Fixed/Predictable Delay
FPGA Segmented Interconnect Structure
A
B
C
Variable/Unpredictable Delay

Fig. 8. CPLD vs. FPGA Routing scheme

Table 1. Summary of programming technologies

Name of the Technology	Re-programmable	Volatile	Process Technology
Fuse	No	No	Bipolar
EPROM	Yes, Out of circuit	No	UVCMOS
EEPROM	Yes, In Circuit	No	EECMOS
FLASH	Yes, In Circuit	No	EECMOS
SRAM	Yes, In Circuit	Yes	CMOS
Antifuse	No	No	CMOS

Table 2. PLD vendors vs. Programming technologies for FPGAs, CPLDs and SPLDs

Vendor	FPGA	CPLD	SPLD
Altera	SRAM	EEPROM, EPROM	EPROM
Xilinx	SRAM	FLASH	-
Lattice	SRAM	EEPROM	EEPROM
Actel	FLASH, Antifuse	-	-
Cypress	-	EEPROM, FLASH, EPROM, SRAM	FLASH, EPROM
Atmel	SRAM	FLASH, EPROM	FLASH, EPROM
Quick Logic	Antifuse	-	-
TI	-	-	Fuse (Bi-Polar)
ICT	-	EEPROM	EEPROM

Table. 3 Reliability data: FIT rate for some of the PLD families

Technology	Fit Rate	Source
0.45 Micro m CMOS Antifuse FPGA from Actel (MX Family)	13.16	Reliability Report Revision 5, April 2007 From Actel Corporation
0.25 Micro m Infineon FLASH CMOS FPGA from Actel (ProAsic A500 K Family)	25.30	
0.50 Micro m CMOS SRAM FPGA from ALTERA (FLEX)	7.7	Reliability Report 45 Q2 2007 from Altera
0.15 Micro m CMOS SRAM FPGA from ALTERA (APEX)	81.45	
0.50 Micro m CMOS EEPROM CPLD from ALTERA (MAX)	7.15	
0.18 Micro m CMOS FLASH CPLD from ALTERA (MAX-II)	16.1	

IV. RADIATION EFFECTS

High energy particles contained in cosmic galactic rays enter the earth's atmosphere and collide with atoms of atmospheric gases. These collisions produce a wide variety of sub-atomic particles, many of which recombine quickly. However, a significant quantity of high-energy neutrons are also produced by these collisions. Neutrons possess no electrical charge, and do not recombine; instead, they are slowly attenuated by the atmosphere. While the greatest quantities of neutrons (called the neutron flux density) occur at an altitude of 60,000 feet, a significant number of neutrons penetrate the atmosphere and reach the earth's surface. These high-energy neutrons can cause flip-flops and memory cells in modern semiconductor electronics to change state. Given this discovery, the effects of neutrons on programmable logic devices, which use memory cells to determine their functionality, is a major concern.

Additionally, plastic package molding compounds contain tiny quantities of radioactive isotopes which emit alpha particles. Many of these alpha particles are sufficiently energetic to cause upsets in data flip-flops and memory cells, including the configuration SRAM memory in programmable logic. Ionizing radiation can cause unwanted effects in semiconductor devices (Refer Fig. 9). Energetic Protons, Neutrons, Heavy Ions, and Alpha particles can strike sensitive regions of the transistor, causing various failures, or Single-Event Effects (SEE), such as:

- Single-Event Upsets (SEUs)
- Single-Event Transients (SETs)
- Single-Event Functional Interrupt (SEFI)

These terms and the failure modes they describe are explained in detail in the Conference Papers below. While SEUs, SETs, and SEFIs cannot be prevented, their effects in CMOS re-configurable logic devices can be mitigated.

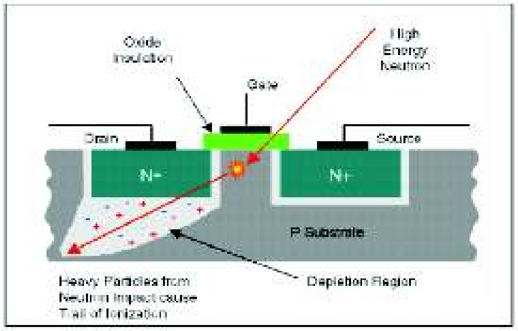

Fig. 9. Interaction of a high-energy neutron and a silicon integrated circuit

Sensitivity to Radiation Effects is dependent on many factors, including transistor geometry and cell layout. Certain CMOS technologies, such as SRAM, are sensitive to SEE.

The SEUs caused by neutrons inside integrated circuits can occur in memory cells or flip-flops. Most attention has been focused on how to mitigate against data corruption as a result of these SEUs, with techniques such as error detection and correction codes (EDAC) and triple-module redundancy (TMR) being used to detect and overcome SEU-induced soft errors.

In some systems SEU detection and correction alone can achieve an acceptable level of reliability. However, for applications where an even higher level of reliability is needed, or simply that any interrupt in service is unacceptable, SEU mitigation techniques may be applied. A good SEU mitigation technique should filter out the effects of upsets, during their short existence, as well as filter out the results of transient upsets or other SEFI effects. A commonly known method for SEU mitigation is “triple module redundancy with voting.” This mitigation scheme uses three identical logic circuits performing the same task in tandem with corresponding outputs compared through a majority vote circuit.

Radiation testing results has shown that antifuse and Flash-based devices are not subject to loss of configuration due to upsets caused by atmospheric neutrons or alpha particles emitted from packaging materials. This makes them eminently suitable for applicationsboth ground-based and airbornewhere high reliability is imperative.

V. VERIFICATION OF HDL-BASED DESIGNS

As the average gate count for designs now approaches or exceeds one million, functional verification has become the main bottleneck in the design process. As designs grow more complex, the verification problems increase exponentially. To eliminate the verification bottleneck, verification engineers have tried incorporating new methodologies and technologies. While various methodologies have evolved, including formal methods, simulation is still the preferred method for verification. High level Verification Languages (HVLs) like *e* / VERA have emerged to solve the functional verification bottleneck. e can be used to construct components to do the following functions in a verification environment.

Generation *e* automates the generation of stimuli. Input stimuli are generated based on the constraints provided by the verification environment.

Driving Stimulus After the test vectors are generated, they must be driven on to the Device Under Test (DUT). *e* provides a simulator interface and the necessary mechanism to drive the DUT.

Collecting Output After the stimulus is applied to the DUT, output is produced from the DUT. This output must be collected and checked. *e* provides a simulator interface and the necessary mechanism to receive data from the DUT.

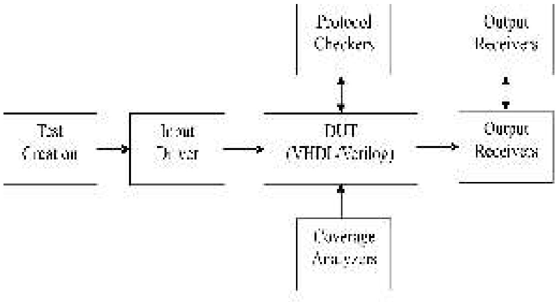

Fig. 10. Components of a verification environment

Data Checking After the output data is received from the DUT, the data must be checked. Data value checks compare the output data values against the expected data. Temporal assertions monitor the functional protocol at important interfaces. Temporal checking constructs are used to build protocol monitors.

Coverage Functional coverage tells the verification engineer if the test plan goats have been met. There are three types of coverage: basic item coverage, transition item coverage, and cross coverage. Basic item coverage tells the engineer if all the legal values of an interesting variable have been covered. Transition item coverage tells the engineer if all legal transitions of a state machine have been covered. Cross coverage allows the engineer to examine the cross product of two or more basic or transition items to check if all interesting combinations of basic and transition items have been covered. Figure-9 shows the components of a verification environment.

VI. CONCLUSION

PLDs are appropriate (and even desirable) to use in safety-critical systems, provided, proper steps are taken to avoid failures. The selection of a particular programming technology has to be made judiciously and proper design methodology has to be followed to address design related problems.

Random failures can be avoided by the selection of proper device technologies. Studies have shown that SRAM based devices are prone to soft errors. On the other hand, antifuse, EEPROM and FLASH based devices are immune to soft errors and radiation effects.

Safety Logic with Fine Impulse Test (SLFIT) for shutdown system-1 of Prototype Fast Breeder Reactor (PFBR) is a safety critical system. This system has the important function of a safe shutdown of the nuclear reactor in the event of any malfunctioning and has been designed using anti-fuse technology based devices (One Time Programmable devices). These devices are highly reliable since during programming the device, the internal structure of the device gets physically changed which can not be altered again.

ACKNOWLEDGEMENT

The authors thank Dr. BaldevRaj, Director, Indira Gandhi Centre for Atomic Research, Kalpakkam for his encouragement and support for this work. We also thank Mr. P. Swaminathan, Director (E&I) Group & Mr. N. Sridhar, EID of our centre for their technical contribution.

REFERENCES

[1] Application-Specific Integrated Circuits By Michael John Sebastian Smith

[2] http://www.actel.com, http://www.altera.com, http://en.wikipedia.org/wiki/Electromigration

[3] http://www.einsite.net/ednmag/archives/1996/101096/21df_07.htm

[4] By John Lach et al, 2005, Accessible Formal Verification for Safety-Critical FPGA Design, Department of Electrical and Computer Engineering University of Virginia at MAPLD /241.

[5] Single-Event Effects in FPGAs,http://www.actel.com/products/rescenter/ser/index.html

[6] FPGA Reliability in Space-Flight and Automotive Applications by Martin Mason, Director, Flash Product Marketing, Actel Corporation and Ken O'Neill, Director, Military and Aerospace Product Marketing, Actel Corporation High-reliability design considerations are fast becoming an art, http://www.fpgajournal.com

[7] Industry Report Sheds Light On FPGA Reliability. iRoC Technologies Study Demonstrates that Neutrons Cause Ground-Level Failures in SRAM FPGAs, http://www.actel.com/company/press/2004pr/iRoC.html, http://www.fpgajournal.com/news_stories2004/Apr/20040419_01.htm

[8] Design Verification with e, by Samir Palnitkar