W. N. Carroll

High-Speed Counter Requiring No Carry Propagation

When very high speed counting is demanded, it sometimes becomes difficult to achieve the objective even with high-speed circuitry, because of the time necessary for carry propagation. The counter described below[1] eliminates carry propagation by a unique system of parallel counting.

Theory

A carry-less method of counting which has been developed[2] is described as follows. Given a register with any number in it; first search for the lowest order zero; then complement the remaining higher order bits; then complement the entire register. See Table 1.

Table 1 **Carry-less counting method.**

1. Original register content	001001	(binary 9)
2. Search for lowest order zero and complement the remaining higher order bits	110101	
3. Complement entire register	001010	(binary 10)

The searching for the lowest order bit containing a zero can be accomplished with a parallel sampling of gates on the zero side of all flip-flops in the register. This method eliminates the need for propagating a carry signal. However, the requirement that the entire register be complemented to complete the process is time-consuming and a disadvantage. To improve the counter, it is necessary to eliminate the need for complementing the entire register.

A logical count can be achieved by proceeding as follows. Given a register with any number in it, first complement the entire register; then search for the lowest order bit containing a one; then complement the remaining higher order bits as shown in Table 2.

Table 2 **Reverse carry-less counting method.**

1. Original register content	001001	(binary 9)
2. Complement entire register	110110	
3. Search for lowest order one and complement the remaining higher order bits	001010	(binary 10)

It can be seen that this method is the reverse of the previous method. However, the disadvantage of having to complement the entire register is still present.

Both techniques perform as follows:

Step 1. Search for lowest order zero

Step 2. Complement remaining higher order bits

Step 3. Complement entire register

Step 4. Complement entire register

Step 5. Search for lowest order one

Step 6. Complement remaining higher order bits.

By combining both methods the requirement for steps 3 and 4 is eliminated since a double complement results in the same answer as before the operation. Thus by alternate counting, first searching for low order zeroes and then lower order ones, the process reduces to steps 1, 2, 5 and 6.

The method used to implement this technique requires a control flip-flop which is complemented by each step pulse to keep track of each mode of counting. Such a method produces the events shown in Table 3.

From the table of events we observe that the even numbers are represented in their complement form while the odd numbers are represented in true form. Also, the lowest order bit is always a one. Thus, no flip-flop is required for this bit. This also means that when we are operating in a mode which is searching for the lowest order bit containing a one we always find one in the first position of the register. Since the first position does not use a flip-flop, the only operation necessary when adding one by this mode would be to complement the register. Thus the table of events reduces to that shown in Table 4.

Note, the sequence of events is identical to that stated in Table 1 except that these steps have resulted in a count of two, and the control bit is always equal to the true form of bit #1.

Circuitry

The circuitry required to implement this counter is shown in Figure 1. That portion of the circuitry shown below

the dotted line is required to do the actual counting. That portion above the dotted line can be used as a method of reading out.

The counter operates as follows:

1. A clear pulse is applied to all the flip-flops setting all bits of the register to one and the control bit to zero.

2. A step pulse is applied to G1 and G2. The control flip-flop being in the zero state allows a pulse to be emitted from G1 which is used to complement all flip-flops. The step pulse is also applied to complement the control flip-flop. The register has now been increased by one, and contains the true representation of binary 1.

3. A second step pulse is applied, with the control flip-flop in the one state, a pulse is emitted from G2. This pulse samples Gates G3 through G7 in parallel. Every flip-flop in the zero state will allow its associated gate to emit a pulse which is applied to the complement inputs of all higher order bits. There may be several zeroes; hence several simultaneous complement pulses being applied to any one flip-flop; however, this produces the same effect as a single complement pulse. The step pulse also complements the control flip-flop. The register will now represent the complement of a binary 2. The operation continues with the next step pulse being emitted from G1. The action continues to alternate with each step pulse applied.

Table 3 **Control flip-flop table of events.**

	Binary Number	*Register Bits*	*Control Bit*
Clear all bits to one and set control bit to zero	0	1111	0
Search for low order ones and complement remaining higher order bits	1	0001	1
Search for low order zeroes and complement remaining higher order bits	2	1101	0
Search for low order ones and complement remaining higher order bits	3	0011	1
	4	1011	0
	5	0101	1
	6	1001	0
	7	0111	1
	8	0111	0
	9	1001	1

Figure 1 **High-speed counter.**

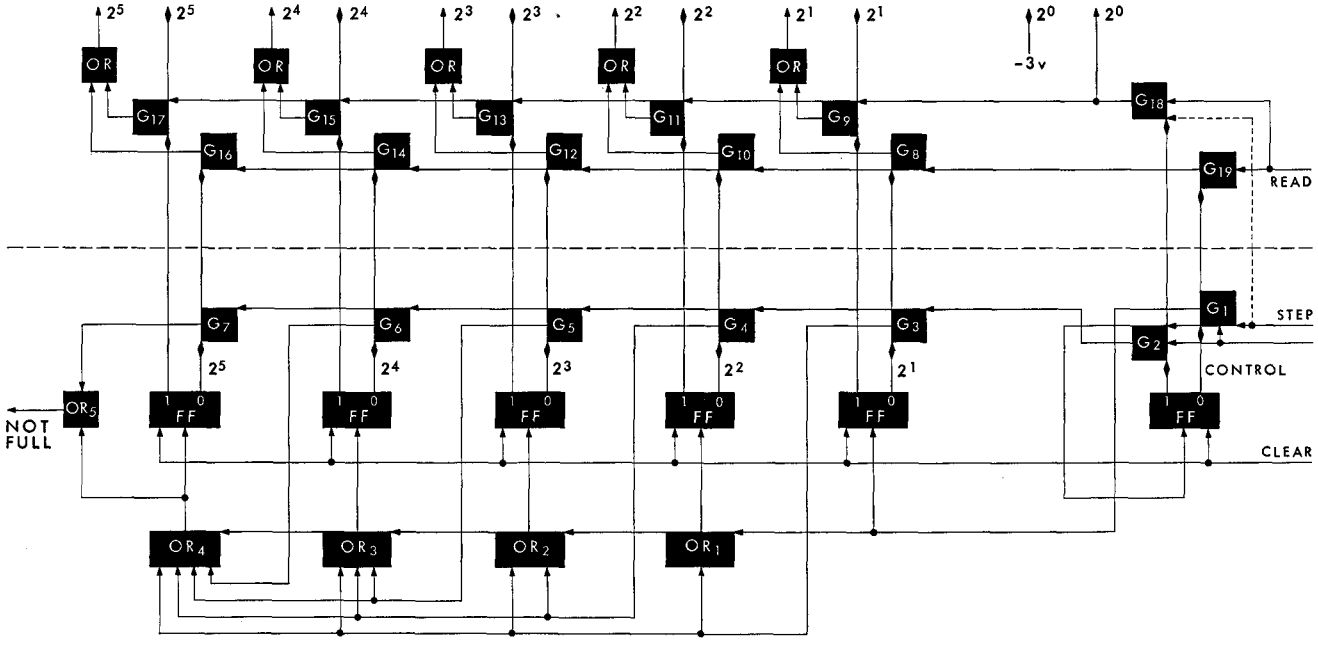

Table 4 **Modified control flip-flop table of events.**

	Register Bits	*Bit 1*	*Control Bit*
1. Original Register Content	00100	1	1 (true binary 9)
2. Search for low order zero and complement remaining higher order bits	11010	1	0 (complement binary 10)
3. Complement entire register	00101	1	1 (true binary 11)

At no time is it necessary to propagate a carry signal. The entire action is parallel, and the time to count is equal to one flip-flop resolution time regardless of the number of bits.

The OR circuit (OR_4) emits a pulse as long as the register is not full. As soon as the register becomes full, no further pulses are emitted from the circuit. This feature is used in error-checking circuitry.

To provide read-out of the counter, the control flip-flop determines which side of the register flip-flops should be read to obtain a true version of the register contents. The OR circuits are used to provide only 1 output line per bit. These read-out gates can be sampled immediately after the flip-flops have resolved.

References

1. IBM Kingston Patent Disclosure, "High Speed Counter" #130582, W. N. Carroll and R. A. D'Antonio, January 27, 1959.
2. E. G. Wagner, Patent #2848166, August 19, 1958.

Received November 16, 1959