AUSTRALIAN POST OFFICE

ETP 0073

TECHNICAL TRAINING PUBLICATION

Engineering Training Group, 6 Flinders Street, Melbourne, Victoria, 3000

# ELECTRONIC LOGIC PRINCIPLES 2

PUBLISHED 1970

(PREVIOUSLY HGP 802)



## 1. INTRODUCTION.

**1.1** The paper "Electronic Logic Principles 1" shows how electronic logic elements are used to make decisions according to the laws of logic. In addition to making decisions, systems using logic elements must be able to control the order in which decisions are made, and store information in binary form.

**1.2** The logic elements used for controlling and storing information in logic systems are,

- Astable Multivibrators, which generate pulses to control time relationships.
- Monostable Multivibrators, which are used to determine pulse lengths.
- Bistable Multivibrators, which store information, or remember decisions.

Of these, bistable multivibrators (generally termed "flip-flops"), have the widest application.

A number of different types of flip-flops exist, and functional diagrams are used in this paper to distinguish between the different operating characteristics of each. A functional diagram is a logic diagram which indicates the logic function performed, but does NOT necessarily indicate the actual circuits or components used.

**1.3** Flip-flops are often arranged in groups to form counters and registers.

- Counters are used to count pulses applied to their inputs, and their outputs provide information indicating the number of pulses received.
- Registers are used to store binary information temporarily, until it is transferred from one section of a system to another.

This paper explains the functional operation of flip-flops, and shows how they are connected to form counters and registers.

The functions represented by the block symbols must be understood thoroughly, so that the operation of logic systems can be followed, and fault conditions analysed.

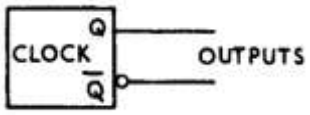
The symbols used in this and other papers of the series are those, which at the time of writing (1970), have been adopted by the A.P.O. from a recommendation by the Standards Association of Australia.

## 2. MULTIVIBRATORS.

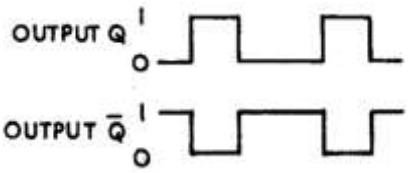
**2.1** GENERAL. A multivibrator is an electronic circuit which has two complementary outputs; when one output is at the voltage level representing logic 1, the other is at the voltage level representing logic 0, and vice versa. Multivibrators are classed as either astable, monostable or bistable, depending on the conditions required to reverse the logic levels at their outputs.

**2.2** ASTABLE MULTIVIBRATORS. An astable multivibrator has no input. The circuit oscillates between two semi-stable states, at a rate determined by the values of C and R in two delay circuits. Complementary outputs of repetitive square or rectangular waveforms are available from the outputs. The symmetry of the output waveforms is dependent on the relationship between the time constants of the two delay circuits. The symbol for an astable multivibrator is shown in Fig. 1, together with typical rectangular wave outputs. The outputs of the symbol are designated Q and $\overline{Q}$ (Q NOT), which indicates that the logic condition on output $\overline{Q}$ is complementary to the logic condition on Q. For example, when Q is logic 1, $\overline{Q}$ is logic 0, and when Q is logic 0, $\overline{Q}$ is logic 1. The state indicator (small circle) on the output of multivibrator symbols signifies that the $\overline{Q}$ output is always in the opposite state to the Q output.

Astable multivibrators are sometimes called "clocks" and are used as square or rectangular wave oscillators in logic equipment. In some systems, a single master astable multivibrator generates clock pulses which are distributed throughout the system to synchronise the transfer of information.




(a) Symbol.

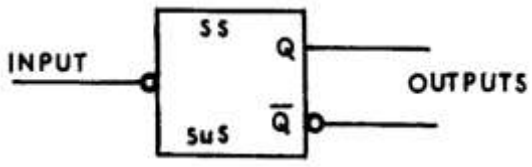



(b) Rectangular wave OUTPUTS.

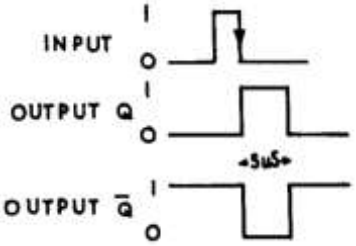
FIG. 1. ASTABLE MULTIVIBRATOR.

**2.3** MONOSTABLE MULTIVIBRATORS. A monostable multivibrator rests in its one stable state, with output Q at logic 0 and output $\overline{Q}$ at logic 1. A pulse on the input causes the outputs to change their state for a period of time determined by the values of C and R in an internal delay circuit. At the end of this predetermined time the multivibrator returns to its stable state. Fig. 2 shows the symbol for a monostable multivibrator and the output waveforms obtained when a pulse is applied to the input. Included in the symbol are the initials SS, which stand for single shot (a name sometimes given to monostable multivibrators), and a figure indicating the time for which the circuit is active. The small circle on the symbol input is a "state indicator", which shows that this multivibrator is activated, or triggered, when the input voltage level goes from logic 1 to logic 0. This is the trailing edge of the input pulse and is indicated by an arrow in Fig. 2b.

A monostable multivibrator may also be triggered on the leading edge of the input pulse, that is, when the input pulse goes from logic 0 to logic 1; in this case no state indicator is shown on the input. Sometimes only one of the monostable multivibrator outlets is used and this is represented with a symbol having one output, as shown in Fig. 3a.




(a) Symbol.




(b) INPUT and OUTPUT Waveforms.

FIG. 2. MONOTABLE MULTIVIBRATOR.

Monostable multivibrators are used in logic equipment to produce single square pulses of some desired length. In some cases, monostables are connected in chains, as shown in Fig. 3a, so that they generate a series of time related pulses to control some sequence of events.

In Fig. 3a, a pulse applied to the input activates each monostable in turn, and produces time related pulses at the outputs of monostables SS1, SS2 and SS3. The timing chart in Fig. 3b shows the relationship between the pulses generated. Monostable chains are used in the sequence control circuits of some mail handling equipment.

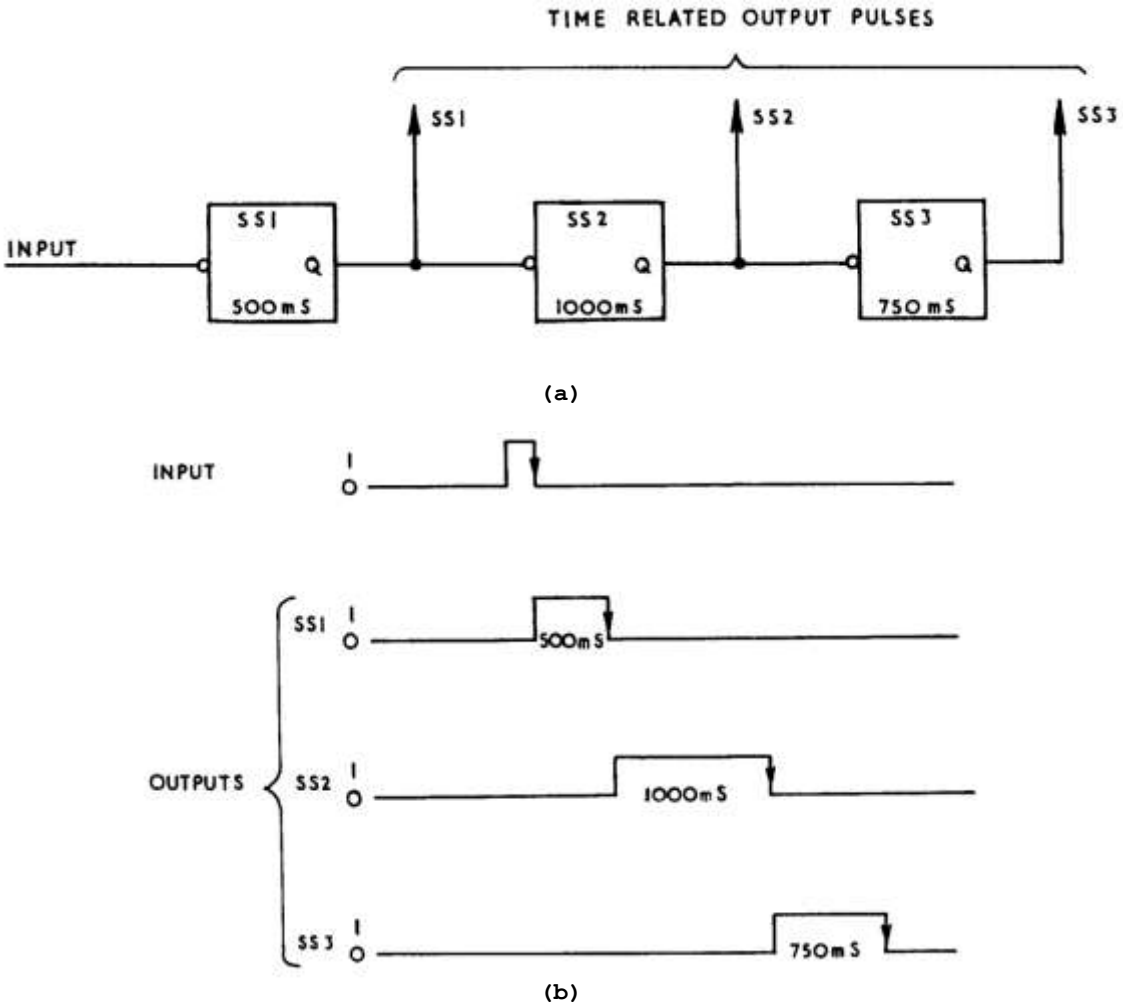



FIG. 3. MONOSTABLE SEQUENCE CONTROL.

**2.4 BISTABLE MULTIVIBRATORS.** Bistable multivibrators have two stable states, and will remain in either state until changed to the other by a triggering signal. For example, assume that a bistable multivibrator is at rest with output Q at logic 1 and output $\overline{Q}$ at logic 0. The outputs will remain in this condition until a triggering signal causes output Q to become logic 0 and output $\overline{Q}$ to become logic 1. The outputs will remain in this state until another triggering signal causes them to revert back to the original condition again, and so on. Because they have two stable states, and will remain in either state indefinitely, bistable multivibrators "remember" the last triggering conditions that occurred on their inputs.

Bistable multivibrators are used extensively in electronic logic equipment as memory elements in registers and counters, or to store information in binary form. They are generally called "flip-flops" in logic applications, although the term "toggle" is also used.

Many variations of flip-flop circuits are encountered in logic equipment. All have two stable states, but they differ in that they require different input conditions to cause them to change from one stable state to the other. The various types of flip-flop, and their application in logic equipment, are discussed in detail in the subsequent sections of this paper.

## 3. SET RESET (SR) FLIP-FLOPS.

**3.1** GENERAL. Fig. 4 is the basic flip-flop (bistable multivibrator) symbol; inputs are added (see Fig. 5) depending on the type of flip-flop required. A flip-flop is either in the SET condition or the RESET condition. It is set when the set output, designated Q, is at logic 1 and reset when the reset output, designated $\overline{Q}$, is at logic 1.

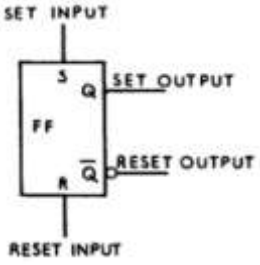



FIG. 4. BASIC FLIP-FLOP SYMBOL.

**3.2** SET-RESET (SR) FLIP-FLOP. (Fig. 5). This types of flip-flop has two inputs designated S (Set Input) and R (Reset Input), together with two outputs designated Q (Set Output) and $\overline{Q}$ (Reset Output). The logic conditions applied to the input determine whether the Q and $\overline{Q}$ outputs take up an active (Set) state, or an inactive (Reset) state.

When Q is logic 1 and $\overline{Q}$ is logic 0, the flip-flop is in an active or "Set" state.

When $\overline{Q}$ is logic 1 and Q is logic 0, the flip-flop is in an inactive or "Reset" state.

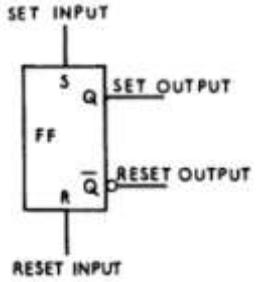



FIG. 5. SR FLIP-FLOP SYMBOL.

When the power is switched on, the flip-flop can take up either a Set or Reset state, but in many cases, associated circuitry is provided to ensure that it initially assumes a Reset state.

Assuming the Reset condition, the inputs S and R are at logic 0, and the outputs Q and $\overline{Q}$ are at logic 0 and logic 1, respectively.

To SET the flip-flop a pulse of logic 1 is applied to the S (Set Input) lead.

In the SET condition Q becomes logic 1 and $\overline{Q}$ becomes logic 0. These conditions remain even after the input on S reverts to logic 0.

To RESET the flip-flop a pulse of logic 1 is applied to the R (Reset Input) lead.

In the RESET condition Q reverts to logic 0 and $\overline{Q}$ reverts to logic 1. These conditions remain until another logic 1 is received on S. In this way the flip-flop "remembers" whether the last pulse of logic 1 occurred on the S or R input. When logic 1 pulses are applied simultaneously to both inputs, the output condition is indeterminate; however this normally does not happen.

**3.3** FUNCTIONAL LOGIC DIAGRAM OF SR FLIP-FLOP. The internal operation of a flip-flop can be explained by examining either its circuit, or its functional logic diagram. In this paper, functional logic diagrams using basic gates and inverters are used. Fig. 6 shows that an SR flip-flop can be made from two OR gates and two inverters.

To set the flip-flop a pulse of logic 1 is applied to the set input and the reset input remains at logic 0. The top OR gate opens to extend logic 1 to the input of inverter I1, Inverter I1 output, which is also the flip-flop reset output, becomes logic 0. Since both inputs to the bottom OR gate are logic 0, the input to inverter I2 is logic 0. I2 output, which is also the flip-flop set output, is logic 1 and the flip-flop is set. When the logic 1 pulse is removed from the set input the flip-flop remains set, because the logic 1 on the set output is fed back to the top OR gate to hold the circuit in this stable state.

Similarly, it can be shown that a logic 1 on the reset input will reset the flip-flop.

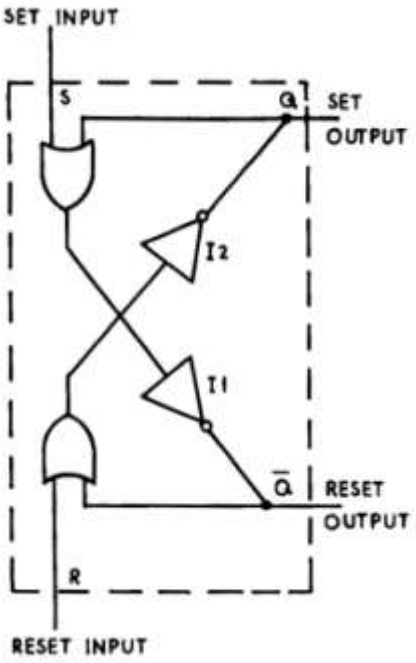



FIG. 6. FUNCTIONAL LOGIC DIAGRAM OF SR FLIP-FLOP.

**3.4** TRUTH TABLE FOR SR FLIP-FLOP. The behaviour of flip-flops is often tabulated in the form of a truth table. Table 1 shows the output conditions resulting from all possible combinations of input conditions.

| INPUTS | | OUTPUT STATE |
|---|---|---|
| S | R | |
| 0 | 0 | Unchanged |
| 0 | 1 | Reset |
| 1 | 0 | Set |
| 1 | 1 | Indeterminate |

TABLE 1. TRUTH TABLE FOR SR FLIP-FLOP.

**3.5** NOR GATES AS SR FLIP-FLOP. Fig. 7 shows how two NOR gates (which combine the OR and NOT functions in the one circuit element) are connected to operate as an SR FLIP-FLOP. The operation of this circuit is similar to that described for Fig. 6.

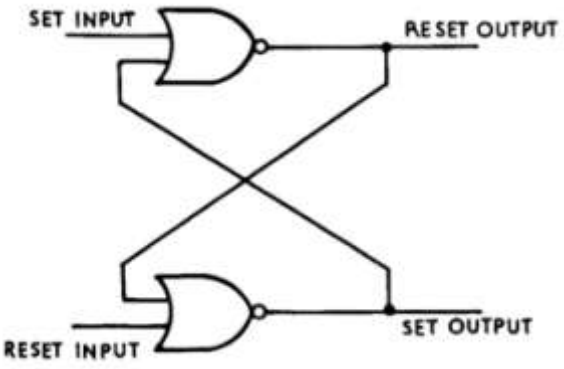



FIG. 7. NOR GATES AS SR FLIP-FLOP.

**3.6** NAND GATES AS SR FLIP-FLOP. Fig. 8a shows how two NAND gates are connected to form an SR

flip-flop. However, in the functional operation of this circuit the inputs are normally at logic 1 and the setting and resetting is done with pulses of logic 0. For example, assume that the flip-flop is set, that is, the set output is logic 1 and the reset output is logic 0. To reset the flip-flop it is necessary for the reset input to go to logic 0. This changes the output of G2 (reset output) to logic 1. Since both inputs of G1 are then at logic 1, its output (the set output) goes to logic 0, and the flip-flop is reset.

Fig. 8b is the symbol for an S.R. flip-flop requiring logic 0 pulses to set and reset it. The state indicators on the inputs show that the circuit is activated when the input goes to logic 0.

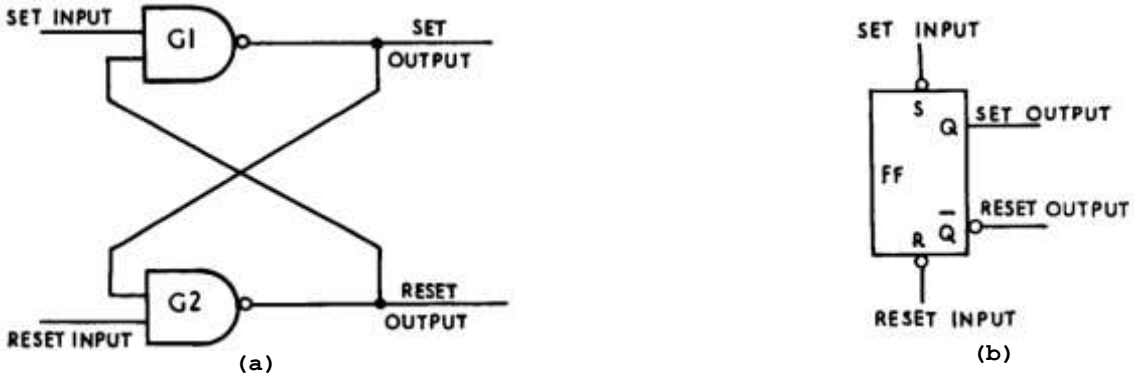



FIG. 8. NAND GATES AS SR FLIP-FLOP.

The truth table for a NAND gate SR flip-flop is given in Table 2.

| INPUTS | | OUTPUT STATE |
|---|---|---|
| S | R | |
| 1 | 1 | Unchanged |
| 1 | 0 | Reset |
| 0 | 1 | Set |
| 0 | 0 | Indeterminate |

TABLE 2. TRUTH TABLE FOR NAND GATE SR FLIP-FLOP.

## 4. PRINCIPLES OF FLIP-FLOPS.

**4.1** The flip-flops considered in section 3 are called "unclocked" (or asynchronous) flip-flops, because they set or reset immediately the required signal conditions are applied to the inputs. However most electronic logic systems use "clocked" (or synchronous) flip-flops. Clocked flip-flops do not set or rest immediately the setting or resetting inputs are applied. Instead, the change of state of the flip-flop is delayed until a clock pulse (sometimes called a "strobe" pulse) is applied to an extra input known as the "clock input".

Clock pulses may be produced by an astable multivibrator or by some other combination of logic elements. The pulses may be of regular occurrence or may depend on some event which occurs at irregular intervals.

Clocked flip-flops are used in electronic logic systems for the following reasons.

- To overcome the problem introduced by "propagation delay" through the system.
- To ensure that information is stepped or transferred through a system in a set order.

**4.2** PROPAGATION DELAY. Modern electronic logic systems operate at very high speeds, and even the small delays introduced by transistors switching from one state to the other have to be considered in their design. An examination of the circuit in Fig. 9 highlights the problem these delays introduce when unclocked flip-flops are used.

Firstly, consider the theoretical operation of the circuit in Fig. 9, assuming that no delays exist and that all flip-flops are reset. When FF1 sets, gates G1 and G2 extend logic 1 to set FF2 and gates G3 to G6 extend logic 1 to set FF3. Since FF2Q is at logic 1 and FF3$\overline{Q}$ is at logic 0, gate G7 does not open and flip-flop FF4 stays reset. FF4 will only set if FF3 does not set, or if it sets later than FF2 (i.e. when FF2Q and FF3$\overline{Q}$ are logic 1 at the same time).

Now consider the practical operation of the circuit, taking into consideration the small delays introduced by each gate and a requirement that FF4 should set only when G7 is activated from a set FF2 and a reset FF3. Because the delay introduced by gates G1 to G2 is shorter than that introduced by G3 to G6, FF2 and FF3 do not set at the same time. FF2 sets before FF3, causing gate G7 to open long enough to set FF4. The setting of FF4 could indicate that FF2 has set and FF3 has not set, when this is not the case.

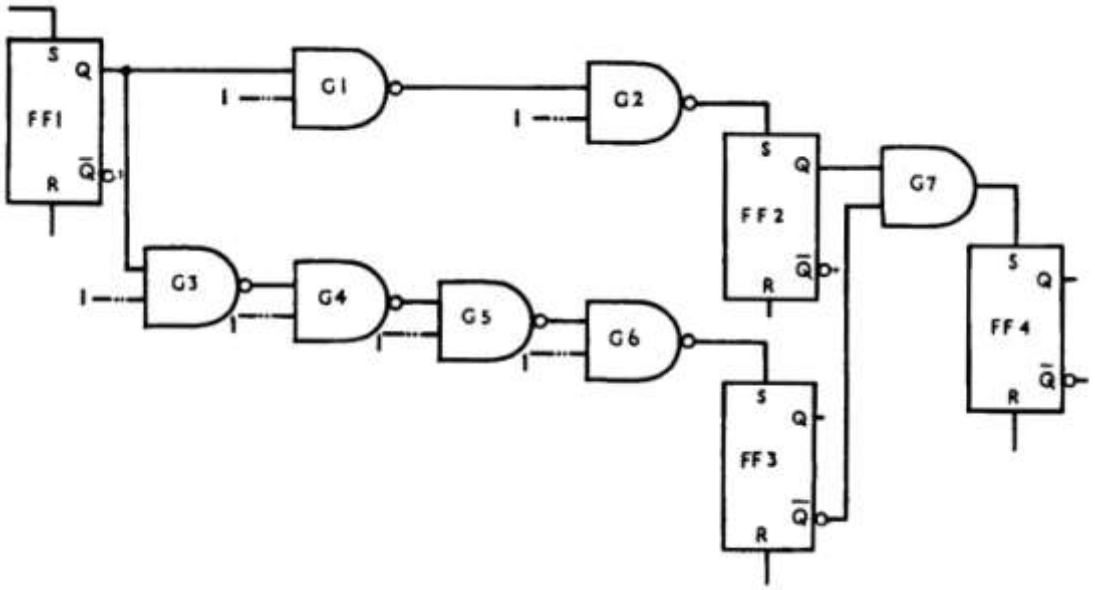



FIG. 9. UNCLOCKED FLIP-FLOP CIRCUIT.

To overcome the problem of varying delays due to switching transients or propagation delays, clocked flip-flops are used. Fig. 10 shows how the theoretical functions of the circuit in Fig. 9 are implemented using clocked flip-flops. Note that the S and R inputs are drawn on the same side of the symbol as the clock input (designated C) to show that setting or resetting of the flip-flop is conditional on a clock pulse. The clock pulses are generated by a common pulse generator circuit having a repetitive square pulse output.

Flip-flop FF1 (Fig. 10) does not set immediately the set input goes to logic 1. It sets when the next clock pulse is applied to the clock lead. The same pulse is also applied to FF2, FF3 and FF4, but since their S inputs are at logic 0 when the clock pulse arrives, they remain reset. The delay between clock pulses is long enough to allow gates G1 to G2 and G3 to G6 to switch and apply logic 1 to the S inputs of FF2 and FF3 before the next clock pulse arrives. The next clock pulse synchronises the setting of FF2 and FF3 so that they set at the same instant. FF4, therefore, cannot set and record an incorrect condition produced by propagation delays through the gate circuits.

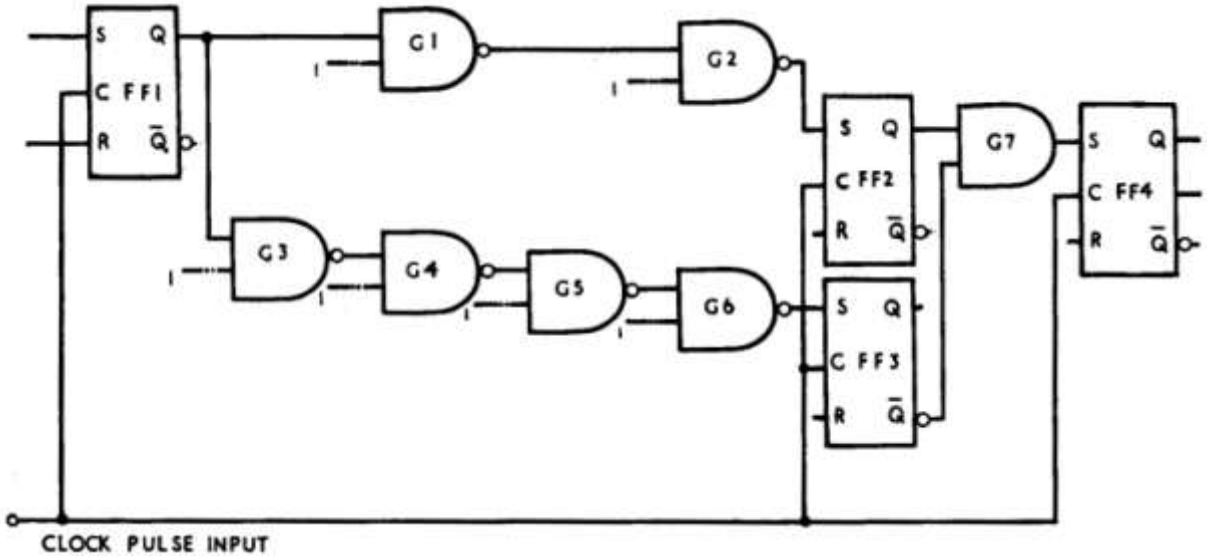



FIG. 10. CLOCKED FLIP-FLOP CIRCUIT.

**4.3** TRANSFER OF INFORMATION. Generally, logic equipment is designed to transfer information in controlled time steps, that is, one step at a time. However, the circuit shown in Fig. 11 transfers information in uncontrolled time steps. For example, if the flip-flops are all reset and a set condition is applied to the set input of FF1, flip-flop FF2 and FF3 will set also. The set output of FF1 sets FF2, and the set output of FF2 sets FF3. Therefore, the set condition applied to the input of FF1 "runs through" to FF3 and sets all the flip-flops in an uncontrolled time period.

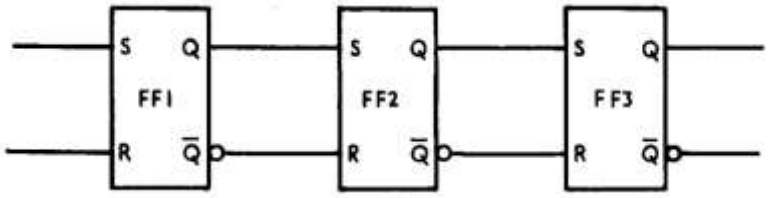



FIG. 11. UNCONTROLLED TRANSFER OF INFORMATION.

By using clocked flip-flops, as shown in Fig. 12, information can be transferred in controlled steps. For example, consider that all flip-flops are reset, and a set condition is applied to the set input of FF1. FF1 does not set until a clock pulse is received on its clock input. When FF1 sets its output applies the set condition to the set input of FF2. However, FF2 does not set until the next clock pulse arrives and is applied to its clock input. When FF2 sets its output applies the set condition to the set input of FF3, which sets when the third clock pulse is received. The clocked flip-flops, therefore, allow the information received on the set input of FF1 to be transferred through the flip-flops in time controlled steps, under the control of clock pulses.

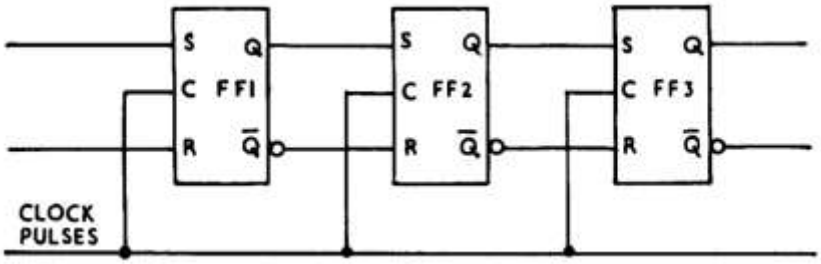
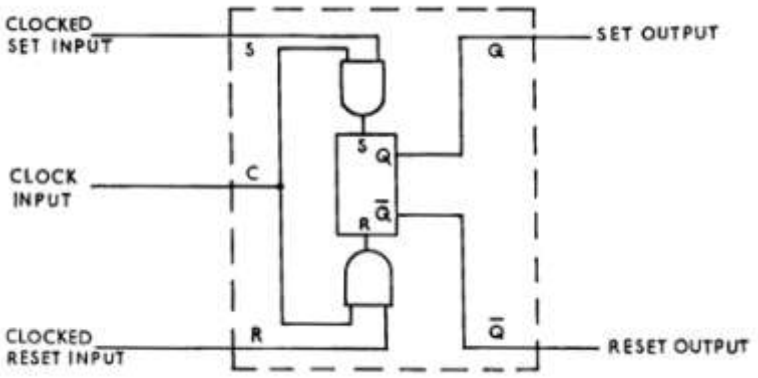



FIG. 12. CLOCK CONTROLLED TRANSFER OF INFORMATION.

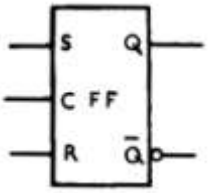
**4.4** TYPES OF CLOCKED FLIP-FLOPS. The three main types of clocked flip-flops are:-

- Clocked SR flip-flop.
- Clocked JK flip-flop.
- Clocked D flip-flop (usually called the D type flip-flop).

**4.5** BASIC CLOCKED SR FLIP-FLOP. The basic clocked SR flip-flop may be considered as containing an SR flip-flop with AND gates in the set and reset leads, as shown in Fig. 13a. A logic 1 connected to either the clocked set input, or the clocked reset input, is not extended to the flip-flop until a pulse of logic 1 is applied to the clock input. The logic 1 clock pulse removes the inhibit from the AND gates (by changing the input condition from logic 0 to logic 1) and allows the logic 1 on the clocked set or reset input to be extended to the SR flip-flop. The inputs are called 'clocked set input' and 'clocked reset input' to indicate that they cannot extend the logic 1 condition to the flip-flop until the clock pulse is received. If both the set and reset inputs of a clocked SR flip-flop are at logic 1 when a clock pulse arrives, setting and resetting pulses are extended simultaneously to the SR flip-flop and the output is indeterminate (unpredictable). In many circuit applications this characteristic is undesirable, and paras. 4.6 and 4.7 describe other types of flip-flop which overcome this problem. Fig. 13b shows the symbol used in logic diagrams to represent clocked SR flip-flops.



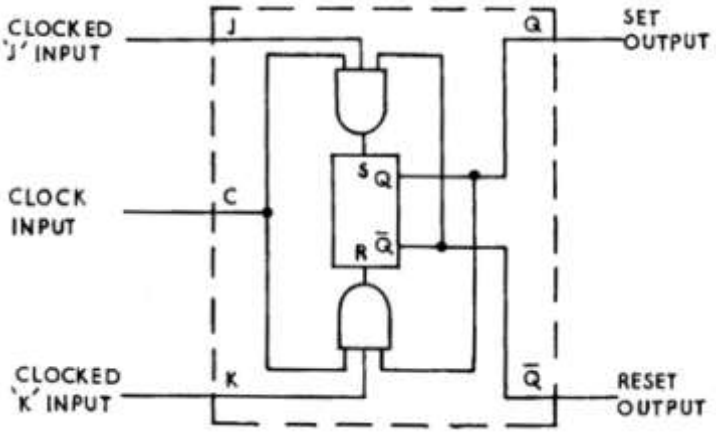

(a) Functional logic diagram.



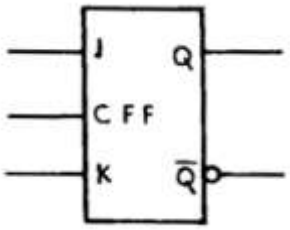

(b) Symbol.

FIG. 13. BASIC CLOCKED SR FLIP-FLOP.

**4.6** BASIC CLOCKED JK FLIP-FLOP. Figs. 14a and 14b, respectively, show the basic functional logic diagram and symbol of a clocked JK flip-flop. A clocked JK flip-flop performs all the functions of a clocked SR flip-flop, and in addition, allows the output condition to be determined if both the J and K inputs are at logic 1 when a clock pulse arrives. If both the inputs are at logic 1 when a clock pulse arrives, the JK flip-flop changes state. For example, assume that the flip-flop in Fig. 14a is set. A logic 1 from the set output is applied to the gate on the K input and a logic 0 from the reset output is applied to the gate on the J input thus inhibiting it. If both the J and K inputs are at logic 1 and the clock pulse arrives, the gate on the K input opens and the flip-flop reverts to the reset state.




(a) Functional logic diagram




(b) Symbol

FIG. 14. BASIC CLOCKED JK FLIP-FLOP.

Tables 3 and 4 show the output conditions which result when a clock pulse is applied for each of the four possible input combinations on clocked SR and JK flip-flops, respectively. Notice that the JK output is determinable when both inputs are at logic 1.

| INPUTS BEFORE Clock Pulse | | Outputs AFTER Clock Pulse |
|---|---|---|
| S | R | |
| 0 | 0 | Unchanged |
| 0 | 1 | Reset |
| 1 | 0 | Set |
| 1 | 1 | Indeterminate |

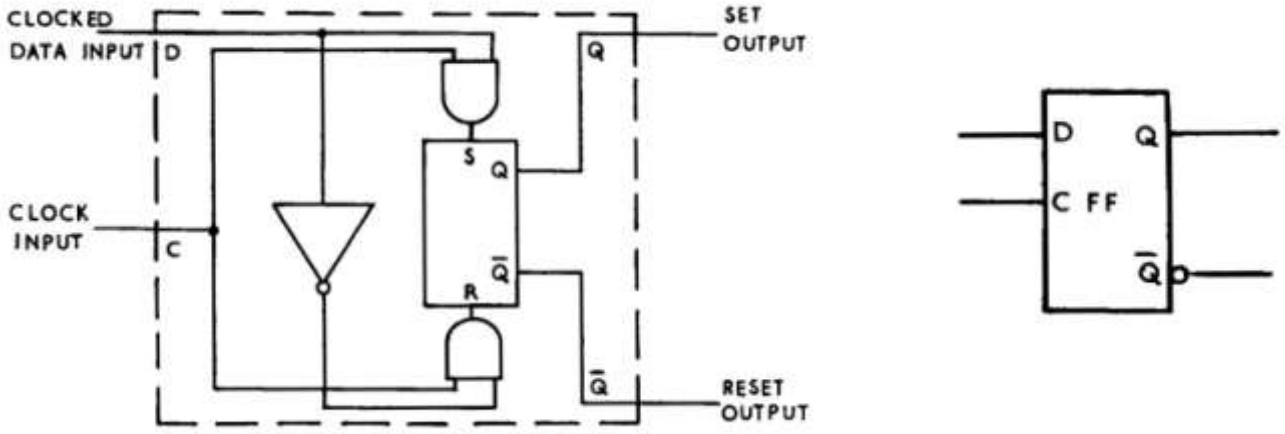
TABLE 3. CLOCKED SR FLIP-FLOP.

| INPUTS BEFORE Clock Pulse | | Outputs AFTER Clock Pulse |
|---|---|---|
| S | R | |
| 0 | 0 | Unchanged |
| 0 | 1 | Reset |
| 1 | 0 | Set |
| 1 | 1 | Changes State |

TABLE 4. CLOCKED JK FLIP-FLOP

**4.7** BASIC D TYPE FLIP-FLOPS. Fig. 15 shows the symbol and basic functional logic diagram of a clocked D flip-flop. This type of flip-flop is used in applications where a single input is used and Table 5 shows the output conditions which result when a clock pulse is applied for each input condition.

Although it has only one clocked input (designated D for Data) the D type flip-flop performs the same general function as a clocked SR flip-flop. The type D flip-flop is set by a clock pulse when the D input is at logic 1 and is reset by a clock pulse when the D input is at logic 0. For example, if the D input is at logic 1 when a clock pulse arrives the top gate extends logic 1 to set the flip-flop. If the D input is at logic 0 when a clock pulse arrives the inverter extends logic 1 to rest the flip-flop.




(a) Functional Logic Diagram. (b) Symbol

FIG. 13. BASIC CLOCKED D FLIP-FLOP.

| D input before clock pulses | Outputs after clock pulses |
|---|---|
| 0 | Reset |
| 1 | Set |

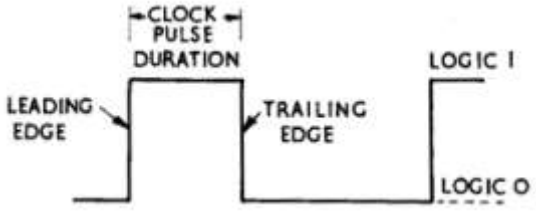
TABLE 3. D TYPE FLIP-FLOP.

## 5. FLIP-FLOPS TRIGGERED ON PULSE EDGES.

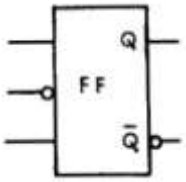
**5.1** LIMITATIONS OF BASIC CLOCKED FLIP-FLOPS. The basic clocked flip-flops shown in Figs. 13 to 15 are limited in their application because "run through" can still occur. This is due to the fact that clock pulse durations are generally longer than the switching delays of flip-flops.

For example, consider that the clock pulse duration in Fig. 12 is longer than the switching delay of FF1. As a result, the clock pulse remains on the clock inputs for a short period of time after FF1 sets, thus allowing the set output of FF1 to set FF2. Therefore, both FF1 and FF2 would set to the same clock pulse. This is prevented by arranging for the flip-flops to be activated, or triggered, on one of the edges of the clock pulse; that is, during the extremely short transition time that the pulse is changing from one state to another. Although flip-flops that are triggered on the edges of clock pulses are more complex than the basic clocked flip-flops described, their functions remain the same.

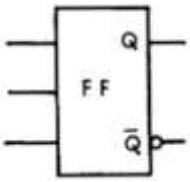
**5.2** TRIGGERING ARRANGEMENTS. Flip-flops may be designed to trigger on either the trailing edge or the leading edge of a clock pulse. Fig. 16a shows the trailing edge, leading edge, and pulse duration of a typical clock pulse. A flip-flop which triggers on the trailing edge of the clock pulse, that is, when the pulse is going from logic 1 to logic 0, has a state indicator on the clock input of the symbol, as shown in Fig. 16b. A flip-flop which triggers on the leading edge of the clock pulse, that is, when the pulse is going from logic 0 to logic 1, is shown in Fig. 16c.




(a) Clock Pulse.




(a) Triggered on Trailing Edge (1-0 transition)




(b) Triggered on Leading Edge (0-1 transition).

FIG. 16. TRIGGERING EDGES OF CLOCK PULSE.

**5.3** FLIP-FLOPS TRIGGERED ON THE TRAILING EDGE. Fig. 17 shows the basic principle of a flip-flop which triggers on the trailing edge of the clock pulse. Each of the clocked inputs has a capacitor associated with it, and for this reason, the flip-flop is called a "capacitive-memory" flip-flop. In general, capacitive memory triggering circuits are used in clocked flip-flops constructed from discrete components. Consider that one clocked input is at logic 1 and the other is at logic 0 when the clock pulse arrives. The capacitor associated with the input at logic 1 charges during the clock pulse, thus remembering the condition on the input. On the trailing edge of the clock pulse the charge on the capacitor is transferred to the appropriate input of the slave flip-flop, which sets or resets accordingly. Note that this circuit will not operate satisfactorily when the conditions on the clocked inputs change during the clock pulse.

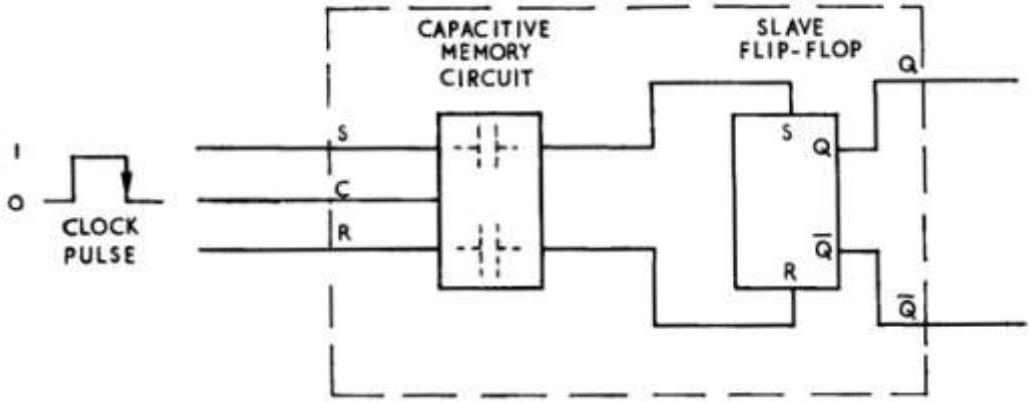


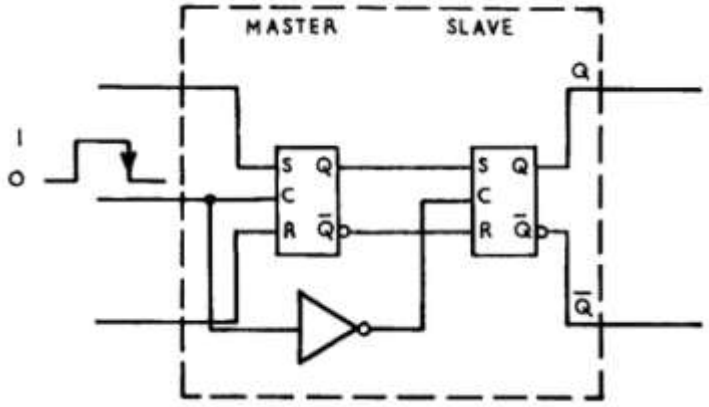

FIG. 17. PRINCIPLES OF CAPACITIVE-MEMORY FLIP-FLOP.

Another type of flip-flop triggered on the trailing edge is shown in Fig. 18. In this flip-flop the capacitive memory circuit is replaced by a master flip-flop stage, which remembers the condition of the clocked inputs during the clock pulse. These flip-flops are called "master-slave" flip-flops and were developed because of the difficulty of including capacitors in integrated circuitry.

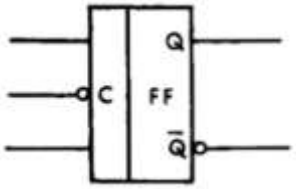
Both the master and the salve flip-flops are functionally the same as the SR clocked flip-flop described in para. 4.5. Fig. 18c is a timing diagram which shows the principle of operation of the master-slave flip-flop.

When the clock pulse goes to logic 1 the inverter applies logic 0 to the clock input of the slave stage, thus isolating the two stages from each other. The logic 1 clock pulse allows the logic conditions on the clocked S and R inputs to be stored on the outputs of the master stage flip-flop. However, the outputs of the master stage flip-flop cannot effect the slave flip-flop while it has a logic 0 on the clock input. When the clock pulse goes to logic 0 (on the trailing edge of the pulse) the inverter applies logic 1 to the clock input of the slave stage. This allows the information stored on the outputs of the master stage to be transferred through the slave stage and stored on its outputs.

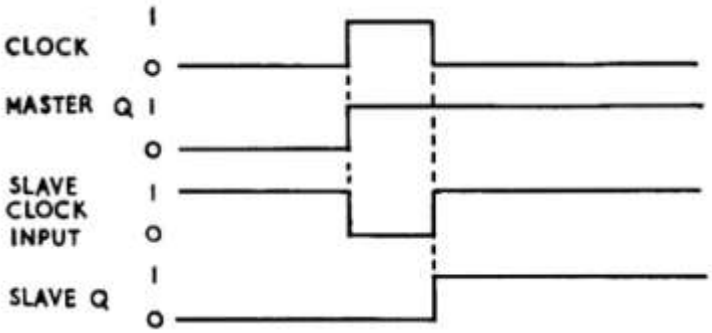
Since the master stage has logic 1 on its clock input for the duration of the clock pulse, any changes of input conditions during the clock pulse changes the conditions on the output of the master stage. Therefore, the input conditions transferred to the slave stage are those which exists at the time the clock pulse reverts to logic 0. Sometimes master-slave flip-flops are represented by the symbol shown in Fig. 18b, instead of the symbol shown in Fig. 16b.




Functional Logic Diagram.




(b) Symbol.




(a) Timing Diagram

FIG. 18. PRINCIPLES OF MASTER-SLAVE FLIP-FLOP.

In practice, the clock pulse takes a finite time to rise and fall. Use is made of this in the design of master-slave flip-flops to ensure that the operations take place in the correct order. The inputs to both stages are level sensitive and Fig. 19 shows the points in the time when the actions take place. The operating sequence is as follows:-

(a) The master is isolated from the slave.

(b) The information on the clocked inputs is stored in the master.

(c) The master input are isolated or inhibited.

(d) The information is transferred from the master to the slave.

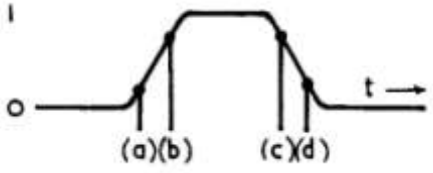



FIG. 19. CLOCK PULSE WAVEFORM.

**5.4** FLIP-FLOPS TRIGGERED ON THE LEADING EDGE. The flip-flops shown in Figs. 17 and 18 do not provide a new output until the clock pulse ends. Fig. 20 is the functional diagram of a flip-flop which gives a new output on the leading edge of the clock pulse.

While the clock input is at logic 0 (that is before the logic 1 clock pulse arrives) the inverter applies logic 1 to the clock input of the master stage. This allows the master stage to store the logic conditions existing on the clocked inputs. Since the slave has logic 0 on its clock input, it is isolated from the master stage and cannot be affected by the new condition on the master output. In effect, the master stage stores the logic conditions existing on the clacked inputs before the logic 1 pulse arrives.

When the logic 1 clock pulse arrives the clock input of the slave stage goes to logic 1, allowing the slave to store the information contained on the outputs of the master stage. This transfer of information from the master stage to the slave stage occurs immediately the clock pulse goes from logic 0 to logic 1, that is on the leading edge of the pulse. Any changes in the input conditions during the clock pulse are ineffective because the master stage clock input is at logic 0 during the period.

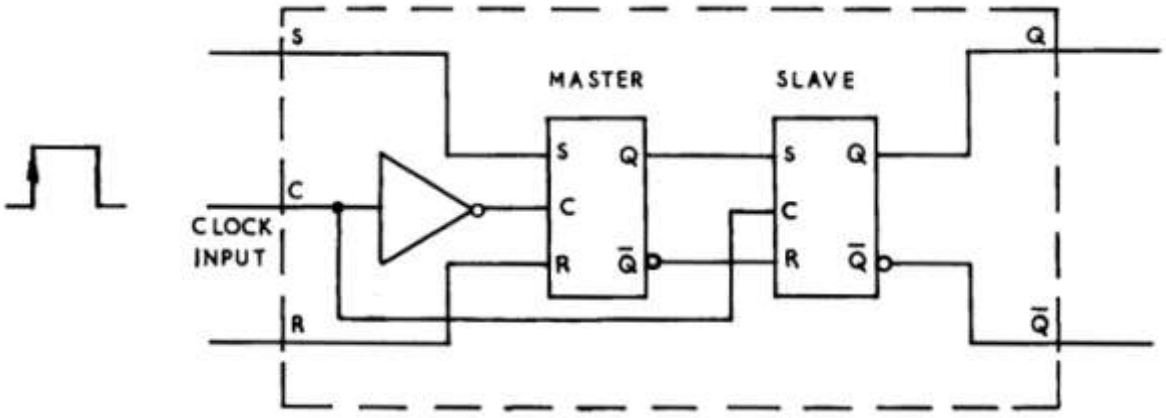



FIG. 20. PRINCIPLES OF FLIP-FLOP TRIGGERED ON LEADING EDGE.

**5.5** Clocked SR, JK or D type flip-flops may have any of the three triggering arrangements described. However, for manufacturing reasons, it is usual for clocked SR flip-flops to have a capacitive-memory circuit when constructed from discrete components and a master-slave arrangement when constructed in integrated circuit form. JK flip-flops are not normally constructed from discrete components, but a large range of JKs are available in integrated circuit form. Generally, type D flip-flops are edge-triggered. A simple form of the D type flip-flop, called a "latch", is arranged so that it does not lock out changes in input data during the clock pulse.

**5.6** MULTI-INPUT FLIP-FLOPS. Some clocked flip-flops have unclocked inputs, called "preset" and "clear" inputs, which can set and reset the flip-flop without being conditional on a clock pulse. The preset input is used to set the flip-flop and the clear input is used to reset the flip-flop. In the symbol of a multi-input flip-flop (Fig. 21), the inputs which are conditional on the clock pulse are shown on the same side of the symbol as the clock input, and those which are not conditional on the clock pulse are shown on the short sides of the symbol. The state indicators on the preset and clear inputs show that these inputs are normally at logic 1 and go to logic 0 to change the state of the flip-flop. In most flip-flops, the direct inputs override the conditions on the clocked inputs, when both are activated together.

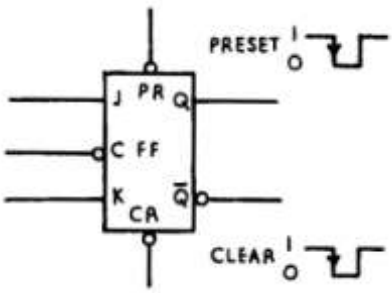



FIG. 21. MULTI-INPUT JK FLIP-FLOP WITH PRESET AND CLEAR.

**5.7** GATED INPUT FLIP-FLOPS. Many integrated circuit flip-flops have a number of input connections with access to the flip-flop via input gates. Fig. 22 is the symbol representing a gated input JK flip-flop. To set the flip-flop with a clock pulse all three J inputs must be at logic 1. Similarly, all three K inputs must be at logic 1 to reset the flip-flop with a clock pulse.

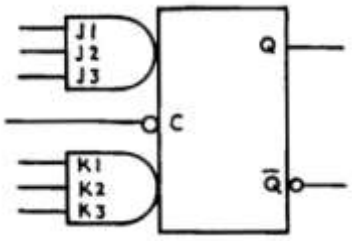



FIG. 22. JK FLIP-FLOP WITH GATED INPUTS.

**5.8** Fig. 23 is a functional logic diagram of the flip-flop symbolised in Fig. 22. Note that the master and slave stages are ordinary SR flip-flops made from NAND gates, (see Fig. 8). The input gates change the stages into clocked flip-flops, and the feedback circuits from Q to K, and $\overline{Q}$ to J, make the flip-flop a JK type (see Fig. 14). The functional diagram, therefore, represents a JK master-slave flip-flop.

Since there are four NAND gates between the inputs and outputs, a logic condition is inverted four times between the input and output. For example, assume that a logic 1 exists on all the J inputs when the clock pulse arrives. This produces a logic 0 on the output of the upper master input gate, and a logic 0 on the upper output of the master stage. At the end of the clock pulse the clock input to the slave input gate changes to logic 1. This produces a logic 0 at the output of the upper slave input gate and a logic 1 at the Q output. Therefore, a logic 1 at the J inputs sets the flip-flop at the end of the clock pulse (trailing edge).

Notice that a NAND gate is used as an inverter in the clock lead between the master and slave stages. The unused input of the NAND gate is permanently tied to logic 1, so that only the condition on the clock input controls the output condition of the gate.

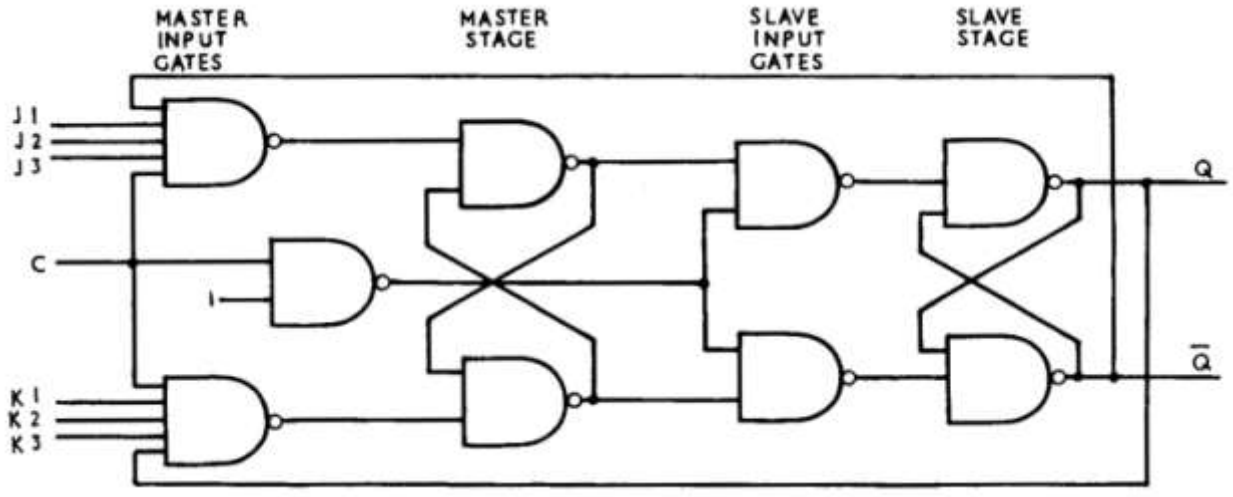



FIG. 23. FUNCTIONAL LOGIC DIAGRAM OF JK MASTER-SLAVE FLIP.

**5.9** CHANGE-OF-STATE FLIP-FLOPS. Fig. 24 shows the logic symbols used to indicate change-of-state flip-flops (sometimes called "toggles"). These flip-flops are designed to change state each time a clock pulse is connected to the clock input. For example, if the flip-flop is reset and a clock pulse is applied, it sets (see Table 4). The next pulse causes it to revert to its original reset state. Since the flip-flop returns to its original condition on every second clock pulse, it provides one pulse of logic 1 on an output for every two pulses of logic 1 on the clock input. The output frequency is therefore half that of the input frequency and the circuit is said to divide by two. This is shown in the timing diagram of the change-of-state flip-flop in Fig. 25. Note that the flip-flop represented in this diagram changes state on the trailing edge of the clock pulse.


(a) Triggered on Leading edge of clock pulse.

(b) Triggered on Trailing edge of clock pulse.

FIG. 24. CHANGE-OF-STATE FLIP-FLOP SYMBOL.

| | |
|---|---|
| Assumed Initial Conditions | Reset |
| After 1st pulse | Set |
| After 2nd pulse | Reset |
| After 3rd pulse | Set |

TABLE 4. CHANGE-OF-STATE FLIP-FLOP.

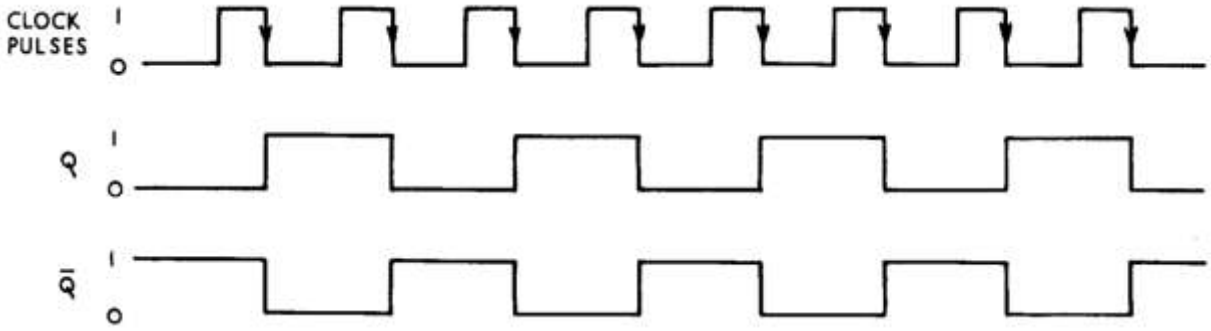
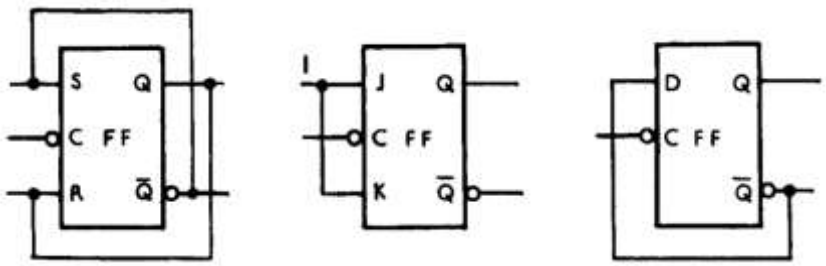



FIG. 25. TIMING DIAGRAM FOR CHANGE-OF-STATE FLIP-FLOP.

Since integrated circuit manufacturers generally do not produce special change-of-state flip-flop elements, clocked SR, JK or D type flip-flops can be used by strapping them externally as shown in Fig. 26. In fig. 26a, the cross connection of the outputs to the inputs of the SR flip-flop causes it to change state on each clock pulse. For example, if the flip-flop is set a logic 1 is connected to the reset input causing the flip-flop to reset on the next clock pulse and vice versa. In Fig. 26b both inputs of the JK flip-flop are strapped together and tied permanently to logic 1. As described in para. 4.6, a JK flip-flop changes state if both inputs are at logic 1 when a clock pulse is applied. The D type flip-flop in Fig. 26c changes state on each clock pulse because the $\overline{Q}$ lead is strapped to the D input. If the flip-flop is reset a logic 1 is applied to the D input causing the flip-flop to set on the next clock pulse, and vice versa. Change-of-state flip-flops may have preset and clear inputs so that their initial condition can be controlled.




(a) (b) (c)

FIG. 26. EXTERNAL STRAPPING FOR CHANGE-OF-STATE OPERATION.

**5.10** Another variation of the clocked flip-flop has two clock inputs. One controls the clocked set input and the other the clocked reset input. The symbol for this flip-flop is shown in Fig. 27.

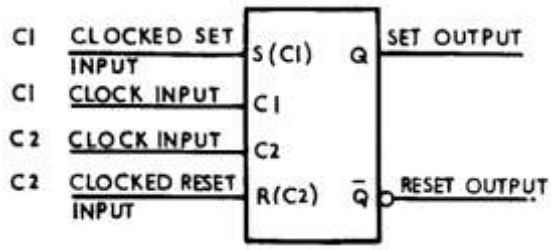



FIG. 27. CLOCKED SR FLIP-FLOP WITH TWO CLOCK INPUTS.

**5.11** PRACTICAL FLIP-FLOPS are sometimes built using the logic elements shown in the functional diagrams in this paper. However, it is more usual to find composite flip-flop circuits constructed either from discrete components, or in the form of integrated circuits. In either case the basic functional operation is the same as that explained in this paper, although the detailed circuit may vary according to the circuit techniques used.

Integrated circuit flip-flops are usually manufactured to supply all the possible facilities for the particular form of flip-flop. When multi-input flip-flops are used to perform only part of the supplied function the inputs which are not used are strapped to one of the two logic levels, or to the clock input, depending on the functional operation required.

## 6. BINARY COUNTERS.

**6.1** BASIC BINARY COUNTER. Binary counters are used extensively in electronic logic equipment for counting pulses applied to their inputs, and are composed of change-of-state flip-flops connected in cascade. Fig. 28 is a basic three stage binary counter which counts the pulses applied to its input and produces the result in binary form at the output. For example, when three pulses are applied to the input of the counter the output reads 011 (the binary number for 3). The output is obtained by noting the conditions on the Q leads of the flip-flops; a set flip-flop produces a logic 1 on the Q lead and a reset flip-flop produces a logic 0 on the Q lead. Therefore, the output of the counter reads 011 when FF3 is reset (logic 0), FF2 is set (logic 1) and FF1 is set (logic 1).

Table 5 shows the outputs obtained from the binary counter following each of 8 successive input pulses. In the conventional circuit formation of Fig. 28, note that the binary counter output reads from left to right, that is, the least significant digit of the binary number is indicated on FF3, whereas the truth table has the least significant digit in the right hand column.

**6.2** OPERATION (refer to Fig. 28 and Table 5). Before counting commences all the flip-flops are reset by a pulse of logic 1 on the clear input. The trailing edge (logic 1 to logic 0) of the first input pulse to be counted causes FF1 to set. The logic 1 on FF1Q is extended to the clock input of FF2, however, FF2 will not change state until activated by a logic 1 to 0 transition. The condition FF1Q=1, FF2Q=0 and FF3Q=0, therefore, indicates that one input pulse has been received. The trailing edge of the second input pulse causes FF1 to revert back to the reset state. FF1 in resetting produces a 1 to 0 transition on the input of FF2, causing it to set. The condition of the count output after the second pulse is received is, FF1Q=0, FF2Q=1 and FF3Q=0. The trailing edge of the third pulse sets FF1 again. Since this produces an 0 to 1 transition on the input of FF2, it is not affected and remains set. The third pulse, therefore, produces the output condition; FF1Q=1, FF2Q=1 and FF3Q=0. The trailing edge of the fourth pulse causes FF1 to reset. This produces a 1 to 0 transition on the input of FF2 which resets. The resetting of FF2 causes a 1 to 0 transition on the input of FF3 which sets. The output condition after the fourth pulse is, therefore, FF1Q=0, FF2Q=0 and FF3Q=1.

The condition of the counter outputs for subsequent pulses are shown in Table 5. A comparison of the output conditions in Table 5 and the counts in the binary number system show that the counter stages change state according to the same pattern which governs the changing of digits in the binary number system.

The number of output combinations possible and, therefore, the number of pulses which can be counted by a binary counter depends on the number of flip-flop stages used and the way they are arranged. The range of the counter in Fig. 28 can be increased by adding further flip-flop stages.

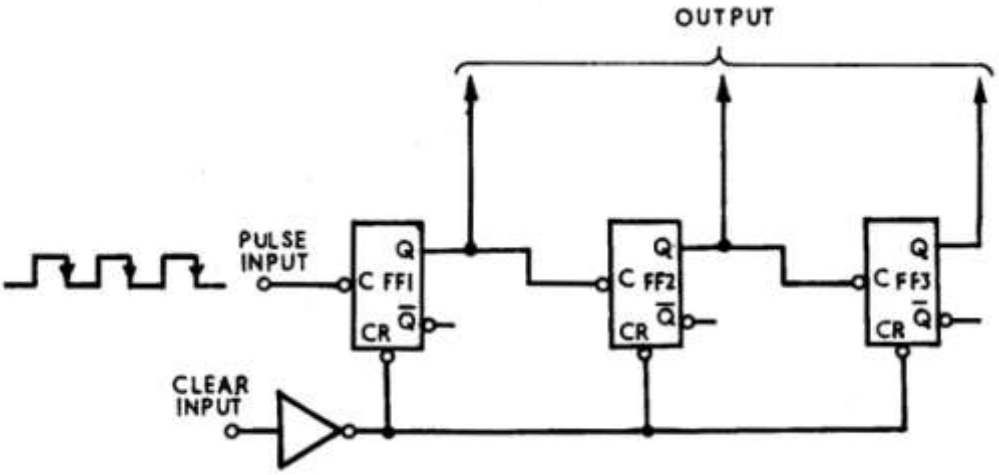



FIG. 28. BASIC THREE STAGE BINARY COUNTER.

| | Output Conditions | | | Binary Number | Decimal Equivalent |
|---|---|---|---|---|---|
| | FF3 Q | FF2 Q | FF1 Q | | |
| Before 1st pulse. | 0 | 0 | 0 | 0 | 0 |
| After 1st pulse | 0 | 0 | 1 | 1 | 1 |
| After 2nd pulse | 0 | 1 | 0 | 10 | 2 |
| After 3rd pulse | 0 | 1 | 1 | 11 | 3 |
| After 4th pulse | 1 | 0 | 0 | 100 | 4 |
| After 5th pulse | 1 | 0 | 1 | 101 | 5 |
| After 6th pulse | 1 | 1 | 0 | 110 | 6 |
| After 7th pulse | 1 | 1 | 1 | 111 | 7 |
| After 8th pulse | 0 | 0 | 0 | 000 | 0 |

TABLE 5. TRUTH TABLE FOR BINARY COUNTER.

**6.3** Fig. 28 is a timing diagram which shows the input pulses and the conditions on the set (Q) output of each stage of the binary counter in Fig. 28. Note that when pulses are applied to a binary counter the first stage changes state every input pulse, the second stage changes state every second input pulse, the third stage changes state every fourth input pulse, and so on. The arrows indicate the logic 1 to 0 edges which trigger the next stage.

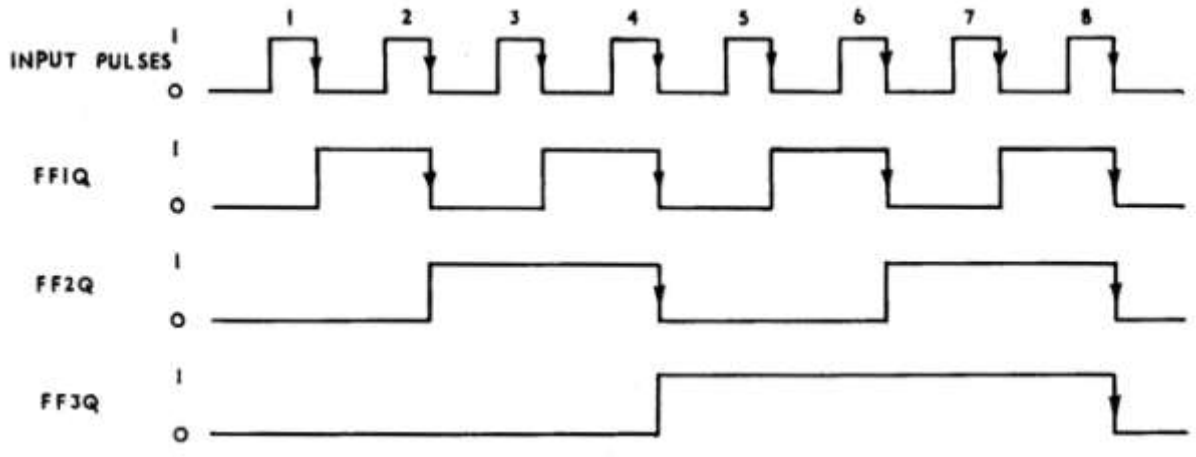



FIG. 29. TIMING DIAGRAM FOR BASIC THREE STAGE BINARY COUNTER.

**6.4 BACKWARD COUNTERS.** Table 5 shows that the output conditions for the counter in Fig. 28 change in accordance with binary numbers when counting forward (or up). For this reason, it is called a forward counter.

A binary counter can also be made to count backwards (or down) by taking the triggering leads from the reset sides of the flip-flops, instead of the set sides, as shown in Fig. 30. Table 6 shows the conditions on the output of a backward counter for each of eight pulses received on the input. The table shows that the outputs change state according to the pattern used when counting backwards (or down) in the binary number system.

Before the first input pulse arrives (Fig. 30 and Table 6) a clearing signal is applied to the clear input to reset all the flip-flops and produce an output reading of 000. The first input pulse sets FF1, causing FF1$\bar{Q}$ to go from logic 1 to logic 0. This 1 to 0 transition is applied to the clock input of FF2, which sets. This 1 to 0 transition from FF2$\bar{Q}$ sets FF3. Therefore, the first input pulse sets the three flip-flops and produces an output of 111 (the binary number for 7). The second input pulse resets FF1, however, the transition from 0-1 on FF1$\bar{Q}$ has no effect on FF2. Therefore, after the second pulse has been received the conditions of the flip-flops are; FF1 – reset, FF2 – set, FF3 – set, which produces an output of 110 (the binary number for 6). The third input pulse sets FF1 again to produce a 1 to 0 transition at FF3$\bar{Q}$, which resets FF2. The 0-1 transition on FF2$\bar{Q}$ has no effect on FF3, so the condition of the flip-flops are, FF1 – set, FF2 – reset, FF3 – set. This produces an output of 101, which is the binary number for 5. Subsequent input pulses cause the counter to produce outputs as shown in Table 6.

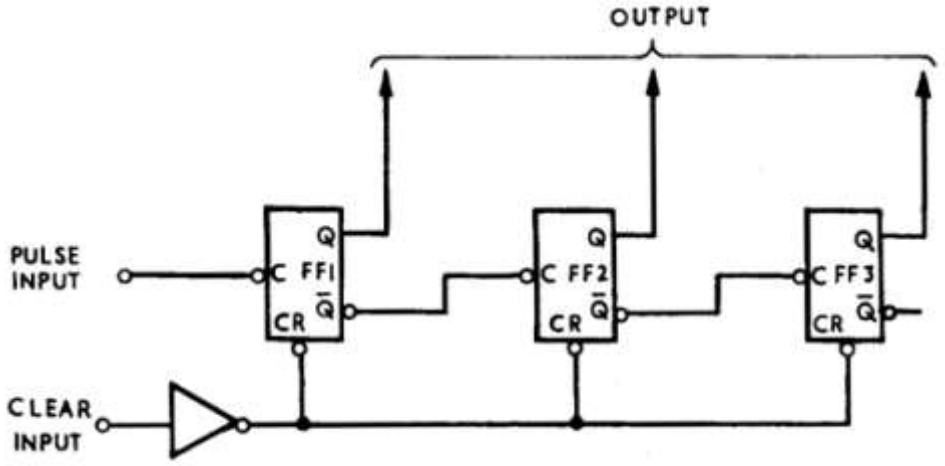



FIG. 30. BACKWARD COUNTER.

| | Output Conditions | | | Binary Number | Decimal Equivalent |
|---|---|---|---|---|---|
| | FF3 Q | FF2 Q | FF1 Q | (Counting Backwards) | |
| Before 1st pulse. | 0 | 0 | 0 | 000 | 0 |
| After 1st pulse | 1 | 1 | 1 | 111 | 7 |
| After 2nd pulse | 1 | 1 | 0 | 110 | 6 |
| After 3rd pulse | 1 | 0 | 1 | 101 | 5 |
| After 4th pulse | 1 | 0 | 0 | 100 | 4 |
| After 5th pulse | 0 | 1 | 1 | 011 | 3 |
| After 6th pulse | 0 | 1 | 0 | 010 | 2 |
| After 7th pulse | 10 | 0 | 1 | 001 | 1 |
| After 8th pulse | 0 | 0 | 0 | 000 | 0 |

TABLE 6. TRUTH TABLE FOR BACKWARD COUNTER.

**6.5** A timing diagram showing the input pulses and the conditions they produce on the $\bar{Q}$ leads of a backward counter, are shown in Fig. 31a. The arrows indicate the logic 1 to 0 edges which trigger the following stages. The conditions on the set (Q) outputs of each stage (which constitute the output of the counter) are shown in Fig. 31(b). Note that the conditions on the Q leads are the complements of the conditions on the $\bar{Q}$ leads.

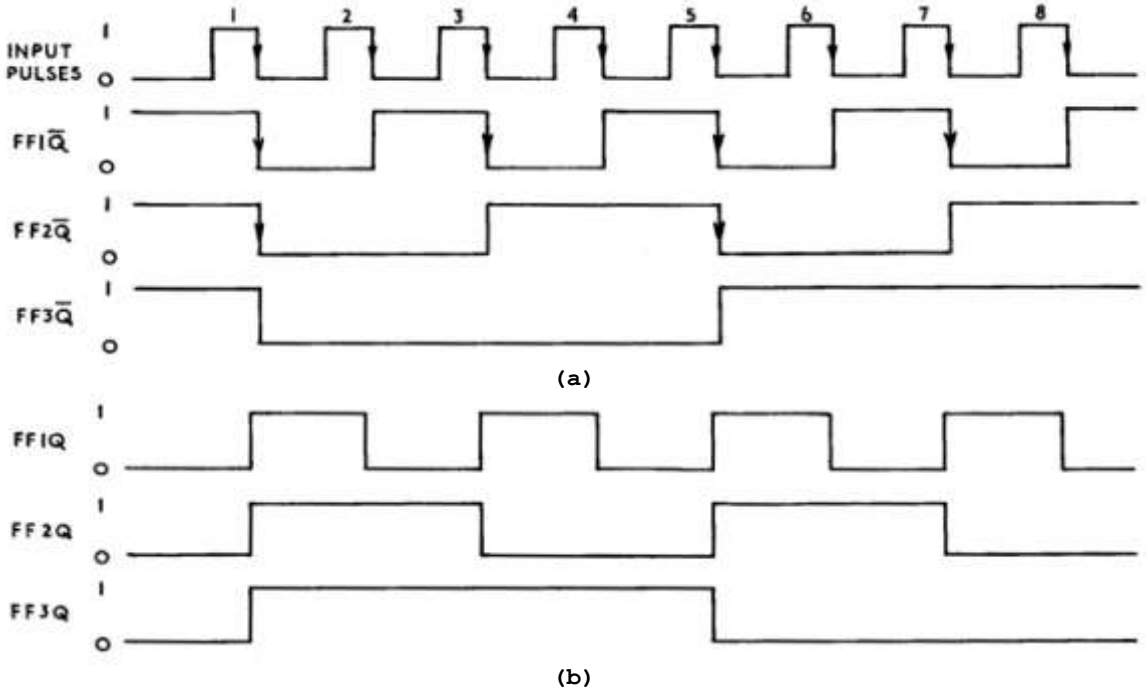



FIG. 31. TIMING DIAGRAM FOR BACKWARD COUNTER.

**6.6** REVERSIBLE COUNTERS. When the triggering leads and the output leads are taken from the same side of the flip-flops the count goes forward, and when they are taken from opposite sides of the flip-flop the count goes backwards. The counter in Fig. 32 can count either forward or backwards by changing the side of the flip-flops which provide the triggering edges to the following stages.

A logic 1 on the count forward input removes the inhibit from the top AND gates so that the triggering edges are extended from the same side of the flip-flops as the output, and the count goes forward.

A logic 1 on the count backward input removes the inhibit from the bottom AND gates so that the triggering edges are extended from the opposite side of the flip-flop to the output, and the count goes backward.

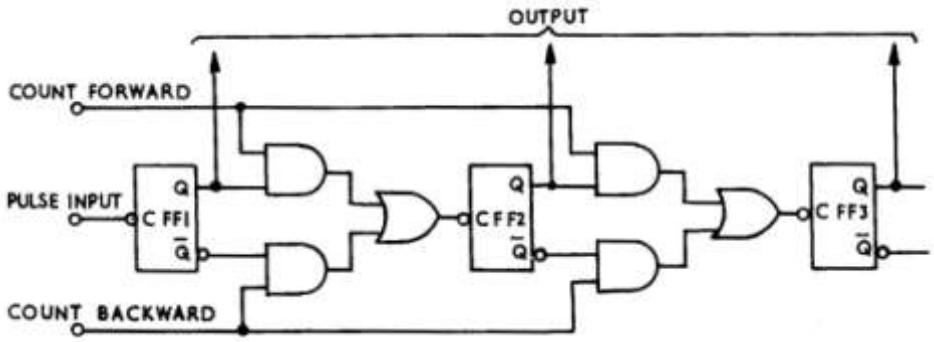



FIG. 32. REVERSIBLE COUNTER.

**6.7** A number of different variations of binary counters exist. For example, all the counters considered so far have used flip-flops which are triggered on the trailing edge. However, leading edge triggered flip-flops are used in some binary counters. If the counter in Fig. 28 employed leading edge triggered flip-flops it would become a backward counter. Similarly, the counter in Fig. 30 would become a forward counter if it used leading edge triggered flip-flops. Also some counters have output leads taken away from both sides of the flip-flops. The classification of these counters depends on the side of the flip-flop which is extended to trigger the following stages, and the type of flip-flops used.

**6.8 NUMBER OF OUTPUT COMBINATIONS FROM BINARY COUNTERS.** The number of different output combinations possible from a binary counter doubles each time a stage is added. The counter in Fig. 28 has three stages and Table 5 shows that it has eight different output combinations. When another stage is added, all those eight combinations are possible with the fourth stage in the set condition and another eight combinations are possible with the fourth stage in the reset condition, making a total of sixteen combinations. The number of output combinations possible from a binary counter is calculated by using the formula -

$$N = 2^{\eta} \qquad \textit{where} \quad N = \textit{Number of Possible output combinations}$$

$$\eta = \textit{number of stages}$$

**6.9 FEEDBACK COUNTERS.** The counters considered so far have a number of output combinations equal to $2^{\eta}$. For example, a four stage counter has 16 ($2^4$) output combinations, a five stage counter has 32 ($2^5$) combinations, and so on. However, these counters may be modified by feedback gating to provide less than the maximum possible output combinations. For example, the four stage counter in Fig. 33 is modified by feedback gating to provide 10 output combinations. Table 7 is a truth table showing the output combinations obtained from this counter.

The counter counts normally, as shown in Table 7, until FF4 sets at the end of the tenth pulse. Since FF2 and FF3 are now both set, the two inputs to the AND gate are both at logic 1 for the first time. The logic 1 output from the AND gate sets FF5, causing FF5$\overline{Q}$ to go to logic 0 and reset all flip-flops via the clear (CR) inputs. Resetting the counter flip-flops returns the output to 0000. The next input pulse resets FF5 and the counter starts counting normally again.

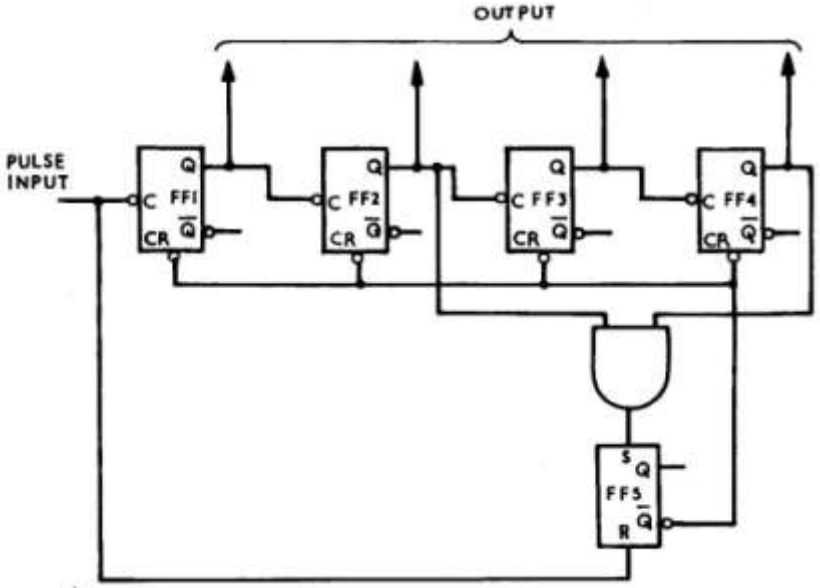



FIG. 33. BINARY COUNTER WITH FEEDBACK GATING.

| | Output Conditions | | | |
|---|---|---|---|---|
| | FF4 Q | FF3 Q | FF2 Q | FF1 Q |
| Before 1st pulse. | 0 | 0 | 0 | 0 |
| After 1st pulse | 0 | 0 | 0 | 1 |
| After 2nd pulse | 0 | 0 | 1 | 0 |
| After 3rd pulse | 0 | 0 | 1 | 1 |
| After 4th pulse | 0 | 0 | 0 | 0 |
| After 5th pulse | 0 | 1 | 0 | 1 |
| After 6th pulse | 0 | 1 | 1 | 0 |
| After 7th pulse | 0 | 1 | 1 | 1 |
| After 8th pulse | 1 | 1 | 0 | 0 |
| After 9th pulse | 1 | 0 | 0 | 1 |
| After 10th pulse | 0 | 0 | 0 | 0 |

TABLE 7. TRUTH TABLE FOR COUNTER IN FIG. 33.

Fig. 34 is a timing diagram showing the waveshape for the counter in Fig. 33. Note how all flip-flops are reset after the 10th pulse and counting starts in the normal manner from one again.

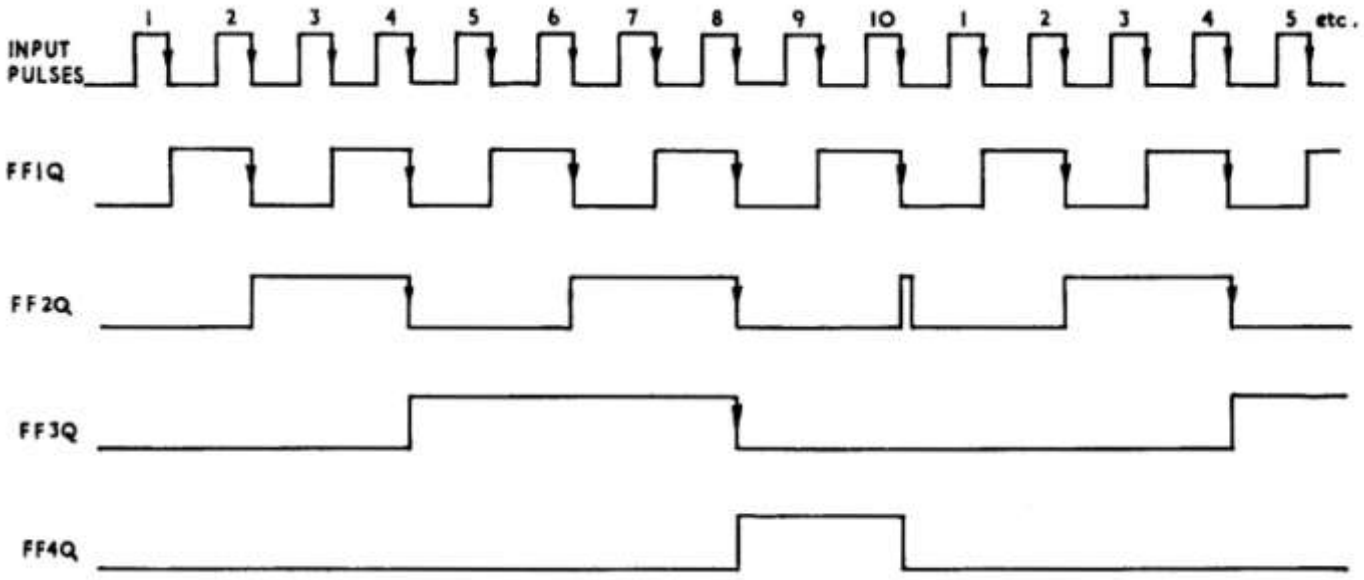



FIG. 34. TIMING DIAGRAM FOR FIG. 33.

**6.10** DECADE COUNTERS. Counters can be made to count to any desired number by selecting the inputs to the feedback gate so that the counter is reset when the desired count is reached. Counters which are designed to count to ten (as in Fig. 33) are called "decade" or "count of ten" counters and are used extensively in logic equipment.

**6.11** DECODING COUNTER OUTPUTS. A binary counter produces a binary number pattern at its output. This type of output is suitable for some applications, but in many cases the output is required in some other form. The process of changing a code into a readily recognisable form, such as decimal, is generally referred to as decoding and the device used is usually called a decoder. Fig. 35 is a block diagram showing a common type of decoder used in the output of a binary counter to change the binary number pattern into a decimal number pattern. This decoder has ten outputs, each one corresponding to a decimal number from 0 to 9. Each output of the decoder becomes activated in turn as successive pulses are fed into the counter. For example, after six pulses have been fed into the counter the output representing the decimal number six is activated.

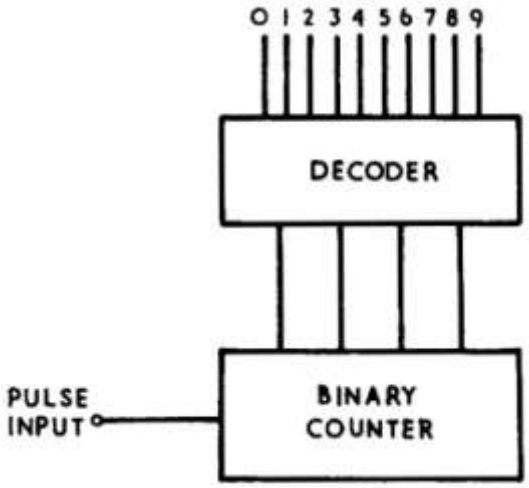



FIG. 35. CONNECTION OF DECODER TO COUNTER OUTPUT.

**6.12** DECODING WITH AND GATES. Fig. 36 shows a binary to decimal decoder connected in the output of a four stage feedback counter designed to count up to ten. Table 7 shows the manner in which the set outputs of the feedback counter flip-flops change, and Table 8 shows the manner in which the decoder outputs change.

The internal circuitry of the decoder (between the gate inputs and the counter outputs) is not shown in Fig. 36. Instead, the designations FF1Q, FF1$\overline{Q}$, FF2Q, FF2$\overline{Q}$, etc, on the gate inputs show how the decoder is wired internally. The inputs to the gates are selected so that each gate provides a logic 1 output only when a particular count condition exists, and at no other time. For example, the output of the AND gate designated 0 is at logic 1 when all counter flip-flops are reset, that is, when no pulses have been counted. After the first input pulse has been received, FF1 sets, and the logic 0 condition on FF1$\overline{Q}$ inhibits the gate designated 0, causing it to produce a logic 0 on its output. However, the gate designated 1 produces a logic 1 on its output because all of its inputs are at logic 1. In this manner, each successive pulse into the counter closes one gate and allows the next to open. Therefore, each decoder output represents the decimal equivalent of one of the binary counter states.

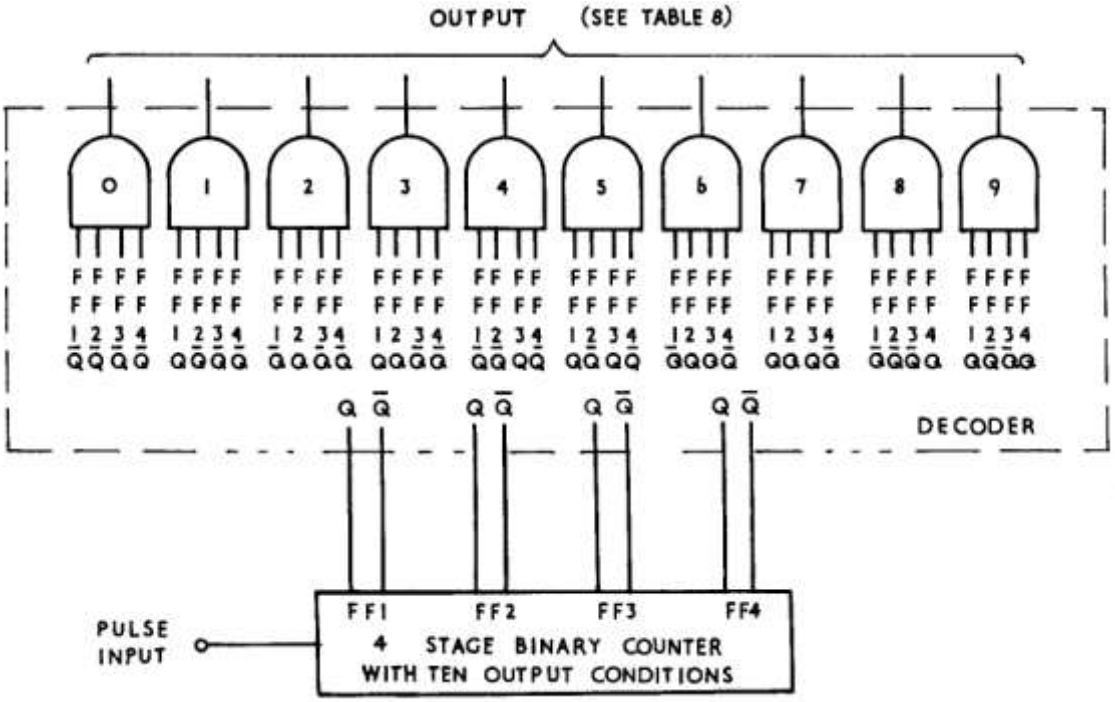



FIG. 36. DECODER USING AND GATES.

| | OUTPUT OF GATES | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Before 1st pulse. | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| After 1st pulse | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| After 2nd pulse | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| After 3rd pulse | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| After 4th pulse | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| After 5th pulse | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| After 6th pulse | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| After 7th pulse | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| After 8th pulse | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| After 9th pulse | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

TABLE 8. TRUTH TABLE FOR DECODER.

**6.13** DECODING MATRIX. Another method of decoding the output of a binary counter is to use a diode matrix, as shown in Fig. 37. This matrix is designed to operate in a logic circuit using a positive voltage (+V) to represent logic 1 and 0 volts to represent logic 0. Both the Q and $\overline{Q}$ outputs of the binary counter flip-flips are connected to the decoder matrix, so that at any one time, half the inputs to the decoder are at logic 1 and the other half are at logic 0.

The outputs from the binary counter are connected through the decoder diodes in such a manner that the positive voltage (logic 1) from behind the resistors can only be extended to one decoder output at a time.

Diodes which have a logic 0 connected to them from the counter outputs will conduct, and diodes which have a logic 1 (+V) connected to them from the counter outputs will be non-conducting. A conducting diode extends the 0V (logic 0) from the counter output to the decoder output and shunts the +V (from behind the resistors) from the decoder output. All decoder outputs, except the one which is required to be at logic 1 for the particular count, have at least one conducting diode connected to them. The decoder output which is to be at logic 1 has all non-conducting diodes connected to it. For example, assume that the binary counter (Fig. 37) has counted three pulses; as a result, FF1 and FF2 are set and FF3 is reset. The darkened diodes have logic 1 (+V) connected to them from the counter outputs and are non-conducting; all others are conducting. Note that there is only one decoder output on which all the diodes are non-conducting and that is the output representing the decimal number 3. The heavy black line in Fig. 37 shows how the +V (logic 1) from behind the resistors is extended to this output. All other decoder outputs are at logic 0 because they have at least one conducting diode connected to them to shunt the +V from the output.

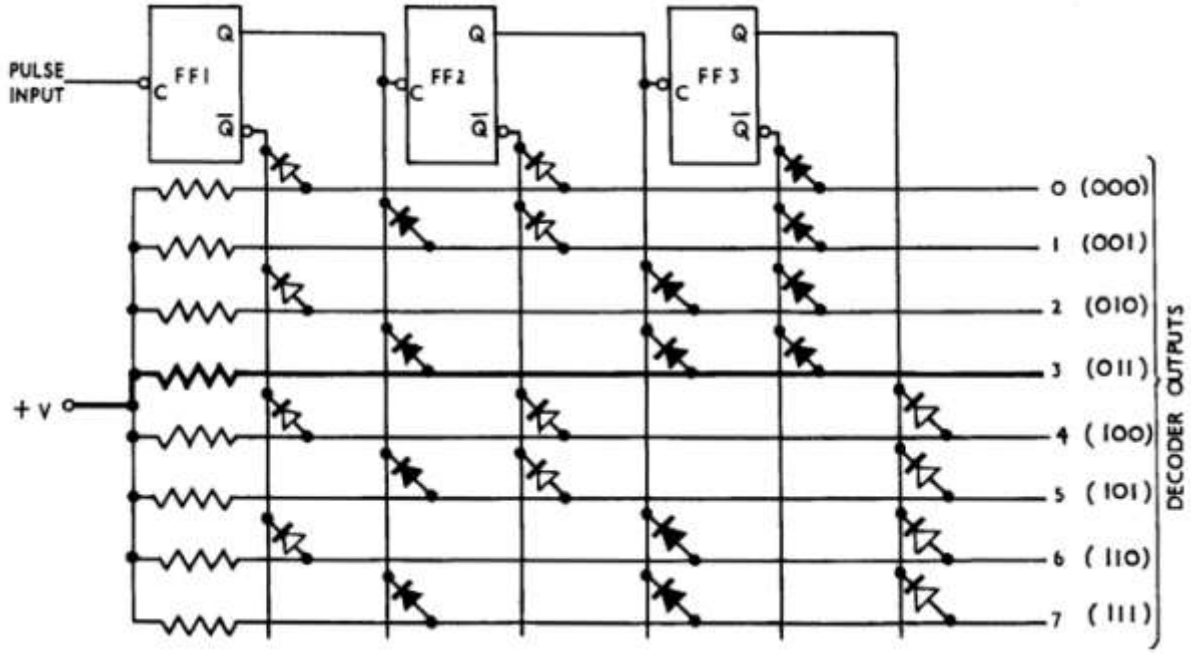



FIG. 37. DECODING MATRIX.

**6.14** NUMERIC INDICATING TUBES. The output leads from decoders are sometimes connected to circuits which enable the count conditions to be read out in a visual decimal form. Gas filled numeric tubes, similar to that shown in Fig. 38, are one type of read out device used for this purpose. These tubes have individual cathodes shaped in the form of numbers, and when the correct voltage condition is applied to a selected cathode, the gas around it ionises causing a glow. The tube shown in Fig. 38 has the cathode corresponding to the decimal figure eight activated.


(a) Construction. (b) Symbol.

FIG. 38. NUMERIC INDICATOR TUBE.

**6.15** **TIME RELATED PULSES.** Gate can be used on the output of a binary counter to produce a train of time related pulses which are used to control an electrical or mechanical sequence. Fig. 39 shows a four stage binary counter with output gating arranged to produce time related pulses of predetermined lengths.

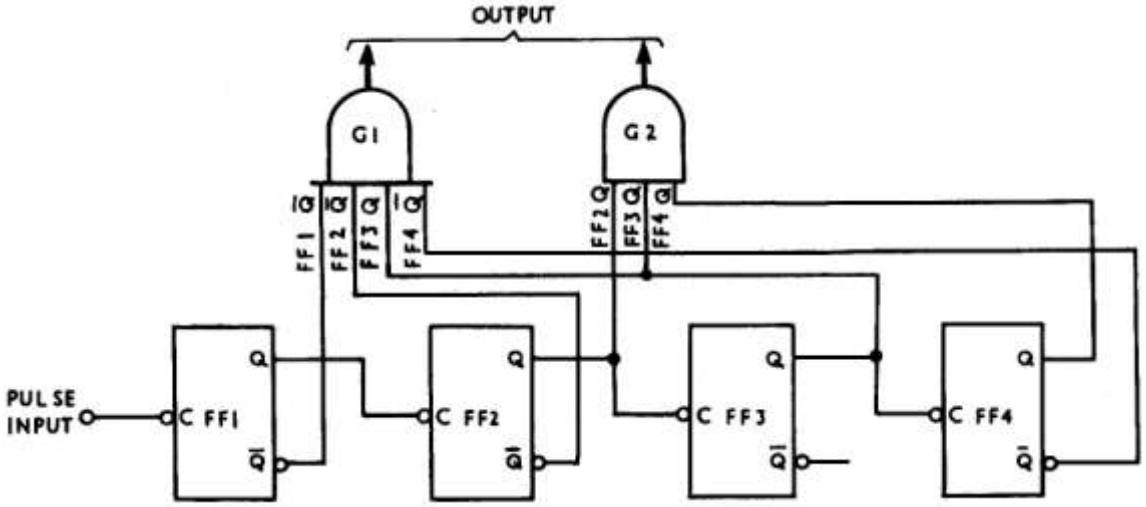



FIG. 39. TIME RELATED PULSES FROM A BINARY COUNTER.

Fig. 40 is a timing diagram showing the operation of the circuit in Fig. 39. It shows that the output of gate G1 goes to logic 1 at the end of the fourth input pulse when FF1$\bar{Q}$, FF2$\bar{Q}$, FF3$\bar{Q}$ and FF4$\bar{Q}$ are all at logic 1, and goes back to logic 0 at the end of the fifth input pulse when FF1Q goes to logic 0. At the end of the fourteenth input pulse FF2Q, FF3Q and FF4Q are all at logic 1 together for the first time and the output of G2 goes to logic 1. Since FF2, FF3 and FF4 stay set for the duration of two input pulses, the length of the pulse from G2 is equivalent to the length of two input pulses. The length and frequency of the input pulses are selected to give the required output pulse lengths and time relationship. Also, the gating may be rearranged to provide other output combinations.

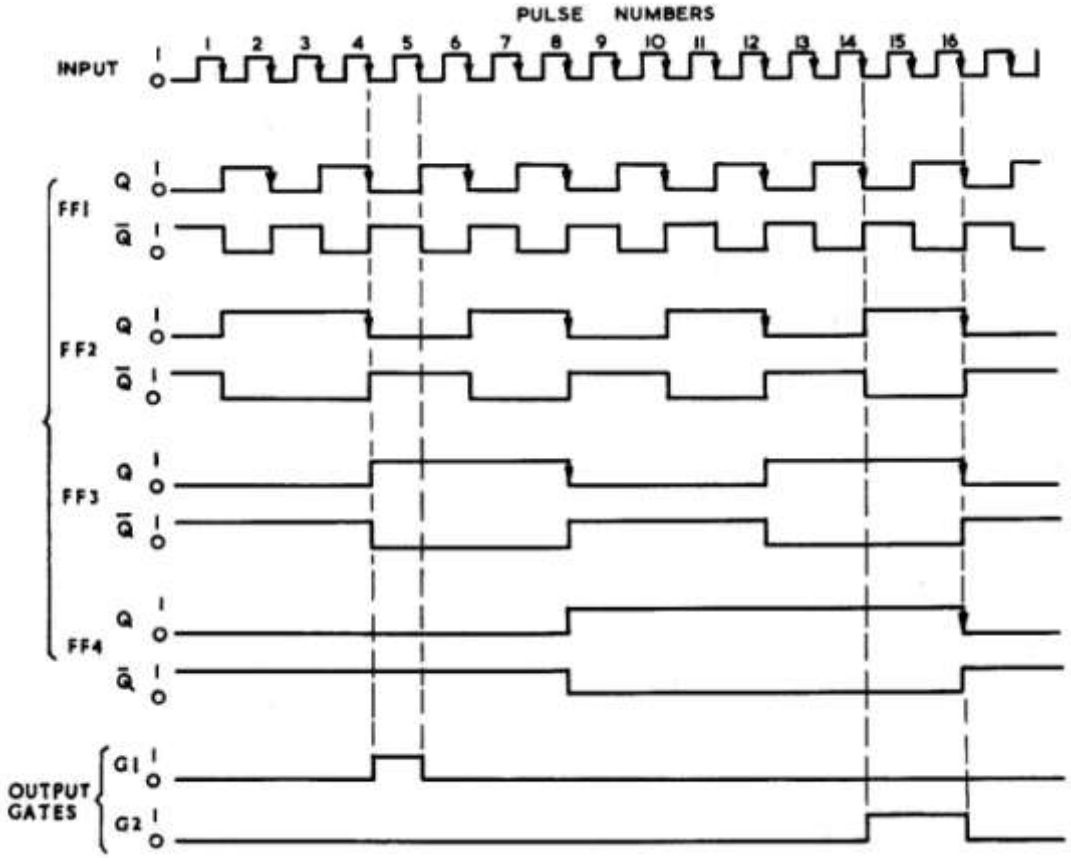



FIG. 40. TIMING DIAGRAM FOR FIG. 39.

**6.16** SYNCHRONOUS AND ASYNCHRONOUS COUNTERS. Counters are classified as synchronous or asynchronous depending on whether the stages are triggered at the same instant (synchronous) or at different instants (asynchronous).

**6.17** ASYNCHRONOUS COUNTERS. In any type of flip-flop there is a small delay between the application of the triggering edge of a pulse and the instant when the stage settles into its new state. In asynchronous or ripple-through counters (typically described in paras. 6.1 to 6.9) this factor results in progressive and cumulative delays in changes-of-state throughout the counter stages, and this is a disadvantage in some applications.

For example, the circuit in Fig. 39 produces a false pulse on the G1 output due to the asynchronous operation of the counter. Fig. 41 is part of the timing diagram for Fig. 39, expanded to show the switching delays, and their effect at the end of the eighth input pulse.

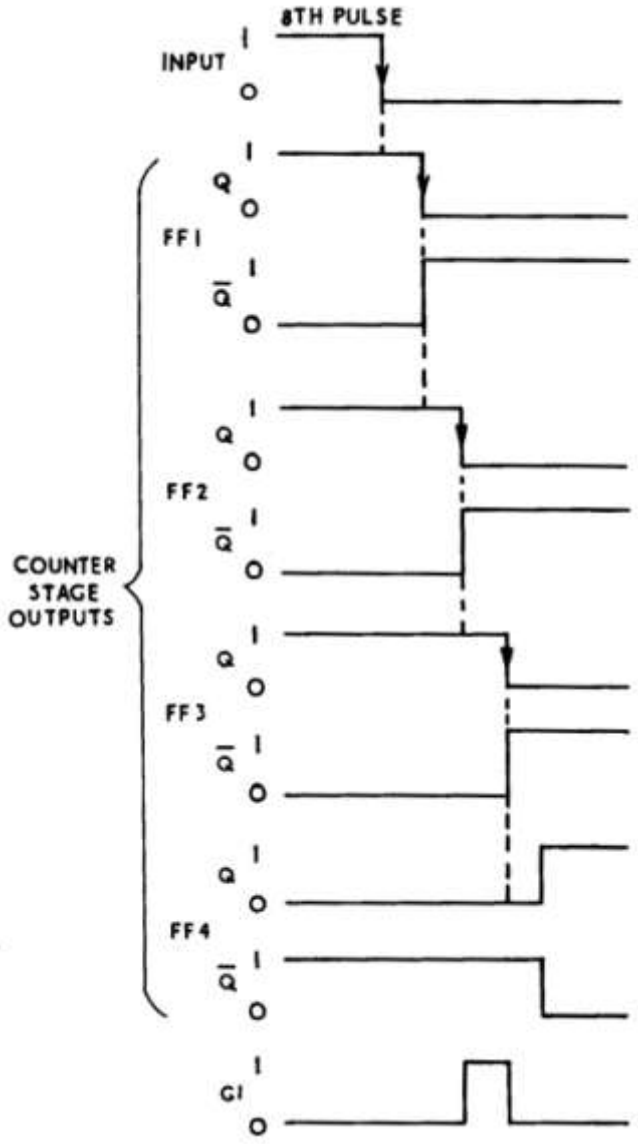



FIG. 41. EFFECT OF SWITCHING DELAYS.

Instead of the counter outputs instantly changing from the condition where FF1Q, FF2Q, FF3Q and FF4$\overline{Q}$ are at logic 1 (Count 7) to the condition where FF1$\overline{Q}$, FF2$\overline{Q}$, FF3$\overline{Q}$, FF4Q are at logic 1 (Count 8), the stages in turn produce a number of intermediate output conditions between the trailing edge of the clock pulse and the time when all stages have settled into their new state. During one of these intermediate stages FF1$\overline{Q}$, FF2$\overline{Q}$, FF3Q, and FF4$\overline{Q}$ are all at logic 1 together and gate G1 will open momentarily to generate a false output pulse.

In some cases, the circuit connected to the output is slow to operate and this pulse is of no consequence, but it could falsely activate a high speed electronic output circuit.

**6.18** Various methods can be used to overcome false signals (false counts) in asynchronous counters. One method is to inhibit the output gates until the counter has had time to settle into its new state. Fig. 42 shows how the inhibiting input is connected to the output gates. This input is maintained at logic 0 until all relative inputs to the gates are at logic 1. The delay period, which is arranged from circuitry external to the counter, exceeds the ripple through time of the counts by a suitable safety margin, before the inhibit is replaced by logic 1 to activate the output gate.

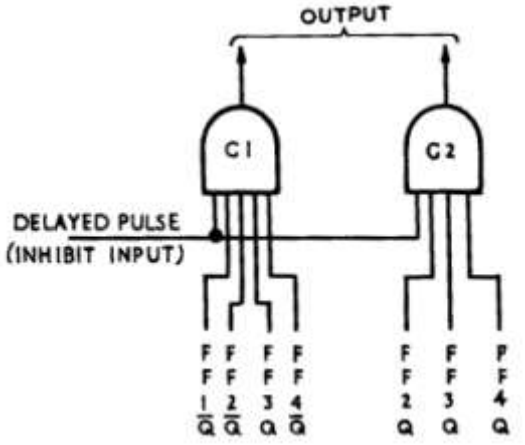



FIG. 42. DELAYED PULSE TO OUTPUT GATES.

**6.19** SYNCHRONOUS COUNTERS. One method of providing the synchronous operation of a binary counter is to gate the input pulses into each stage as shown in Fig. 43. Fig. 44 is a timing diagram showing the operation of this type of counter. During the second input pulse the output of G1 is at logic 1 because both the input pulse and FF1Q are at logic 1. At the end of the second input pulse the output of G1 goes to logic 0 to provide a logic 1 to 0 transition on the input of FF2, which changes state. Notice, that although a delay exists between the end of the input pulse and FF1Q going to logic 0, the output of the AND gate goes to logic 0 immediately the input pulse ends. Therefore, the delay in FF1 has no effect and the end of the input pulse provides the logic 1 to 0 transition to the next stage.

During the fourth input pulse both FF1Q and FF2Q are at logic 1, therefore, the two inputs of Q1 are at logic 1 and the three inputs of G2 are at logic 1. At the end of the fourth pulse the outputs of both gates go from logic 1 to logic 0 at the same instant, causing FF2 and FF3 to change state.

The output of gate G1 goes to logic 1 during every second input pulse and the output of G2 goes to logic 1 during every fourth input pulse. In this way a triggering transition is extended to FF2 at the end of every second input pulse and to FF3 at the end of every fourth input pulse. Since the triggering transitions are timed by the end of the input pulses, any delay in the resetting of the flip-flops is of no consequence.

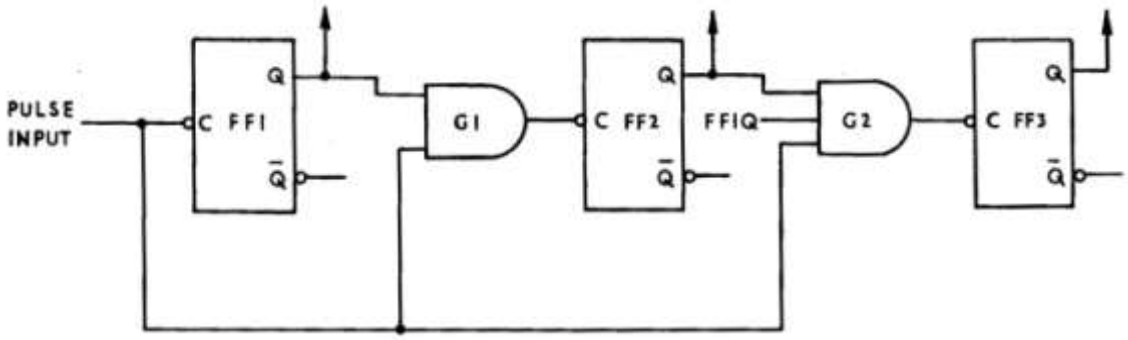



FIG. 43. THREE STAGE SYNCHRONOUS BINARY COUNTER.

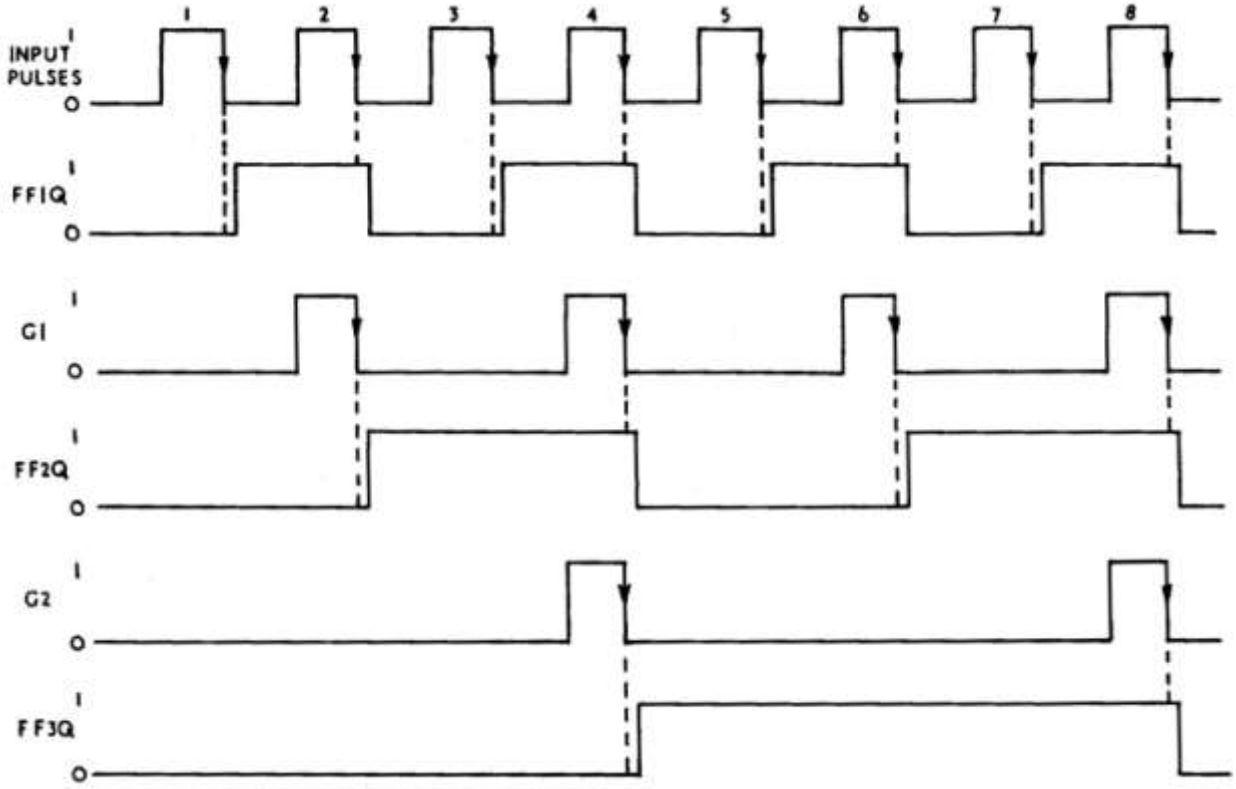



FIG. 44. TIMING DIAGRAM FOR FIG. 43.

## 7. DECADE COUNTERS IN CASCADE.

**7.1** GENERAL. There are many examples in electronic logic equipment where it is necessary to count a large number of pulses and produce the result in a decimal read-out, for example, as in digital measuring instruments. In some cases this type of equipment is required to count many millions of pulses. The type of binary counter described in paragraph 6.2 would be impractical for this purpose because of the extremely complex decoding arrangements needed to convert the binary output to a decimal number. However, decade counters connected in cascade provide a counter with an output that can be decoded in a relatively simple manner.

**7.2** DECADE COUNTERS IN CASCADE. Fig. 45 shows four decade counters connected in cascade, that is, they are connected in such a way that each operates the next one in turn. The first decade counter is used to count units, the second to count tens, the third to count hundreds and the fourth to count thousands. Each decade counter counts to ten and produces a binary output, as described in para. 6.9. However, when a decade counter resets at the count of ten, it sends a triggering pulse to the input of the next decade counter.

After every ten input pulses the units counter resets and sends a triggering pulse to the tens counter. The tens counter reaches the count of ten on every hundred input pulses. It then resets and sends a triggering pulse to the hundreds counter which reaches a count of ten after every thousand input pulses. At every thousand input pulses the hundreds counter resets and sends a triggering pulse to the thousands counter, and so on. For every thousand pulses counted, the units counter counts to ten and resets 100 times, the tens counter counts to ten and resets 10 times and the hundreds counter counts to ten and resets once. If, for example, 1274 pulses were counted, the outputs of the cascade counters would be:

Thousands counter .... 0001 (Binary number for 1)

Hundreds counter .... 0010 (Binary number for 2)

Tens counter .... 0111 (Binary number for 7)

Units counter .... 0100 (Binary number for 4)

The counter shown in Fig. 45 is capable of counting up to 10,000 pulses. The capacity of this counter can be increased by connecting additional decade counting stages in cascade. Each additional decade counting stage increases the counting capacity ten times.

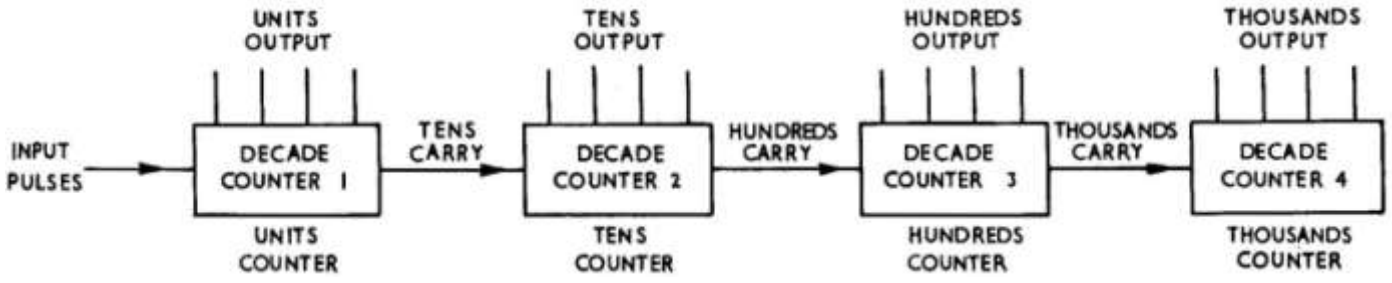



FIG. 45. DECADE COUNTERS IN CASCADE.

**7.3** BINARY CODED DECIMAL (B.C.D.). The output of the cascaded decade counters shown in Fig. 45 is expressed in Binary Coded Decimal (B.C.D.) form. In the B.C.D. system each digit of a decimal number is represented by its equivalent binary number. For example, the decimal number 1824 is represented in B.C.D. by 0001 1000 0100 as follows:

```
0001  1000  0010  0100
  ↑     ↑     ↑     ↑
  1     8     2     4
```

Note that in B.C.D. a four bit binary counter is used to represent each digit in the decimal number. This is different to expressing a decimal number in pure binary form. For example, 1824 is represented by 11100100000 in the binary number system and as 0001 1000 0010 0100 in the B.C.D. system. Some examples of decimal numbers expressed in B.C.D. are as follows:

| Decimal Number | Binary Coded Decimal |
|---|---|
| 21 | 0010 0001 |
| 35 | 0011 0101 |
| 264 | 0010 0110 0100 |
| 2781 | 0010 0111 1000 0001 |

**7.4** DECODING CASCADED DECADE COUNTERS. Since the output of cascaded decade counters is in B.C.D. form, each counter is decoded in the normal manner, as shown in Fig. 46. The decoders produce the decimal equivalent of the binary output of each counter. Some type of numeric indicating device is connected to the output of the decoder in order to display the count in decimal form.

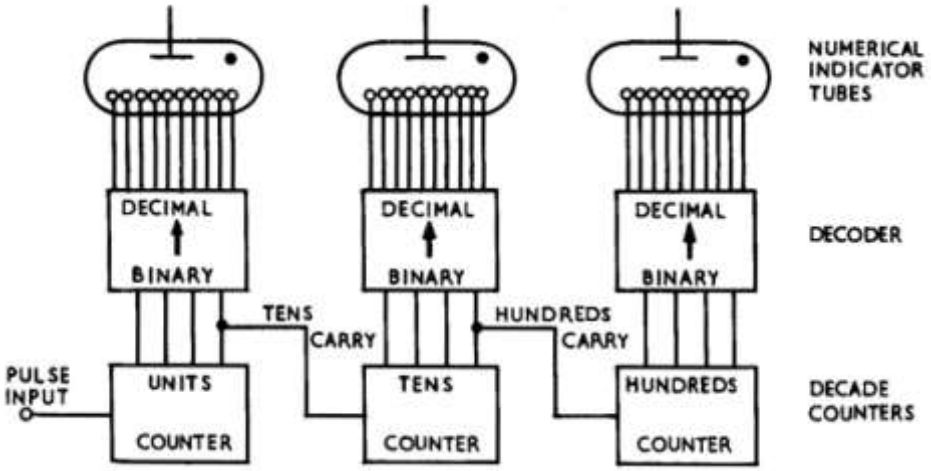



FIG. 46. DECODING CASCADED DECADE COUNTERS.

As an example of decoding cascaded decade counters, assume that 2916 pulses have been counted. The outputs of the counters and their associated decoders are as follows:

| Thousands Counter Output is | Hundreds Counter Output is | Tens Counter Output is | Units Counter Output is |
|---|---|---|---|
| 0010 | 1001 | 0001 | 0110 |
| ↓ | ↓ | ↓ | ↓ |
| Decoded to | Decoded to | Decoded to | Decoded to |
| 2 | 9 | 1 | 6 |

**7.5** B.C.D. COUNTERS. Counters which have four flip-flop stages and produce a four bit binary output are often called B.C.D. counters. The B.C.D. counters described so far in this paper have an output which follows the binary number system and are called "8421 B.C.D." counters (or sometimes "8421 decade" counters). They are given this name to signify that the four binary bits have a value (or weight) equal to the decimal digits 8, 4, 2, and 1, respectively. The decimal equivalent to the binary number on the output is obtained by adding the weighting numbers corresponding to any output leads at logic 1. For example, assume that the output of an 8421 B.C.D. counter is 0101. Since the weighting of the first output is 1 and the third output is 4, the decimal equivalent of the binary output is 0+4+0+1 = 9.

Some types of B.C.D. counters produce output patterns which do not conform to the binary number system. For example, the 2421 B.C.D. counter has its four outputs weighted as 2, 4, 2 and 1, respectively. In this counter the decimal equivalent of 1011 on the output is 2+0+2+1 = 5. Other examples of B.C.D. counter output weighting are, 1242, 1245 and 7421. The reasons for this type of weighting and the circuit arrangements used to achieve them are not covered in this paper.

## 8. RING COUNTERS.

**8.1** GENERAL. Ring counters are synchronous counters which use clocked flip-flops connected in a continuous ring. Although they require more stages than binary counters to provide the same number of output conditions, they are preferred in some applications because of the simplicity of their output circuits.

**8.2** RING COUNTER. In ring counters only one stage is set at any time and all other stages are reset. Each successive input pulse moves the set condition to the next stage. In this manner, the set condition moves through the counter, one step at a time, under the control of the input pulses. Fig. 47 shows a three stage ring counter using type D flip-flops, which set if the D input is logic 1 and reset if the D input is at logic 0. A timing diagram of this counter is shown in Fig. 48.

Before counting commences a pulse of logic 1 is applied to the clear input to set FF1 and reset FF2 and FF3. The logic 1 on FF1Q indicates that zero pulses have been counted (See Table 9). Under this condition FF1Q extends logic 1 to the D input of FF2, FF2Q extends logic 0 to the D input of FF3, and FF3Q extends logic 0 to the D input of FF1. The input pulses are applied simultaneously to all stages. The first input pulse causes FF1 to reset and FF2 to set; FF3 remains reset. A logic 1 now exists on FF2Q to indicate that one input pulse has been received. Now logic 1 is applied to the D input of FF3 and logic 0 to the D inputs of FF1 and FF2. The second input pulse sets FF3 and resets FF2; FF1 remains reset. Similarly, the third input pulse sets FF1, (because a logic 1 is extended from FF3Q to the D input of FF1) and resets FF3.

Table 9 shows that after three pulses the counter returns to its initial condition. Therefore, the number of different output conditions from a ring counter is equal to the number of stages used. Decade ring counters have 10 stages and are sometimes called "one out of 10 counters". Although ring counters use more stages than other counters, they do provide logic 1 on a different lead for each count condition without using decoding gates.

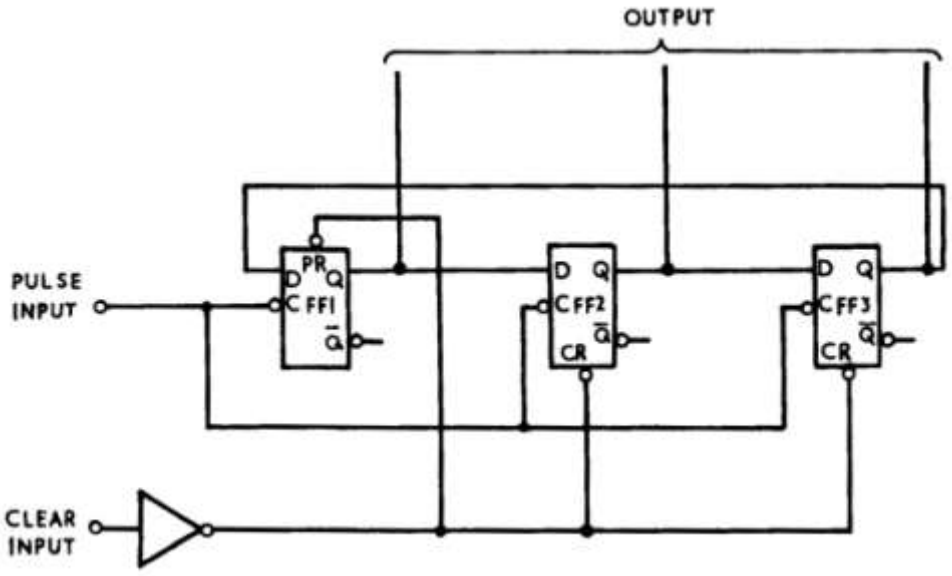



FIG. 47. RING COUNTER.

| | Output Conditions | | |
|---|---|---|---|
| | FF1Q | FF2Q | FF3Q |
| Before 1st pulse. | 1 | 0 | 0 |
| After 1st pulse | 0 | 1 | 0 |
| After 2nd pulse | 0 | 0 | 1 |
| After 3rd pulse | 1 | 0 | 0 |

TABLE 9. TRUTH TABLE FOR RING COUNTER.

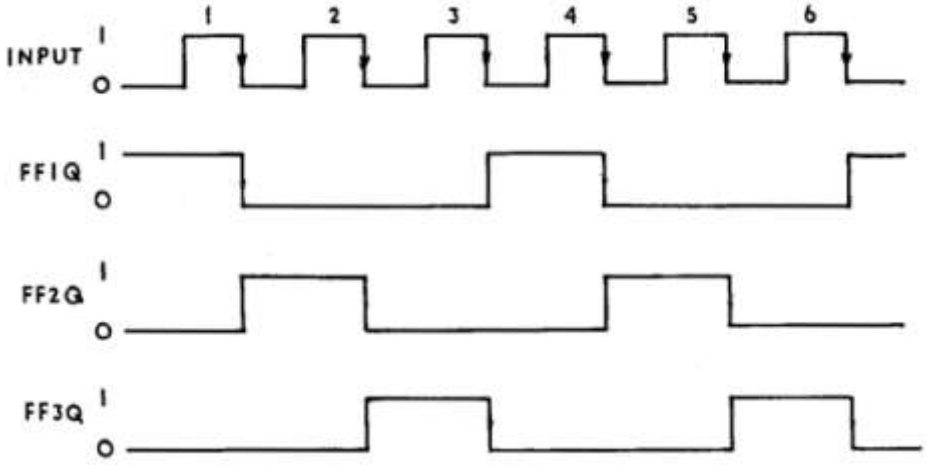



FIG. 48. TIMING DIAGRAM FOR FIG. 47.

8.3 Clocked SR and JK flip-flops are also used as ring counter stages. Fig. 49 shows a three stage ring counter with clocked SR flip-flop stages.

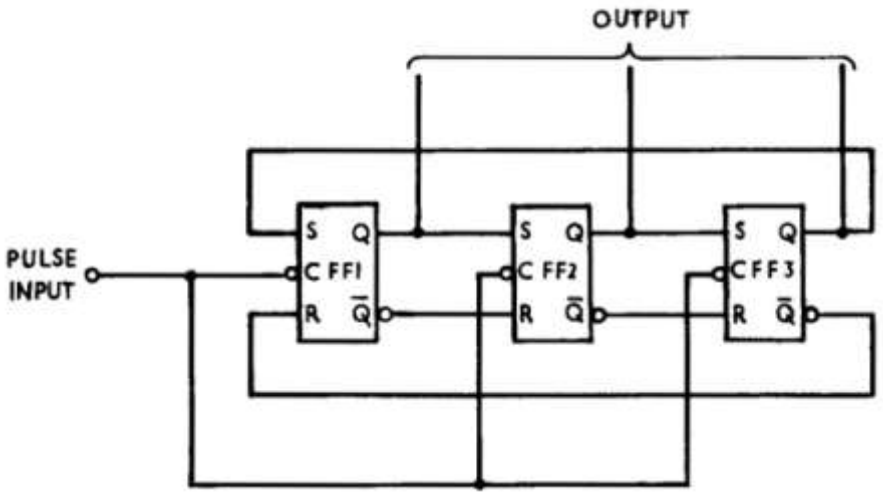



FIG. 49. RING COUNTER USING SR FLIP-FLOPS.

8.4 Ring counters using relays (cyclic call distributors in crossbar exchanges), thyristors and unijunction transistors are also encountered. Also, cold cathode counting tubes, which combine the functions of a ring counter and decimal readout, are used in some test instruments.

8.5 TWISTED RING COUNTER. Twisted ring (or Johnson) counters have the advantage that they use fewer stages than ring counters to provide the same number of count conditions. However, they require simple output gating when the count conditions are required as logic 1 signals on individual leads. Fig. 50 shows a twisted ring counter using type D flip-flops, and Fig. 51 is the timing diagram for this counter. The coupling between the reset side of the last stage and the D input of the first stage distinguishes the twisted ring counter from the normal ring counter.

Before counting commences all stages are reset by a pulse of logic 1 on the clear input, and the initial condition FF1Q=0, FF2Q=0, and FF3Q=0, is recorded in Table 10. In this condition, logic 0's are extended from FF1Q and FF2Q to the D inputs of FF2 and FF3, respectively, and logic 1 is extended from FF3$\overline{Q}$ to the D input of FF1. On the first input pulse FF1 sets and FF2 and FF3 remain reset. This causes the D input to FF2 to go to logic 1; the D input to FF1 remains at logic 1 because FF3 is still reset, and the D input to FF3 remains at logic 0 because FF2 is still reset. The second input pulse sets FF2; FF1 remains set and FF3 remains reset. All flip-flops now have a logic 1 on their D inputs. The third input pulse sets FF3; FF1 and FF2 remain set. The setting of FF3 changes the D input of FF1 to logic 0, so that the fourth pulse resets FF1. This changes the D input to FF2 to logic 0 so that it resets on the fifth pulse. The resetting of FF2 prepares the input to FF3 so that it resets on the sixth pulse.

Table 10 shows that the input pulses set each stage in turn, and then reset each stage in turn, so that the number of output combinations is equal to twice the number of stages. For example, a twisted ring counter which counts to ten has five stages and is called a twisted ring decade counter.

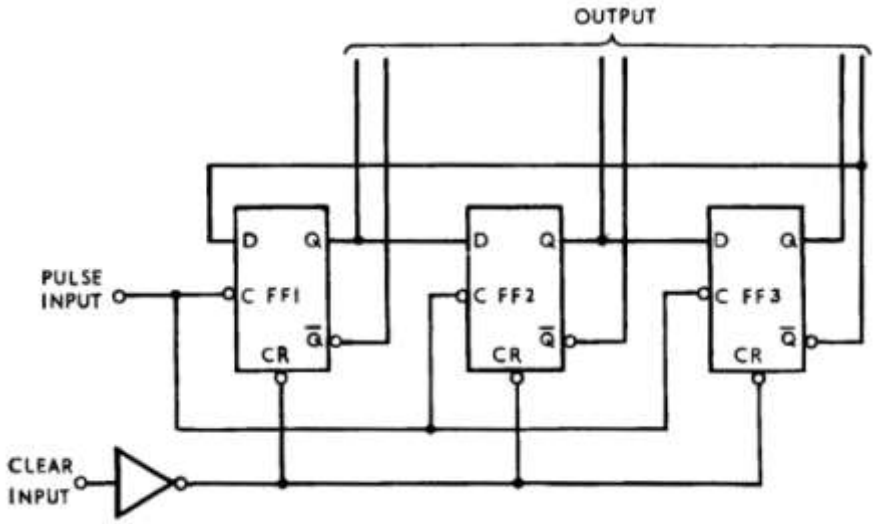



FIG. 50. TWISTED RING COUNTER.

| | Output Conditions | | |
|---|---|---|---|
| | FF1Q | FF2Q | FF3Q |
| Before 1st pulse. | 0 | 0 | 0 |
| After 1st pulse | 1 | 0 | 0 |
| After 2nd pulse | 1 | 1 | 0 |
| After 3rd pulse | 1 | 1 | 1 |
| After 4th pulse | 0 | 1 | 1 |
| After 5th pulse | 0 | 0 | 1 |
| After 6th pulse | 0 | 0 | 0 |

TABLE 10. TRUTH TABLE FOR TWISTED RING COUNTER.

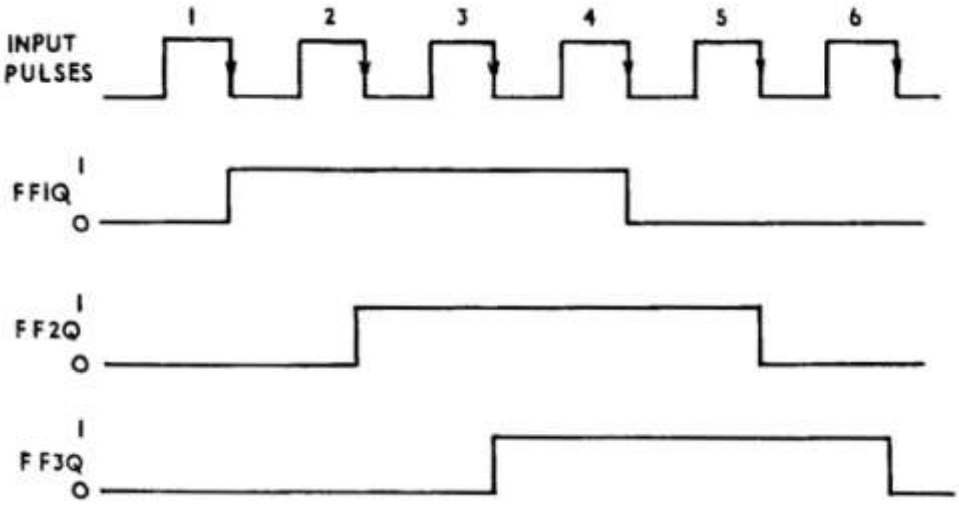



FIG. 51. TIMING DIAGRAM FOR TWISTED RING COUNTER.

**8.6** DECODING TWISTED RING COUNTER OUTPUTS. Fig. 52 shows the decoding gate arrangement required on the output of a twisted ring counter to provide logic 1 signals on individual leads. Two input AND gates are used for decoding in this type of counter.

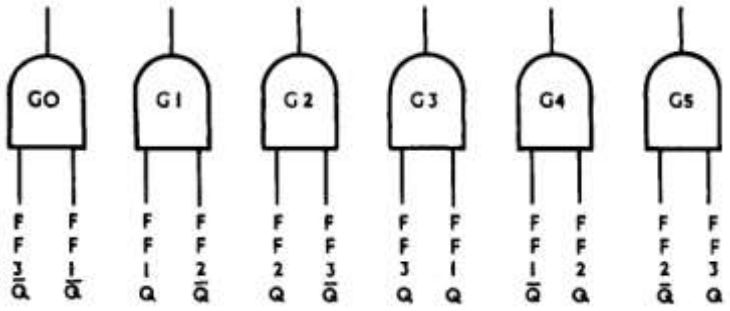



FIG. 52. DECODER FOR TWISTED RING COUNTER.

When the counter is at count zero, all flip-flops are rest, FF3$\bar{Q}$ and FF1$\bar{Q}$ are at logic 1, and the output of gate G0 is logic 1. All other gates are inhibited. The first input pulse sets FF1 and FF1$\bar{Q}$ becomes logic 0 to inhibit gate G0. Logic 1 on FF1Q and FF2$\bar{Q}$ activates gate G1 and its output becomes logic 1. With each successive input pulse applied to the counter, each gate is activated in turn as its preceding one is inhibited, thus providing the required decoded outputs from the gates.

**8.7** Clocked SR or JK flip-flops are also used in twisted ring counters. In these counters the set output of the last stage is connected to the R or K input of the first stage, and the reset output of the last stage is connected to the S or J input of the first stage, as shown in Fig. 53.

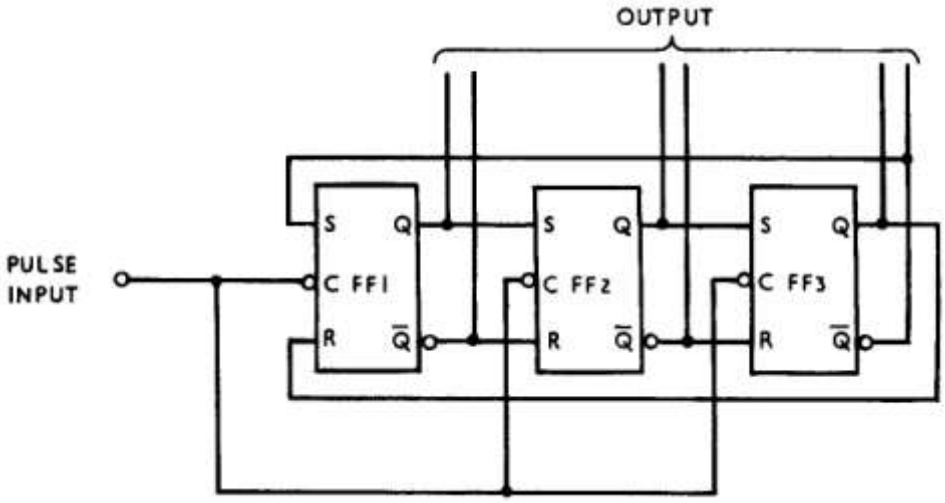



FIG. 53. TWISTED RING COUNTER USING SR FLIP-FLOPS.

**8.7** CASCADED TWISTED RING COUNTERS. Twisted ring decade counters may be connected in cascade in a similar manner to that shown in Fig. 45. When a decade has counted ten pulses on its input, a single carry pulse is applied to the following decade to step it on one count. In this manner cascaded twisted ring decade counters count in units, tens, hundreds, etc. Each decade or stage has five bistable elements, the ten outputs of which are gated in a similar manner to that shown in Fig. 52, to provide a decimal read-out of the number of pulses counted. The outputs of the gates may be connected to a numeric indicator tube.

## 9. REGISTERS.

**9.1** GENERAL. Registers are used in electronic logic equipment to store a small amount of information for a short time. They contain a number of flip-flop stages, or other bistable devices, and each stage stores one bit of information. The register as a whole is said to store one *word*. *Bit* and *word* are units used to indicate a quantity of information and are defined as follows:

- A BIT is the smallest unit of binary information and is represented by the binary digits 1 or 0. For example, the binary number 1011 contains four bits. The word BIT is an abbreviation for Binary Digit.

- A WORD is a number of bits representing some desired amount of information and is the normal unit in which information is stored and transferred through logic equipment. For example, 1001110110 is a word containing ten bits. The number of bits contained in a word depends on the system design.

Flip-flop registers are used as intermediate or buffer stores between sections of a logic system.

**9.2** PARALLEL AND SERIAL TRANSMISSION OF INFORMATION. Information or data can be transmitted to or from registers in one of two forms. The information is said to be transmitted in parallel form when all the bits of a word are fed at the same time into a register. Parallel transfer of information requires a separate lead for each bit.

Information is fed in serial form by transmitting the bits one after the other over a single lead. Although many registers receive and transmit information in the same form, that is either serial or parallel, others are used primarily to change from one form of transmission to the other.

**9.3** PARALLEL IN, PARALLEL OUT REGISTERS. Fig. 54 shows a register which can store a four bit word fed into it in parallel form. Type D flip-flops, or latches as they are sometimes called, are used in this example, but unclocked SR flip-flops are also widely used in parallel to parallel registers.

To store the information available on the input leads a pulse of logic 1 is applied to the clock inputs of the register stages. The flip-flops then set or reset depending on the conditions on the input leads. Even though the input condition may change, the word stored is held in the register until a clock pulse initiates the storage of the new word. The conditions on the Q leads are extended to the output to provide a parallel output from the register.

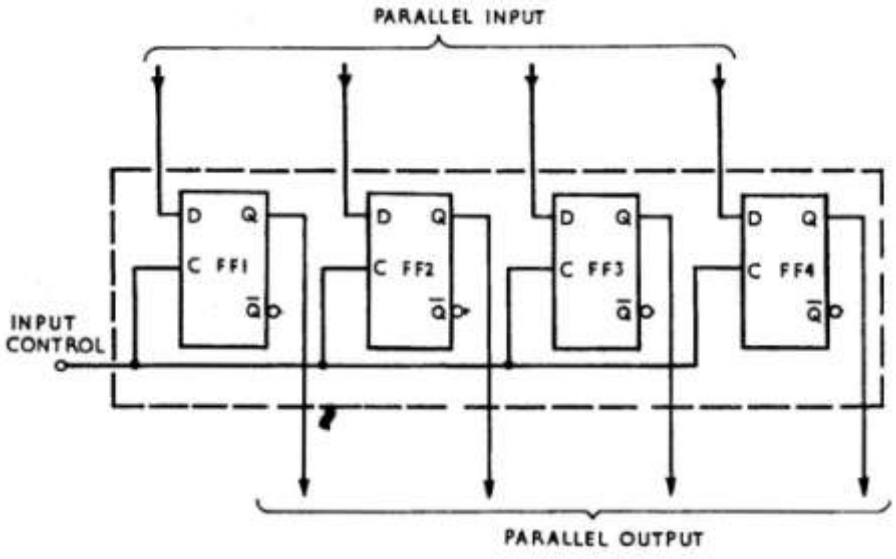



FIG. 54. PARALLEL IN, PARALLEL OUT FOUR BIT REGISTER.

**9.4** PARALLEL IN, SERIAL OUT REGISTERS. Fig. 55 shows a four bit register which receives information in parallel form and transmits it in serial form. This type of register is sometimes referred to as a "serialiser".

Assume that all stages are reset and the word 1001 is present on the input leads. A pulse of logic 1 on the input control lead removes the inhibit from the input gates. This causes FF1 and FF4 to set, and FF2 and FF3 to remain reset. Since type D flip-flops are used and the stages are connected as an unbroken ring, each stage sets or resets to the condition on the set output of the preceding stage when a shift (clock) pulse is applied. The first shift pulse, therefore, resets FF1 (there is a permanent logic 0 on the D input of FF1), sets FF2 and resets FF4; FF3 remains reset. These conditions are recorded in Table 11, which also shows the operation of the register on subsequent shift pulses.

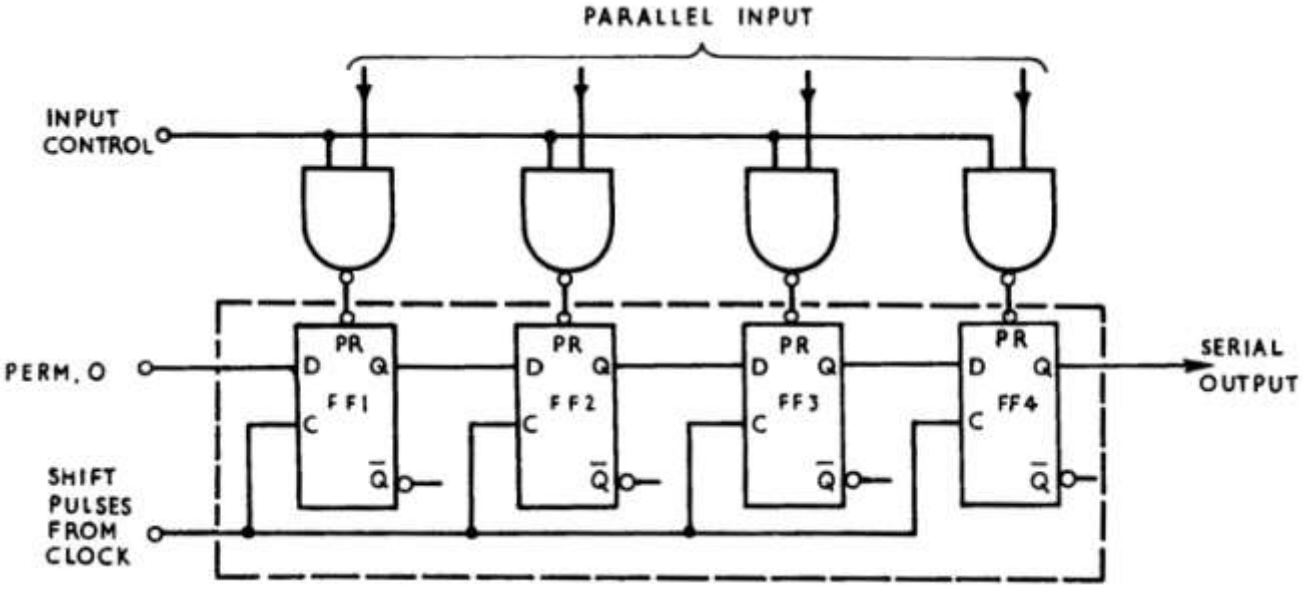



FIG. 55. PARALLEL IN, SERIAL OUT FOUR BIT REGISTER.

| | Output Conditions | | | | Register Serial Output |
|---|---|---|---|---|---|
| | FF1Q | FF2Q | FF3Q | FF4Q | |
| After input control pulse. | 1 | 0 | 0 | 1 | 1 |
| ( | 0 | 1 | 0 | 0 | 0 |
| ( | 0 | 0 | 1 | 0 | 0 |
| After subsequent clock pulses. ( | 0 | 0 | 0 | 1 | 1 |
| ( | 0 | 0 | 0 | 0 | 0 |

TABLE 11. TRUTH TABLE FOR FIG. 54.

Each shift pulse shifts the information one stage to the right. The serial output from the register is taken from the set side of FF4. The output voltage level on FF4Q changes to either logic 0 to logic 1 as each bit is shifted into FF4. Fig. 56 shows the waveform generated on the output lead.

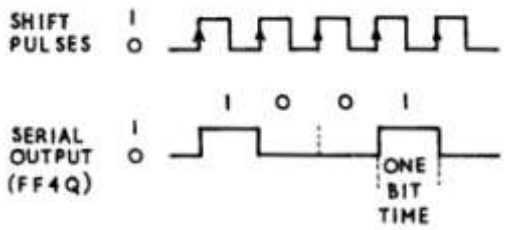



FIG. 56. GENERATION OF SERIAL OUTPUT.

Since a new bit of information is shifted into FF4 on each clock pulse, and FF4 remains unchanged until the next pulse, a time equal to the pulse period of the clock output is allocated for the transmission of each bit. This time is known as one "bit time", and is shown in Fig. 56.

Registers of this type, are often called "shift registers" because of the way information is shifted sideways through the register stages. Shift registers can also be constructed from clocked SR or JK flip-flops.

**9.5** SERIAL IN, PARALLEL OUT REGISTERS. Although serial transmission of information is slower than parallel transmission, it is used where data has to be transmitted over long distances, because only one lead (bearer circuit) is required. Serial-in, parallel-out registers, similar to that shown in Fig. 57, convert the serial information on the input lead into parallel information at the receiving end. Shift registers of this type are sometimes referred to as 'staticisers'.

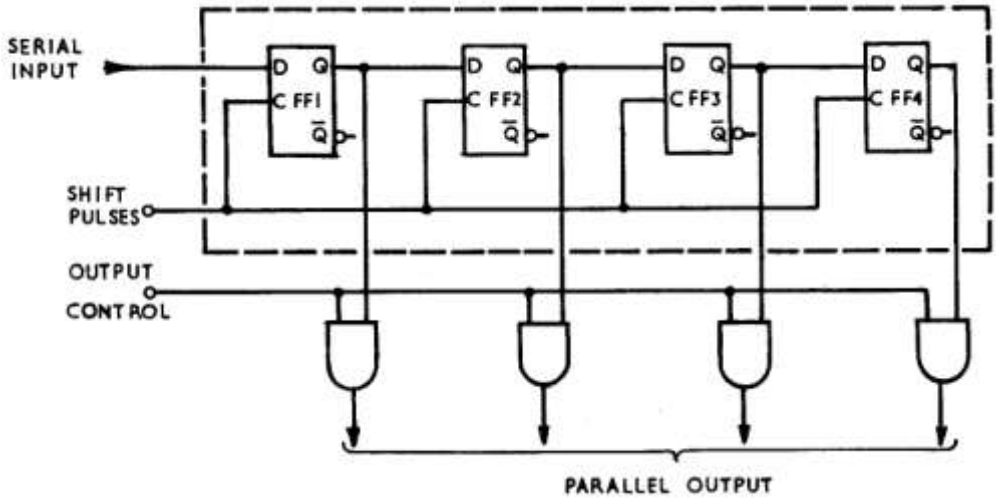



FIG. 57. SERIAL-IN, PARALLEL-OUT FOR BIT REGISTER.

Assume that the register is reset and that the word 1001 is received in serial form. The shift pulses are synchronised to be in step with the digital information on the serial input lead, so that the information can be shifted progressively through the register. When the input lead goes to logic 1 on the first input pulse and a shift pulse is applied, FF1 sets. The next shift pulse moves this 1 into FF2 and FF1 resets because an 0 is transmitted during the second bit time. Since an 0 is transmitted during the third bit time FF1 remains reset, FF2 resets, and the 1 is shifted into FF3 on the third shift pulse. The fourth shift pulse shifts the first 1 received into FF4 to set it, resets FF3 and shifts the 1 on the input lead into FF1. Therefore, after four shift pulses FF1Q=1, FF2Q=0, FF3Q=0 and FF4Q=1 and the whole word is stored in the register. To extract the word in parallel form the inhibit is removed from the output control gates.

**9.6** SERIAL IN SERIAL OUT REGISTER. Fig. 58 shows a serial-in, serial-out register which is similar in configuration to Fig. 57, except that a single wire output is taken from the set output of the last flip-flop, and parallel output gates are not required. When a word is stored in the register, it is extracted by applying four successive shift pulses to progressively shift the information out of the register in serial form, that is, each bit of information appears, in turn, at the output as the shift pulses are applied. Note, it is possible to insert a new word into the register in synchronism with the shift pulses that are moving a stored word out.

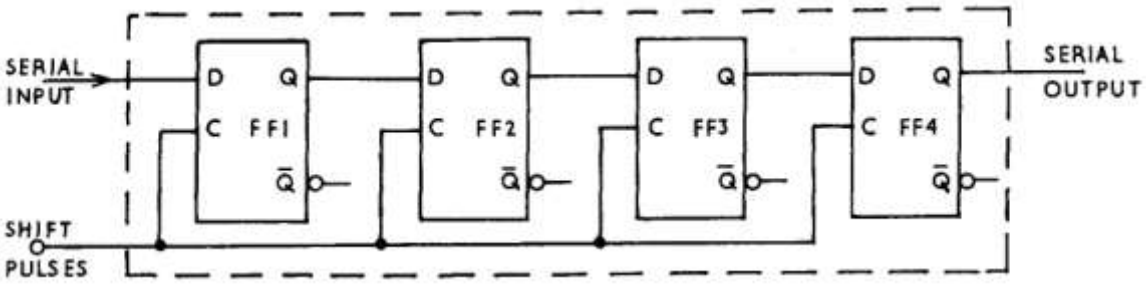



FIG. 58. SERIAL-IN, SERIAL-OUT REGISTER.

**9.7** REGISTER SYMBOLS. In some logic diagrams a single block symbol is used to represent a register containing a number of flip-flop stages. Fig. 59 shows two typical symbols for registers. In Fig. 59 the notation 'n' is replaced by the number which represents the number of bits that the register is capable of storing, and the lines showing the parallel inputs and outputs represent the same number of wires. Note that registers are usually manufactured to provide more than one facility, especially those manufactured in integrated circuit form. For example, the register represented in Fig. 59a provides the facility of parallel-in, parallel-out, or parallel-in, serial-out.

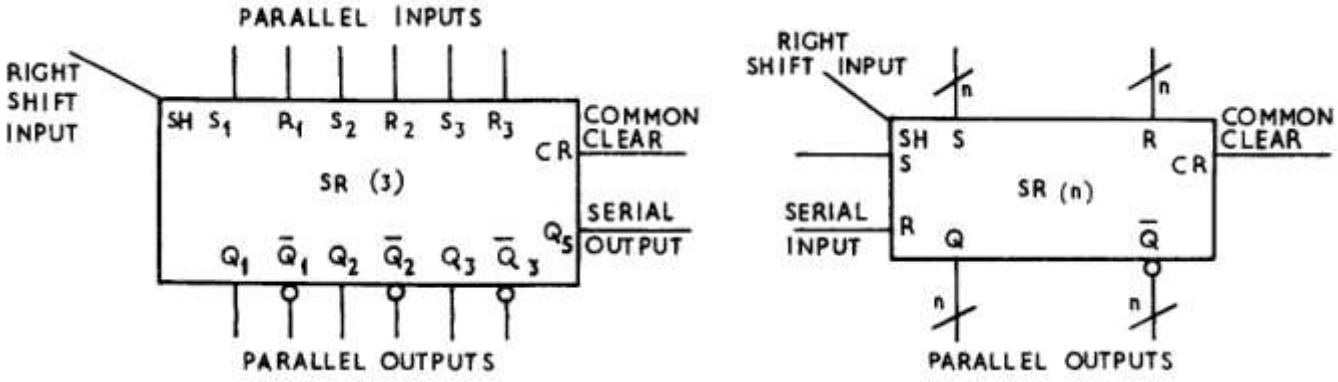



FIG. 59. REGISTER SYMBOLS.

**9.8** Registers are manufactured which are capable of shifting serial information in either direction and are often referred to as "shift-right shift-left" registers. The shift-right, shift-left designation has reference to arithmetical functions performed in certain applications and does not necessarily indicate which direction the information moves through the register circuit. The circuits and details of these registers are not dealt with in this paper, although, their principles of operation are similar to those already discussed.

NOTES

## 10. TEST QUESTIONS.

1. *Briefly explain the operation of the elements shown in Fig. 60 when a pulse of logic 1 is applied to their inputs.*


*(a)* *(b)*

*FIG. 60.*

2. *(a) Draw the functional logic diagram of an SR flip-flop using inverters and OR gates, and explain how it is set and reset.*

   *(b) Draw the functional logic diagram of an SR flip-flop constructed from NAND gates.*

   *(c) Draw the symbol for a NAND gate SR flip-flop.*

3. *(a) What is the difference between the functional operation of clocked SR and JK flip-flops?*

   *(b) Draw the functional logic diagram of a type D flip-flop, and explain how it is set and reset.*

4. *Sketch a functional logic diagram of a master-slave flip-flop and describe its operation.*

5. *Show how a JK flip-flop, and a type D flip-flop are strapped externally to operate as change-of-state flip-flops.*

6. *Briefly explain the functional operation of each of the flip-flops symbolised in Fig. 61. Include all conditions which will set or reset the flip-flops and state which of the clock pulse edges control the timing of the flip-flop operation.*

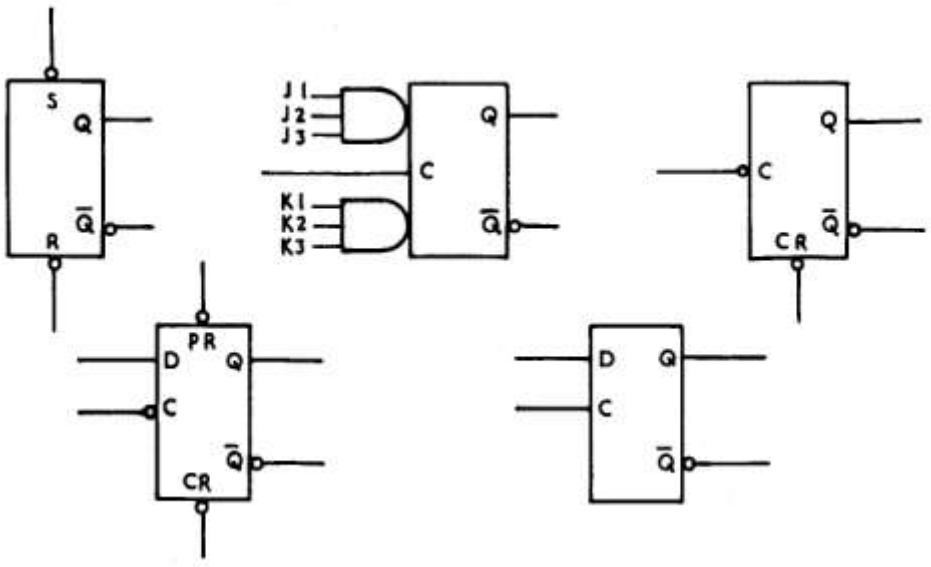



*FIG. 61.*

7. *(a) Draw the circuit of a three stage binary counter using change-of-state flip-flops.*

   *(b) Draw a timing diagram which shows the input pulses and the output conditions of the counter.*

   *(c) Assuming that the count numbers are derived from the conditions on the set side of the flip-flops state whether the counter you have drawn is a forward or backward counter.*

   *(d) How many output conditions are available from a five stage binary counter?*

8. *(a) Draw the circuit of a binary counter which uses feedback gating to provide 10 output conditions.*

   *(b) Show where outputs may be taken from this counter to provide frequencies which are one half input frequency.*

9. *Draw the functional logic diagram for a binary to decimal decoder which could be used on the output of a decade counter.*

10. *What is meant by the terms synchronous and asynchronous when used to describe binary counters?*

11. *(a) Draw the circuit of a ring counter which has four output conditions and explain its operation.*

    *(b) Draw the circuit of a twisted ring counter and the decoding gates required to provide logic 1 pulses on four output leads in turn. Explain its operation.*

12. *Draw the circuit of a three-bit parallel in, serial out, shift register and explain its operation.*

13. *The circuit in Fig. 62 is a typical sequence control circuit using binary counters. Assume all flip-flops are reset and use a timing diagram to explain its operation when a start pulse is applied. (Assume the start pulse has the same duration as the clock pulse).*

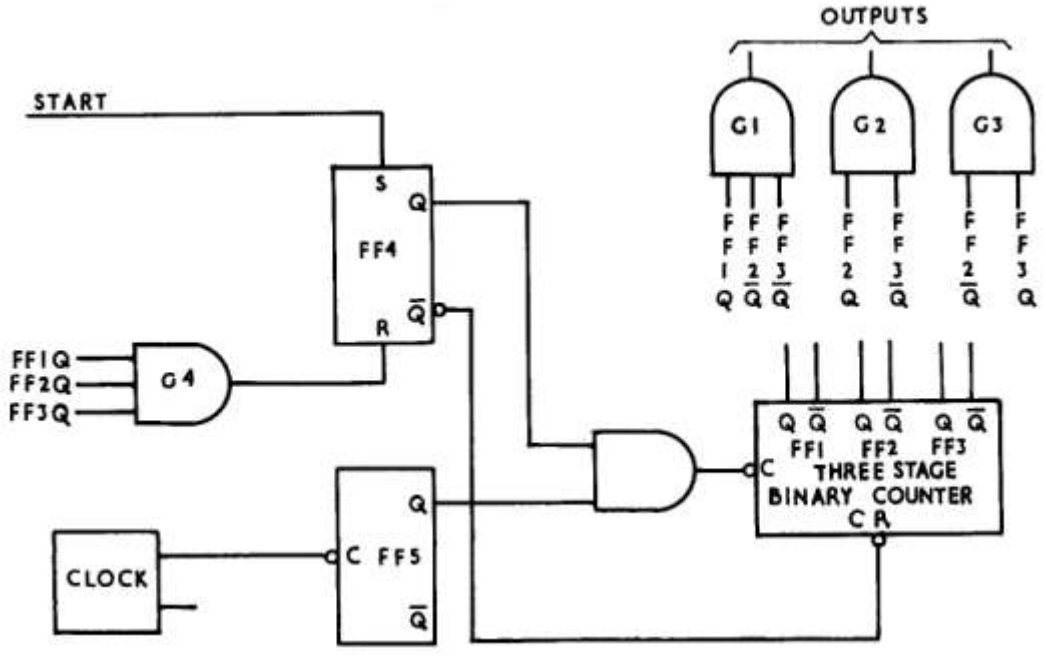



*FIG. 62.*

*14. The circuit in Fig. 63 is an asynchronous 1248 BCD counter. Explain its operation.*

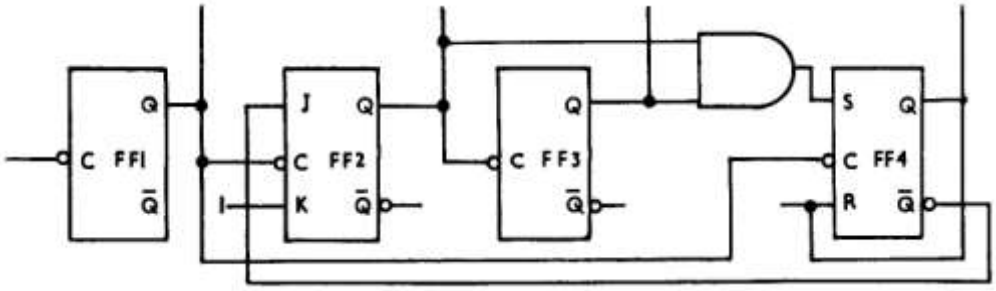



*FIG. 63.*

*15. Describe the operation of the counter shown in Fig. 64.*

*How many states does the counter have?*

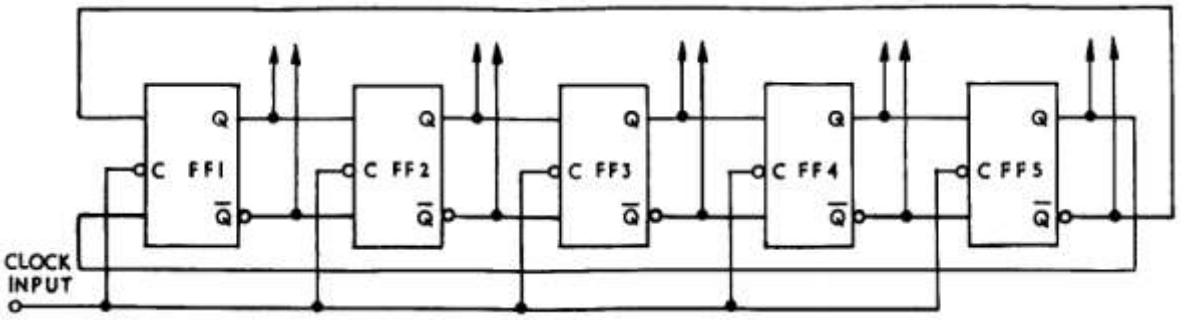



*FIG. 64.*

*16. (a) What is meant by the term Binary Coded Decimal?*

*(b) Show how a number of decode counters can be connected to count up to 10 million pulses.*

*(c) Assuming 1248 BCD decade counters are used, what is the binary number output on each stage when 13647 pulses have been counted?*

*(d) Describe the decoding arrangement required to provide a visual decimal readout on each stage.*

*17. Explain the operation of the shift register in Fig. 65 when the three input pulses shown are received. State the final condition on the output.*

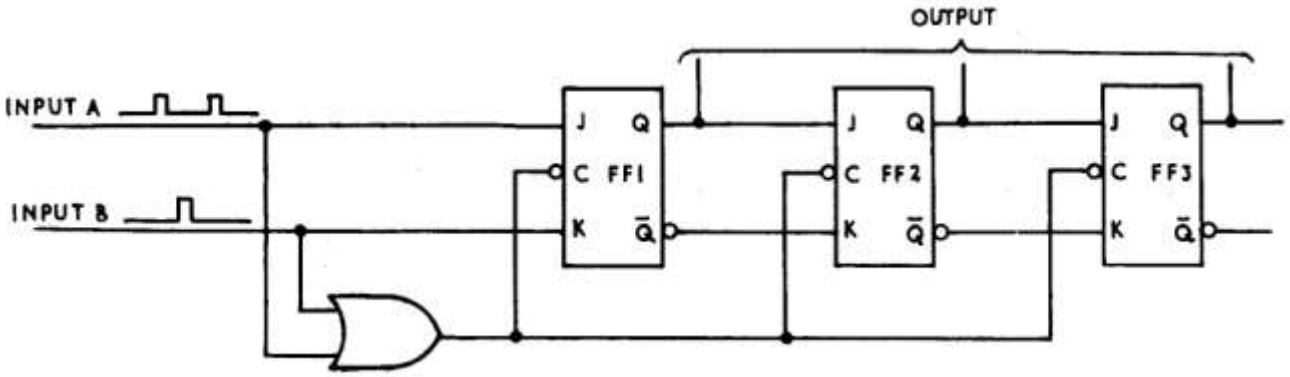



*FIG. 65.*

*END OF PAPER.*