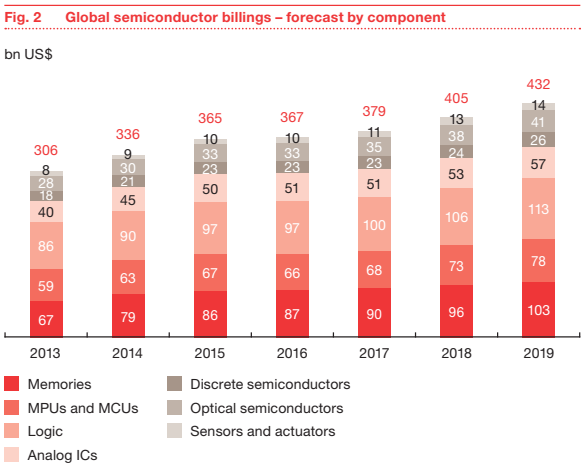
PLD Applications and Architectures

Digital Integrated Circuits

Sales figure can be a useful way to estimate the importance of a technology. World Gross Domestic Product (GDP) is (very) roughly about $100 trillion of which about $30 trillion is industrial output. Of this, about about $350 billion (or about 1%) is semiconductors of which about a quarter is logic ICs. The graph below shows the breakdown of the semiconductor market[1]. FPGAs/CPLD sales represent about $7 billion of the $100 billion logic IC market.



Exercise 1: What are world-wide per-capita PLD sales?

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 transitor 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 interfaces, including power electronics.

Digital ICs can be classified by 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. an MPEG decoder for a DVD player) as opposed to being suited to many different applications (e.g. a microcontroller). A System on a Chip (SoC) is an ASIC that includes a general-purpose CPU as well as memory and peripheral interfaces and application-specific components such as graphics or signal processors.

Digital ICs are also classified by the feature size of the transistors measured in nanometers. As of 2016, ICs with 14 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 crystaline silicon. Each wafer is put through dozens of manufacturing steps where dopants are diffused into the silicon and many alernating 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. $\$10^6$) because of the very small dimensions involved.

Exercise 2: Approximately how many 5x5 mm die fit on a 300 mm wafer?

[1] Data from Gartner via PWC.

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, 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 Altera[2], each with about half of the market. Both are fabless.

PLD Selection

FPGA vs Software

When would you use an FPGA instead of software?

For the majority of applications the required functionality can be achieved by writing software. However, there are two cases where digital logic is required:

- where the required response time is short for computer software to respond. A typical example is a high-speed interface to a peripheral transferring data at GHz rates.
- when the amount of processing required exceeds the capability of a sequential processor. An example is a graphics processor that accelerates image processing by processing many pixels in parallel.

FPGA vs ASIC

When would you use an FPGA 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 graphics 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:

- 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.
- Risk. Errors in an IC design will require a "respin" of the IC resulting in additional costs and delays. For a small company this can be fatal.
- 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.

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).

PLD Architectures

PAL

The first popular programmable logic devices were Programmable Logic Arrays (PALs[3]). Each output implemented a sum-of-products combination of the inputs. The connections to the product terms could be programmed one time (OTP) by "blowing" fuses with a special device programmer.

Many PALs were registered (included flip-flops on output pins, all driven from a common clock).

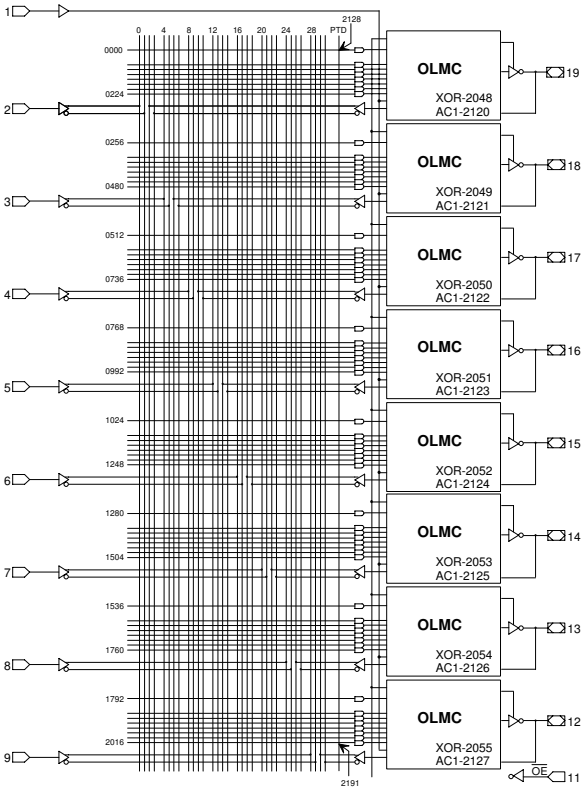
[2] Recently bought by Intel.

[3] PAL is actually a trademark for a specific family of PLAs but it's the name commonly used for these devices.

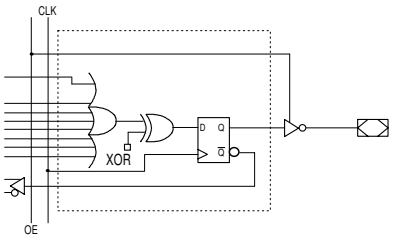
More complex designs required that some of the outputs be fed back to the inputs to implement more than one level of logic. PALs were typically designed "by hand" and "assembled" into a fuse map by relatively simple software.

An example of a PAL is the Lattice 16V8 GAL which is a registered PAL that is electrically-erasable[4]. It features 16 I/O pins of which 8 can act as inputs or registered sum-of-product outputs and 8 are dedicated inputs. These devices are obsolete and have been replaced with CPLDs which can implement more complex logic functions and consume less power.

The schematic of the Lattice GAL16V8[5] is shown below:



The registered PAL output:

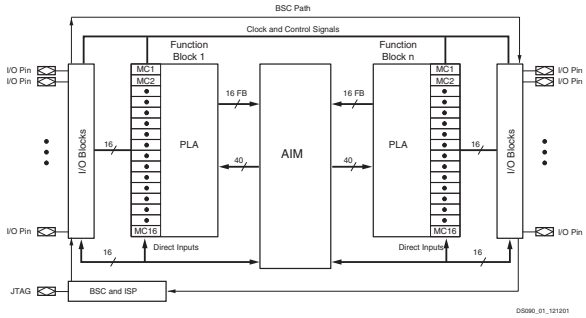


Exercise 3: What component implements the product terms? What component implements the sum of products? Where are the fuses located?

CPLD

A CPLD extends the PAL architecture by adding an interconnect matrix that allows more flexible interconnections between the sum-of-product results. However, unlike the FPGA described below, propagation delays from input to output are fixed and easily predicted.

A typical example is the Xilinx Coolrunner-II CPLD which consists of PAL-like functional blocks (FB) with 16 I/O pins each. Each FB can interconnect with between 16 and 32 other FBs using up to 40 signals. A block diagram of the CPLD is shown below[6]:



A small CPLD (Xilinx XC2C32A, 32 pins, 2 FB) sells for about $1.50 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.

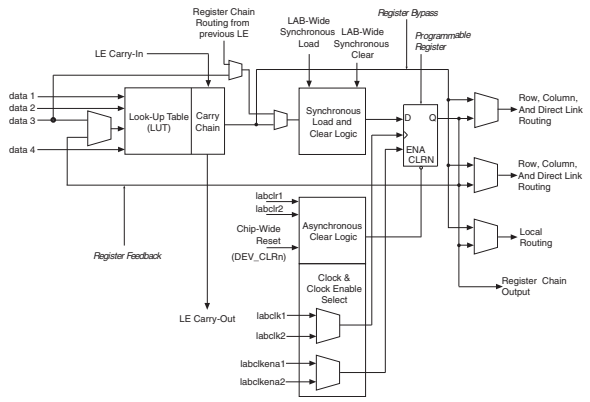
[4] This is the IC which was used to implement the lab test ICs you used in first-year digital electronics labs.

[5] Diagrams from the Lattice 16V8 datasheet.

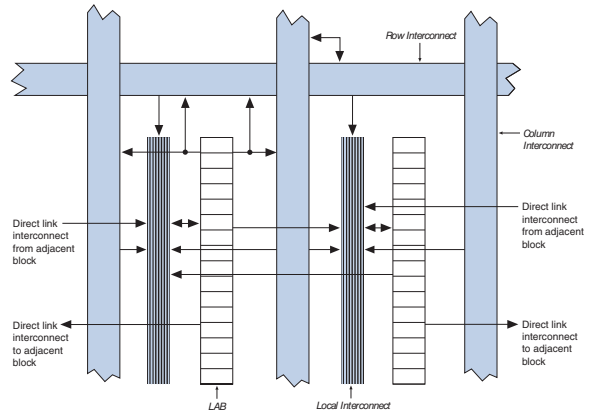
[6] From Xilinx Coolrunner II datasheet.

A Field-Programmable Gate Array is a PLD modelled on the gate array idea. It contains a large number (thousands to hundreds of thousands) of simple logic elements (LEs). Each logic element in a modern FPGA consists of a look-up table (e.g. 4x1 or 4x2) driving an associated one- or two-bit flip-flop. Note that modern FPGAs have no gates – the logic for each LE is implemented using small memories called look-up tables (LUTs).

The following diagram shows a Cyclone IV LE[7]:



However, an FPGA's interconnection resources are more limited than 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 and to vertical and horizontal 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.

[7] Diagrams from the Altera Cyclone IV manual.

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. A relatively recent development is a "programmable SoC" (PSoC) which combines an FPGA and a (hardware-based, typically ARM) CPU core.

A typical FPGA is the Altera Cyclone IV family which we have used in the lab. Depending on the version, the FPGA can contain between 6 k and 114 k logic elements and between 180 and 530 I/O pins. A small FPGA (EP4CE6, 6K LE, 91 I/O pins) costs about $12 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.