Digital Period Detector Oscilloscope Trigger

W. A. Lushbaugh
Communications Systems Research Section

Due to the increased complexity of new digital equipment, there has arisen a need for more sophisticated test equipment. This article describes a piece of equipment for obtaining an optimum trigger for an oscilloscope. This equipment accepts a periodic digital sequence and its associated clock, and outputs a single pulse once per period. This output is intended to be used as the external trigger for an oscilloscope. A digital readout of the numerical value of the period is also provided to enable determination of the correct trigger to be used for a multitrace display.

I. Introduction

Due to the increased complexity of new digital equipment, there has arisen a need for more sophisticated test equipment. The digital period detector oscilloscope trigger (DIPDOT) is a piece of test equipment which accepts a periodic digital sequence and its associated clock, derives the period of this sequence, and outputs a single pulse once per period. A digital readout of the number of clock pulses in the period appears on the front panel. The output pulse is intended to be used as the external trigger input to an oscilloscope, thereby enabling the display of sequences for which no other sync is available and which will not self-trigger. The digital readout can be used to check that the external trigger being supplied to a multi-trace display has the correct period necessary to properly display all traces in their true phase relationship. Use of the DIPDOT to detect the period of the longest length sequence of a multi-trace display will also ensure the maximum brightness possible for such a display.

II. Design Aims

It was desired to have the DIPDOT use as little hardware as possible. Obviously an easy method of finding the period of a sequence is to store a number of bits of the sequence larger than the greatest expected period and do a simple correlation on these bits until the minimum period is found. However, since it was decided that the device would not be useful unless it could determine periods of at least several thousand bits, the mass memory approach was abandoned and a serial scheme adopted. The serial version uses a minimum of sequence memory (actually only one bit) but instead, observes the sequence over many of its periods to extract the necessary information.

III. The Algorithm

A sequence $f(n)$, $n = 1, 2, \cdots$, has period P if

$$f(n) = f(n + P)$$

for all n and P is the smallest such number for which this equation is satisfied. To ensure that the P found by the DIPDOT is indeed the smallest such value, the first hypothesis H is one, i.e., it is first assumed that all the bits of the sequence are equal. This assumption is held and every bit of the sequence examined until a difference is observed. At this point the hypothesis is set to 2 and the sequence is searched for an adjacent 1–0 combination of bits.

After the 1–0 is found every (non-overlapping) pair of bits following is examined for agreement with 1–0. This mode of operation will continue until a disagreement is found, at which time H is set to 3, the device waits for an adjacent 1–0 and then checks every pair of bits spaced three units from the 1 for the 1–0 agreement. Iteration continues in this manner until an H is found such that a 1–0 combination is found M times spaced H apart, where M is the largest period expected. The search up to this point will be referred to as Mode I.

The 1–0 window was chosen because every sequence of period greater than 1 has such a combination and because many digital sequences encountered in practice have a low density of *ones* or *zeros* leading to a low number of transitions. Thus the DIPDOT locks onto a significant point in low density sequences (i.e., the probability of passing a large number of tests when H is not correct is low) while in more random sequences nothing is lost since all two-bit windows would have approximately the same density.

The job of finding the correct period is not completed when an M is found such that M consecutive tests show no errors. However it is certain that H and P, the actual period, have a common factor. Thus the H-2 bits between the 1–0 windows must be checked for agreement. This second part of the algorithm, which will be referred to as Mode II, uses a time-saving method developed by Dr. E. Rodemich and is described below:

IV. Rodemich Verification Method

Theorem. *If a periodic sequence $f(n)$, $n = 1, 2, \cdots$, has period $P < M$ and it satisfies the following set of relations:*

$$f(kH + a_\ell) = f(a_\ell), \qquad 0 \leq k \leq \frac{M}{\ell} - 1$$

$$\ell = 1, 2, \cdots, H$$

with $a_1 = 0$ and

$$a_{\ell+1} = a_\ell + \left(\frac{M}{\ell} - 1\right) H + 1$$

then $P|H$.

Proof: If $P = ab$ and $H = ac$ with $(b, c) = 1$ notice that the relations

$$f(kH + a_\ell) = f(a_\ell), \qquad 0 \leq k \leq \frac{P}{a} - 1$$

$$1 \leq \ell \leq a$$

are included in the above.

Define $f_\ell(m) = f(am + a_\ell)$ and note that f_ℓ has a period dividing b, i.e., $f_\ell(m + \lambda b) = f_\ell(m)$ now

$$f_\ell(kc) = f_\ell(0) \qquad 0 \leq k \leq \frac{P}{a} - 1 = b - 1$$

since

$$f_\ell(kc) = f(kca + a_\ell) = f(a_\ell) = f_\ell(0)$$

observe $\{kc\} \equiv \{0, 1, \cdots, b - 1\}$ mod b because if $k_1 \neq k_2$

$$k_1 c - k_2 c \neq \alpha b$$

because $(b, c) = 1$ and $|k_1 - k_2| < b$

$$\therefore f_\ell \text{ is constant for } 1 \leq \ell \leq a$$

By definition $a_\ell \equiv \ell - 1 \pmod{a}$ so that if $y \equiv z \pmod{a}$ any such y can be expressed as $y = y_1 a + a_\ell$ for some a_ℓ and $z = z_1 a + a_\ell$

$$\therefore f(y) = f_\ell(y_1) = f_\ell(z_1) = f(z)$$

which means that the period of f divides a, i.e.,

$$P|A \Rightarrow b = 1 \Rightarrow P|H$$

V. Consequences of Theorem

The first set of tests for a given hypothesis H is given by

$$f(kH) = f(0) \qquad 0 \leq k \leq M - 1$$

which amounts to the Mode I algorithm described above. The Rodemich Theorem now says to move over 1 bit in the sequence, i.e., starting at $a_2 = (M - 1)H + 1$ and verify that

$$f(kH + a_2) = f(a_2) \qquad 0 \leq k \leq \frac{M}{2} - 1$$

i.e., only do half as many tests as were done the first time. After this move over one bit and do $M/3$-1 tests, then $M/4$-1 etc. Thus the total number of observed bits to verify that $P|H$ is

$$T = \sum_{\ell=1}^{H} \left(\frac{M-1}{\ell}\right) H + 1 \approx MH \ln H$$

But due to the way in which the hypotheses are formed, i.e., starting at $H = 1$ and incrementing by one each time any test fails, the first H found is actually P. Thus

$$T \approx MP \ln P$$

which is the lowest value found to date for this quantity.

VI. Calculation Time in Mode I

The calculation time for H to go from 1 to P in mode I can be significant. The time is not only a function of P but of the structure of the particular sequence. A lower bound for the length of time can be calculated for sequence with only one 1–0 transition. In this case all hypotheses except the correct one fail the first test in the series. Since the device then waits for the 1–0 transition (or P time units) to test the next hypothesis this minimum time is approximately $(P-1)^2$ units. Actually a pseudorandom sequence with a probability of 1/4 of finding a 1–0 window is slightly faster and has a Mode I computation time of approximately

$$T_{M_I}, (Pn) \approx \sum_{n=1}^{P} \left[\left(\frac{4}{3} n + 4\right)\right]$$

$$= 4P + \frac{4}{3}\frac{P(P+1)}{2} = \frac{2P^2}{3} + 4\frac{2}{3} P$$

The worst-case sequence is not known, but the following example takes particularly long.

Consider the sequence

$$f(0) = f(2) = f(4) = \cdots = f(M-4) = 0$$
$$f(1) = f(3) = f(5) = \cdots = f(M-1) = 1$$

and $f(M-2) = 1$, i.e.,

n	0	1	2	3	4	5	$\cdots$	$(M-4)$	$(M-3)$	$(M-2)$	$(M-1)$
f_n	0	1	0	1	0	1	$\cdots$	0	1	1	1

Every even hypothesis for this sequence will look good and conceivably pass most of the series of tests. Thus the upper bound on the total acquisition time is given by

$$T_{\text{MAX}} \approx (M-1) \sum_{h=1}^{M/2} \sum_{\ell=1}^{h} \frac{2h}{\ell} \approx \frac{M^3}{4} \ln M$$

and for $M = 10^4$ as in the final design T_{MAX} could be on the order of 10^{12} clock periods of the sequence.

VII. Hardware

Figure 1 shows a block diagram of the DIPDOT. There is a hypothesis register and two countdown circuits which deliver pulses at a rate determined by the number held in the hypothesis register. Two countdown circuits were used so that one of them may be held fixed during Mode II to provide a useful sync to the scope earlier than if only one device were used. This second countdown circuit is not completely extraneous because it is the phase difference between its output and the output of the first countdown circuit that enables the $\div \ell$ feature of the Rodemich method. During Mode I, the two countdown networks are held in the same phase, and the $\div \ell$ flip-flop sets on CD1 = 1 and resets on CD2 = 0; i.e., one time unit later so that only one clock pulse gets to the M counter every H clock periods. In Mode II, which is entered when the M counter reaches full scale for the first time, one clock pulse is deleted from the CD1 circuit, and a new one-bit sample of the sequence is taken at this new phase. The $\div \ell$ flip-flop now is set for 2 clock pulses every H times, causing the M counter to count twice as fast as it did in Mode I. After $M/2$ observations have been made, the M counter reaches full-scale, causing CD1 to shift over another unit in the sequence, a new sample to be taken and the $\div \ell$ flip-flop to stay up three time periods every time CD2 reaches 1. In general then, M/ℓ samples are taken at the ℓth iteration, in accordance with the above theorem.

When the CD1 and CD2 outputs finally get back to their original phase, it means that all the prescribed tests are finished and that the hypothesis has been verified. The completion of this verification is communicated to the operator of the device by the shutting off of the decimal point in front of each digit of the digital readout.

A. Start Sequence

Since the DIPDOT never reduces the number in the hypothesis register, a start button is provided to restart the search. The start button produces the following sequence of events: all registers are reset to zero, then a single pulse is supplied to the hypothesis register to advance its count to 1, and sequence clock is supplied only to the M counter. The system then checks to see if the sequence actually does have period 1, i.e., that all bits are equal. If two different bits are found (i.e., an adjacent 1–0 combination) the start sequence is over and operation, described in the foregoing as Mode I, starts. If the sequence does have all bits equal, this start mode will never be terminated, but after M consecutive equal bits have been observed, the decimal points on the display will go out, signifying that the period has been verified.

It should be noted at this point that the machine actually never stops checking the input sequence and that, if it has verified period 1 and at some later time the period changes, the device will automatically find the new period if it is less than M.

B. Return to Mode I

Actually many sequences can pass all the tests of Mode I with a wrong hypothesis. This results in the discovery of an error in Mode II which entails a slightly different sequence of events to occur than if this happened in Mode I. Actually it is very much like the start sequence in that all the registers except the hypothesis register have to be reset. This realigns the CD1 and CD2 circuits to their original phase and puts the system back into Mode I.

VIII. Improved Methods of Period Detection

It is obvious that the method used for period detection can be improved at the cost of system complexity. If each test, described above, tested N consecutive bits of the sequence, the search time in Mode I would obviously be less than at present (especially if the N-bits examined were constrained to have at least one 1–0 combination), and the time to verify (Mode II) the hypothesis would be divided by at least N. The major drawback to such a design is that all periods less than N would become special cases in the logic design of the device.

Other improvements can easily be thought of, e.g., checking the parity of the number of *ones* in the hypothesized period as well as looking at the bits every H time units. Every approach of this type examined to date seems only to enable some new sequence to be found that would cause the calculation time in Mode I to remain excessive.

IX. The Prime Method

A completely different approach to the problem would utilize a property of prime numbers. An easily proved theorem is the following:

THEOREM. *If M is a prime number $> P$ then the set*

$$\{kM\} \equiv \{0, 1, 2, \cdots, P-1\}$$

mod P, $k = 0, 1, 2, \cdots, P-1$.

Proof. Suppose $k_1M \equiv k_2M$ mod P, i.e.,

$$k_1M = \alpha_1P + B_1$$
$$k_2M = \alpha_2P + B_1$$

then

$$P(\alpha_1 - \alpha_2) = M(k_1 - k_2)$$

and since $P \nmid M$ and $|k_1 - k_2| < P$, this implies

$$\alpha_1 = \alpha_2$$

This theorem implies that the set of equations

$$f(kM) = f(kM + H) \qquad k = 0, \cdots, M-1$$

reduce mod P for any $P < M$ to the set of relations

$$f(0) = f(H)$$
$$f(1) = f(H+1)$$
$$\cdot$$
$$\cdot$$
$$f(H-1) = f(2H-1)$$

so that if H is the lowest number that satisfies this set of relations we have by definition $H = P$ for the sequence in question. This theorem implies that a period detector could be built that takes a new sample of an input sequence, e.g., every 10,007 (the smallest prime greater than 10^4) time units, verifies that $f(10{,}007 \cdot k) = f(10{,}0007 \cdot k + H)$ for $k = 1, 2, \cdots, 10{,}007$ and if all tests are satisfied, the period is verified. The time to verify a given hypothesis is seen to be approximately

$$T \cong M^2$$

which is independent of the period and a smaller time than the present design if $P > 1382$. For periods in the range of 10^4 the prime machine approaches 9.2 times (i.e., ln 10^4) the speed of the present design.

X. Conclusions

The DIPDOT was designed in support of the Viterbi Decoder project (Ref. 1) and was used extensively in the debugging stage of that project. Since the decoder uses 10-bit serial arithmetic, many small period data sequence inputs would lead to arithmetic register periods of 1024 nodes (10240 bits) or some multiple thereof. By looking at the sign bit of these circulating numbers (using a word marker as clock) the DIPDOT was able to obtain sync to display extremely long bit streams. At one point in the debugging, it appeared that the decoder had a hardware malfunction, but by obtaining the proper sync on a bit stream, it was found that an oversight in the design had permitted the machine, when first turned on, to enter and hang up in an undesired, incorrect mode of operation.

In summary, when a digital machine is misperforming, some part or parts of it are not operating with their designed periods, and a device such as the DIPDOT is essential in order to give a proper oscilloscope display of what is happening. Figures 2 and 3 are photographs of the DIPDOT assembly. Figure 2 is the original prototype which was later modified for box mounting with integral power supply.

Reference

1. Lushbaugh, W. A., "Information Systems: Hardware Version of an Optimal Convolutional Decoder," in *The Deep Space Network Progress Report*, Technical Report 32-1526, Vol. II, pp. 49–55. Jet Propulsion Laboratory, Pasadena, Calif., April 15, 1971.

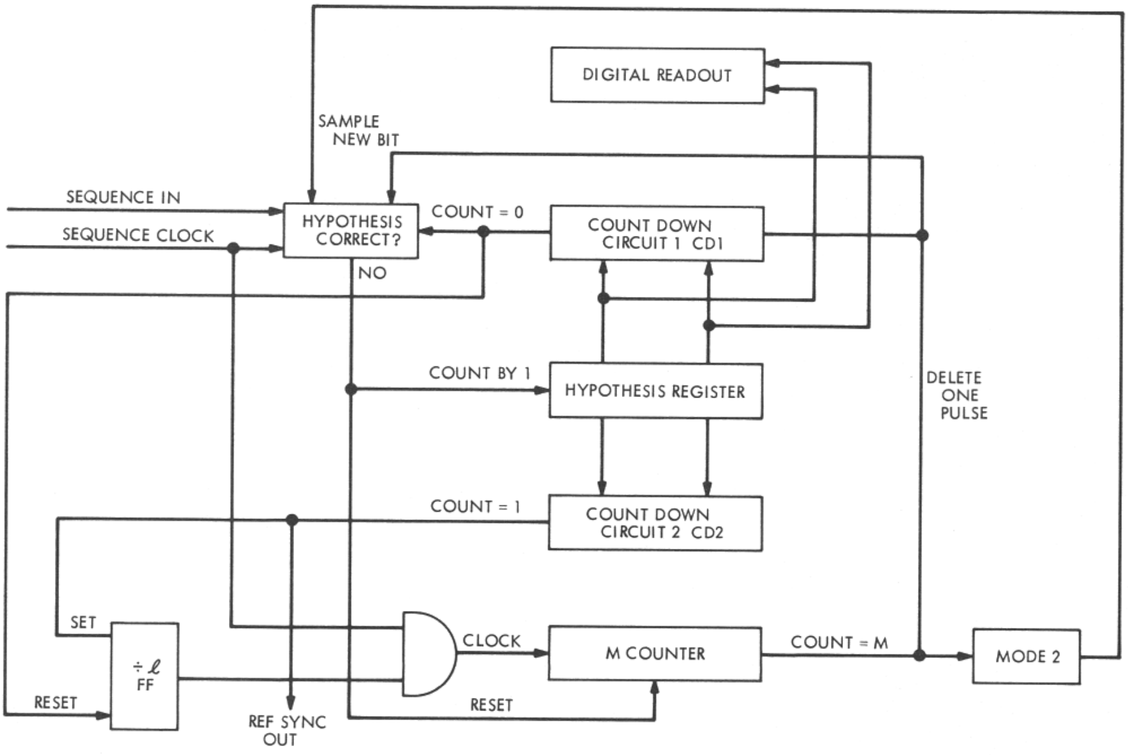

Fig. 1. DIPDOT block diagram

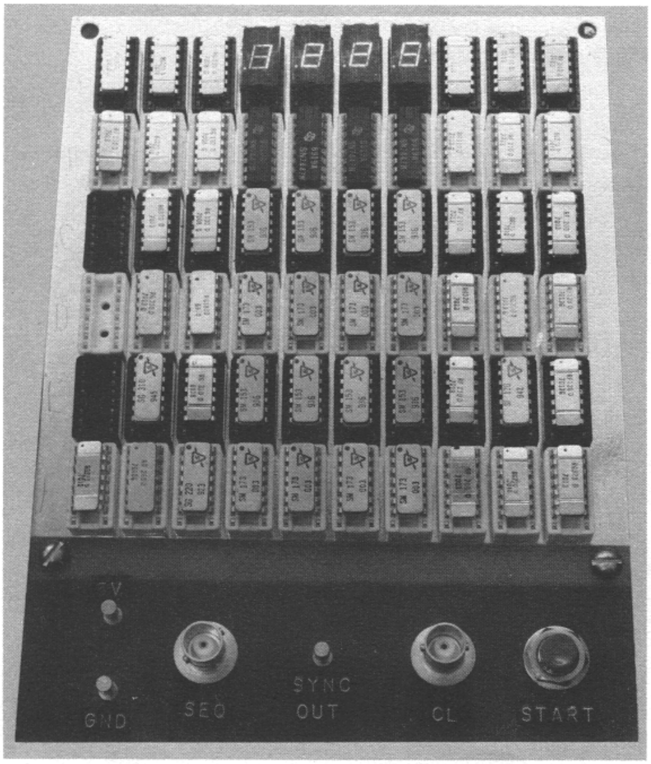

Fig. 2. DIPDOT prototype assembly

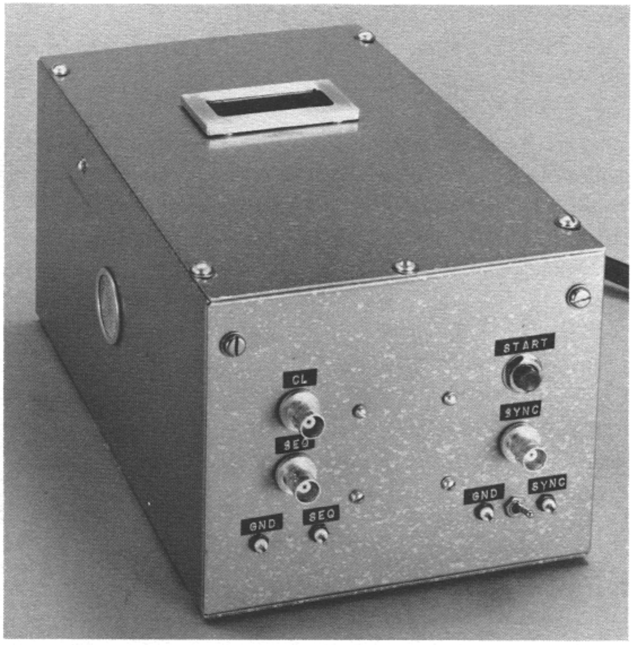

Fig. 3. DIPDOT assembly with box mounting and integral power supply