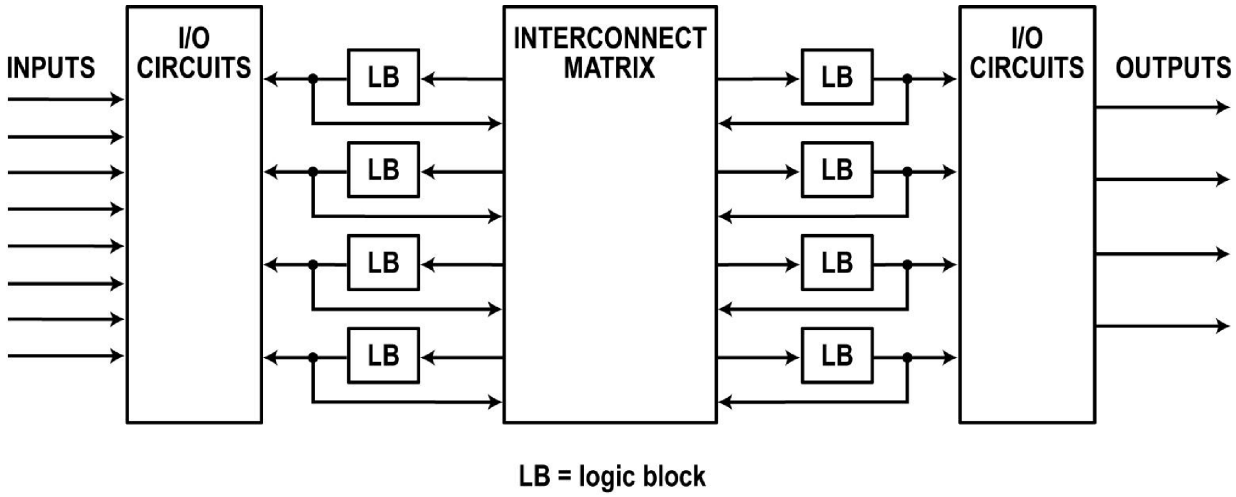
Complex Programmable Logic Devices (CPLD)
and
Field Programmable Gate Arrays (FPGAs)

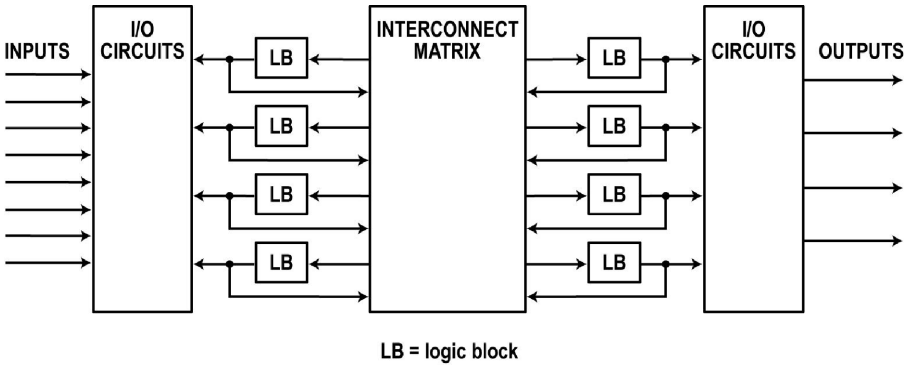
CPLD Organization



CPLDs are larger more sophisticated PLDs designed for very large projects. These devices can contain up to several thousand gates and also flip flops and other logic devices.

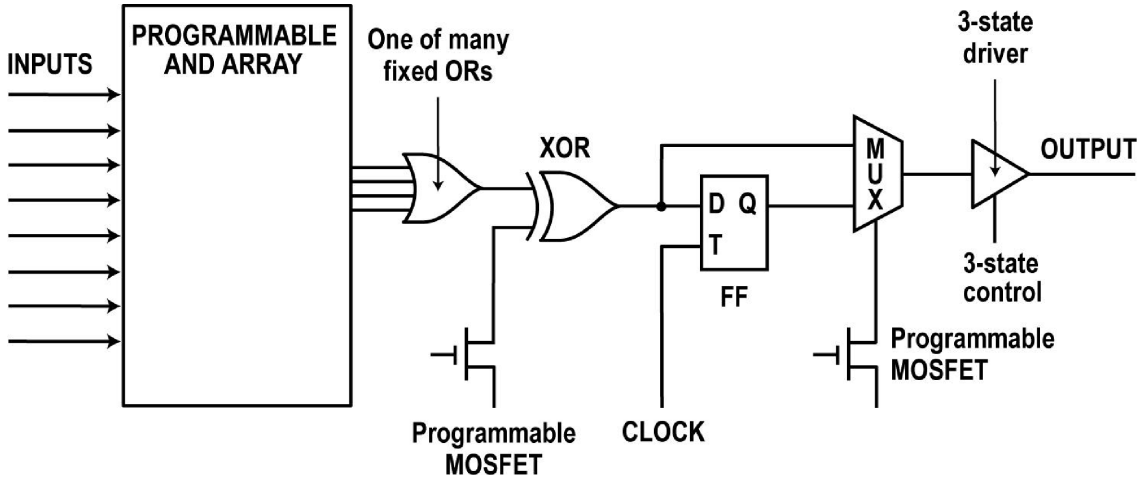
A CPLD is essentially multiple PALs arranged so that they can be conveniently connected to one another and to input/output circuits as shown in here.

CPLD Operation



The figure shows multiple inputs coming into the device. They are conditioned by the I/O circuits then applied to multiple logic blocks (LBs) that are PALs. The PAL outputs then go to an interconnect matrix with rows and columns. At the intersection of each row and column is a MOSFET switch that can be turned off or on by the programming of the device such as a floating gate MOSFET or a SRAM enabled MOSFET. The outputs can then go to other LBs and then on to output circuits where they are conditioned. In some designs the outputs contain three state drivers or flip flops.

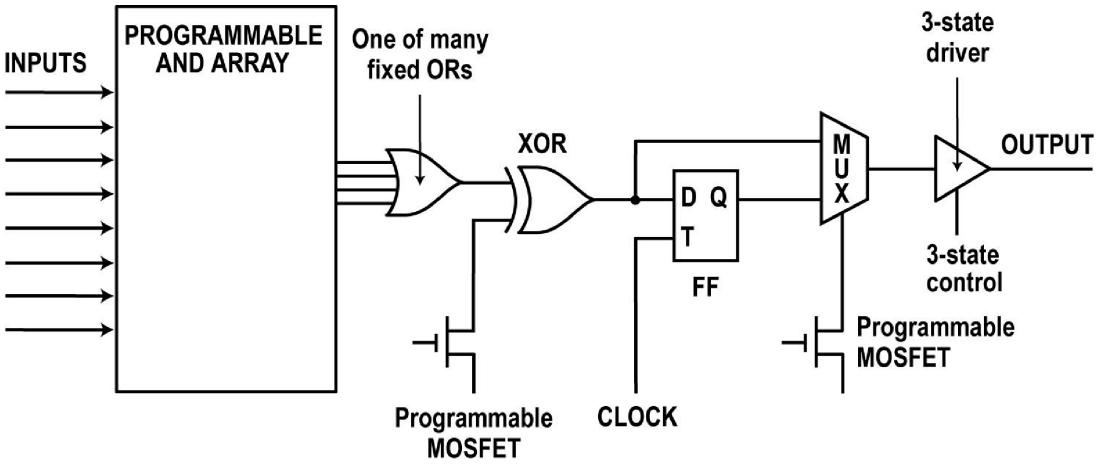
Logic Block



The logic block (LB) shown in the previous figures is a PAL with some interesting features. A generic example is given here.

Remember that in a PAL, the AND array is programmable but the OR array is fixed. The fixed OR outputs are routed to other circuits that may or may not be used depending upon the programming.

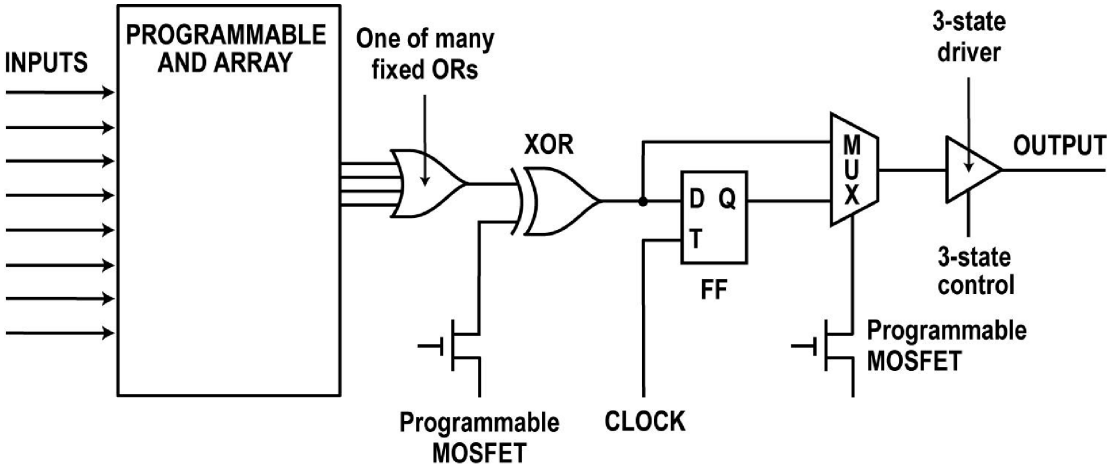
Logic Block Example



As an example, the OR output is fed to an XOR. If the XOR programmable input is binary 0, the XOR just repeats its input to the output. If the programmable input is binary 1, the XOR output is inverted.

The XOR output goes to the input of a FF where the data may be stored or where multiple FF may be used to create a counter, shift register, or some special sequential state machine.

Logic Block



To bypass the FF, the MUX can be programmed to pass the FF input instead of its output. The output may be a three state driver.

Logic blocks vary widely in terms of what circuits they contain and how they can be configured. Multiplexers are typically used to select one programming feature or another by programming.

Introduction to FPGAs

Field programmable gate arrays (FPGAs) are the newest, largest, and most complex of the PLDs. While they have been available since the 1980s, it has only been within the past five years or so that they have emerged as an alternative to ASICs and other PLDs.

FPGAs are used primarily for the very large complex logic designs. Most FPGAs, even the smallest ones contain tens of thousands of gates and other logic elements. The largest models have as many as a million gates. Because of its size, it fits between CPLDs and ASICs in term of number of available gates.

Advantages of FPGA over ASICs

ASICs are the IC of choice when a custom design is necessary or desirable and when the expected volume of use is very high, (over 100,000 units or so depending on the application).

The main disadvantage of ASICs is their very high design cost. Furthermore, they are difficult and expensive to change or update, requiring new semiconductor processing masks and other expensive changes. The initial design and any subsequent design change are very time consuming.

The FPGA overcomes these problems.

FPGA Features and Characteristics

Early FPGAs did not have enough gates to be competitive with ASICs. And they were more expensive even in high volume. Today's FPGAs are larger and the prices have declined to make them an alternative to custom ASICs in some applications. As a result, their use is increasing.

FPGAs are field programmable meaning that they can be programmed by the user one at a time or on a large scale if high volume is needed. This makes correcting design errors fast and easy. It also means that changes, additions, and desirable improvements can be implemented rapidly and incorporated immediately through reprogramming.

Programming a FPGA

Programming a FPGA is done by downloading patterns of binary 0s and 1s into SRAM storage cells that tell the gates and other logic elements what to do. This makes the FPGA a volatile device in that when power is removed the SRAM data is lost so the chip is no longer programmed. For that reason, upon power up, the FPGA must be preloaded from some other memory source such as an accompanying ROM or a down load from a PC or embedded controller in the system.

Disadvantage of SRAM-based FPGA

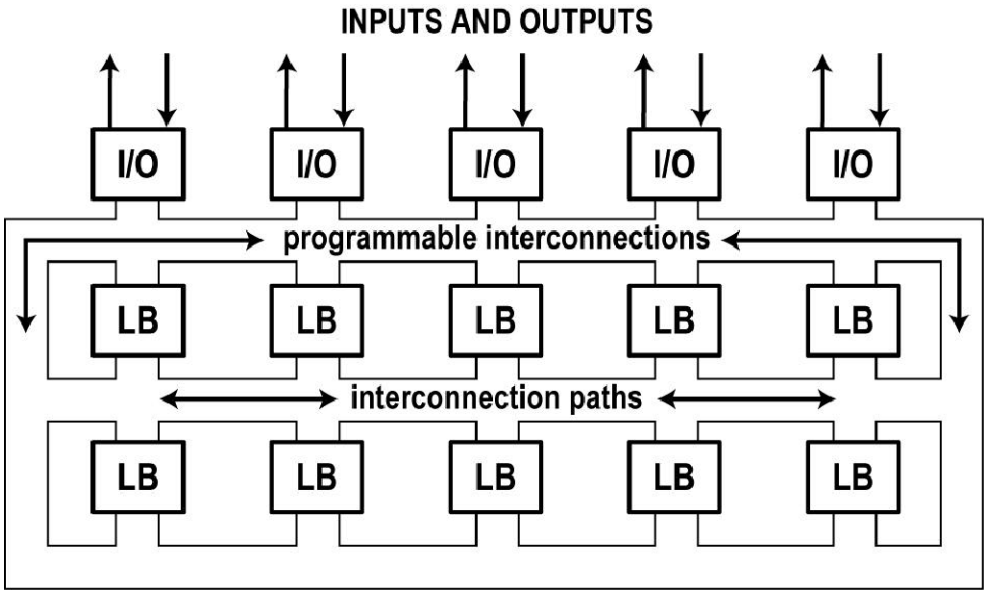
Another disadvantage of the SRAM-based FPGA is that the pattern of 1s and 0s defining the design (the configuration program) stored in an external ROM or embedded controller is subject to theft. That program is what makes the FPGA into whatever unique system or circuits that implements the design. That design is what is called intellectual property (IP). Companies go to great lengths to protect their designs but SRAM FPGAs are vulnerable.

To overcome this problem, FPGAs began using antifuse and EEPROM or Flash memory. These FPGAs are programmed by an external programming device that forms the fuse connections where needed internally to implement the logic. Once programmed, the device is fixed and cannot be changed. However, the IP of the design is protected since the antifuse connections cannot be identified externally.

Using EEPROM

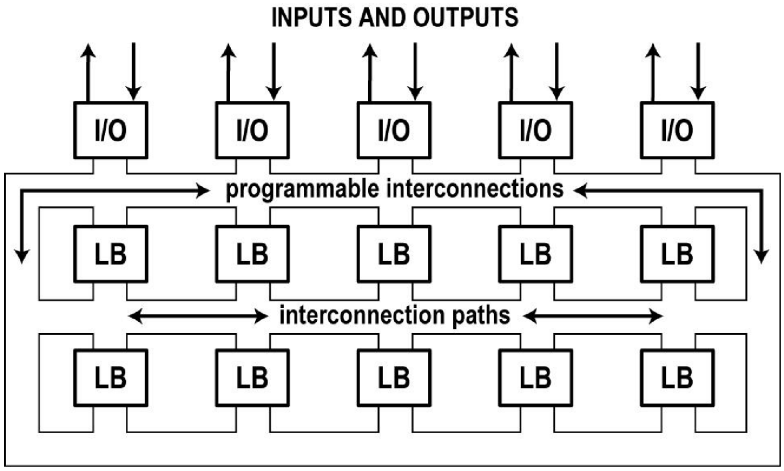
EEPROM FPGAs are a good compromise for those wishing to protect their IP and eliminate the need for an external ROM to initialize the FPGA upon boot up. The program is loaded into the EEPROM which stores it even if power is removed. EEPROMs or flash memory can be erased and reprogrammed if necessary.

FPGA Architecture



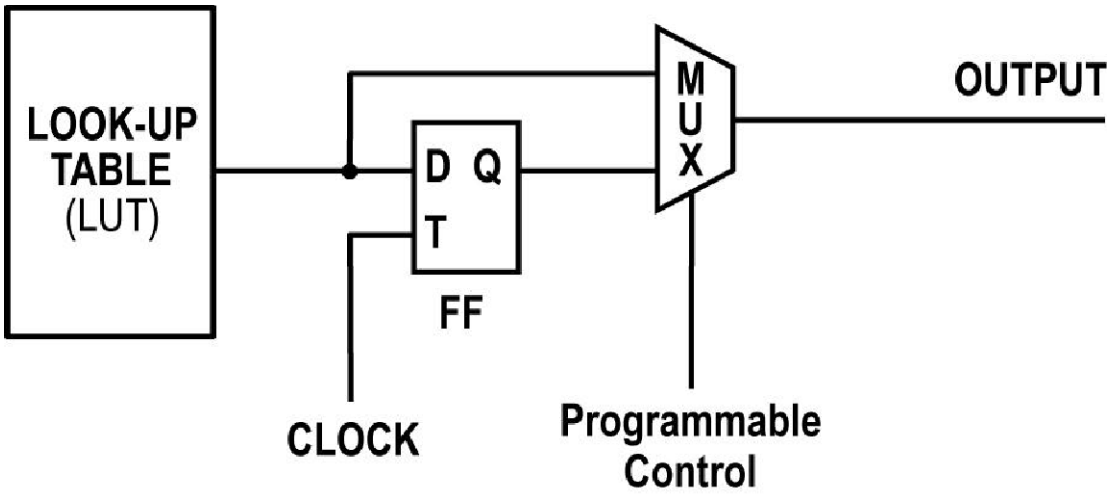
Like the CPLD, the FPGA consists of multiple logic blocks (LB) that can be interconnected to one another and to available I/O circuits with on-chip interconnect blocks. Here is one generic arrangement.

FPGA Architecture: Interconnecting Paths



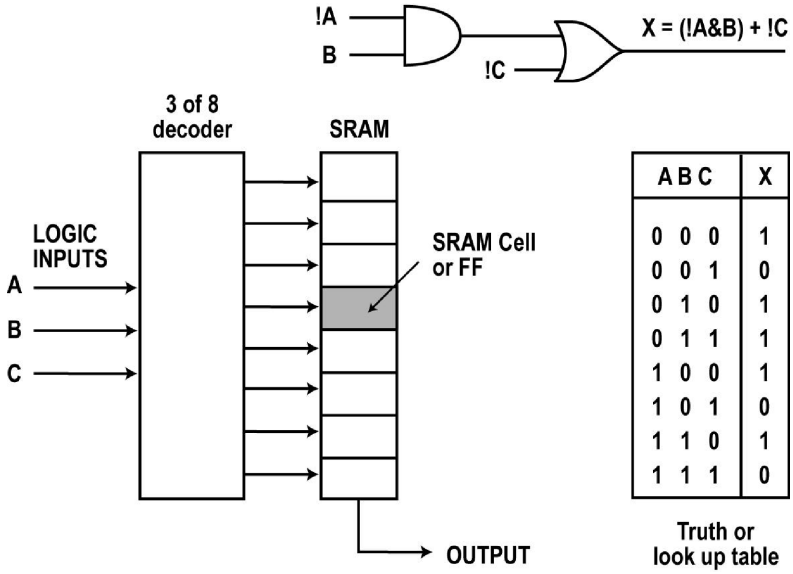
The physical architecture varies from manufacturer to manufacturer. In any case, the designer doesn't need to understand the interconnect scheme to use it to create a working design. The interconnecting paths may be row and column matrices or chains of multiplexers controlled by an SRAM or EEPROM or other programming element. In general, any I/O may be connected to any LB and any LB to any other LB.

FPGA Logic Block Look-Up Table



This figure shows a simplified and generic logic block (LB) used in most FPGAs. Instead of a PAL, the logic is implemented in a look-up table (LUT). A LUT is essentially a programmable memory like a PROM but with a SRAM storage element such as a flip flop (FF).

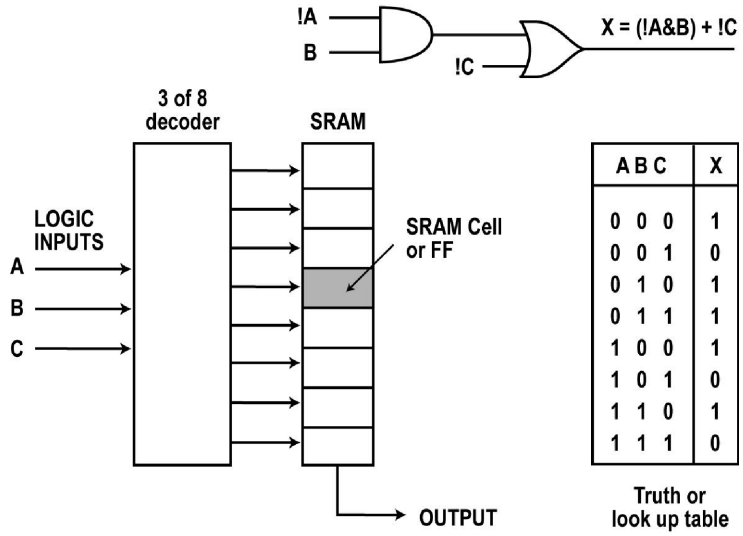
FPGA Logic Block Truth Table



This figure illustrates the LUT concept with a 3-input LUT with eight storage locations. The 3 to 8 decoder is used to enable one of the eight SRAM cells. Each SRAM cell is a flip flop.

The desired logic operation is expressed in a truth table. The LUT is programmed by setting or resetting the FFs according to the desired output.

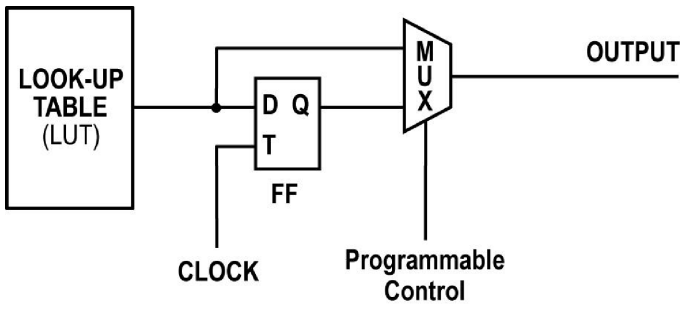
FPGA Logic Block Example



For instance, assume a logic operation expressed by the expression X = (!A & B) + !C. The equivalent logic diagram is also shown in the figure. The SRAM cells are programmed with a binary 1 or 0 corresponding to the output desired for each of the 3-bit input combinations.

When each 3-bit input combination appears, the SRAM output is the desired bit.

Other Logic Block Features



Most FPGAs use a 4-input LUT with 16 SRAM cells but larger FPGAs may use 5 and 6 input LBs.

A special feature of the LUT is that in most cases the SRAM cells can be used separately as a 16-bit memory as the designer may need. The 16 FF may also be connected as a storage register, counter or shift register giving the designer even more design flexibility.

Here each logic block includes a FF for storing the logic outcome or to use in forming a register or counter. Multiplexers (MUX) are used to select the FF or bypass it.

Additional FPGA Circuits: SRAM Blocks

FPGAs may also contain one or more of the either SRAM blocks, arithmetic circuits, or one or more microcomputer cores in addition to multiple logic blocks.

An example of a SRAM block is a 128-bit block that can be configured as a 128 x 1, 64 x 2, 32 x 4, or 16 x 8 memory.

The first number in the notation above refers to the total number of memory locations while the second number refers to the number of bits in each memory location. A 16 x 8 memory has 16 locations for byte length words.

Normally the SRAM cells can be configured as a shift register of any desired length.

Arithmetic Circuits and Microcomputer Cores

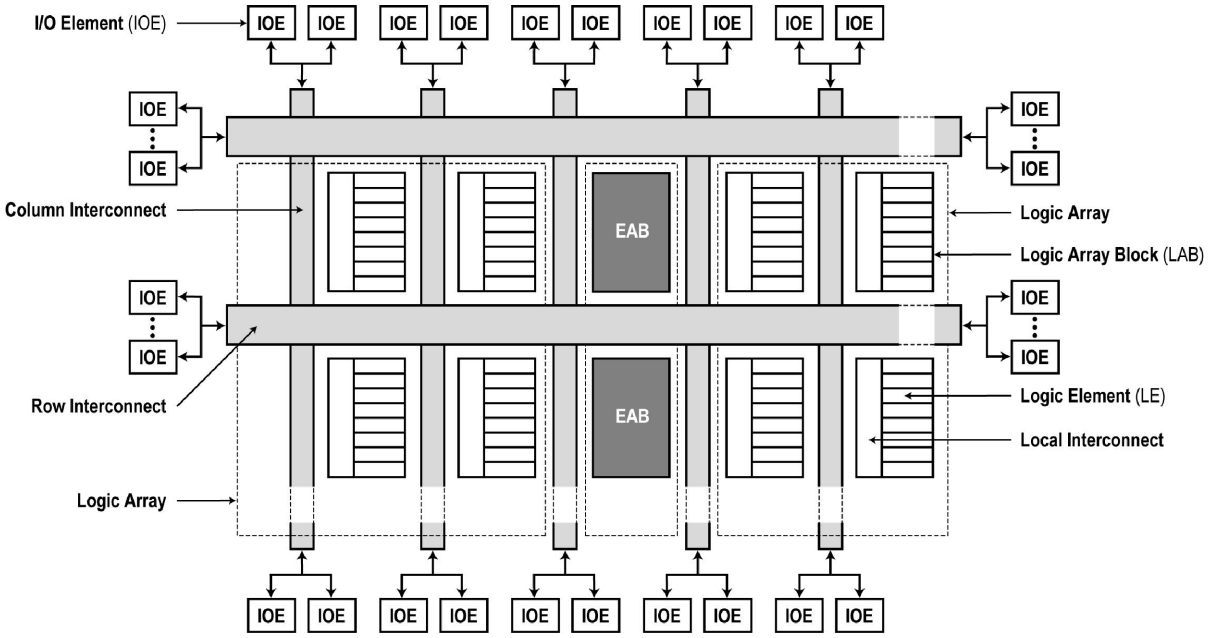
Arithmetic circuits are circuits that perform addition and subtraction or multiplication and division. Newer FPGAs include an arithmetic unit called a multiply and accumulate circuit (MAC) which multiplies two input numbers and adds the product to an output accumulator register. MACs are the circuit used most often in digital signal processing (DSP) operations.

Some FPGAs may also contain one or more microcomputer cores. This is an embedded controller with a limited amount of RAM, ROM, and I/O circuits that can be connected to any of the other on-chip logic circuits. This processor can be programmed to perform specific operations instead of implementing them in hard logic of the FPGA.

The Altera FLEX 10K: a Typical FPGA

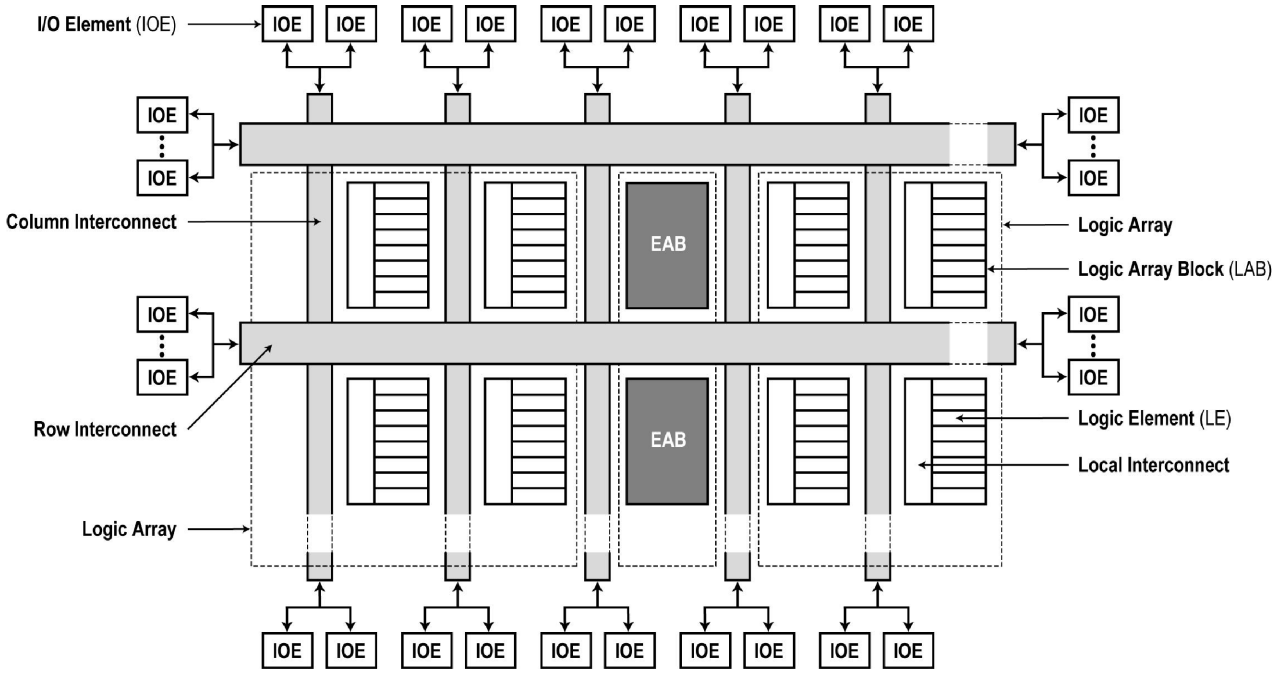
There are about a dozen companies that make FPGAs. Two of the largest are Altera and Xilinx which hold the greatest market share. Other prominent FPGA vendors are Actel and Lattice Semiconductor. An example of a representative FPGA is the Altera FLEX 10K devices. FLEX means Flexible Logic Element MatriX and refers to a family of FPGA chips. They are also called embedded PLDs. There are nine basic versions of the FLEX family with the main difference between them being the total number of total logic gates, RAM cells, and other elements available for programming. The smallest is the EPF10K with a total of 10,000 gates and logic circuits. The largest is the EPF10K250A with a maximum of 250,000 circuits. Versions are available to operate from a 5 volt or a 3.3 volt DC supply. Different chips also have different speed ratings from about 60 MHz to over 200 MHz. Various IC packages are also available.

FLEX 10K Architecture



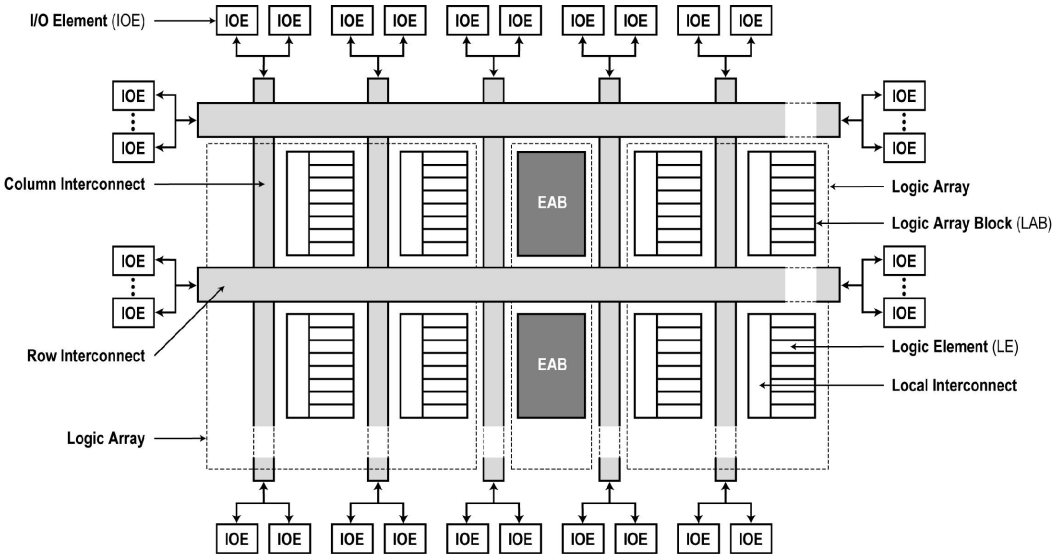
This figure shows a generalized block diagram of the FLEX 10K. It is make up of embedded array blocks (EABs), logic array blocks (LABs), and input/output elements (IOE) plus rows and columns of programmable interconnection paths. The local interconnect blocks allow programming within the EAB or LABs.

EAB



The EAB is a flexible block of SRAM cells with registers on the input and output. These cells can be used as RAM or a block of registers. Each EAB block contains 2048 bits which can be configured as 256 x 8, 512 x 4 1024 x 2 or 2048 x 1 SRAM.

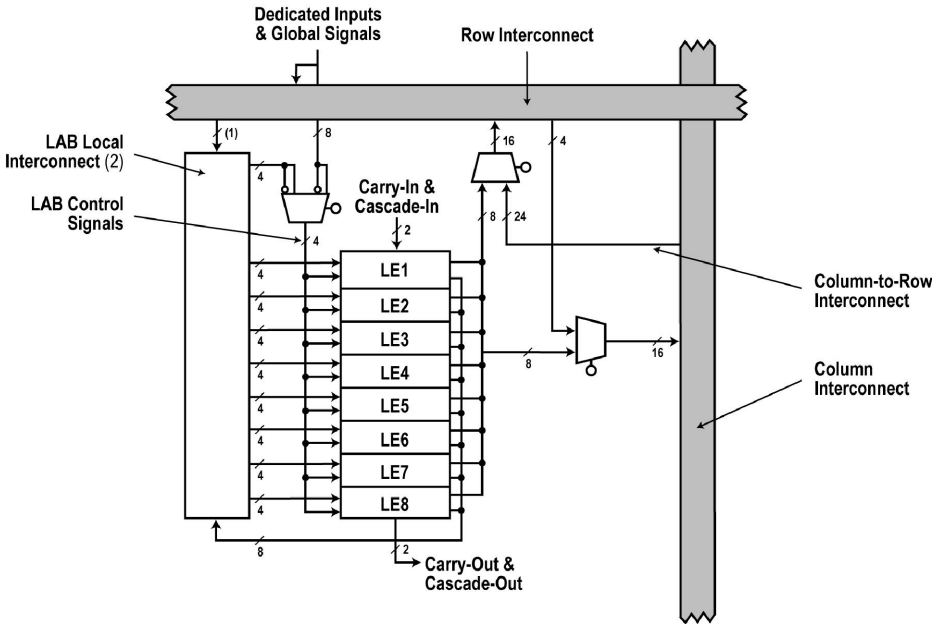
Linked EABs



EABs may also be linked together to form even larger memories. There is a data input buffer register, a data output buffer register, and an address register. Remember that it takes an 11-bit address to identify 2048 RAM cells ($2^{11} = 2048$).

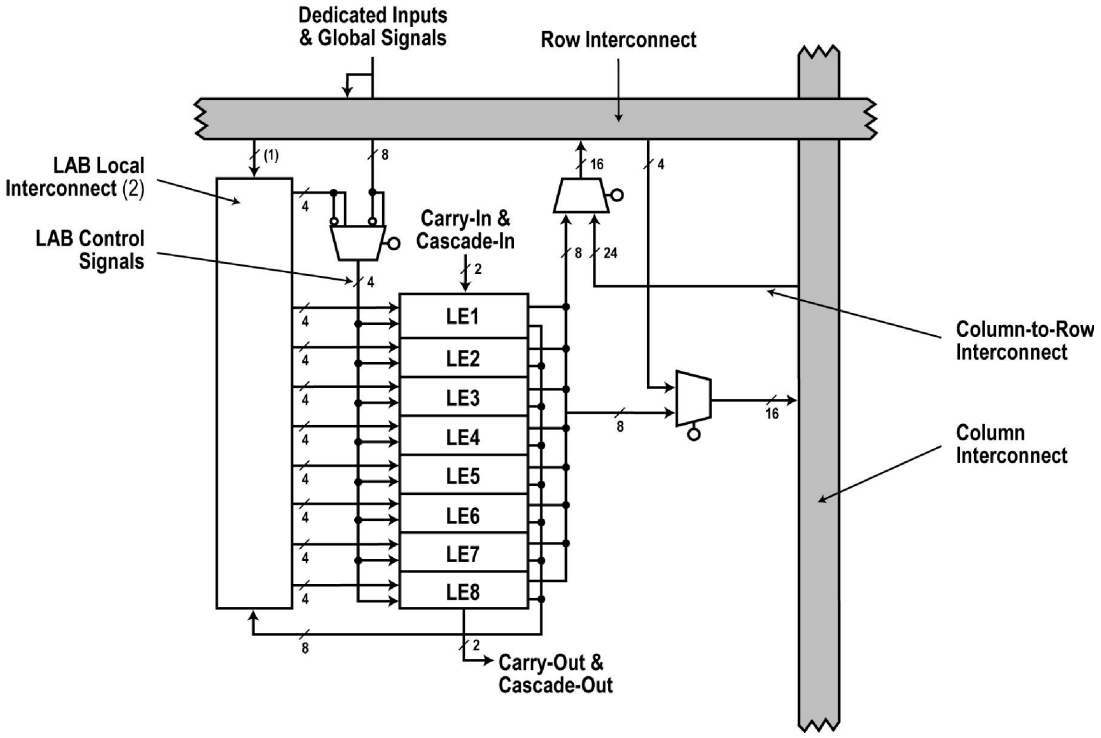
The EAB can also be used to implement logic functions by using the RAM cells as a very large LUT.

Logic Array Block



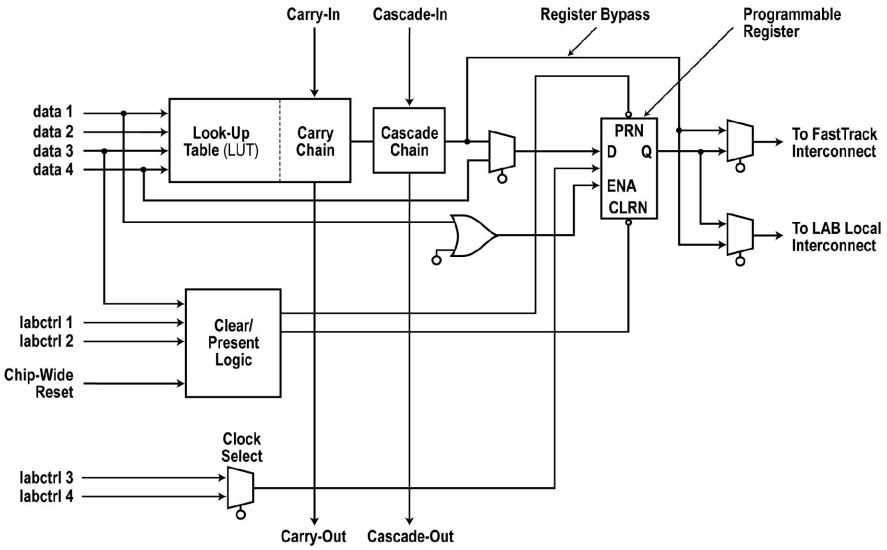
A logic array block (LAB) is made up of eight logic elements (LEs) and the supporting circuits. Multiple inputs are applied to the LEs from different sources and multiple outputs are developed. Local interconnects permit programmed connections within the block. The inputs and outputs connect to the row and column interconnect system through multiplexers.

Control Signals



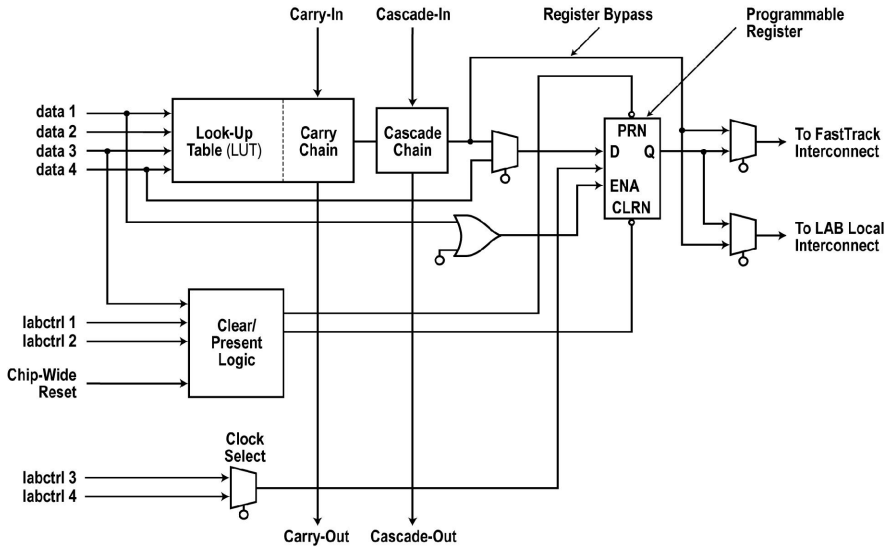
A variety of control signals are also present. The logic elements each have carry out and carry in lines to make arithmetic circuits easy to implement.

Logic Elements



The logic element (LE) is the basic circuit for implementing random logic functions. A block diagram of a FLEX 10K LE is shown here. The basic logic functions are created with a 4-input LUT. The LUT contains carry in and carry out logic making arithmetic circuits easy to create. The LUT output goes to a cascade chain which provides interconnections to combine multiple LEs into larger circuits with more inputs and outputs.

Logic Elements: Flip Flop



Each LE also provides a FF that can be configured as RS/latch, D, or JK type with programmable positive or negative-going clock and set/resets. Note the clear/preset logic available to the FF.

Several multiplexers allow the designer to select the use of FF storage or no storage. The inputs and outputs attach to the row and column connections.

Test your knowledge

Programmable Logic Devices
Knowledge Probe 4
Complex Programmable Logic Devices (CPLD)
and
Field Programmable Gate Arrays (FPGAs)

Click on Course Materials at the top of the page.

Then choose **Knowledge Probe 4**.