May 1968

Brief 68-10067

NASA TECH BRIEF

Self-Correcting, Synchronizing Ring Counter Using Integrated Circuit Devices

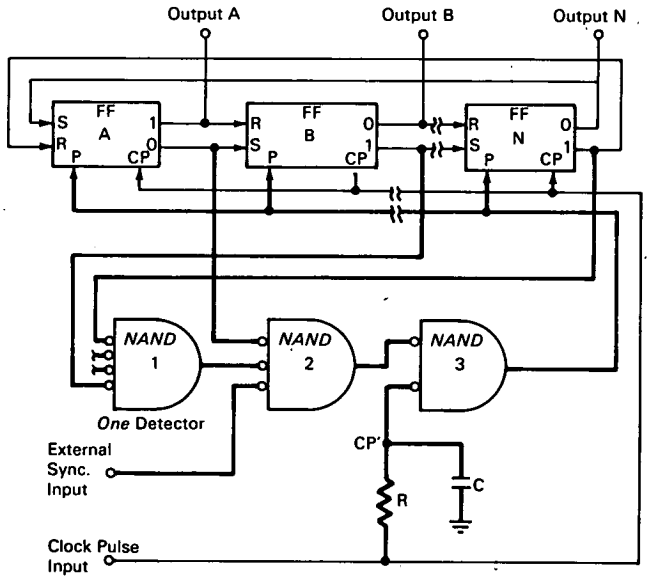

FIGURE 1. SIMPLIFIED DIAGRAM

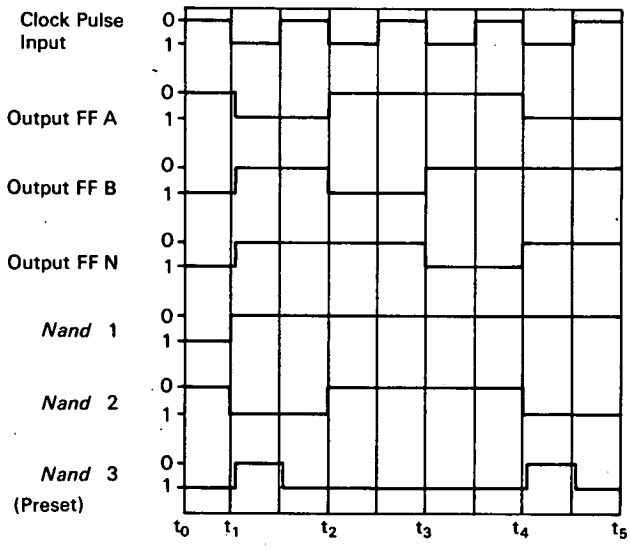

FIGURE 2. PRESET AND TIMING SIGNAL SEQUENCE OF CIRCUIT USING NEGATIVE TRUE LOGIC

The problem:

To initiate and retain the correct binary state in the flip-flop circuits of a ring counter so as input signals are counted, the position of the specified state moves in an ordered sequence around the circuit loop. When power is turned on, the circuit is likely to be in an incorrect state. Error occurs at power-on if any flip-flops other than the first flip-flop contain an *on* (i.e., using negative true logic, a logical *one* state). Error also occurs during operation if a flip-flop other than the specified circuit contains an *on* or logical *one* state. Conventional monitoring techniques require complex circuit configurations and a large number of gate circuits to detect an error.

The solution:

Add error detection and reset logic circuitry using only three *nand* gate circuits.

How it's done:

Figure 1 shows conventional ring counter circuitry with error detection and reset logic circuitry added (in heavy lines). The necessary initial circuit condition is to have the output of first integrated circuit flip-flop (A) in the logical *one* condition. When this is so, a preset pulse P is simultaneously applied to all flip-flops of the ring counter from *nand* gate 3 resulting in any extraneous logical *one's* being reset to *off* (i.e., a logical *zero* state). Thus, the circuit is ready for operation, the logical *one* condition of flip-flop A will

(continued overleaf)

advance to each successive flip–flop in the loop as the clock pulse (CP) input signals are counted.

By adding a single input to the second *nand* gate (2), an external signal may be applied to synchronize the ring counter with another external source.

The CP input is applied to the final gate (3) so that the preset pulse P will not last into the following clock time and hold the ring counter in the initial state. The CP input to the last gate is slightly delayed by an RC network; so the flip–flops have time to settle before being error-detected.

Considering the worst possible turn-on circuit condition, where output A is a logical *zero* and outputs B and N are logical *one's*, a signal sequence is shown at t_0 in Figure 2. Since the flip–flop outputs are in these conditions, *nand* gate 1 output will be in the logical *one* state, because of the redundant logical *zero* input detection, ("1" = $\overline{B \cdot N}$), *nand* gate 2 will have a logical *zero* output because of the *one* input from *nand* gate 1, "1" = (NB) A (the same as positive logic A + BN). At t_1, *nand* gate 3 will have a logical *zero* output, because of the logical *one* input from *nand* gate 2 and the logical *one*-going portion of the clock pulse ("1" = A + B · N · CP'). Therefore, a preset signal (P) is sent to the flip–flops until *nand* gate 3 returns to *one*, during the logical *zero* swing of the clock pulse. The preset signal forces output A to become logical *one* and outputs B through N to become logical *zero's*. The preset input is the only part of the circuit using positive logic.

The preset pulse changes each flip–flop circuit from the worst-case condition; output A goes to a logical *one*, and outputs B through N go to logical *zero's*. When the clock pulse returns to the logical *zero* state, *nand* gate 3 returns to a logical *one* and the preset pulse is terminated. *Nand* gate 2 retains its logical *one* state because of the *one* input from flip–flop A.

Nand gate 1, no longer sensing logical *one's* in flip–flops B through N, changes to a logical *zero*. Thus, normal operation begins with the preset pulse application at t_1.

At t_2, output A will change to a logical *zero* and output B to the logical *one* state. *Nand* gate 2 no longer senses a logical *one* in flip–flop A and returns to a logical *zero*.

The next count, beginning at t_3, the CP and the output of flip–flop B causes flip–flop N to set to a logical *one*. *Nand* gate 1, still sensing a logical *one* from flip–flop N, holds the state preventing any change in the preset circuitry. Flip–flop N outputs are returned to the inputs of flip–flop A, and in sequence with the *zero*-going CP, at t_4, flip–flop N passes the *one* to flip–flop A, accepts the *zero* in flip–flop B, and completes a correct cycle of ring counter operation.

A redundant preset pulse will be generated when *nand* gate 2 senses a logical *one* in flip–flop A. If an extraneous logical *one* were to appear in the ring counter for any other reason, either *nand* gate 1 or 2 would have sensed it and caused *nand* gate 3 to preset the counter back to t_1 condition.

Note:

Inquiries concerning this circuit may be directed to:

Technology Utilization Officer
Marshall Space Flight Center
Huntsville, Alabama 35812
Reference: B68-10067

Patent status:

No patent action is contemplated by NASA.

Source: W. A. Maasberg
of International Business Machines Corporation
under contract to
Marshall Space Flight Center
(MFS-13901)