Field Upgradability Using Actel One-Time-Programmable FPGAs

The conventional wisdom has been that one-time-programmable (OTP) FPGAs can't be used in applications for which logic needs to be upgraded in the field. This application note will show that in many cases, OTP FPGAs can be used in applications requiring this capability.

Introduction

In some applications, detailed specifications for functionality requirements are not available until well into the design process—in some cases, even after the design process. Usually the data path portion of the design is well defined at the beginning of a project, but some of the details of the control logic are not well defined. With a reprogrammable technology, the designer can reprogram the device in the field to adapt to different control requirements. An OTP device can also be reprogrammed in the field to accomplish similar objectives if the designer plans for it.

A typical electronic system is shown on the left side of Figure 1. A main CPU board communicates with a variety of peripheral boards, perhaps in an equipment bay for a telecommunication system. The block diagram on the right of Figure 1 shows the communication path between the CPU and an FPGA. The CPU can write data into the FPGA's on-chip SRAM or registers to alter the operation of the FPGA's control logic.

Application Example

The operation required on a data packet might be well known, but the number of data words and the size of the header might not be specified. These parameters cannot be embedded in a hardwired state machine because they are undefined. These parameters can be stored in read/write registers, however, and can be used to load counters in conjunction with the state machine. The counters can then be used to step through the header and the data packets. This capability allows complete programmability with respect to data and header size. A block diagram implementation of this application is shown in Figure 1.

The programmable header logic in Figure 2 demonstrates how to make the header size an in-system reconfigurable value. The CPU can write any desired value into the 8-bit register Reg8. The state machine will load the value into the counter and then enable the counter in the state associated with the header. When the counter has counted down and the Done signal goes active, the header state is exited. Similar logic can be used for data packet control.

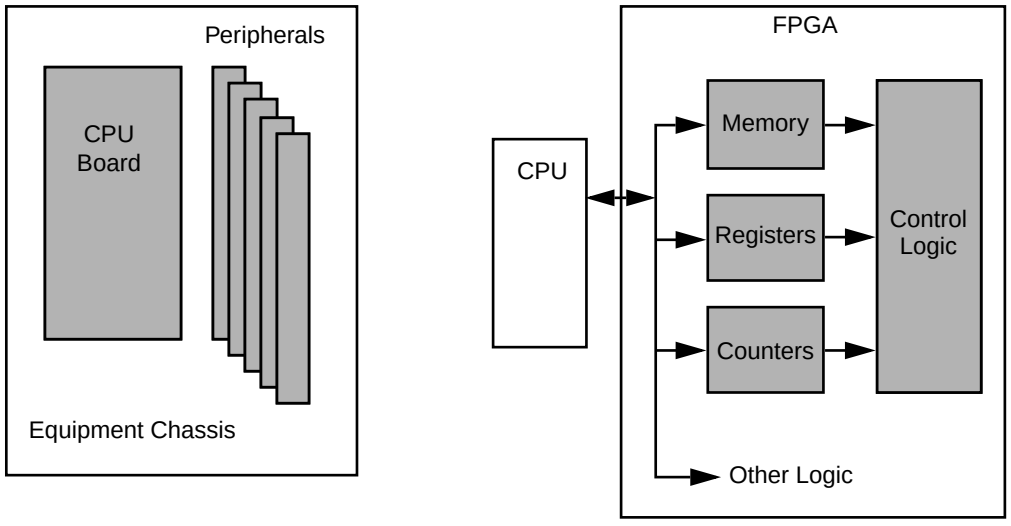

Figure 1 • *Field Upgrading with OTP FPGAs: Block Diagram*

© 1996 Actel Corporation

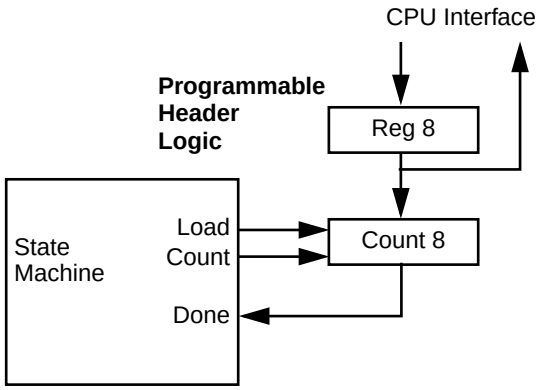

Figure 2 • Programmable Header and Data Length State Machine

The programmable state variable techniques used in Figure 1 can be applied to many different applications. Programmable delays can be important in microprocessor peripherals for wait-state generation and for burst length of synchronous processor interfaces. In networking and communication systems, values for packet size, header size, and time delays can all be programmable.

Another approach to field upgradability is to use on-chip SRAM to create state machines that can be modified once the system has been installed in the field. Actel's 3200DX family has on-chip dual-port SRAM, which can be used to create microprogrammed state machines, using the dual-port SRAM to store next-state and control-output signals. The dual-port nature makes it easy to load the SRAM from an initialization path (perhaps from an on-board processor or memory) by using the write port and to access the microcode stored in SRAM via the read port. Figure 3 shows an example application in which the dual-port SRAM is used to create a microprogrammed machine that controls a complex data path that, in turn, processes an incoming data stream.

The on-board processor loads the SRAM data from the processor interface (a simple multiplexed address and data bus). The next-state logic is predefined and can be a simple next-state address generator (perhaps similar to a bit-slice device). Test conditions are signals generated by the data path used to modify the state sequence and can be a combination of fixed function logic and programmable state variable functions. Programmable delays, for example, can use data from the SRAM to initialize compare registers or other programmable functions.

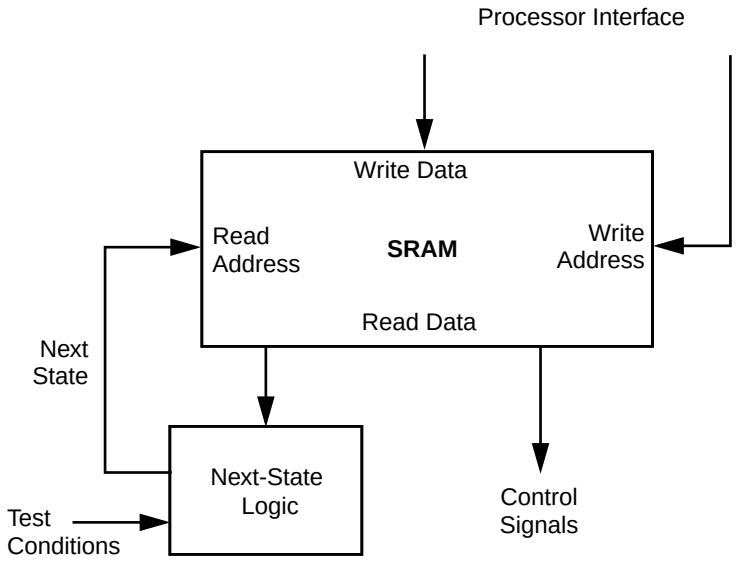

Figure 3 • Microprogrammed State Machine Using 3200DX Dual-Port SRAM

Other Programmable Functions

The following are other examples of programmable control functions that can be used in OTP FPGAs to support field reconfigurability:

- Use of comparitors and variable registers to find a programmable value in a data stream (e.g., for start and stop symbols)
- Use of dual-port RAM for programmable code conversion (for multiple-code support in the same hardware at different times)
- Use of counters and variable registers for programmable delay (when the size of a data packet is unknown)
- Use of XOR gates and variable registers for programmable signal polarity (when you don't know the polarity of an incoming signal)
- Use of multiplexers and variable registers for programmable data flow between buses (when you don't know to which bus result data will be routed)

Summary

OTP FPGAs can be used in many applications for which field reconfigurability of system logic is required. Programmable delays, RAM-based code conversions, RAM-based state machines, and programmable comparitors are all possible in OTP FPGAs.