

Programmable Logic Applications and Architectures

Digital ICs have increased in complexity at an exponential rate over the last 40 years. This growth has been at a rate predicted by “Moore’s Law” – an observation that digital IC complexity per unit area seems to double every 2 to 3 years.

Moore’s law does not apply to analog ICs. This is because the die area required for an analog IC (e.g. an op-amp) is determined by factors such as its voltage and power rating rather than by the minimum transistor size.

The steady decline in the cost of digital relative to analog electronics has resulted in modern electronic devices implementing almost all functionality using digital rather than analog electronics. The main exception is interface electronics, including power electronics.

Digital ICs can be classified by the number of gates or transistors on an IC (e.g. SSI, LSI and VLSI standing for small, large and very large scale integration). Application-specific ICs (ASICs) are ICs designed for a specific application (e.g. a graphics processor for a video card or the WLAN transceiver in a cell phone) as opposed to being suited to many different applications (e.g. the CPU in a PC or the microcontroller in an appliance). Many ASICs include a general-purpose CPU as well as memory and peripheral interfaces and application-specific components such as graphics or signal processors. These are often called “System on a Chip” (SoC).

Digital ICs are also classified by the feature size of their transistors measured in nanometers. As of 2020, ICs with **7 nm features sizes** were being manufactured.

Due to the cost of the equipment required to manufacture ICs with such small feature sizes, only a handful of companies own fabrication plants (“fabs”) that manufacture digital ICs. Instead, most semiconductor companies are “fabless.” These companies design and sell their ICs but use “fabs” to manufacture them.

IC Manufacturing

IC’s are manufactured on (typically) 300 mm diameter wafers of crystalline silicon. Each wafer is put through dozens of manufacturing steps where dopants are diffused into the silicon and alternating layers of insulating (dielectric) and conductive (metal) materials are deposited to build the circuit. Each step requires a “mask” that is used to expose a photoresist so the processing can be limited to specific areas.

Preparing a mask set for a digital IC is very expensive (e.g. \$10⁶) because of the very small dimensions involved.

Exercise 1: Approximately how many 5×5 mm die fit on a 300 mm wafer?

PLDs

Programmable Logic Devices (PLDs) are Integrated Circuits (ICs) that can be configured after manufacturing to implement different logic functions. Unlike software that consists of a sequence of instructions drawn from a limited instruction set, a PLD’s logic functions can be defined in fine detail and are performed in parallel.

PLDs are often categorized into three major types: PALs (Programmable Array Logic, now obsolete), CPLDs (Complex PLDs) and FPGAs (Field-Programmable Gate Arrays). CPLDs have limited functionality and are often used as “glue logic.” FPGAs are more capable and can often replace ASICs. The different types of PLDs are described below.

The two largest PLD companies are Xilinx and Intel¹, each with about half of the market.

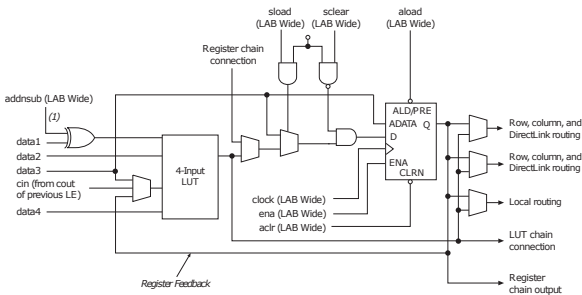
PLD Architectures

CPLD

A CPLD is composed of small logic blocks called Logic Elements (LE, Intel terminology) or Functional

¹Formerly Altera.

Blocks (FB, Xilinx terminology). Each logic block can implement an arbitrary combinational logic function and has one flip-flop at its output. An example is the Intel MAX-V LE:



The Xilinx FB architecture is slightly different and consists of sum-of-products functional blocks with 16 inputs and outputs. Each FB can interconnect with between 16 and 32 other FBs using up to 40 signals.

In both cases a programmable interconnect matrix allows the inputs and outputs of each logic block to be interconnected. By configuring the look-up tables in each logic block and by programming the interconnections between the logic blocks results a wide range of logic functions can be implemented.

Small CPLDs (e.g. a Xilinx XC2C32A with 32 i/o pins and 2 FB or the Intel 5M40 with 40 LE and 64 i/o pins) sell for about \$1 or \$2 in small quantities.

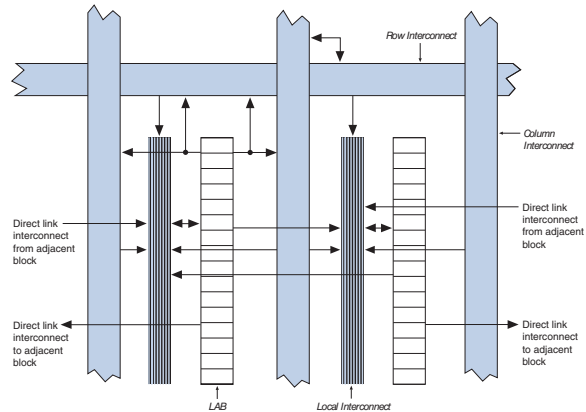
FPGA

Gate arrays were an attempt to bridge the gap between fully custom ICs and PLDs. The idea was that gates would be laid out in predefined locations on the die and only the last few layers of interconnect would need to be customized for each IC thus reducing the number of custom masks needed for each IC and thus the NRE. This approach is no longer popular.

This idea developed into a Field-Programmable Gate Array. An FPGA is a PLD that contains a large number (thousands to hundreds of thousands) of simple logic elements similar to those in an Intel CPLD. Note that modern FPGAs have no gates – the logic for each LE is implemented using small memories called look-up tables (LUTs).

However, an FPGA's interconnection resources are more limited than those in a CPLD. In keeping with the idea that multiple LEs will be combined to form multi-bit logic functions, logic elements can be connected to their neighbours in logic array blocks

(LABs) and these to row and column interconnect buses:



Complex software is required to fit a design into an FPGA and route the signals between logic elements. Propagation delays are harder to predict and some designs may not 'fit' into an FPGA because of insufficient routing resources.

Modern FPGAs include special-purpose components such as RAM, multipliers, PLL clock generators and high-speed serial I/O in addition to general-purpose logic elements.

Most modern FPGAs have enough logic elements and memory that they can be configured with a "soft" CPU (e.g. Altera's Nios and Xilinx's MicroBlaze). This allows the FPGA to include both software and hardware functions. Some FPGAs include a (hardware-based, typically ARM) CPU core for applications that require both an SoC and programmable logic.

Depending on the version, the FPGA might contain between 6 k and 114 k logic elements and between 180 and 530 I/O pins. Smaller FPGAs sell for under \$10 in small quantities. However, large high-performance FPGAs, often used for ASIC prototyping, can cost many thousands of dollars. Due to the high I/O count most FPGAs use ball grid array (BGA) packages.

PLD Configuration

Although most CPLDs have on-board non-volatile configuration memory, most FPGAs use volatile configuration memory which must be reloaded each time the device powers up. The FPGA can load itself from an external, typically serial, EEPROM or it

can be configured through the JTAG interface. On larger systems that include processors the FPGA is often configured by software running on the processor and in this case the FPGA configuration can be changed as part of a firmware update.

Technology Selection

Hardware vs Software

When would you use hardware instead of software?

In most cases a solution can be implemented by writing software that runs on a general-purpose processor or microcontroller. However, there are two common situations where custom digital hardware is required:

- when the required response time is too short for computer software to respond. A typical example is an interface to a peripheral which needs to transfer data high speed.
- when the computational complexity exceeds the capability of a sequential processor. An example is image processing by a graphics processor that processes many pixels in parallel.

PLD vs ASIC

When would you use an programmable logic instead of designing a custom IC?

In most cases the required functionality will already be available in an ASIC or SoC. Examples include many common peripheral interfaces and computational accelerators.

However, in some cases the required hardware function is not available and in this case a decision must be made whether to design a new IC or use an FPGA. The decision depends on the following factors:

- Performance. This includes clock speed and power consumption. An ASIC (or custom full-custom IC) will execute faster and use less power than the same function programmed into an FPGA.
- Sales volume. The NRE (non-recurring engineering) costs for designing the IC (e.g. masks)

must be recovered from sales. If the sales volumes are low then the NRE costs per unit may be too high.

- Time to Market (TTM). It takes many months to design, verify and manufacture a custom IC. If TTM is short then an FPGA may a better option.
- Flexibility. If the functionality of the IC is likely to change then a PLD may be a better option. The design can even be changed in the field after the product has been delivered to the customer.
- Risk. Errors in an IC design will require a “re-spin” of the IC resulting in additional costs and delays. For a small company this can be fatal.

Because of the above considerations, FPGAs tend to be used in non-consumer products that that have relatively low volumes and are relatively cost-insensitive. Examples of typical application areas include telecommunications infrastructure (e.g. base stations, large routers), medical equipment (e.g. imaging) and military/aerospace (e.g. avionics).