7.12 Field programmable devices

(Original section[1])

Within a custom or semicustom chip, designers can make use of the flexibility of the underlying structure to easily implement combinational or sequential logic. How can a designer who does not want to use a custom or semicustom IC implement a complex piece of logic taking advantage of the very high levels of integration available? The most popular component used for sequential and combinational logic design outside of a custom or semicustom IC is a *field programmable device* (*FPD*). An FPD is an integrated circuit containing combinational logic, and possibly memory devices, that are configurable by the end user.

Field programmable devices (***FPD***): An integrated circuit containing combinational logic, and possibly memory devices, that are configurable by the end user.

FPDs generally fall into two camps: *programmable logic devices* (*PLDs*), which are purely combinational, and *field programmable gate arrays* (*FPGAs*), which provide both combinational logic and flip-flops. PLDs consist of two forms: *simple PLDs* (*SPLDs*), which are usually either a PLA or a *programmable array logic* (*PAL*), and complex PLDs, which allow more than one logic block as well as configurable interconnections among blocks. When we speak of a PLA in a PLD, we mean a PLA with user programmable and-plane and or-plane. A PAL is like a PLA, except that the or-plane is fixed.

Programmable logic device (***PLD***): An integrated circuit containing combinational logic whose function is configured by the end user.

Field programmable gate array (***FPGA***): A configurable integrated circuit containing both combinational logic blocks and flip-flops.

Simple programmable logic device (***SPLD***): Programmable logic device, usually containing either a single PAL or PLA.

Programmable array logic (***PAL***): Contains a programmable and-plane followed by a fixed or-plane.

Before we discuss FPGAs, it is useful to talk about how FPDs are configured. Configuration is essentially a question of where to make or break connections. Gate and register structures are static, but the connections can be configured. Notice that by configuring the connections, a user determines what logic functions are implemented. Consider a configurable PLA: by determining where the connections are in the and-plane and the or-plane, the user dictates what logical functions are computed in the PLA. Connections in FPDs are either permanent or reconfigurable. Permanent connections involve the creation or destruction of a connection between two wires. Current FPLDs all use an *antifuse* technology, which allows a connection to be built at programming time that is then permanent. The other way to configure CMOS FPLDs is through a SRAM. The SRAM is downloaded at power-on, and the contents control the setting of switches, which in turn determines which metal lines are connected. The use of SRAM control has the advantage in that the FPD can be reconfigured by changing the contents of the SRAM. The disadvantages of the SRAM-based control are two fold: the configuration is volatile and must be reloaded on power-on, and the use of active transistors for switches slightly increases the resistance of such connections.

Antifuse: A structure in an integrated circuit that when programmed makes a permanent connection between two wires.

FPGAs include both logic and memory devices, usually structured in a two-dimensional array with the corridors dividing the rows and columns used for global interconnect between the cells of the array. Each cell is a combination of gates and flip-flops that can be programmed to perform some specific function. Because they are basically small, programmable RAMs, they are also called *lookup tables* (*LUTs*). Newer FPGAs contain more sophisticated building blocks such as pieces of adders and RAM blocks that can be used to build register files. A few large FPGAs even contain 32-bit RISC cores!

Lookup table (***LUTs***): In a field programmable device, the name given to the cells because they consist of a small amount of logic and RAM.

In addition to programming each cell to perform a specific function, the interconnections between cells are also programmable, allowing modern FPGAs with hundreds of blocks and hundreds of thousands of gates to be used for complex logic functions. Interconnect is a major challenge in custom chips, and this is even more true for FPGAs, because cells do not represent natural units of decomposition for structured design. In many FPGAs, 90% of the area is reserved for interconnect and only 10% is for logic and memory blocks.

Just as you cannot design a custom or semicustom chip without CAD tools, you also need them for FPDs. Logic synthesis tools have been developed that target FPGAs, allowing the generation of a system using FPGAs from structural and behavioral Verilog.

(*1) This section is in original form.

Provide feedback on this section