

## A MORE WATCHFUL WATCHDOG FOR MICROCOMPUTERS

Paul Newland, ad7i  
Post Office Box 205  
Holmdel, New Jersey 07733

### Introduction

Many hardware/software watchdog timers consist of a software routine that repetitively triggers a hardware retriggerable monostable. If the monostable ever times out, the computer is reset. This technique, although useful, is not extremely reliable under most software/hardware insane conditions. This paper discusses an alternative approach that may prove to be more reliable under various fault conditions.

### Discussion

The improved system offered here involves the use of a serial pseudo-random number (PN) generator, implemented in software, and a serial pseudo-random number detector, implemented in hardware. During the execution of the system program, a small portion of the CPU's time is allocated to generating another bit of the PN sequence and dumping it (and an associated clock signal) out a parallel port to the hardware PN detector.

When the clock lead from the parallel port goes high, the hardware checks to see if the data bit is a valid part of the PN sequence. If so, it does nothing. If it is incorrect, it resets the computer and doesn't allow the PN detector to issue another reset for a pre-determined period of time. This time delay is necessary to allow the computer to restart all processes and get the PN generator and PN detector back into synchronization.

While this PN checking is going on, additional hardware checks a tap from the hardware shift register to ensure that data is changing state occasionally. A failure of data change would indicate that either the computer is not clocking the external hardware or the computer is "fooling" the PN detector by giving it a degenerate PN sequence[1].

### Implementation

Figure 1 shows a subroutine, written in Zilog mnemonics for the Z80(tm)

processor, that implements a serial PN generator. The describing polynomial is:

$$D_o = 1 \oplus D_o^3 \oplus D_o^7$$

where "Do" means the Data-output value for the current bit time and  $D_o^n$  means the Data-output value "n" bit times ago. The " $\oplus$ " operator is the modulo-two sum (exclusive or) operator.

Every call of this routine generates a new bit output and an associated clock signal. Figure 2 shows hardware for the PN detector and the timing circuitry.

### Conclusion

It is apparent from the discussion that any perturbation in the PN sequence (i.e., wrong bit at the wrong time, data stuck high or low, slow clock) will cause the reset signal to be issued. However, this "watchdog" can not protect against a computer malfunctioning in a way that causes the software-driven PN sequence to continue being generated. Programmers who might rely on a system such as this to protect against software "getting loose" should carefully consider where and when the PN generator routine is called.

### Notes

[1] Typically, for PN detectors implemented using shift-registers and exclusive or gates, the detector can be fooled into thinking that the PN sequence received is correct if that sequence is either an all ones or all zeros pattern, depending on the construction of the detector. For this discussion, consider a degenerate PN sequence to be one of either all ones or all zeros.

