

Summary

Each state of a state machine can be represented with a unique pattern of high (1) and low (0) register output signals, a process called "encoding." The two primary encoding methods are binary and one-hot encoding. This application brief describes both methods and discusses how to select the encoding scheme that best suits your design, so that you can ensure efficient performance and resource usage.

Binary Encoding

The relationship between the number of state bits (B) and number of states (S) in a binary-encoded state machine is represented by the following equation:

$$B = \log_2(S)$$

With this formula, you can easily determine the minimum number of state bits required for a binary-encoded state machine. For example, to implement a 4-state state machine with a binary encoding scheme, you can use two flipflops (i.e., state bits) to uniquely define the four states as follows:

```
state1 = "00"  
state2 = "01"  
state3 = "10"  
state4 = "11"
```

This example implements the state machine with a left-to-right sequential binary encoding. You can also use a Gray code binary encoding scheme, in which only one state bit changes at a time:

```
state1 = "00"  
state2 = "01"  
state3 = "11"  
state4 = "10"
```

Gray code binary encoding is especially useful when the outputs of the state bits are used asynchronously. For example, if a state machine switches from "01" to "10"—as it does in sequential binary encoding—and the registers do not switch outputs at exactly the same time, temporary outputs of either "11" or "00" can exist. This type of fluctuation can cause unpredictable results throughout your circuit.

One-Hot Encoding

A one-hot encoding scheme uses one register for each state—e.g., four registers for a 4-state state machine—with only one state bit at a high logic level at one time. You can implement a 4-state state machine with a one-hot encoding scheme as follows:

```
state1 = "0001"  
state2 = "0010"  
state3 = "0100"  
state4 = "1000"
```

Selecting an Encoding Method

You should choose an encoding method based on the complexity of your state machine, the target device family, and requirements for recovering from illegal states.

Complex State Machines

Binary encoding uses fewer registers than one-hot encoding. Thus, binary encoding requires only seven registers to implement a 100-state state machine, whereas one-hot encoding needs 100 registers. On the other hand, although one-hot encoding requires more registers, the logic is generally less complex. In a binary-encoded state machine, the logic that controls the transitions from state to state depends on all seven state bits as well as the inputs to the state machine. This type of logic typically requires high-fan-in functions to the inputs of the state bits. In a one-hot-encoded state machine, however, the inputs to the state bits are often simply the functions of other state bits.

Device Architecture

Different architectures favor certain types of encoding. The MAX+PLUS II Compiler automatically selects the most appropriate encoding method for the targeted device family, unless you specify a particular scheme in one of your design files. For example, since the Altera FLEX 8000 device family is register-intensive, state machines targeted for these devices are best implemented with a one-hot encoding scheme. Since a one-hot-encoded state machine reduces the complexity of the logic feeding the state bits, one-hot encoding can increase the performance of your state machine design for FLEX 8000 devices.

The MAX 5000 and MAX 7000 device families are best suited to a binary state machine encoding scheme. Both of these device families can efficiently implement complex combinatorial logic with shared and parallel expander product terms. Thus, devices in these device families can accommodate complex combinatorial logic functions without wasting resources or sacrificing performance.

Recovery from Illegal States

When choosing an encoding method, you must consider the number of potential illegal states your state machine can enter. Your design can end up in an illegal state if you violate the setup or hold times of the state-bit registers and have not defined all possible states. MAX+PLUS II design entry methods allow you to define illegal states and to specify how your state machine should recover from them.

For example, a 14-state state machine implemented with a binary encoding scheme requires four state bits, for a total of 16 possible states. In this case, the state machine has only two possible illegal states. One-hot-encoded state machines, however, generally have more potential illegal states. A 14-state one-hot encoded state machine requires 14 state bits. The number of illegal states for a one-hot encoded state machine is determined by the equation $(2^n) - n$, where n equals the number of states in the state machine. Therefore, a 14-state state machine with one-hot encoding has 16,370 possible illegal states. However, as long as your design does not violate the setup or hold time of the state bit registers, your state machine should not enter an illegal state.

Copyright © 1995, 1996, 1997, 1998 Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, USA, all rights reserved.

By accessing this information, you agree to be bound by the terms of Altera's Legal Notice.