Fault Tolerant State Machines

State Machines

Background

In ASIC/FPGA design state machines are useful to create sequential logic. Figure 1 shows 2 types of state machine, Mealy and Moore. In the Moore state machine, the state of the sequential machine is held in a state register. The next state is determined by a logic combination of inputs and the old state. The state changes on a clock. Outputs are determined by decode logic on the state register.

The Mealy state machine is similar, except the outputs are determined by both the current state and the inputs.

State Encoding

The state register transitions through a sequence of defined states. In the simplest case, the encoding for these states is binary. For example for 4 states the encoding would be:

state	encoding
0	00
1	01
2	10
3	11

However this is not the only method of encoding these states. In '1-hot' encoding the states are encoded as follows:

state	encoding
0	0001
1	0010
2	0100
3	1000

This method has the advantage that is efficient in 'register rich' technologies such as gate arrays. It also is more efficient in decoding the outputs, in some cases.

Another method is grey code. In this code only 1-bit changes for each transition.

state	encoding
0	00
1	01
2	11
3	10

This method has slight advantage in power reduction since only one bit changes at a time. There are also circuits that can benefit by the glitch free decoding of the states.

Fault Tolerant encoding

In a radiation filled environment such as Space, flip/flops may change state randomly. This referred to as Single Event Upset (SEU)
If a SEU changes the state of a flip/flop in the state register, the state will represent a different state than expected. This can cause unexpected results, and in the worse case cause the state machine to 'hang' ie remain in undefined states indefinitely.
To ensure a reliable state machine, the state encoding should be chosen such that:

1: All possible states are defined
2: An SEU brings the state machine to a known state
3: No possibility of a hang state
4: No false state is entered
5: An SEU has no effect on the state machine

The binary encoding can meet criteria 1-3 if all possible states are defined. This is usually achieved in a High Level Design Language (HDL) by defining a 'default' state. This is the state the state register will go to if there are no other matches.

Grey code can also meet 1-3 if all states are specified. This has some small advantage over binary in that the next state after an SEU is limited to adjacent states.

One-hot encoding can meet criteria 1-4. An SEU will set the state machine to an invalid state. This can be detected and used to generate a known next state at the next clock. However, implementations of 1-hot take advantage of its structure to simplify the logic, and do not fully decode the one hot state. If SEUs were not a problem, then detection of a state can rely on decoding just one bit of the state register. Modern synthesis tools take advantage of this to simplify the logic. The other problem with 1-hot is that the size of the state machine becomes large for a large number of states. For example a 4000 state state-machine needs only 12 bits in H2 encoding, but 4000 bits in 1-hot encoding.

H2 encoding is a more efficient way to meet criteria 1-4.

<u>H2 encoding</u>

In H2-encoding the state machine runs through states that are guaranteed to be different by 2 bits. This is known as a hamming distance of 2. For example:

state	encoding
0	000
1	101
2	110
3	011

From any state an SEU will give an illegal state. The illegal state will take the state machine to a defined state. No incorrect state has been entered and no incorrect outputs have been generated.
Note that this encoding is easily generated by adding a parity bit to the binary encoding. So that a 4000 state state-machine will require only 13 bits.

This idea can be extended to meet criteria 5 – this is the H3 encoding.

H3 encoding

In H3 encoding there is a hamming distance of 3 between all states. This means that 3 bits need to change in any state before it will change to another legal state.
In the case of an SEU, the change in state will be to a state which is adjacent and unique to the original state. An SEU in any other legal state will not cause the state register to take on this value. Because of this property, it is possible to generate a state machine which is immune to an SEU change of state. An SEU will bring the state machine to the state which was originally intended. From this point the state machine is automatically 'fixed'.

Generating the H3 encoding is not as easy as generating the H2 encoding, and the author GRB has devised an algorithm of generating H3 encoding.

Summary

A new type of state machine encoding H2 and H3 is presented, which have advantage in fault tolerant state machines, especially in a radiation environment.