Design of a Self Checking Up-Down Counter

AMAL J. MAHFOUD   KHADIJA F. ALGHEITTA   ALI H. MAAMAR
Department of Computer Engineering
Faculty of Electronics Technology
Beni Waled, LIBYA
ali_h_maamar@yahoo.co.uk

Abstract: A register that goes through a sequence of distinct states upon the application of a sequence of input pulses is called a counter. Counters which count upward from zero to maximum are called up counter, while down counters count from maximum to zero. It is possible to construct an up/down counter that will either count up or down, depending on the control signals, in this counter the ability of faults are available [1]. The characteristics of these types of faults render them undetectable by standard test strategies. The detection of intermittent faults requires the use of Concurrent Error Detection (CED) technique, which continuously monitors the operation of circuits and compares them with some known reference. This is achieved by incorporating some form of redundancy into the system [2]. One method of implementing CED in VLSI circuit is through the use of hardware redundancy. This paper investigates the use of extra hardware into unchecked system as a mean of incorporating CED into a self-checking up-down counter.

Key-Words: up/down counter, Self checking, Concurrent Error Detection, Hardware Redundancy.

1. Introduction
Unfortunately as the scale of integration has increased so also has the occurrence of intermittent faults. The characteristics of these types of faults render them undetectable by standard test strategies. The detection of intermittent faults requires the use of Concurrent Error Detection (CED) techniques, which continually monitor the operation of the circuit and compared it with some known reference; this is achieved by incorporating some form of redundancy into the system [2]. One method of implementing CED in VLSI circuit is through the use of hardware redundancy. This paper investigates the use of extra hardware into unchecked system as a mean of incorporating CED into a self-checking up-down counter. Self-checking circuit can be defined as the ability to verify automatically whether there is any fault in the circuit (chips, boards, or assembled system), thus, self-checking circuits allow on-line error detection, which means faults can be detected during the normal operation of the circuit [3]. The self-checking could be achieved by the redundancy techniques; one way to achieve self-checking design is through the use of extra resources (the hardware redundancy technique) [4].

2. up-down counter
An up-down counter goes through the binary states in reverse order from maximum down to zero and back to maximum to repeat the count [5]. When the counter counts down, the bit in the least significant position is complemented with each pulse, a bit in any other position i, is complemented if all lower significant bits are equal to 0. For example, if the present state of the counter is 0100, then the next state of the counter will be 0011, the least significant bit is always complemented. The second significant bit changes because the first bit is 0. But the fourth bit does not change because not all lower significant bits are equal to 0 [6].

Figure 1, shows a 4 bits up-down counter. The up-down counter is a normal counter, it updates itself according to the control signals (up, down, and reset), when it receives up signal the present state of the counter is incremented by 1, and when it receives down signal, the present state of the counter decremented by 1. The reset signal sets all bits of the counter to zero.

Figure 3, shows the typical waveform of the up-down counter.
3 Self Checking Circuits

Self-checking circuits allow on-line error detection, that means faults can be detected during the normal operation of the circuit. It can detect the presence of both transient and permanent faults. A self-checking circuits consists of a functional circuit, which produces normal output, and a checker circuit, which checks the output of the functional circuit to determine if an error has occurred. The checker has the ability to give an error indication even when a fault occurs in the checker circuit itself.

The hardware of the checker circuit depends on the type of the redundancy to be used, whether it is a hardware redundancy, information redundancy, or time redundancy. In this paper a hardware redundancy will be used in the design of self checking up-down counter. Figure 2 shows a block diagram of a self checking up-down counter. The extra hardware needed as shown in figure3, consists of up counter, down counter, and two checkers.

3.1 Up counter

The up counter is a normal up counter, that counts up or incremented its present state when it receives count up signal, its present state at any time is equal to the present state of the up-down counter +1, for example if the present state of the up-down counter is 1001, then the present state of the up counter is 1010. The up counter is active only when the up-down counter is counts up, as the new value of the up-down counter has to be compared with the value of the up counter, if the two values are not match an error signal is generated, which means the up-down counter did not counts up correctly, or the up counter itself did a mistake, in both cases an error signal should be activated. When the up counter receives reset signal it goes back to zero not to 1, as its present state should be always one step above the present state of the up-down counter.

3.2 Down Counter

The down counter is counter that counts down from maximum to zero and back to maximum to repeat count. The down counter used in this paper, it keeps its present state one step below present state of the up-down counter, for example if the present state of the up-down counter is 1001, then the present state of the down counter should be 1000. In general, at any time if the present state of up-down counter is X (represented in binary form), the state of the down counter should be X-1. The down counter is active only when the up-down counter is counts down, as the new value of the up-down counter has to be compared with the value of the down counter, if the two values are not match an error signal is generated, which means the up-down counter did not counts down correctly, or the down counter itself did a mistake, in both cases an error signal should be activated.
3.3 Checker Circuit
There are two identical checker circuits used in the self checking up-down counter, one checker is used when the up-down counter counts up, it receives the output of the up-down counter and the complement of the output of up counter. The other checker is used when the up-down counter counts down, it receives the output of the up-down counter and the complement output of the down counter. The checker circuit is a normal Two-Rail checker (TRC) figure 4, the TRC is used to compare two complementary binary values. The checker determines whether the output of the functional circuit is a valid or invalid. Two-rail checker unit has two groups of inputs: \((x_1,x_2 \ldots x_n)\) and \((y_1,y_2 \ldots y_n)\). It also has two outputs: \(f\) and \(g\). The signals observed on the outputs should always be complementary. Consider a two rail checker with \(n=2\), as shown in Figure 4, the two input groups are \((x_1,x_2)\) and \((y_1,y_2)\). In a non-error situation where \((y_1=x_1')\) and \((y_2=x_2')\), the result of this is \((f=g')\). In situation where due to a fault where \((y_1=x_1)\) or \((y_2=x_2)\), this will then produce \((f=g)\), that means a valid output thus giving an error signal.

![Fig.4 Two rail checker with 2 pairs of inputs](image)

Figure 5, shows the typical waveform of the counters. CLK signal is free running. The counters counts the new values by a rising edge on the CLK line and up or down signals are either active high depending on the status of the counters.

4 Conclusion:
In this paper a self checking up-down counter was designed and simulated using HDL. The counter is error free. Hardware redundancy technique is used to achieve a concurrent error detection, where errors can be detected during normal operation of the counter. The penalty of implementing concurrent error detection is an extra hardware, as an up counter, down counter and two checkers were used, also an extra time (delay) which is the time for comparison is added to the normal time of the un checked up-down counter.

References: