**15/ENG02/005**

**AKEGH PETER TERKUMA**

**COE506 (DIGITAL SYSTEMS DESIGN WITH VHDL)**

**Question 1:** Define the following acronyms as they apply to digital logic circuits:

• ASIC

• PAL

• PLA

• PLD

• CPLD

• FPGA

Solution

* ASIC (Application Specific Integrated Circuit): This is an IC chip customized for a particular use, rather than intended for general purpose.
* PAL (Programmable Array Logic): PAL is a family of programmable logic device that that provides only single level programmability. It consists of a programmable wired AND-plane that feeds fixed OR-gates. It usually contains flip-flops connected to the OR gate circuits so that the sequential circuits can be realized.

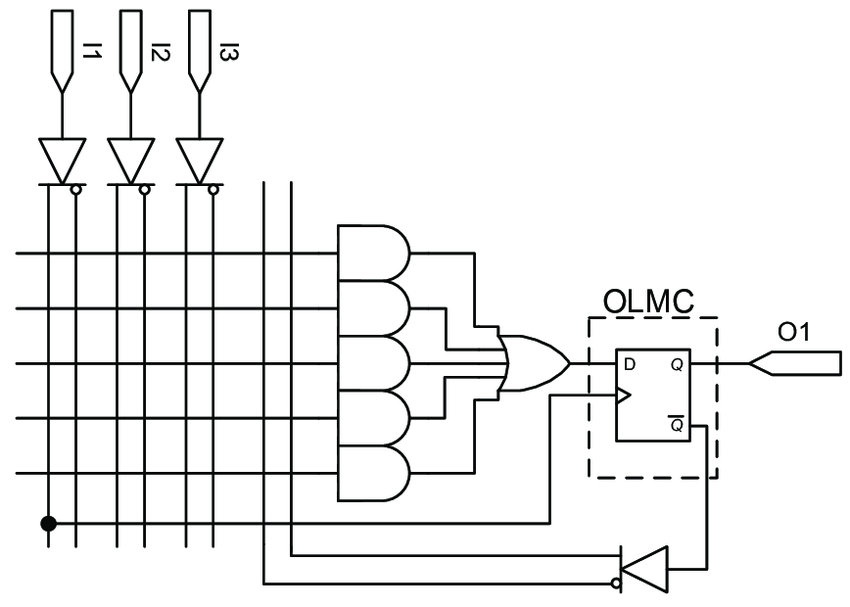


Figure 1.1: Structure of a PAL device

* PLA (Programmable Logic Array): PLA is a type of programmable logic device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output. PLAs differ from PALs and in that both the AND and OR gate planes are programmable.

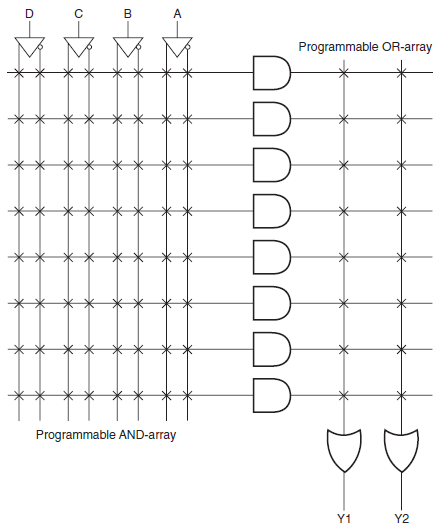


Figure 1.2: Structure of a PLA device

* PLD (Programmable Logic Device): PLD is an electronic component used to build reconfigurable digital circuits. Before the PLD can be used in a circuit it must be programmed (reconfigured) by using a specialized program.
* CPLD (Complex Programmable Logic Devices): CPLDs are multiple PLDs with some programmable interconnect on a single chip. Logically larger CPLDs allow implementation of more complicated designs. CPLDs are chosen over FPGAs when high-performance logic is required; this is because the delay through a CPLD is more predictable and usually shorter.

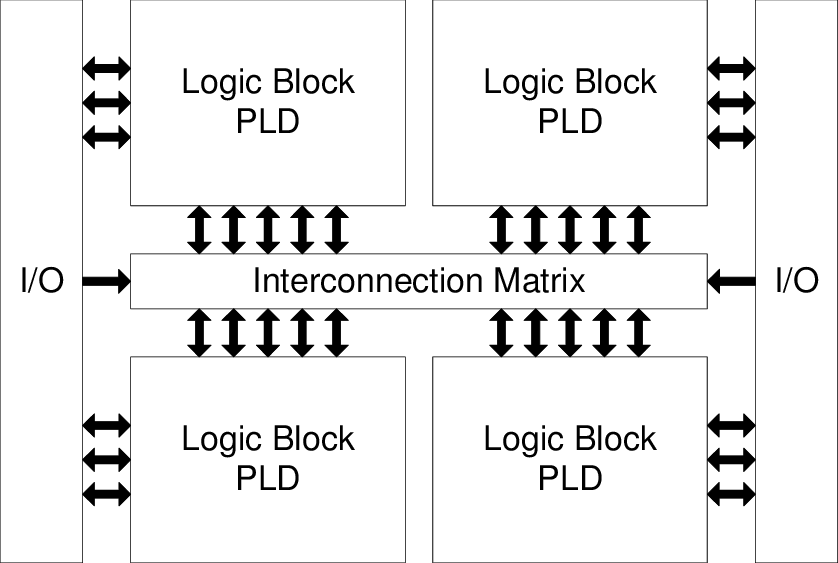


Figure 1.3: Internal Structure of CPLD

* FPGA (Field Programmable Gate Array): An FPGA is an integrated circuit designed to be configured by a customer or a designer after manufacturing. Once it is configured, the internal circuitry is connected in such a way that creates a hardware implementation of the software application. They use dedicated hardware for processing logic and do not have operating system (unlike processors). FPGAs are parallel in nature, hence multiple control loops can run on a single device at different rates.

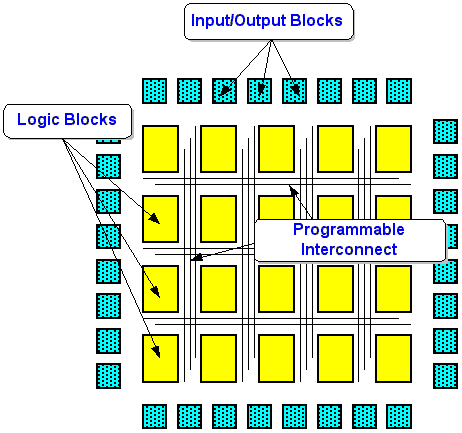


Figure 1.4: FPGA Internal Architecture

**Question 2:** How granularity of logic block influences the performance of an FPGA?

**Solution**

Higher granularity level results in lesser delay between input and output. Increase in granularity of logic block causes the number of levels of logic in critical path to decrease, hence delay in critical path decreases. On other hand, increase in granularity level causes average fan out to increase and also the number of switches as each block has more pins.

**Question 3:** Why would anyone use programmable logic devices (PLD, PAL, PLA, CPLD, FPGA, etc.) in place of traditional "hard-wired" logic such as NAND, NOR, AND, and OR gates? Are there any applications where hard-wired logic would do a better job than a programmable device?

**Solution**

* They are smaller, and consume less power.
* The devices typically offer a higher degree of integration as compared to discrete 7400-series or 4000-series components.
* Some of them are reprogrammable, they can be upgraded without changing the PCB.
* FPGAs have large amount of reprogrammable logic; it is not feasible/ not convenient to build that much logic in discrete components.

1. Yes.

Most of such cases are in embedded systems, are due to is a critical timing constraint. Working on a motor controller where the current in the coils has to be measured controlled, real time, in nanoseconds whilst other measurements and algorithms are being calculated. This is easy to do in hardwire, but impossible to do with current chips in software; hard wired logic will do a better job.

**Question 4:** Some programmable logic devices (and PROM memory devices as well) use tiny fuses which are intentionally "blown" in specific patterns to represent the desired program. Programming a device by blowing tiny fuses inside of it carries certain advantages and disadvantages - describe what some of these are.

**Solution**

The stored program is non-volatile, but it also read-only. This is why fuse programmed devices are sometimes called” OTP” (One-Time Programmable)

OTP (one time programmable) memory is a special type of non-volatile memory that permits data to be written to memory only once. Once the memory has been programmed, it retains its value upon loss of power.

**Question 5**

