NASA CONTRACTOR REPORT

NASA CR-1035

NASA CR-1035

FREQUENCY CODED THRESHOLD LOGIC UNIT FOR PATTERN RECOGNITION APPLICATION

by Rob Roy and David Hinds

Prepared by

RENSSELAER POLYTECHNIC INSTITUTE

Troy, N. Y.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1968

NASA CR-1035

FREQUENCY CODED THRESHOLD LOGIC UNIT FOR PATTERN RECOGNITION APPLICATION

By Rob Roy and David Hinds

Prepared under Grant No. NGR-33-018-014 by
RENSSELAER POLYTECHNIC INSTITUTE
Troy, N.Y.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The objective of this project was the development of a fast, accurate and reliable method of weight adaptation to be used in trainable pattern recognition systems. The sensory nervous system was investigated as being a possible model for the proposed device. The speed with which complex patterns are recognized and the binary nature of sensory systems indicated that the model would be worthy of further consideration.

The concept of frequency coding, as found in the sensory nervous system, provides a method in which modern digital techniques may be applied to incorporate speed and reliability into the design. Coding the input patterns as a pulse frequency, allows the use of binary equipment to instantaneously transmit an analog signal without quantization error or coding time delay. The concepts are then developed in the form of a Frequency Coded Threshold Logic Element.

The design presented in this project develops a discriminant function which is a linear combination of weighted frequencies and computed simply as a pulse count. Briefly, each input voltage is converted to a pulse frequency by using a voltage controlled astable multivibrator. Each input pulse stream is then scaled using a Binary Rate Multiplier, which allows the storage of each weight as the contents of a binary register. Summation and threshold detection are accomplished by counting pulses in a binary counter over a short period of time, at the end of which a classification is made based on the contents of the counter. The design presented here allows for both positive and negative inputs and weights. Further, provision is made for an adaptable threshold by considering the threshold as an additional input.

The design is applicable to the current pattern recognition systems and training procedures. It is capable of speed and accuracy in both classification and weight adaptation that cannot be obtained by using analog devices. The hardware requirements of the proposed design are given in detail and require only a moderate expenditure of equipment. The design concepts presented in this project should prove useful in the construction of practical pattern recognition systems.

PART (I.)

INTRODUCTION

A. Purpose

The purpose of this project is the investigation of the principles on which the sensory nervous system functions, and their application to pattern recognition problems. By sensory nervous system, we mean the peripheral nervous system through which environmental information and body conditions are transmitted to the central nervous system. It is the purpose of the sensory system, such as the visual, auditory and kinestetic systems, to supply the animal with specific pattern information. The recognition, evaluation and eventual action taken by the animal is determined by the central nervous system.

We are considering the sensory nervous system as a model for a pattern recognition device because of its information transmission characteristics. When one considers the amount of information and the efficiency by which it is transmitted from the eye to the brain, it is evident that these characteristics would be advantages to any pattern recognition problem.

We will explore the basic concepts of the sensory nervous system, and then apply them to the current concepts in the pattern recognition field. The final result will be the development of a pattern recognition device, employing the advantageous techniques of information transmission exhibited by the sensory nervous system, to improve the performance of pattern recognition systems based on the Threshold Logic concept.

B. Historical Review

The concept of artificial intelligence in machines is not new. Rosenblatt's PERCEPTRON was among the first attempts to design a trainable pattern classification machine.[11] It was Rosenblatt's idea to use the nervous system as a model for his design. Other pattern classification systems, such as the Adaline and Madaline networks, have also been proposed and are well covered in the literature.[13, 14]

Pattern recognition is based on the concept of the discriminant function, which defines the behavior of the categorizer. The machine is considered trainable when the discriminant function can be modified by an iterative procedure. Nilsson presents an excellent review of the current state of trainable pattern recognition devices.[10] Hawkins presents a comprehensive survey of the early work done in the field.[4]

There exists ample literature available in the pattern recognition field, and we shall assume the reader is familiar with the fundamental concepts of Threshold Logic. Included in the Selected Reference, are a number of publications dealing with specific classification systems and techniques, training methods and convergence proofs, and various implementations of analog memory devices. No attempt is made in this project to discuss these papers. We will present only those concepts that are used to explain the work done here.

PART (II.)

THEORY

A. Nervous System

Physiological research into the neural structure of the vertebrates has not been able to give a complete picture, or explanation, of the nervous system. This is partly due to the complexity of the system and partly to the minute structure of the components. The research reports that do exist are often contradictory and inconclusive. The following discussion is a summary of the basic concepts accepted by the majority of researchers.

There are two distinct types of nerve cells known to exist. They are characterized by the manner in which they respond to stimuli. These responses can be classified as the "all or nothing" principle and the "pulse rate" principle. The first, is found in the cells of the central nervous system. A cell will produce a pulse only if there is a sufficient number of incoming pulses arriving approximately at the same time. The central nervous system is located in the brain and spinal structure. Operation is dynamic in nature, responding to system changes rather than static conditions. It is capable of responding to an internal or environmental change rapidly and in a complex manner. The advantages of the method of operation seems to lie in the versatility of the system in handling a wide variety of recognition and decision problems.

The "pulse rate" principle is found in cells of the peripheral or sensory nervous system. When stimulated, they will generate a series of impulse spikes whose frequency is proportional to the intensity of the stimulus. The amplitude of these impulses is observed to be relatively constant over all frequencies. The upper frequency at which the cells

will operate is determined by the refractory period of the cell membrane. In the human the maximum frequency is about 300 pulses / second.[12]

Studies of the visual systems of the cat and horseshoe crab have shown the impulse frequency to be proportional to the cell membrane potential and resistance.[5,12] Incident energy, in the form of light, chemically changes the cell resistance and the frequency of impulse generation. The function of the "pulse rate" neuron is shown in figure 1. The output frequency is a linear combination of the input frequencies. The respective weighting of each input is done in the synaptic junction of the nerve cell and may be either positive or negative. However, existence of a threshold below which the cell will not fire is questionable. Kuffler has shown that the retinal ganglion cells of the cat discharge at a constant rate without stimulation. The effect of stimulation may either increase or decrease this rate, depending on where in the visual field the stimulus falls.[5] This phenomenon can be explained by considering the retinal structure. The retinal ganglion cell is a third layer cell receiving impulses from the bipolar cells. By stimulating the receptors of a particular bipolar cell, and then weighting the synaptic junction of this cell as an inhibitory input to the retinal ganglion cell, an increase in stimulus energy will be observed as a decrease in firing rate.

Hubel has shown that there exists a high degree of rigid organization in the visual nervous system.[5] However, cell adaption is also evident. For the most part, it is a means of varying resolution to maintain pattern recognition over a wide range of stimulus variation. Perhaps, the best example is the ability of the eye to recognize patterns in bright light as well as semi-darkness, with the only loss of information being in the resolution of details. The cell does this by adapting

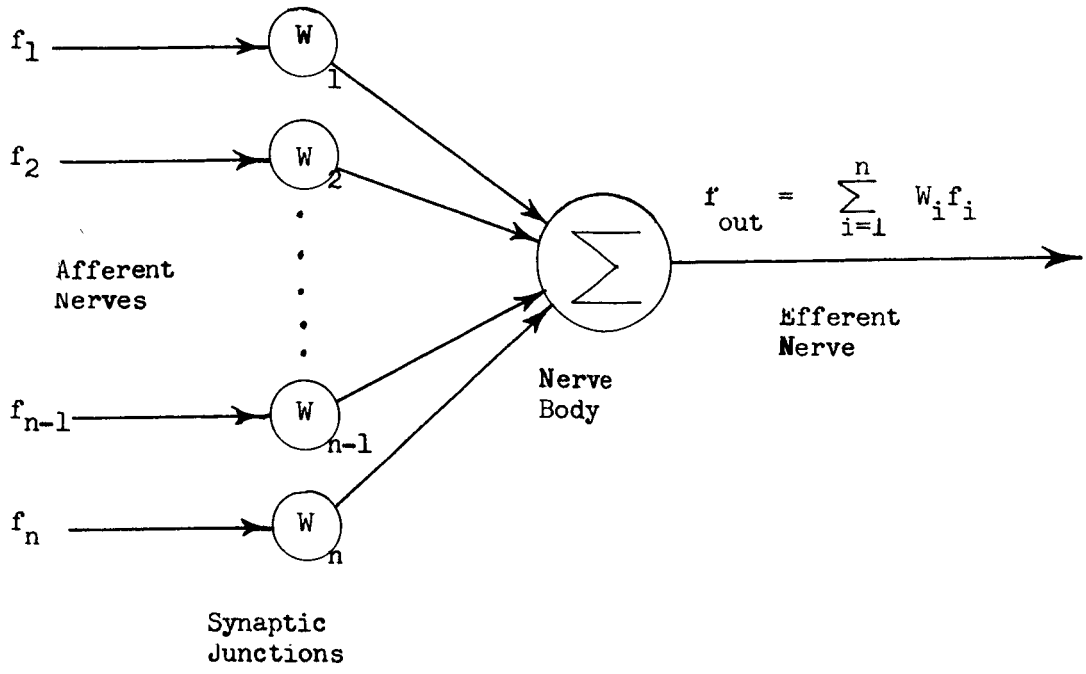


Figure 1 Operational Model of a Pulse Rate Neuron

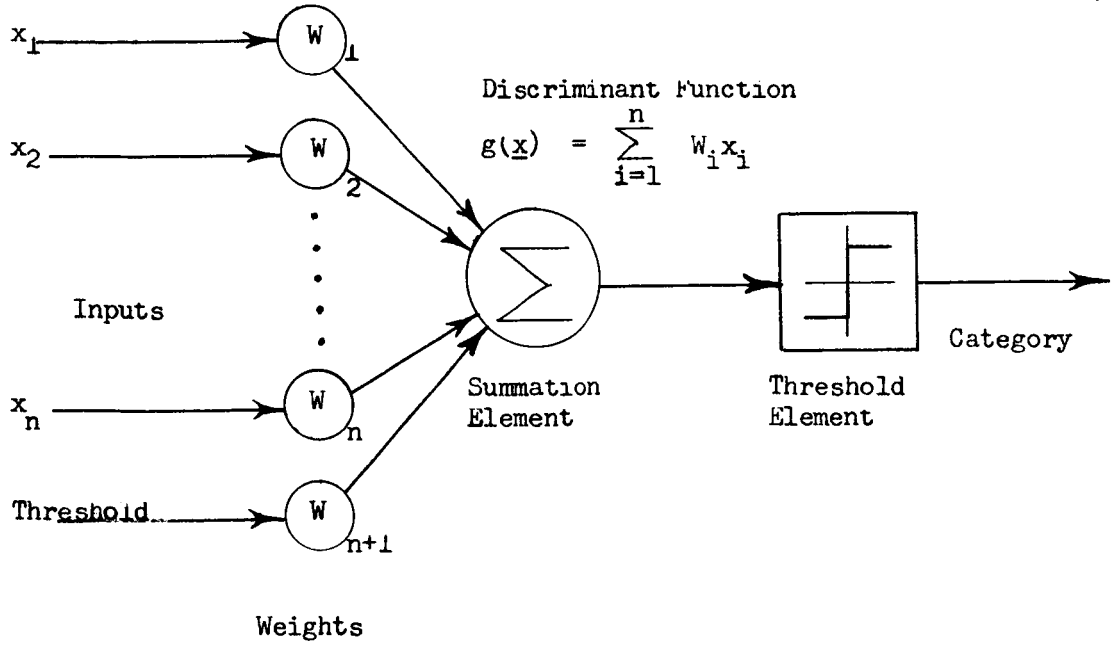


Figure 2 Structure of the Threshold Logic Unit

weights to maintain the firing frequency approximately in the center of its linear range.

The role of the "pulse rate" neuron in the physiological nervous system is that of information transmission. Information is sent continually to the central nervous system, where it is processed. It is the changes in the environment that the central nervous system acts on. However, it is a constant monitoring of the environment that is carried out by the sensory neuron.

B. Problem Definition

The main objective of this project is the development of a pattern recognition device based on the principle of the sensory nervous system. The remainder of this project report will be devoted to that goal. We will not attempt to simulate a sensory neuron, but rather, will use the concept of frequency coding information to develop a device applicable to the current pattern recognition techniques.

The basic element of a pattern recognition system, and the simplest form of a complete system, is the Threshold Logic Unit. Extension of the Threshold Logic Unit to the more complex forms of pattern recognition devices is fairly straightforward. For this reason, and to keep the project to reasonable proportions, we will confine our attention to the development of a Frequency Coded Threshold Logic Unit.

The Threshold Logic Unit functions, as previously discussed, by generating a discriminant function, which is a linear combination of the weighted inputs. Classification is then accomplished by comparing the value of the discriminant function with a threshold value. In order to allow for the adaptability of the threshold, it is considered as an

additional input. This procedure permits the decision to be based on a constant value, which we will assume to be zero. Thus, those patterns whose discriminant functions are less than zero are in category one, and those greater than zero are in category two. The structure of a Threshold Logic Unit is shown in figure 2. It can be considered as consisting of four distinct parts; the input sensor, the weight, the summation junction and the decision element.

Each part will be developed to function similarly to its physiological counterpart in the sensory nervous system. In addition to the physiological motivation for the project, consideration has also been given to the engineering aspects and applications for the Frequency Coded Threshold Logic Unit. To be of practical significance in the pattern recognition field, its design should use concepts and components within the present state of practical engineering, be capable of reasonable speed and accuracy, and should exhibit distinct advantages over the present existing methods and devices. We will give a theoretical discussion of the proposed device, detailed equipment requirements, and finally demonstrate the use of the Frequency Coded Threshold Logic Unit by a computer simulation of a pattern recognition problem.

C. Design Considerations

1. Sensor Implementation

The first stage of the Threshold Logic Unit, the sensor, converts the features of the pattern into input information in a form acceptable to the system. In the sensory nervous system, stimulus energy is converted to impulse frequency. The conversion is not linear, but rather logarithmic, as Rushton has shown.[12] However, this non-linear aspect of the conversion process can be considered as a method of

extracting invarient features from the pattern. As such, it is another problem, and we will not consider it here. We shall assume, instead, that our pattern inputs are the desired features and will use a linear conversion of input voltage to frequency. We shall further restrict our allowable pattern inputs to a specified voltage range centered about zero.

By converting the input voltage to a pulse frequency, we are essentially coding the information contained in each feature of the pattern. The coding is carried one step further in that the magnitude of the voltage will produce a pulse frequency independent of sign. The sign is also retained and used in the determination of the discriminant function.

2. Weight Implementation

Before considering the methods used to weight the input frequencies and sum the results, we must look at the decision process in its entirety. The decision of the Frequency Coded Threshold Logic Unit is based on the discriminant function,

$$g(f) = \sum_{i=1}^{n} W_i f_i + W_{th}$$

where:

N = Number of input lines
W_i = Are the weights
W_{th} = Is the threshold weight
f_i = Are the input frequencies

compared to a threshold value. We shall allow both positive and negative weights as well as positive and negative inputs. This requires retaining and processing sign as well as magnitude information.

In order to identify a specific pulse frequency, the pulse train must be observed for at least one period. We propose to observe each input

pulse train for a time ΔT , called the decision time. A decision, or categorization of an input pattern will occur at the end of each ΔT interval. The discriminant function thus becomes,

$$g'(f) = \sum_{i=1}^{n} W_i f_i + W_{th} .$$

Further, $f_i \Delta T = n_i$ and represents a specified number of pulses occuring on the i^{th} input line in one decision interval.

This leads to the formulation of the discriminant function,

$$g'(f) = \sum_{i=1}^{n} n_i W_i + W_{th} .$$

We now have a linear combination of pulses rather than frequencies. The threshold weight is chosen as being independent of decision time and is given as a pulse count. Considering the entire decision process in this light, allows the weighting and summation processes to be implemented as manipulations of a pulse count, rather than attempting to maintain the frequency concept throughout the device.

Perhaps, the most important part of any trainable pattern recognition system is the method of implementing the weights. The physiological counterpart is the synaptic junction of the nerve cell, where weighting each incoming nerve is a complex electro-chemical process. The effect of an incoming nerve can be excitory or inhibatory. Correspondingly this proposal will allow positive and negative values for the weights. Among the major problems of pattern recognition systems, is input modification and the associated storage and adaptability of the modifying weights. The input weighting process must be accurate, reliable and rapid. Weight

storage must be constant over relatively long periods of time and capable of reversible adaptation,at a reasonably rapid rate and by accurately controlled amounts.

The proposal of this project is to use the techniques of modern digital circuitry to develop an adaptive system meeting the above requirements. It is in this respect that the concept of frequency coding becomes advantageous. Binary devices can be used without first quantizing and coding the inputs. The weights are quantized, however, there are no theoretical restrictions on the number of levels. Practically, a maximum would be 10 bits, or to within one tenth of one percent. The weights are allowed to range from -1 to 1 in steps of $1/2^n$, where n is the number of binary bits.

The accuracy with which a weight is implemented is discussed under Error and Accuracy Considerations. The accuracy to which the weight can be adopted is limited only by the quantization level, since the weight can be adapted exactly to any desired level. In these respects, the digital weight is superior to any analog memory device currently available for adaptive application.

The proposal presented in this project requires the weighting element to scale the incoming pulse train and also retain sign information. The sign is stored as a single bit, which is the method used to handle signs throughout the design of the device. There were two methods considered for implementing the scaling of the pulse train. The first was the Digital Differential Analyzer.[2,6]

The structure of Digital Differential Analyzer is shown in figure 3. The device operates in such a manner that an input pulse causes the addition of the contents of the weight register to be added to the

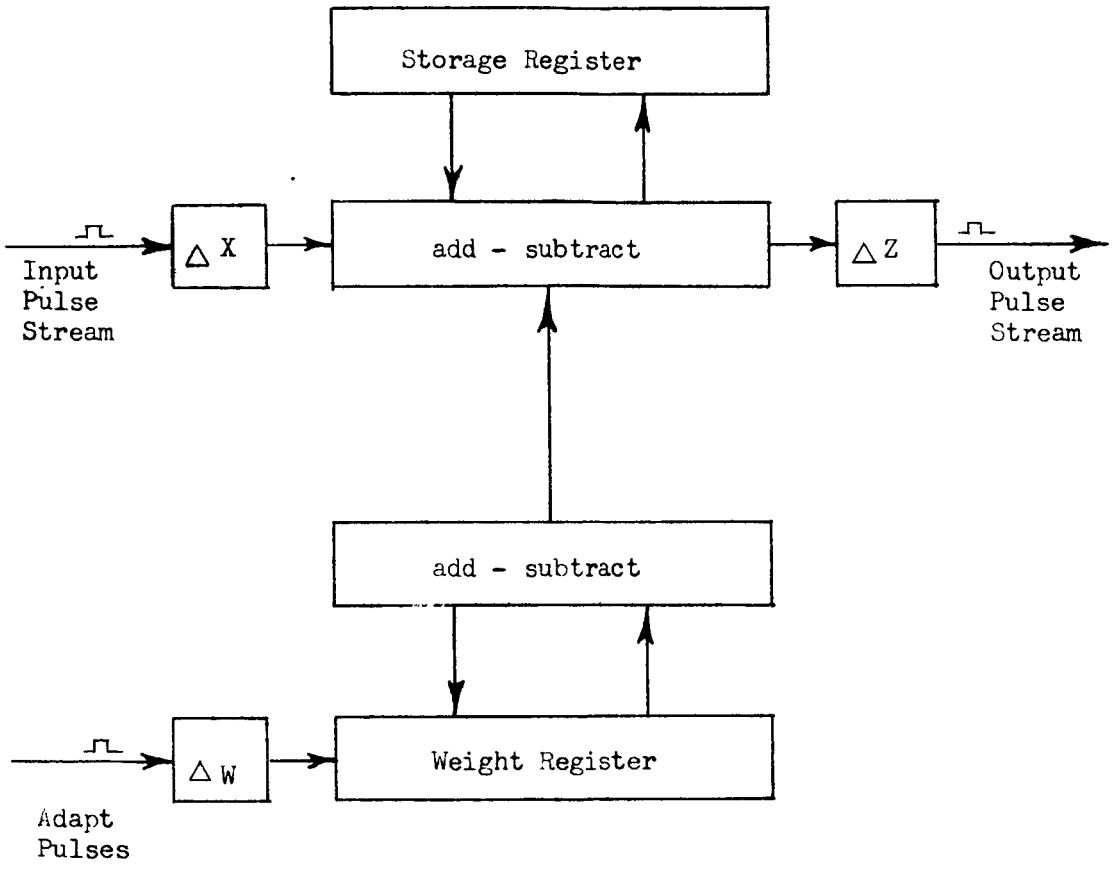


Figure 3 Digital Differential Analyzer

contents of the storage register. The result of this addition is then entered in the storage register. An output pulse is produced each time the storage register overflows. The number of output pulses, $\triangle Z$, is determined by the number of input pulses, $\triangle X$, the weight, W, represented by the contents of the Binary register, and the number of stages, n, in the register. The equation is $\triangle Z = (1/2^n) \times W$. The result is that the input pulse train is scaled by a factor of $W/2^n$. Adaptation is accomplished by adding or subtracting $\triangle W$ bits, one at a time, from the weight register. The Digital Differential Analyzer is used in the design of incremental computers and, therefore, has a wide variety of application. It is felt that the equipment required to obtain this versatility is far too complex for application in the Frequency Coded Threshold Logic Unit. Hence, it will not be considered further in this design.

The second method, and the one chosen for this application, is the Binary Rate Multiplier.[1,3,7,8,9] The basic structure of the Binary Rate Multiplier is shown in figure 4. The equation for the device is the same as given for the Digital Differential Analyzer. However, the implementation is quite different.

The input pulse train is presented to an n stage binary counter, called the input counter, where n is the same as the number of bits used in the weight register. The input pulses ripple through the counter as each stage makes the transition from the 1 to the 0 state. The ripple is accomplished by the generation of a carry pulse, as the stage makes the 1 to 0 transition, which then acts as an input to the next stage. The carry pulse is propagated until a transition from 0 to 1 occurs. At this point no carry pulse is generated, but a pulse is transmitted from that stage to its associated scaling gate. At each scaling gate there appears

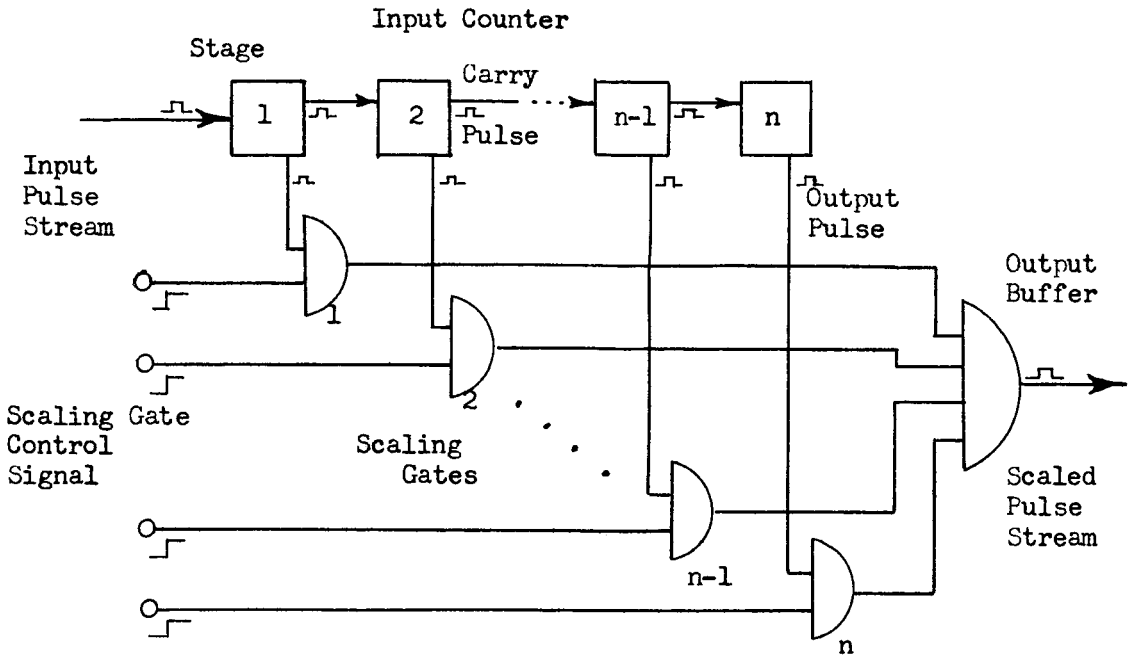


Figure 4 Binary Rate Multiplier

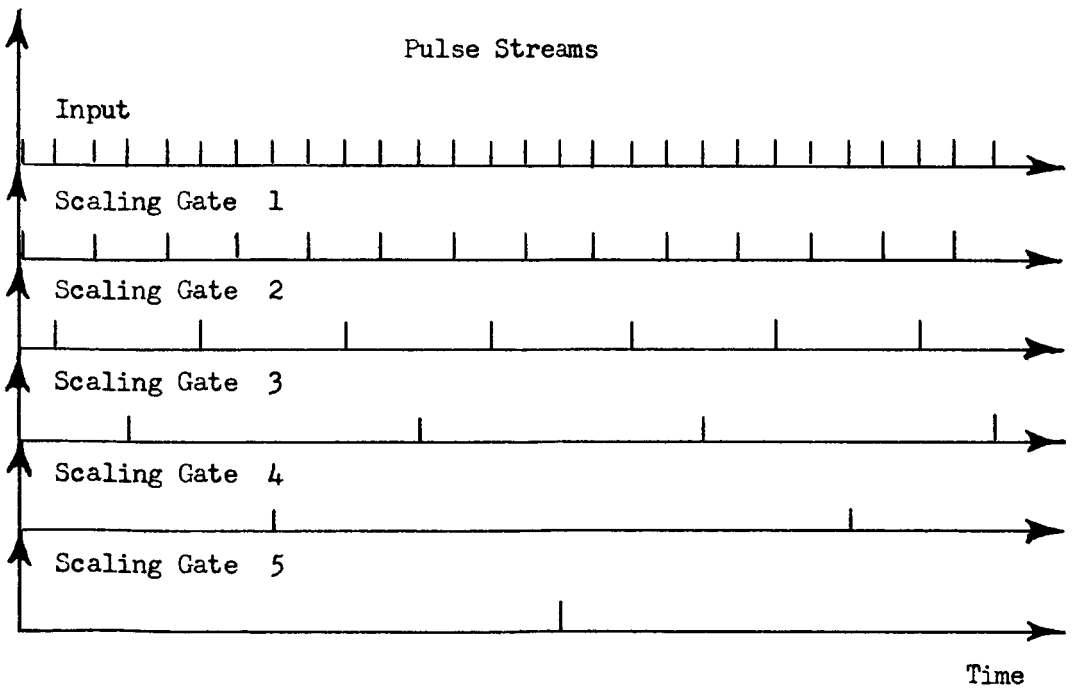


Figure 5 Timing Diagram for the First 5 Stages of the Binary Rate Multiplier

a specific fraction of the input pulse train, determined by the number of input pulses required to cause the 0 to 1 transition of that stage. The k_{th} stage will scale the pulse train by a factor of $1/2^k$ since it takes 2^k input pulses to cause the 0 to 1 transition.

The actual fraction of the input pulse train that is transmitted to the output buffer is determined by the combination of scaling gates that are open. The condition of the scaling gate is controlled by the associated stage of the weight register. A 1 on the k_{th} stage of the weight register permits the transmission, to the output buffer, of the pulses arriving at the k_{th} scaling gate. The total scaling fraction, then, is the number represented by the binary contents of the weight register divided by 2^n, for an n bit weight. By scaling the pulse train, we have accomplished the desired scaling of the pulse count over the decision interval.

The timing diagram, figure 5, shows the fractional scaling operation at each stage of the input counts. From the timing diagram, it is evident that there can be no simultaneous occurrence of pulses at the output buffer, thus presenting no synchronization problems. The form of the Binary Rate Multiplier used in the final design has an additional stage in the weight register and an associate scaling gate. The additional equipment is optional and is used to transmit the overflow pulse from the last stage of the input counter to the output. It allows the entire input pulse train to be transmitted to the output and corresponds to a maximum weight of unity. Adaptation is accomplished by pulsing the weight register to count up or down. The adaptation algorithm is discussed in conjunction with the training procedure.

3. Summation Element Implementation

When considering the implementation of the summation junction, we again use the concept of pulse count to determine the discriminant function. To obtain the summation of pulses occurring on all input lines, we need only feed the pulse stream from the output buffer of each Binary Rate Multiplier into a single counter. The sign of the input is used to determine whether the count is up or down. The contents of the counter will be the signed value of the discriminate function after the decision time has elapsed.

Provision must be made, however, for the simultaneous occurrence of pulses on two or more lines. It will be recalled that simultaneous occurrence of pulses on a single line cannot occur. This provision entails the synchronization of the imput lines by delaying the pulses and presenting each line in sequence to either the up or down counter buffer. The only restriction in this procedure is that the entire set of inputs, including the threshold, are sequenced to the counter at a rate at least equal to the maximum allowable input frequency. This will ensure that no pulse will be lost by the successive arrival of two pulses on the same line. Actual equipment used for this procedure will be discussed later.

4. Decision Element Implementation

The final stage of the Threshold Logic Unit is the decision element. The decision process, whether neural or electronic, entails the comparison of the discriminant function with a predetermined threshold value. In the nerve cell this threshold is the firing potential of the cell, below which the cell will not produce an impulse. In the decision

process it must be remembered that, it is the central nervous system and the "on-off" principle being considered. In the pattern recognition problem, the threshold is determined by a fixed value to which the discriminant function is compared.

In this project we propose to use zero as the fixed threshold value. This allows making the final decision using only the sign bit of the summation counter. There is provision for threshold modification during the training period. This is done by considering the threshold weight as an additional input. The input frequency is a constant and the weight is allowed to be positive or negative and contains the same number of bits as the other weights. In order that the threshold exert an equal effect on the discriminant function in comparison with the other weights, the input frequency should be limited to the same range as the other input lines.

There is a classification or decision made only at the end of the decision interval. This implicitly assumes that the inputs are approximately constant over the decision interval. This is not a serious restriction since decision times of less than a tenth of a second are readily attainable, with good accuracy, for a problem with 100 inputs. Further, even if several inputs should vary the effective input to the system will be an average of the actual input over the decision interval.

D. Training Procedure

The previous discussion has dealt with the pattern classification aspect of the pattern recognition problem. We now turn our attention toward training procedure and weight adaptation. We will assume that the training patterns and the proper category have been determined prior to training, and that they will be presented in sequence by an external agent,

called the "teacher". The method of presentation depends on the particular problem and will not concern us here. We need only assume that the "teacher" will introduce a new pattern upon receiving a signal from the classification device indicating that the prior adaptation has been completed. It is also assumed that the "teacher" will continue the training procedure until satisfied with the performance of the machine.

As was previously stated, the weights are stored as the contents of a binary register and are modified by pulsing the register to count up or down. The following discussion will develop an algorithm for systematic adaptation of the weights, which is consistent with current non-parametric training procedures. Specifically, we will use the fixed increment correction rule described by Nilsson.[10]

Adaptation is only allowed if the classification of a training pattern is erroneous. Should a pattern be incorrectly classified, the weights are modified using the following rule:

$$\underline{W}' = \underline{W} + c\ \underline{Y}$$

where $\underline{W}$ is the weight vector whose first n components are the input weights and n + 1 component is the threshold weight.

$\underline{Y}$ is the augmented pattern vector whose first n components are the pattern inputs and whose n + 1 component is the relative input to the threshold weight.

c is the constant which determines the increment of adaption.

For our purposes c will be considered the number of bits each weight is adapted per volt input, and will be used to determine the number of pulses gated to the adapt input of each weight. Whether the weight is adapted up or down is determined by the sign of the input and the category into which the pattern was erroneously classified.

The sign of the constant c is determined by the value of the

discriminant function. The sign will be opposite to the sign of the discriminant functions produced by either positive or negative incorrect classification. This allows the reinforcement of the weights which will tend to correct the discriminant function for that particular input pattern, and decrease the effect of those inputs which tend to adversly effect the value of the discriminant function. The procedure has been shown to be convergent, but speed will depend on the choice of c.[10]

E. Error and Accuracy Consideration

In the preceding discussion we developed a discriminant function in terms of a pulse count. In order to control the accuracy of the Frequency Coded Threshold Logic Unit, we will derive a bound on the maximum deviation of the actual pulse count from the desired. We have previously indicated that the weights can be quantized to any desired accuracy and adapted to any desired quantization level. Now we will consider the error introduced in the process of determining the final pulse count.

We note that the input frequency generation and weighting are both done considering only the absolute value of the quantities concerned. Likewise, the error in the summation process is independent of sign. In discussing the maximum error bound, we need only consider the magnitude of the maximum deviation. The actual error will be well below the bound developed here, since positive and negative errors will tend to cancel in the final decision. There are three independent sources of error to be considered. First, is the linearity of the input voltage to frequency converter. Second, the error introduced by the Binary Rate Multiplier in scaling the pulse train. Finally, the error introduced by the summation process.

The accuracy of the pulse count is determined by the deviation of the actual pulse count from the value of the discriminant function.

Total pulse deviation = $\left| g'(f) - \text{pulse count in the summation counter} \right|$,

where $g'(f) = \sum_{i=1}^{n} W_i f_i + W_{th}$.

The bound developed here is for a single input line and must be multiplied by the number of input lines to obtain the total deviation. However, the figure of most interest is not the pulse deviation, but rather the percent deviation, measured as the pulse deviation with respect to the maximum number of pulses possible. This figure is given as a percentage and is independent of the number of input lines. The figure is important because it gives a measure of the precision of the final count, which should be approximately the same as the precision with which the weights are quantized.

Table 1 shows the maximum error bounds for the specific design indicated, in pulse deviation per input line and in percent deviation, for each stage of the device and for the total design. The input error is given as a percent deviation from linearity. The maximum pulse deviation is dependent on the maximum input frequency, f_{max}, and the decision interval, ΔT. It is computed as follows:

$$\text{Maximum input pulse deviation} = \frac{\% \text{ Deviation from linearity}}{100} \times f_{max} \times \Delta T$$

The maximum pulse deviation introduced by the Binary Rate Multiplier is dependent on the initial state of the input counter, the contents of the weight register and on the number of stages used in the register. Moshos has computed various bounds on the maximum pulse deviation

with respect to these variables.[9] He shows that for an arbitrary weight and initial counter values, the maximum deviation depends on the number of stages, n, in the following manner:

$$\text{Maximum Binary Rate Multiplier pulse deviation / input line} = \frac{7}{9} + \frac{n}{3} + \frac{(-1)^n}{9(2)^{n-1}}$$

Figure 6 shows the error characteristic of the Binary Rate Multiplier. As is indicated, the pulse scaling is exact after each 2^n input pulses, where n is the number of stages. In the computation of the maximum percent deviations in table 1, we have given the results as a function of cycles of 2^n pulses. What this means is that each input line sees a minimum number of pulses equal to 2^n x the number of cycles. We have also computed the decision time, since the number of cycles is considered as the system parameter, rather than the decision time.

The summation process introduces a maximum deviation of one pulse per input line. This error arises when a pulse arrives at the input to the summation buffer after that input line has been sequenced into the summation counter. This can occur only on the final input sequencing iteration before the end of the decision period. Should this occur on any iteration prior to the last, the pulse will be picked up on the next iteration. For proper operation of the summation element, each input must be sequenced at least once between each input pulse.

We compute the percent deviation by first calculating the total range per input line that the summation counter can assume. For a weight range of 0 to 1, the maximum number of pulses per input line is

$$\text{Maximum number of pulses / input lines} = f_{max} \times \Delta T$$

The percent deviation is:

$$\text{Maximum percent deviation} = \frac{\text{Maximum pulse deviation / input line}}{\text{Maximum number of pulses / input line}} \times 100$$

Table I Summary of Error Characteristics for

Variation in the System Design Parameters

Number of Binary Stages		Bit Weight (percent)			Binary Rate Multiplier Deviation (pulses/line)		
7		.78			3.11		
Cycles of 2^n	Counter Iterations	Decision Interval (10^{-3}sec)	Pulse Deviation		Percent Deviation		
			Input	Total	Weight	Summation	Total
2	320	6.40	1.60	5.71	.97	.31	1.78
3	480	9.60	2.40	6.51	.65	.21	1.36
4	640	12.80	3.20	7.31	.49	.16	1.14
5	800	16.00	4.00	8.11	.39	.13	1.01
6	960	19.20	4.80	8.91	.32	.10	.93
7	1120	22.40	5.60	9.71	.28	.089	.87
8	1280	25.60	6.40	10.51	.24	.078	.82
9	1440	28.80	7.20	11.31	.22	.069	.79
10	1600	32.00	8.00	12.11	.19	.063	.76

Number of Binary Stages		Bit Weight (percent)			Binary Rate Multiplier Deviation (pulses/line)		
8		.39			3.45		
Cycles of 2^n	Counter Iterations	Decision Interval (10^{-3}sec)	Pulse Deviation		Percent Deviation		
			Input	Total	Weight	Summation	Total
2	640	12.80	3.20	7.65	.52	.16	1.19
3	960	19.20	4.80	9.25	.36	.10	.96
4	1280	25.60	6.40	10.85	.27	.078	.84
5	1600	32.00	8.00	12.45	.22	.063	.78
6	1920	38.40	9.60	14.05	.18	.052	.73
7	2240	44.80	11.20	15.65	.15	.045	.70
8	2560	51.20	12.80	17.25	.13	.039	.67
9	2880	57.60	14.40	18.85	.12	.035	.65
10	3200	64.00	16.00	20.45	.11	.031	.64

Number of Binary Stages		Bit Weight (percent)			Binary Rate Multiplier Deviation (pulses/line)		
9		.19			3.77		
Cycles of 2^n	Counter Iterations	Decision Interval (10^{-3}sec)	Pulse Deviation		Percent Deviation		
			Input	Total	Weight	Summation	Total
2	1280	25.60	6.40	11.18	.30	.078	.87
3	1920	38.40	9.60	14.38	.20	.052	.75
4	2560	51.20	12.80	17.58	.15	.039	.69
5	3200	64.40	16.00	20.78	.12	.031	.65
6	3840	76.80	19.20	23.98	.10	.026	.62
7	4480	89.60	22.40	27.18	.08	.022	.61
8	5120	10.24	25.60	30.38	.07	.020	.59
9	5760	11.52	28.80	33.58	.07	.017	.58
10	6400	12.80	32.00	36.78	.06	.016	.57

Number of Binary Stages		Bit Weight (percent)			Binary Rate Multiplier Deviation (pulses/line)		
10		.10			4.11		
Cycles of 2^n	Counter Iterations	Decision Interval (10^{-3} sec)	Pulse Deviation		Percent Deviation		
			Input	Total	Weight	Summation	Total
2	2560	51.20	12.80	17.91	.16	.039	.70
3	3840	76.80	19.20	24.31	.11	.026	.63
4	5120	102.40	25.60	30.71	.08	.020	.60
5	6400	128.00	32.00	37.11	.06	.016	.58
6	7680	153.60	38.40	43.51	.05	.013	.57
7	8960	179.20	44.80	49.91	.05	.011	.56
8	10240	204.80	51.20	56.31	.04	.010	.55
9	11520	230.40	57.60	62.71	.04	.009	.54
10	12800	256.00	64.00	69.11	.03	.008	.54

Computations based on the following assumed parameters:

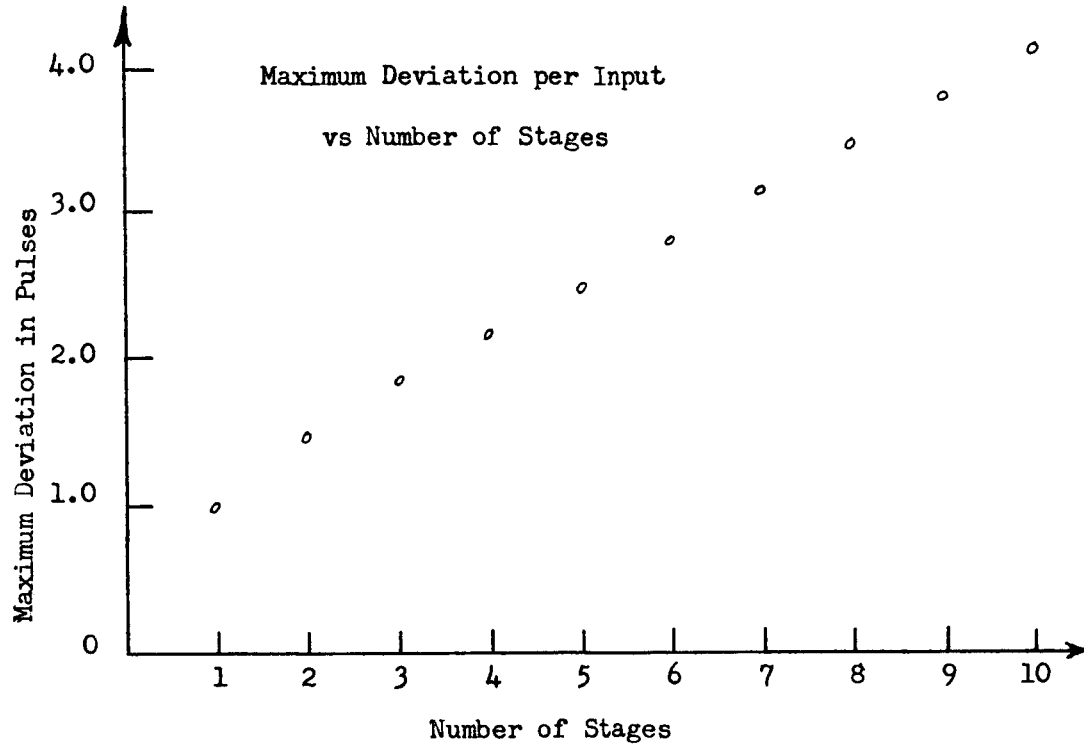
Maximum input frequency = 50,000 cps

Minimum input frequency = 40,000 cps

Percent deviation from linearity of the input frequency generator = .5 percent

Maximum pulse deviation/input line of the summation element = 1 pulse

Weight range from -1.0 to 1.0



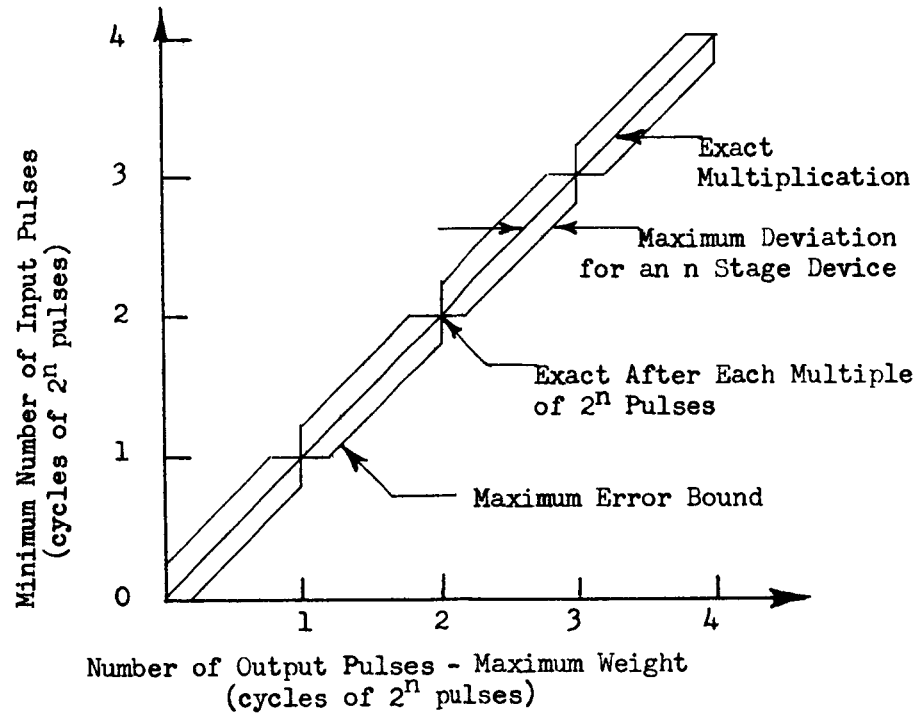


Figure 6 Error Characteristics for the

Binary Rate Multiplier

The percent deviation is independent of the number of input lines and is used as the figure of merit for the device.

From the results of table 1, the error is within acceptable limits for most application with a reasonable expenditure of time and hardware. We further mention that these are worst case considerations and actual results could be expected to be far better.

F. Speed of Operation

Throughout the development of the design of this pattern recognition device, it is obvious that there exists a strong interdependence between speed and equipment requirements. As in any engineering design, one must be traded for the other.

Some mention must be made, however, as to the limiting factors where operating and adapting speeds are concerned. For a given accuracy requirement, we need a sufficiently high number of pulses on which to base our decision. There are two ways to obtain more pulses. First, we can increase the decision time by using more cycles of 2^n pulses. Second, we can increase the operating frequency of the input transducers. In the second case, there is a maximum rate at which we can operate the summation counter, that must be considered. The maximum allowable input frequency is directly proportional to the maximum summation counter operating rate and inversely proportional to the number of inputs.

It should be observed, that while there are practical limitations to both speed and accuracy, there are no fundamental or theoretical limitations. Any speed and degree of accuracy desired can be obtained, provided that the equipment is available.

PART (III.)

EQUIPMENT DESIGN

In the preceding sections we have confined our attention to the theoretical aspects of the design of the Frequency Coded Threshold Logic Unit. In this section we will give consideration to the equipment requirements for the implementation of the device. Although an actual hardware design is not attempted, the performance requirements of each piece of equipment are discussed in detail, and where appropriate, a method of implementation is suggested and indicated by the dashed figures in the equipment diagrams.

A. Input Transducer

The basic structure of the input transducer is drawn in figure 7. The input voltage is assumed to range from $-v_{max}$ to $+ v_{max}$. The input transducer is required to produce a pulse frequency proportional to the absolute value of v_i. In addition the sign of the input must also be transmitted. Voltages and frequencies used will depend on the type of equipment employed. The only requirements are that $v_i = 0$ correspond to f_{min}, and $v_i = \pm v_{max}$ correspond to f_{max} and that the voltage to the frequency conversion be linear. The deviation from linearity should be limited to less that .5 percent, since this is the major source of error in the Frequency Coded Threshold Logic Unit. The suggested implementation of the frequency generator is to use a voltage controlled astable multivibrator. The reference voltage V_{th} is included to provide a method of obtaining the desired input frequency range. The sign bit is recorded by a threshold detector which makes a transition when the input voltage passes through zero. The absolute value function can be implemented with

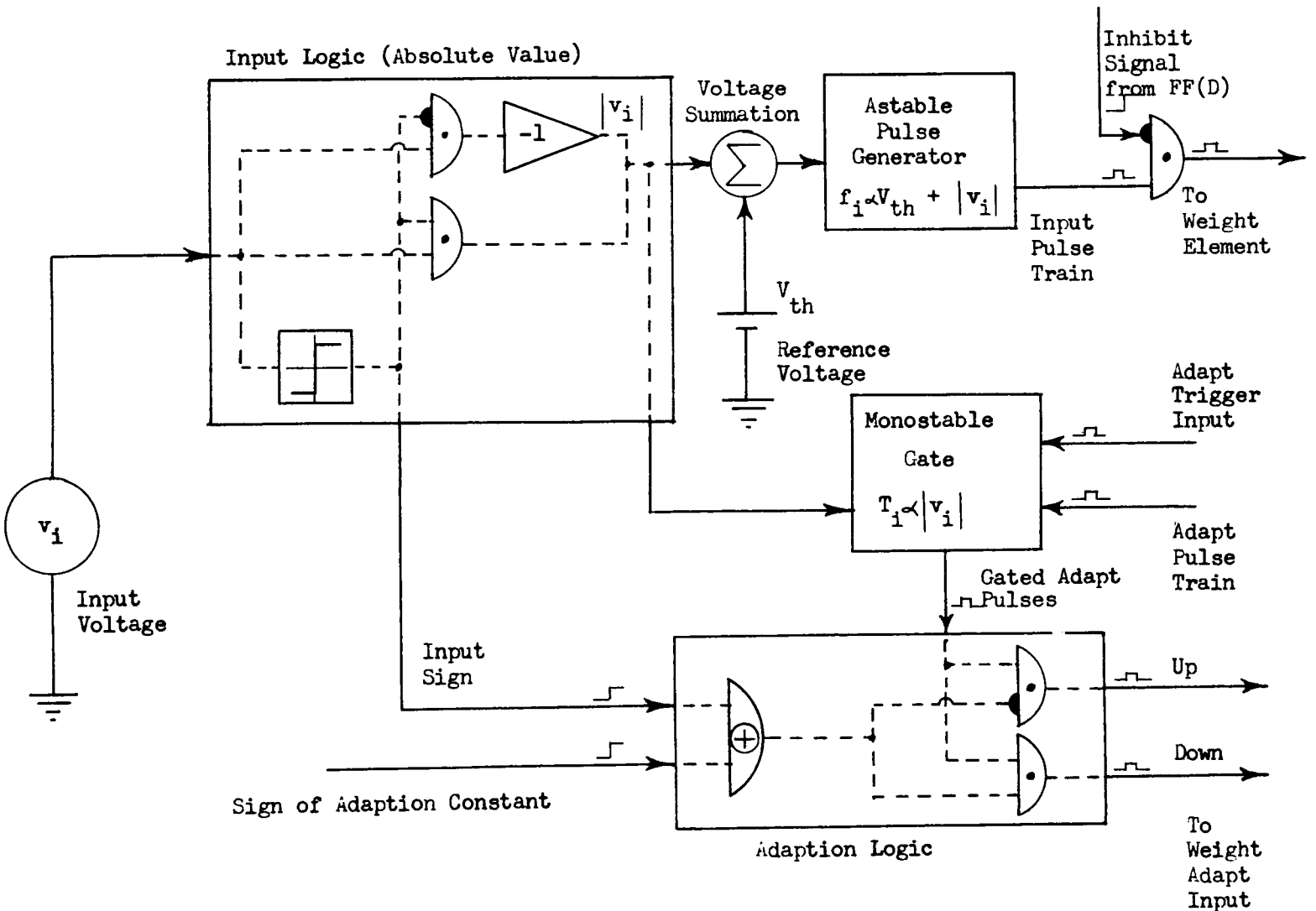


Figure 7 Input Equipment Diagram for a Single Input

an invertion amplifier and the logic shown in figure 7.

The pulse duration here, and throughout the system, need only be sufficient to trigger the next stage.

B. Weighting Element

The implementation of the weighting element is a modified form of the Binary Rate Multiplier discussed earlier. The structure of the proposed weighting element is shown in figure 8. There are several functions which must be performed by the Binary Rate Multiplier. It must scale the incoming pulse train, compute the sign of the weighting input and finally, must be capable of weight adaptation. The choice of hardware will depend on the speed and accuracy requirements of the particular application. Only the basic structure and operating principles will be given here. For generality we will assume a weight quantization of n bits throughout the discussion.

The input pulse train enters an n stage binary counter. The counter is shown as a series of cascaded flip flops each operated as a trigger. Each input pulse to a particular stage causes a transition of the flip flop and produces an output pulse. The output pulses are of two types depending on the transition. As the flip flop makes a transition from the 1 to the 0 state, a carry pulse is generated and acts as the input pulse to the next stage. If the transition is made from the 0 to the 1 state, the pulse is channeled to the scaling gate for that respective stage. The operating speed and ripple time requirements for the counter will be determined by the maximum input frequency to which the counter is required to respond. A reset provision is used to initalize the counter to zero during the training phase, as will be discussed in the training procedure.

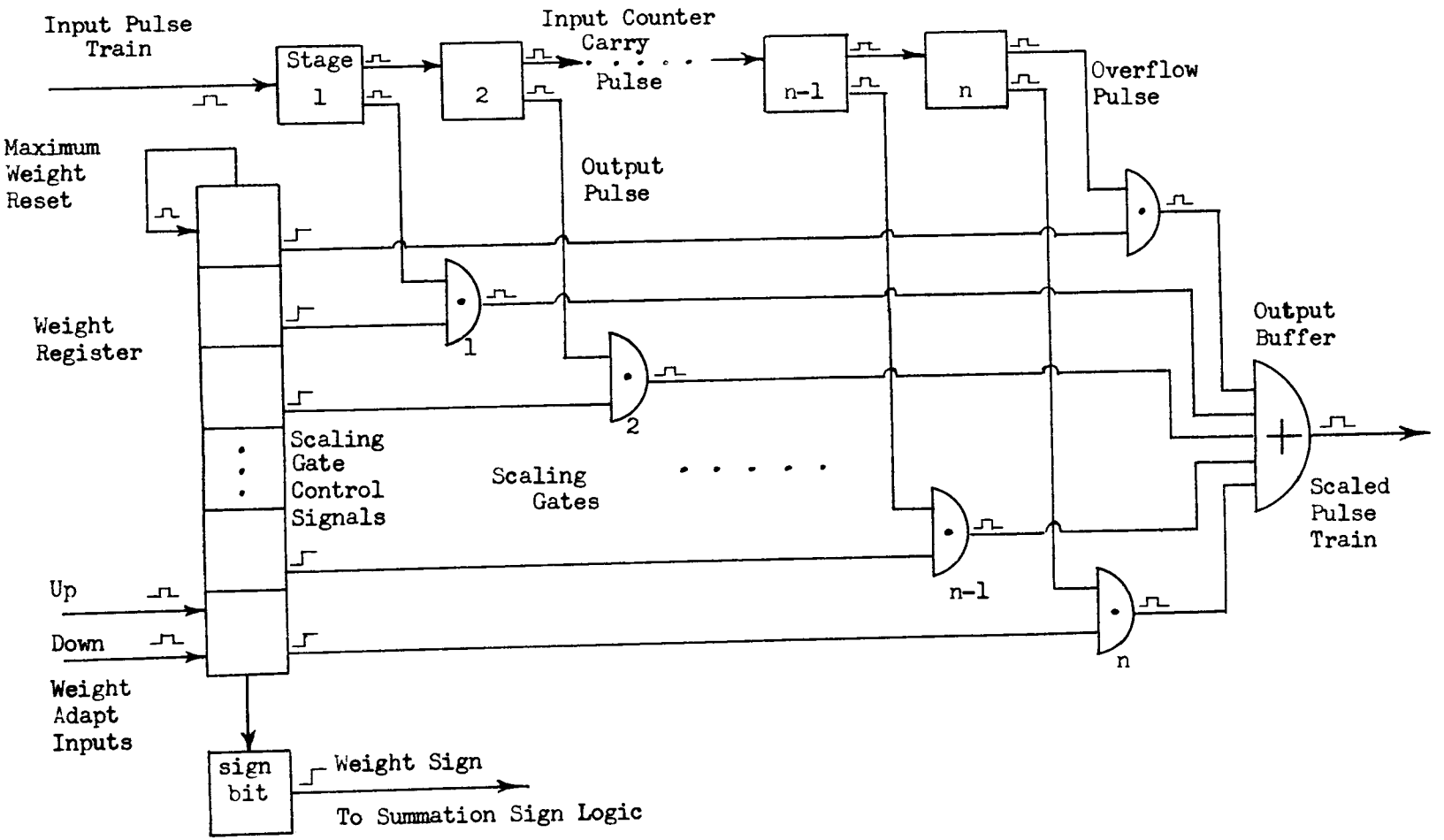


Figure 8 Weight Element Equipment Diagram for a Single Input

The scaling gates are simple AND gates controlled by the state of the respective stage of the weight register. They are used to transmit or block the pulse flow to the output buffer, as previously discussed. The n+1st gate is provided to transmit the overflow pulse from the nth stage of the counter to the output buffer. This allows the weight to assume a value of one, but need not be included if the maximum weight is restricted to $(2^n-1)/2^n$, rather than one.

The output buffer is shown as an OR gate and is simply a way of channeling pulses through the scaling gates to a single output line. There need be no provision for simultaneous occurrence of pulses on two or more lines, since these conditions cannot occur.

Weight storage and modification in the Frequency Coded Threshold Logic Unit is accomplished using an n stage binary register or n+1 if the maximum weight = 1. The control signals for the scaling gates are simply the states of each weight register stage. For example, a condition of 1 on the first stage of a weight register, corresponding to a weight of $1/2^n$, allows pulses to be gated from the nth stage of the input counter. This effectively scales the pulse train by a factor of $1/2^n$ since only one pulse is transmitted from the nth scaling gate for each 2^n pulses presented to the input counter.

Adaptation is accomplished by operating the weight register as a counter, capable of counting up or down. Speed at which the weight can be adapted is limited only by the speed at which the counter can be driven. There are several additional provisions which must be included in the design of the weight register. First, a reset should be provided so that the weights can be initialized to zero before training begins. Second, there must be a sign bit included as part of the weight. The sign bit is

stored in a flip flop which is set (1 state). When the counter passes through zero while operating in the up mode, and reset (0 state) when the counter passes through zero while operating in the down mode. Finally, provision must be made for the possibility of an adapt pulse on the up mode of a weight register that is already at its maximum value (a 1 on all stages). This pulse would tend to overflow the counter and return it to a zero condition. The overflow pulse should be used to reset the register to the maximum condition.

C. Summation Element

The purpose of the summation element is to accept the weighted pulse train from each input line and compute the value of the discriminant function. Since the discriminant function is in the form of a pulse count, the summation process is one of obtaining this pulse count. Figure 9 shows the implementation of the summation process.

We have allowed positive and negative values for both inputs and weights. Thus we must consider both signs to determine whether pulses for an input line are to be added or subtracted. This is done by the logic shown in the sign determining block. It entails determining the sign of the product $w_i x_i$, where w_i is the weight and x_i is the value of the ith input. Then the pulse train is submitted into the up (+ sign) or into the down (- sign) summation counter input buffers.

Both buffers may receive pulses from each input line, although no input may feed both buffers simultaneously. However, there is the problem of the simultaneous occurrence of pulses from two or more lines at a buffer. In order that all pulses may be counted, the input lines must be synchronized so that only one pulse is presented to a buffer at a time. The method and equipment used to synchronize the input lines is

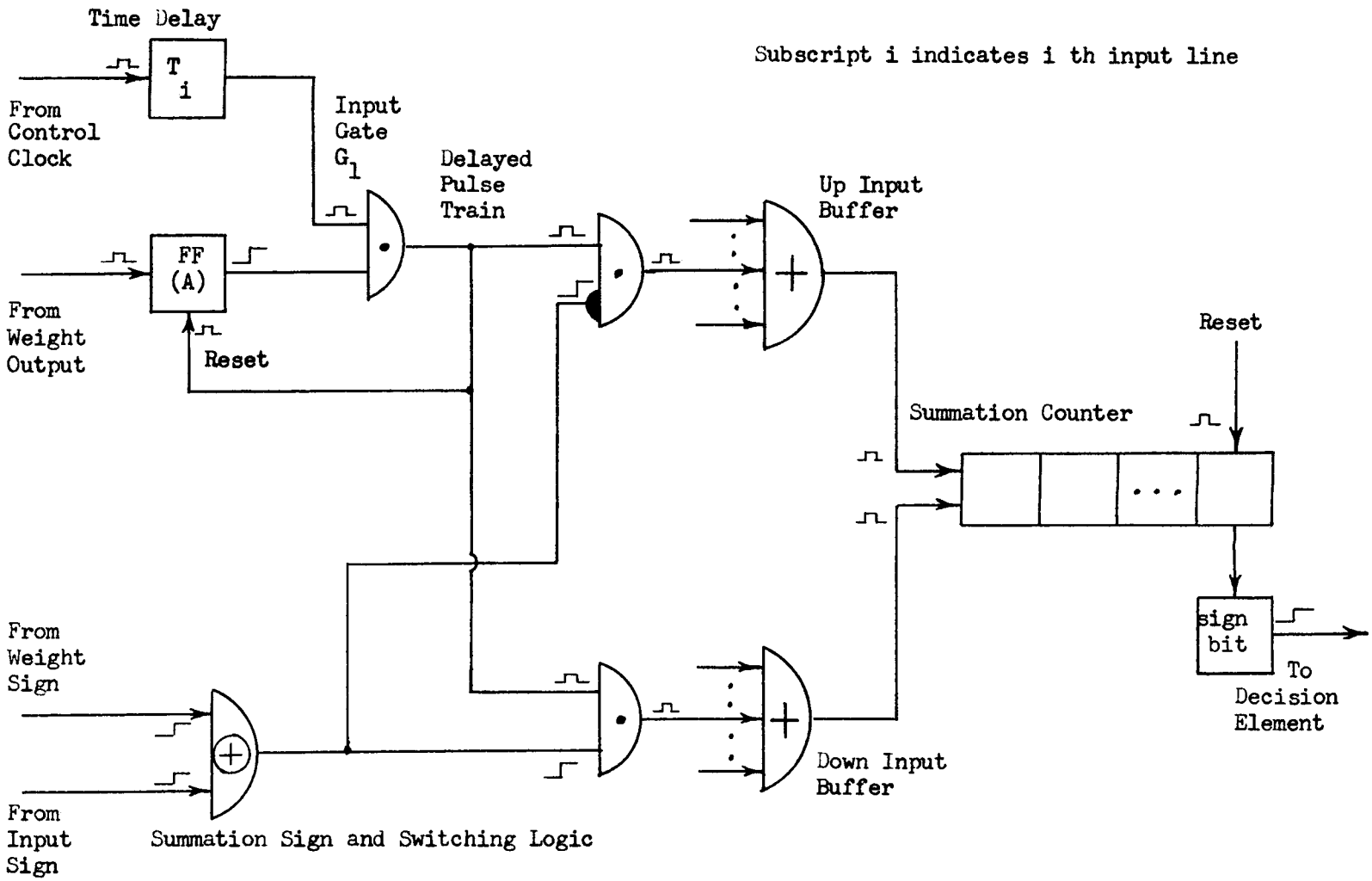


Figure 9 Summation Element Equipment Diagram

now described.

A pulse arrives at the output buffer of a Binary Rate Multiplier causes flip flop (A) to be set (1 state). The appearance of a 1 on this flip flop in turn opens the input gate (G_1). The synchronization is accomplished by using a control clock which presents a single pulse simultaneous to each input line. It is this pulse that is gated through input gate (G_1) to advance the summation counter, rather than the actual pulse from the Binary Rate Multiplier. Sequencing the input lines so that the pulses from several lines do not overlap is done by delaying the control pulse by an amount T_i. Figure 10 shows the timing diagram for this process. The reset pulse shown on the diagram is the same pulse being sent to the counter and is used to reset flip flop (A) so that the next Binary Rate Multiplier output pulse can be counted in the same manner. To insure that all pulses can be counted, it is only necessary to sequence the control pulse and flip flop reset pulse over all input lines before the next input pulse arrives at the Binary Rate Multiplier. This concept is clearly shown on the timing diagram, figure 10.

Once the pulse is sequenced and switched to the proper buffer, it is presented to the summation counter. The counter must be of sufficient capacity (stages) to allow a maximum of pulses to be counted. The actual size should be sufficient to handle a number of pulses equal to the product of the maximum input frequency, decision time interval and number of input lines, assuming a maximum weight of one. The counter must be able to count up and down, but is not required to handle pulses on both lines simultaneously, since the synchronization procedure eliminates this possibility. A sign bit is used to allow the counter to count through zero and to register the sign of the total pulse count. The sign

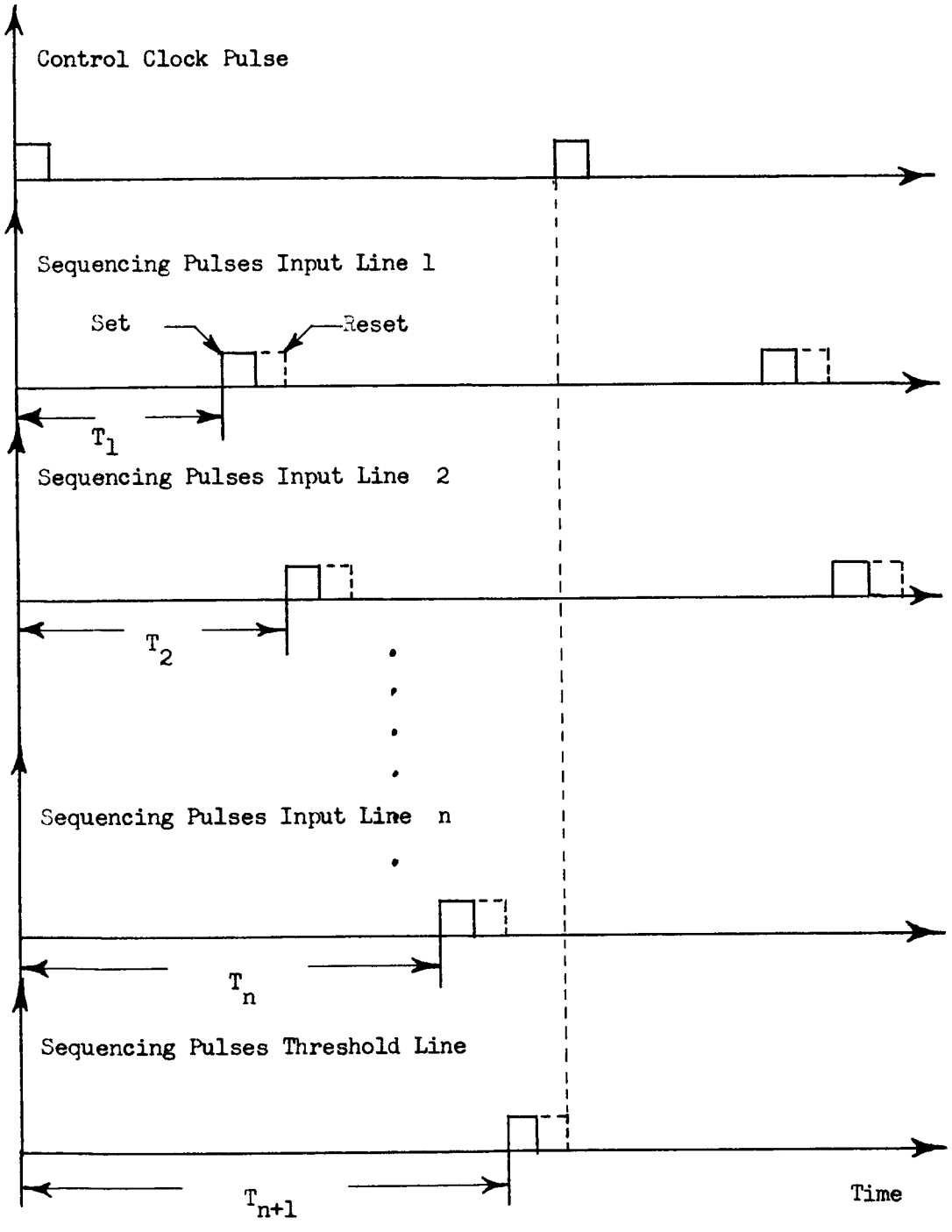


Figure 10 Timing Diagram for Summation Counter

bit is implemented in the same manner as the sign bits for each weight register. The final value of the discriminant function at the end of the decision period is the signed contents of the summation counter.

D. Decision Element

Before considering the implementation of the decision element, we must describe the method of handling the variable threshold. The method chosen for this design is to consider the variable threshold as an additional input weight. This allows using the same weight implementation and adaptation techniques previously developed. The input to the threshold weight is implemented by using a pulse generator of constant frequency. Perhaps, the most economical implementate would be to use the control clock as the input to the threshold weight. Since the control clock is essentially operating at the maximum allowable input frequency, this implementation corresponds to a threshold input of +1, on an input scale of -1 to +1. The implementation of the threshold weight is identical to the input weights shown in figure 7.

The implementation of the threshold in this manner, in addition to design simplification, has another important advantage. By considering the threshold as an input, we modified the discriminate function so that the threshold is zero. This allows the use of a simple sign detection procedure for the final decision. The result is that we can use the summation counter sign bit as the threshold detector. A classification is made only at the conclusion of decision time interval. This classification is then stored until the next decision time. Figure 11 shows the equipment required for the decision element.

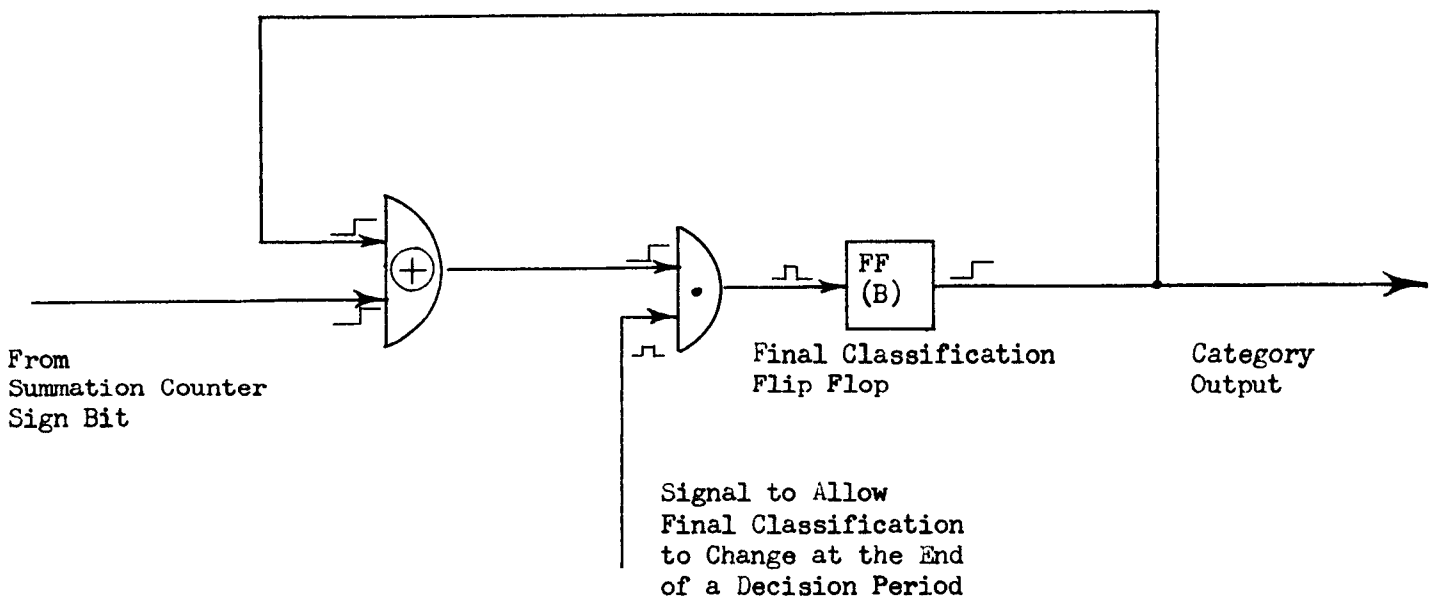


Figure 11 Decision Element Equipment Diagram

E. Control and Timing

The controls considered for the Frequency Control Threshold Logic Unit are of two types: those involved with operation in the classifying mode, and those involved with operation in the training mode. The control aspects of the design are best illustrated by the timing diagrams of figures 12 and 13. The implementation is shown in figure 14. The classification mode is designed for continuous operation with a classification at the conclusion of each decision period. The key to the entire timing control is the control clock. As was previously mentioned, the control pulse is used in the summation element and as the input to the threshold weight. The control pulse is also used to determine the decision period. This is done by channeling the control pulse into the decision interval timing circuit, which can be simply a preset counter that will produce an output pulse after the desired count has been reached. The preset count will depend on speed and accuracy requirements as previously discussed and is considered as the basic design parameter rather than the actual decision interval. The timing diagram of figure 12 shows several time delays that are critical to the correct operation of the device. The delay in the decision pulse T_a is the inherent delay in the decision timing circuit. It is shown to indicate the actual order of events. The reset pulse delay T_b is a delay purposely introduced to allow the final output classification flip flop (B) to change states before the summation and decision counters are reset. The pulse duration is shown to indicate that it must be long enough to permit the proper transition to occur. The summation counter is shown as being reset, counting slightly negative and then going positive, to illustrate the sign bit transition. The large time delay T_1 of the first input into the summation element is

purposely introduced in the summation synchronization scheme. This is done to allow the completion of the reset and classification procedures before any input pulses are allowed to enter the summation buffers.

The control aspects of the training mode are slightly more complex. Figure 13 shows the additional timing considerations that occur when weight adaptation is allowed. The training procedure used in this project has previously been described and only implementation will be discussed here. At the end of a decision period the decision pulse occurs as in the classification mode. In the training mode the decision pulse is also used to inhibit the input pulse generators and to prevent the starting of the decision timing circuit. The inputs are inhibited to allow the input counters to be reset while the training pattern is being used in the adaptation routine. The reset pulse to the input counter is delayed T_c, to allow the inputs to be inhibited before reset occurs.

Adaptation is accomplished by gating a specific number of pulses into the adapt line of each weight register. The adaptation delay T_d is introduced to prevent the modification of weights before the completion of the classification process. Adapt pulses are produced by a clock, whose frequency will determine the speed with which adaptation occurs. The control clock could be used for this purpose. The magnitude of each weight adaptation is governed by the number of adapt pulses that are gated into the weight register. The fixed increment training procedure requires that the adaptation be proportional to the magnitude of that particular input. This is accomplished by using a monostable gate whose pulse duration is proportional to the absolute value of the input voltage. The constant used in the fixed increment adaptation scheme will determine the maximum gate duration required for a specific adapt pulse frequency.

Threshold adaptation is accomplished in the same manner with one exception. This is that the gate duration is a constant determined by a fixed reference voltage. Note that the training pattern is required to be a constant and to be presented over the entire classification and adaptation periods. At the completion of the adaptation period T_e, which will be determined by the maximum value of the gate duration, a pulse is sent to signal the "teacher" that adaptation has been completed.

When the "teacher" has presented a new training pattern, a pulse is generated by the "teacher" which will remove the inhibit signal from the control clock. The first control pulse then removes the inhibit signal from the input lines, thus beginning the next classification period. In the event a training pattern is classified correctly, no adaptation is performed and the signal for a new pattern to be presented occurs after a time delay T_f. The delay is introduced to allow reset and initialization procedure to be completed.

The sign of the adaptation is determined by the logic operations shown. First, we determine the sign of c, the fixed adaptation increment, from the correct category previously introduced by the "teacher". This sign is then compared with the sign of the pattern input. Adapt pulses are then channeled into the proper weight adapt input line.

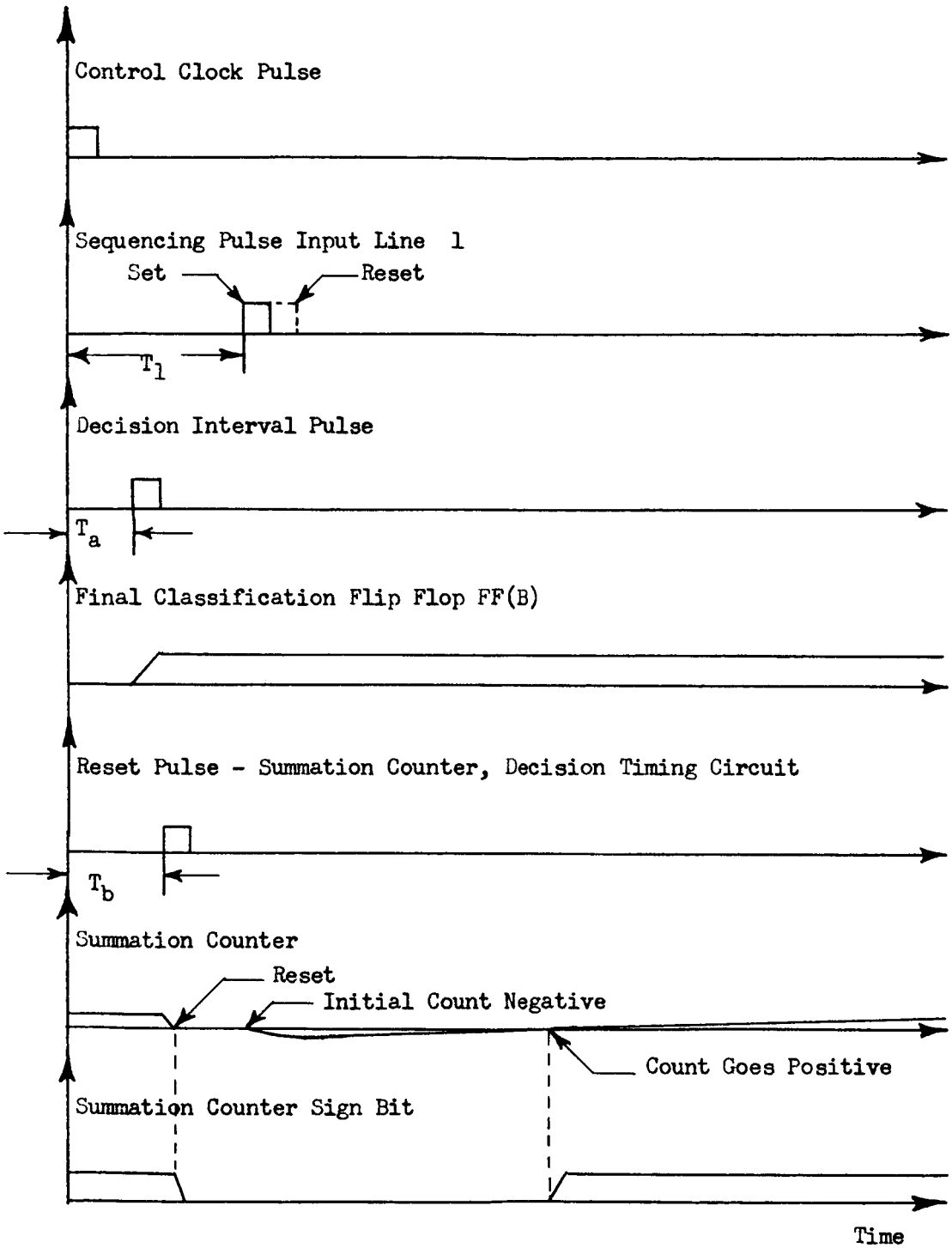


Figure 12 Timing Diagram for Operation in the Classification Mode

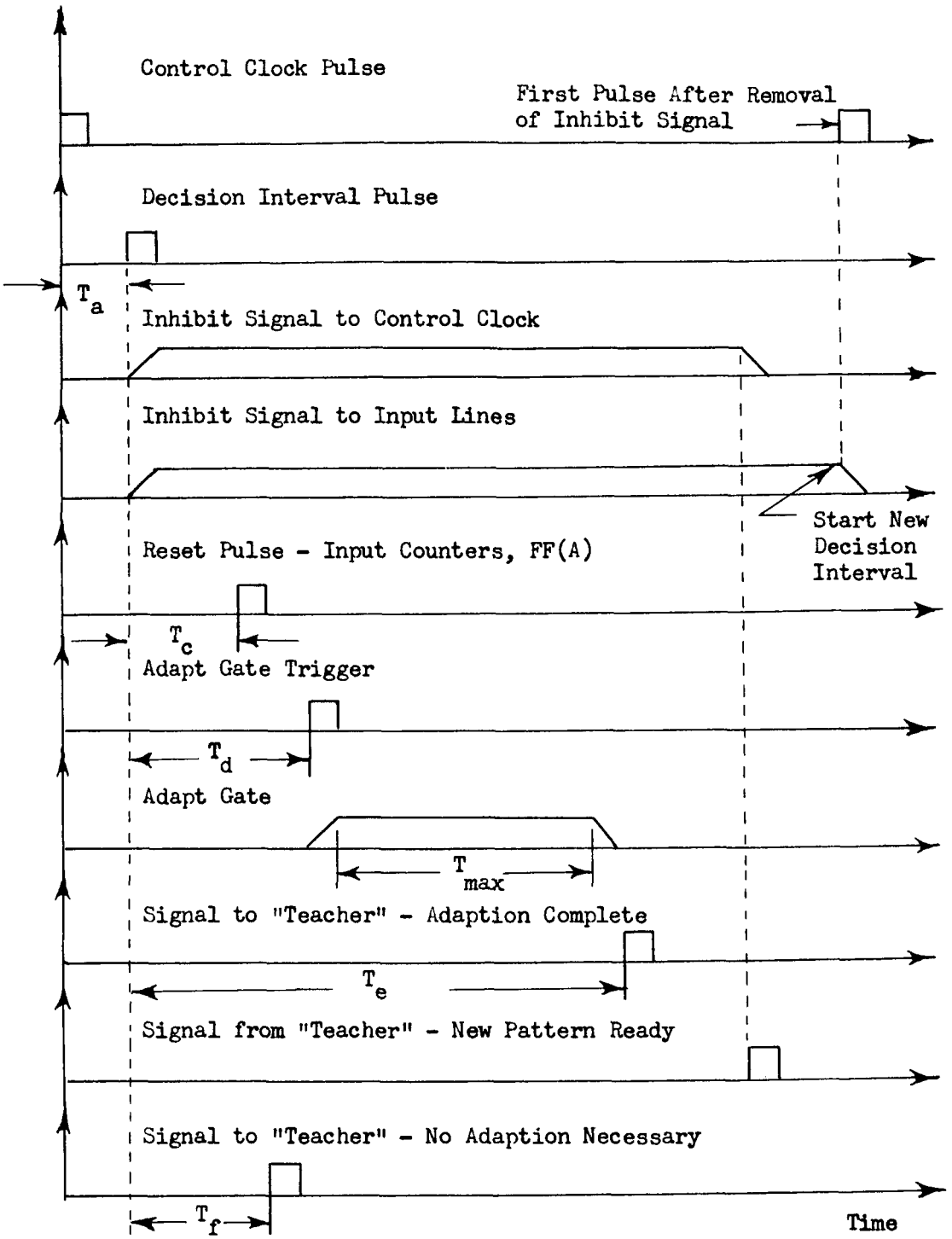


Figure 13 Timing Diagram for Operation in the Training Mode

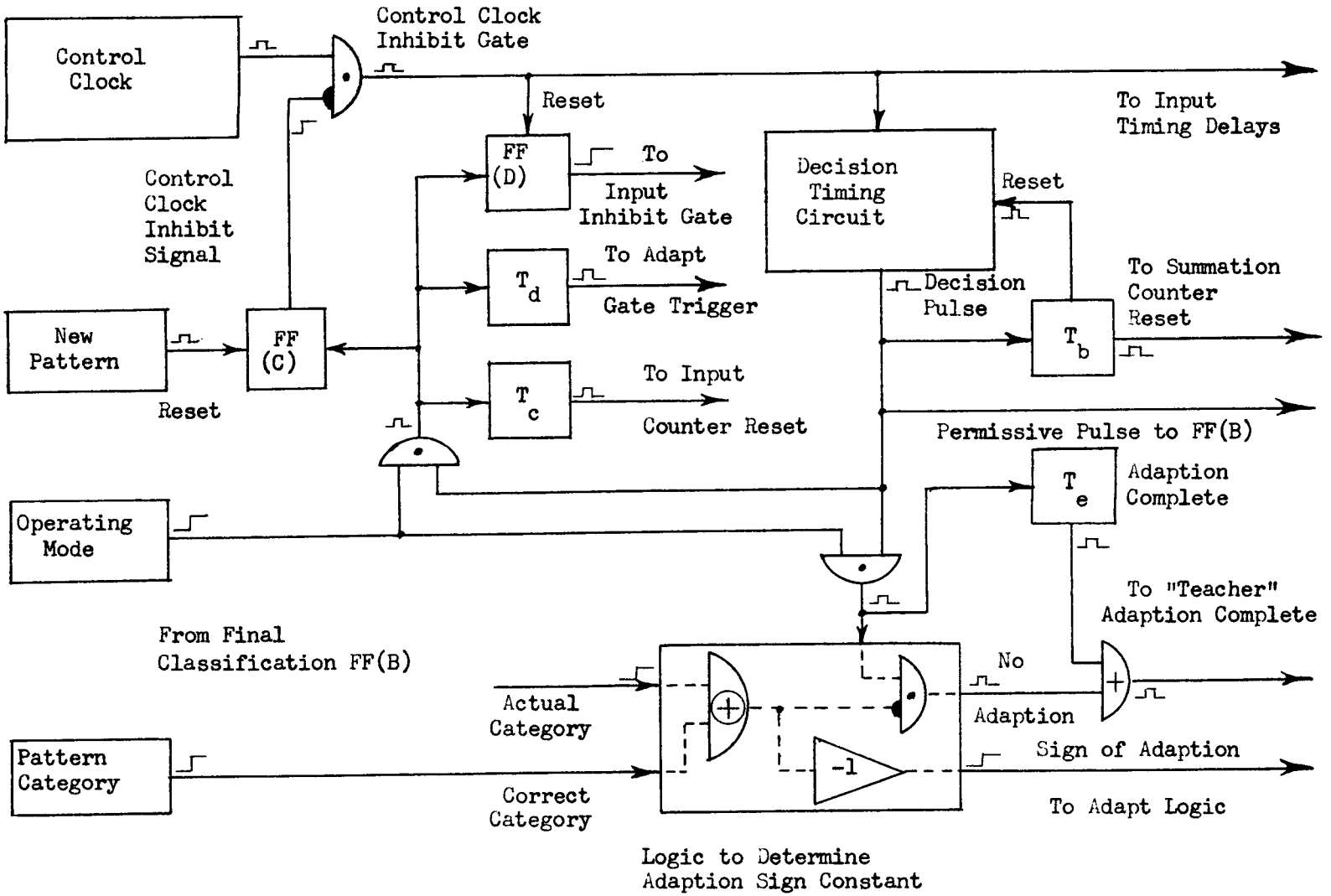


Figure 14 Control Equipment Diagram

PART (IV.)

APPLICATIONS

A. General Discussion

The design presented in this project is concerned with the simplest trainable pattern recognition system known, the Threshold Logic Unit. It is only capable of handling two category problems in which a linear discriminant function can be used. There is no restriction on the size of the problem, since any number of inputs can be considered with no loss of accuracy, and only a slight decrease in speed. Extension of the concepts presented here can be made to the multicategory classification problem. The two category decision is made using only the sign bit of the summation counter. The contents of the counter are a measurement of the distance from the pattern point to the decision hyperplane. For multicategory classification we need only compare contents of the summation counter with a set of threshold values. Implementation of each threshold weight could be accomplished by storing it as the contents of an adaptable register, rather than as a pulse stream scaling device. The final decision would entail comparing the contents of the summation counter with those of each threshold weight, either by digital computation or by digital to analog conversion and analog comparison.

The extension of the Frequency Coded Threshold Logic Unit to nonlinear classification problems is also possible. Generation of a Φ machine requires the analog multiplication of the input voltages to obtain the desired products, which are then used as the input voltages to the device. A piecewise linear machine can be constructed by the proper interconnection of elementary Threshold Logic Units.

The development of this extension was not attempted in this

project. The major concern was the development of the frequency coding principle and its application as a method of weight implementation. The design concepts presented here have several advantages over presently existing methods. The major advantage is the accuracy and speed with which the weights can be adapted. The use of high speed digital equipment in place of an analog memory device is the key to this design. In addition to adaptation accuracy, the weight implementation, discriminant function computation and the decision process are all done with sufficient precision for most applications.

The use of digital equipment has added advantages. It is highly reliable, capable of miniaturization and is readily available for engineering application. The final choice of equipment in any pattern recognition problem will depend on the specific requirements of the intended application. It is felt that the frequency coding principle has advantages which warrant its consideration as a suitable solution to many pattern recognition problems.

B. Computer Simulation

The operation of the Frequency Coded Threshold Logic Unit was simulated on a digital computer. The simmulation was then applied to a single two character recognition problem. The continual aspect of the operation and the two category aspect of the design, makes it more applicable to a quality control problem than a character recognition problem. However, the simulation serves to illustrate the method by which the frequency principle can be applied to a wide variety of problems.

The continuous nature of the input variation is used in the recognition of the characters A and P. We use a 3 by 3 grid to determine our input pattern. The percentage of each grid sector that is blacked

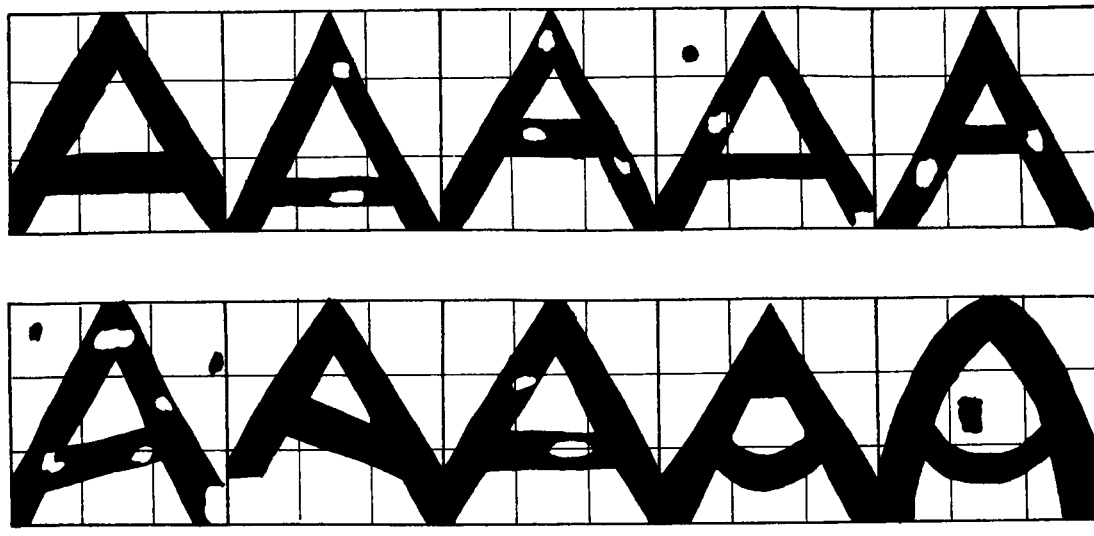
out is considered as the input feature. The scale is centered about a 50 percent blacking of the grid. Thus, no marks in a grid sector will correspond to a maximum negative input, 50 percent to zero volts input and total blackout will correspond to a maximum positive voltage. The training set used is indicated in figure 15.

The simulation program is written to generate the same errors that would be encountered in the operation of the actual device. Flow charts are given in the Appendix. Provisions are made for the variation of design and training parameters to allow a certain degree of versatility in simulation. The input patterns are presented to a program that converts them to the desired frequencies. The program operates on the period rather than the frequency, since it is the time between pulses that carries the information.

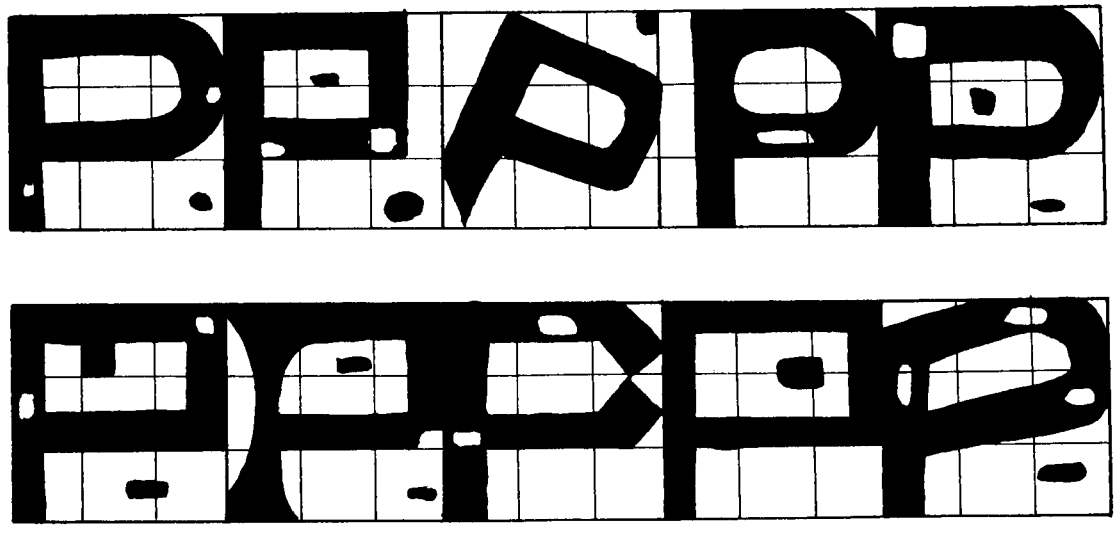
The results of the character recognition problem are given in table 2. The results are not given to indicate the advantages of a particular method of character recognition. They serve only to illustrate the application of the frequency coding principle. The significant results are the speed with which the patterns would be classified in the actual device, the accuracy of the weights, and the rate at which adaptation would occur. These parameters are given in table 2 for the particular design indicated in the table.

Table II Results of the Sample Character Recognition Problem

System Design Parameters	
Number of Inputs	9
Number of Binary Stages	7
Minimum Input Frequency	40000 cps
Maximum Input Frequency	50000 cps
Threshold Frequency	45000 cps
Number of Control Pulses / Decision Interval	480
Fixed Increment Correction Constant	5 pulses
Relative Value of the Threshold Adaptation	.20
Number of Iterations of the Training Set	20
Number of Patterns Used in the Training Set	20
Number of Patterns Used in the Trial Set	30
Recognition Results for the Trial Set	
Total Percent Recognition	78 percent
Percent Recognition of the A	60 percent
Percent Recognition of the P	96 percent
Timing and Speed Considerations	
Decision Interval	.0096 sec.
Maximum Time Required for one Complete Adaptation 5 Bits at a Frequency of 50000 cps	.0001 sec.
Total Time Required for the Training Procedure 60 Adaptations	3.84 sec.
Time Required for the Trial Classifications	.288 sec.



Category 1



Category 2

Figure 15 Training Patterns Used in the Sample Character Recognition Problem

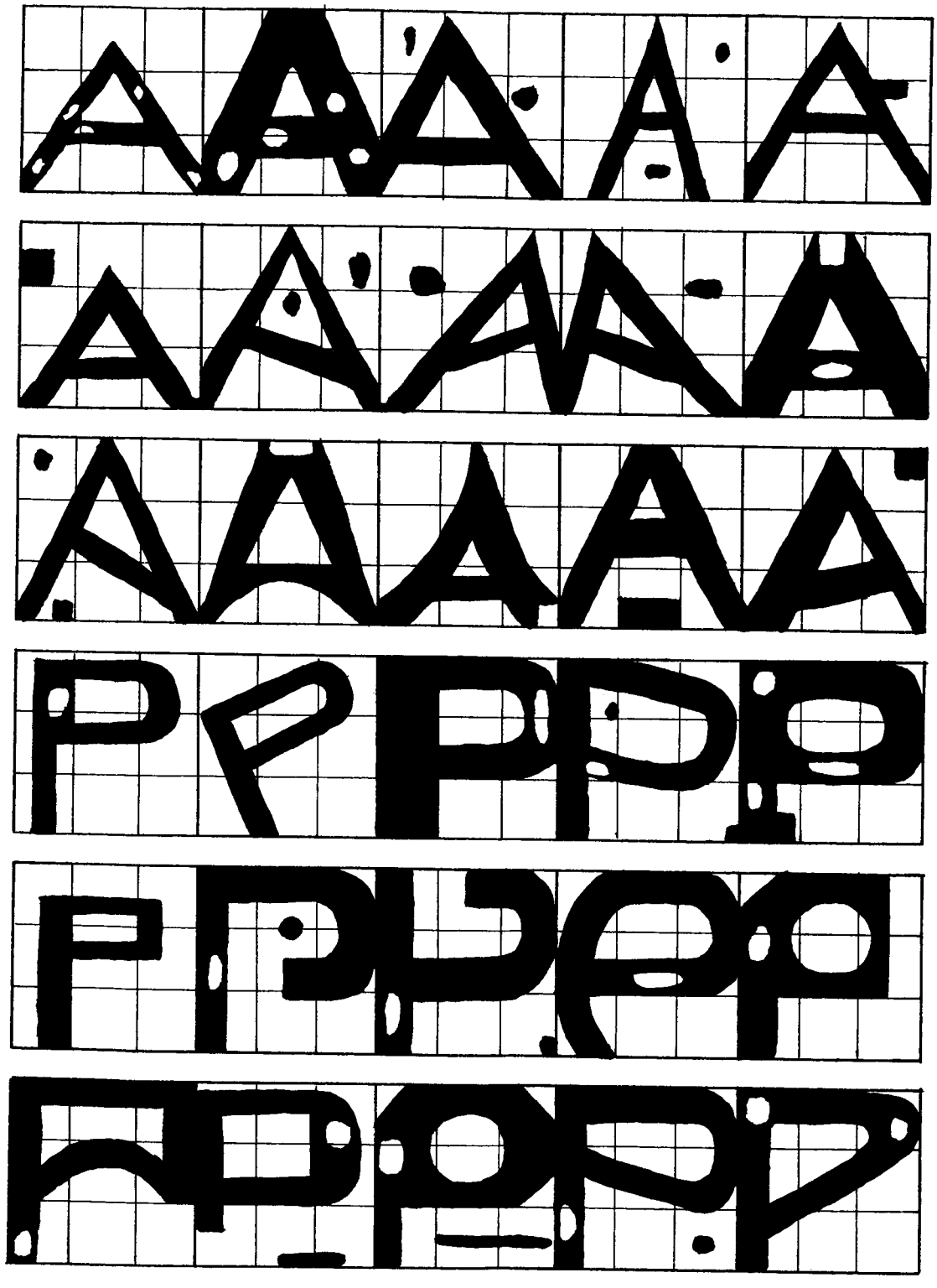


Figure 16 Patterns Used to Test the Recognition Ability of the Sample Design After Training was Complete

PART (V.)

SUMMARY

The motivation for this project was the need for a fast, accurate and reliable method of weight adaptation to be used in trainable pattern recognition systems. The sensory nervous system was investigated as being a possible model for the proposed device. The speed with which complex patterns are recognized and the binary nature of sensory system indicated that the model would be worthy of further consideration.

Modern digital techniques were applied in an attempt to incorporate speed and reliability into the design. It was in the application of the digital techniques that the frequency coding of information, found in the sensory nervous system, became important. Coding the input patterns as a pulse frequency, allows the use of binary equipment to instantaneously transmit an analog signal without quantization error or coding time delay. The concepts were then developed in the form of a Frequency Coded Threshold Logic Element.

The device was shown; to be applicable to the current pattern recognition systems and training procedures, to possess desirable speed and accuracy characteristics for both classification and adaptation, and to require only a moderate equipment expenditure. As such, the design concepts should prove useful in the construction of practical pattern recognition systems.

PART (VI.)

LITERATURE CITED

1. Arnstein, W., Mergler, H.W., and Singer, B.: "Digital Linear Interpolation and the Binary Rate Multiplier," Control Engineering, vol. 11, No. 6, pp. 79-83, June, 1964.

2. Braun, Edward L.: "Design Features of Current Digital Differential Analyzers," IRE Convention Record, part 4, pp. 87-97, 1954.

3. Gordon, Bernard M.: "Adapting Digital Techniques for Automatic Controls," Electrical Manufacturing, vol. 54, No. 5, pp. 136-143, November, 1954.

4. Hawkins, J.: "Self-organizing Systems: A Review and Commentary," Proceedings of the IRE, vol. 49, No. 1, pp. 31-48, January, 1961

5. Hubel, David H.: "The Visual Cortex of the Brain," Science American, vol. 209, pp. 54-62, November, 1963.

6. Mayorov, F.V.: Electronic Digital Integrating Computers: Digital Differential Analyzers, American Elsevier Publishing Co.,Inc., New York, 1964.

7. Meyer, M.A.: "Digital Techniques in Analog Systems, "IRE Transactions on Electronic Computers, vol. EC-3, pp. 23-29, June, 1954.

8. Moshos, George J.: "Design of Real Time Computers Utilizing Counting Techniques," NASA Technical Note TN D -3042, Washington, D.C., October, 1965.

9. Moshos, George J.: "Error Analysis of Binary Rate Multiplier," NASA Technical Note TN D -3124, Washington, D.C., December 1965.

10. Nilsson, Nils J.: Learning Machines: Foundation of Trainable Pattern-Classifying System, McGraw-Hill, New York, 1965.

11. Rosenblatt, Frank: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, Washington, D.C., 1962.

12. Rushton, W.A.: "Peripheral Coding in the Nervous System," Sensory Communication, Rosenblith, W.A., Ed., Wiley and Sons, New York, 1961.

13. Tou, J.T., Wilcox, R.H.: Computer and Information Science Symposium - Northwestern University, 1963, Spartan Books, Washington, D.C. 1964.

14. Widrow, B. and Hoff, M.E.: "Adaptive Switching Circuits," Stanford Electronics Laboratories Technical Report 1553-1, Stanford University, Stanford, California, June 30, 1960.

PART (VII.)

SELECTED REFERENCES

1. _____,: "Adaptive Memory Research," Technical Document, Report No. RADC-TDR-63-371, November, 1963, Information and Processing Lab., Rome Air Development Center, Griffin Air Force Base, N.Y..

2. Adrian, E.D.: The Mechanism of Nervous Action, University of Pennsylvania Press, Philadelphia, 1932.

3. Crafts, H.S.: "A Magnetic Variable Gain Component for Adaptive Networks," Technical Report No. 1851-2 Stanford Electronics Lab., Stanford University, Stanford, Calif;, December, 1962.

4. Deutsch, Sid;: "Conjectures on Mammalian Neuron Networks for Visual Pattern Recognition," IEEE Transactions on Systems Science and Cybernetics, vol. SSC-2, No. 2, December, 1966.

5. Enslein, Kurt, Ed.: Data Acquisition and Processing in Biology and Medicine; Proceedings of the 1961 Rochester Conference, Pargamon Press Inc., New York, 1962.

PART (VIII.)

APPENDIX

Flow Charts for the Computer Simulation of the Frequency Coded Threshold Logic Unit

Clarification of Symbols

Input Parameters

Design

F_{min}	Minimum input frequency
F_{max}	Maximum input frequency
N	Number of binary stages
I	Number of input lines
NCI	Number of control pulses per decision interval
THI	Relative value of the threshold input

Training

C	Fixed increment correction
THC	Relative value of the threshold adaptation
MAX	Maximum allowable iterations of the training set

Patterns

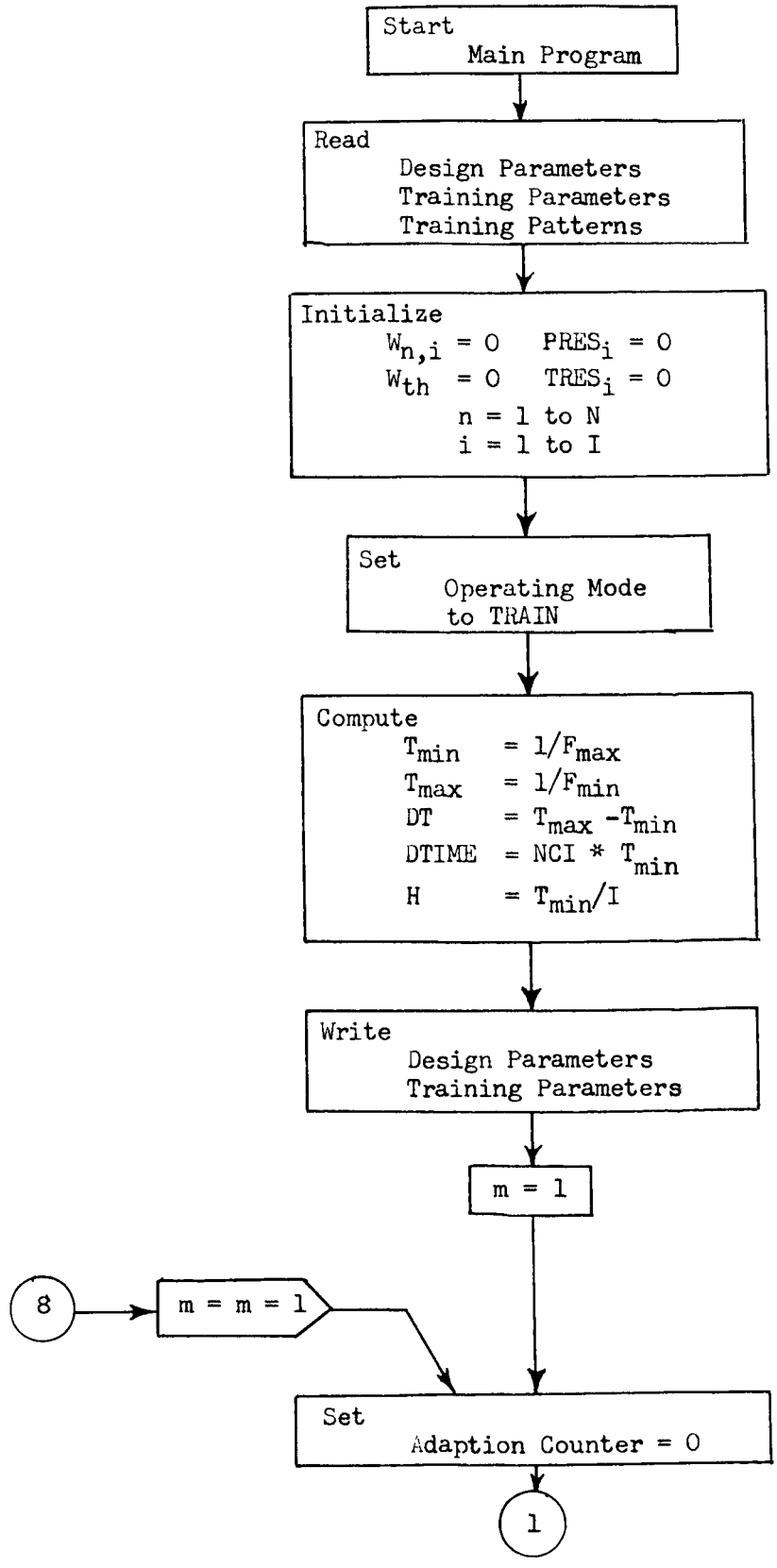
$TSET_{i,j}$	Training patterns	Index i over the input lines
$CSET_{i,j}$	Trial patterns	
J	Number of patterns	j over the patterns

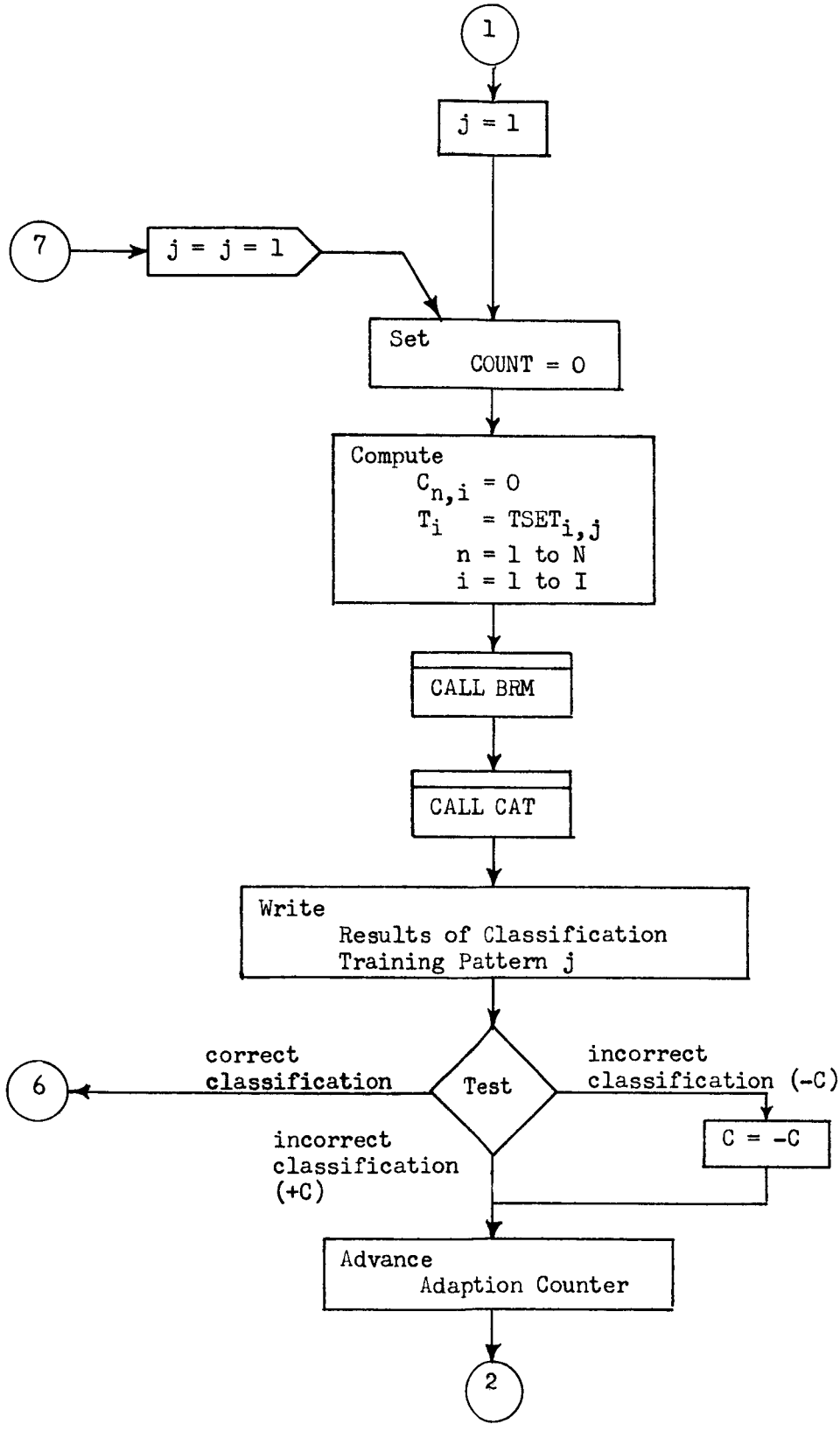
Subroutines

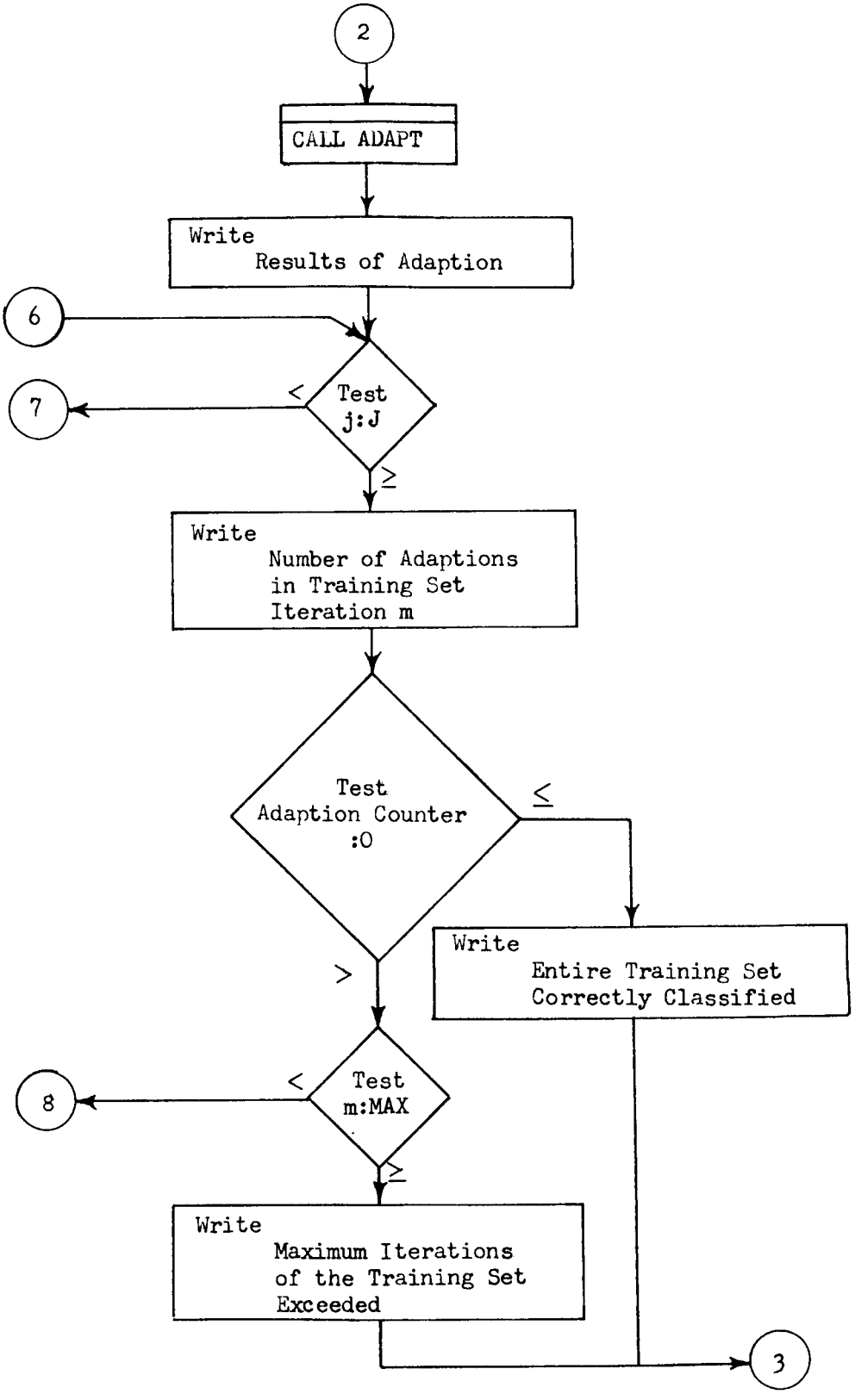
BRM	Simulation of the Binary Rate Multiplier
CAT	Simulation of the Categorization routine
ADAPT	Simulation of the adaptation routine

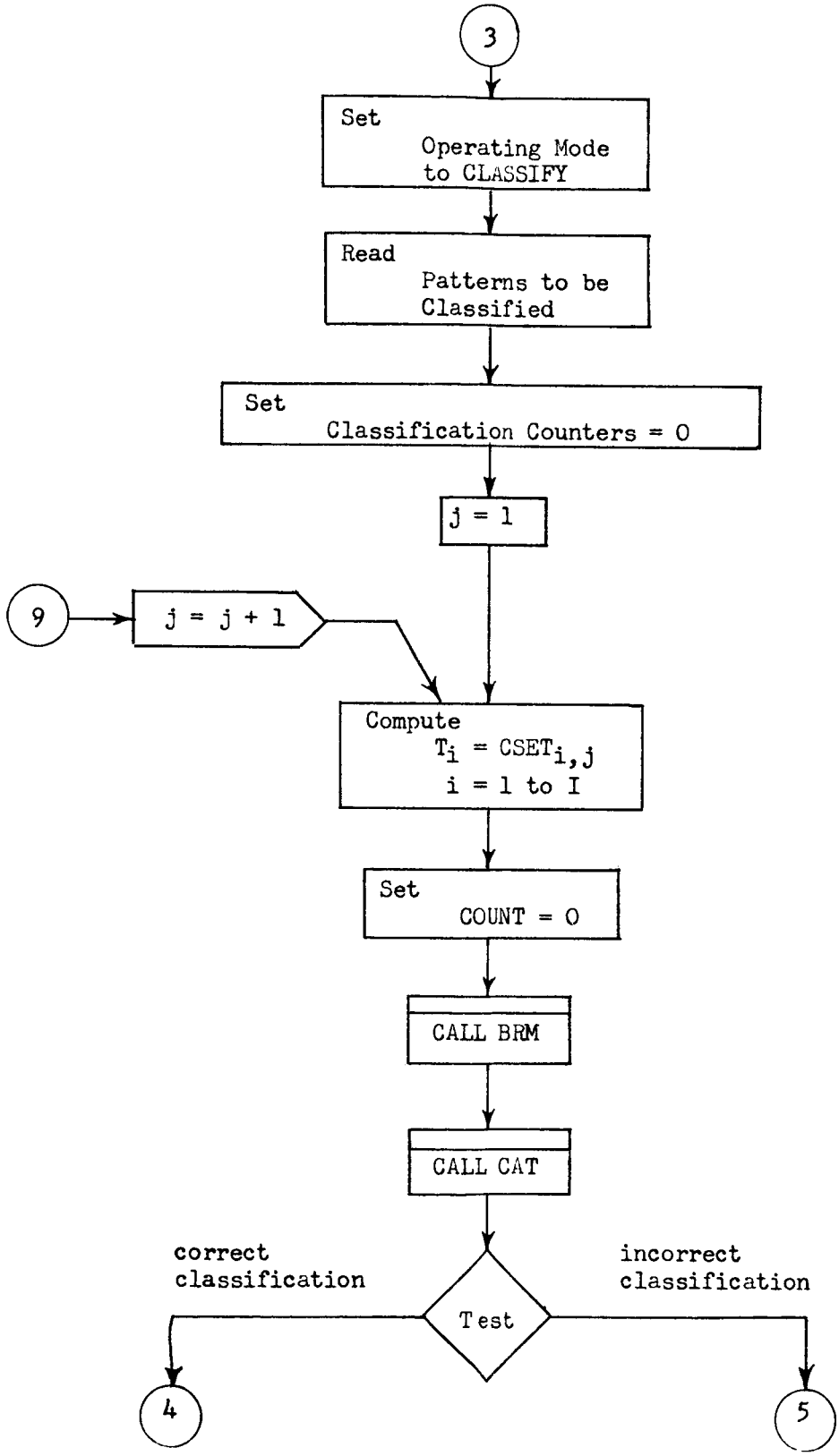
Symbols used in the Simulation

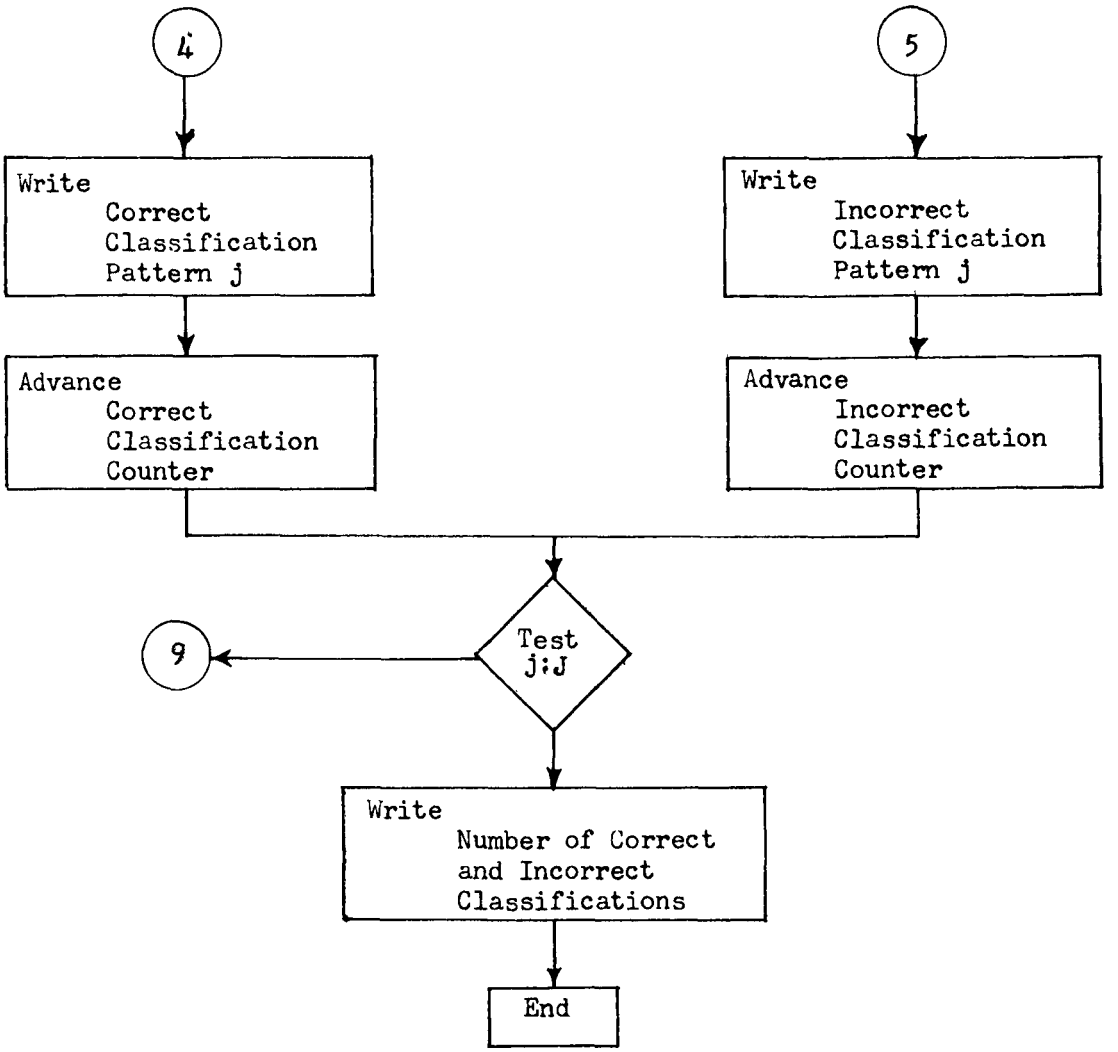
$W_{i,j}$	Weight register	Index i over the register stages
W_{th}	Threshold weight	
$W_{sign,j}$	Sign bit for associated weight register	j over the input lines
$C_{i,j}$	Input counter	
$PRES_j$	Condition of flip flop (A) at the start of a decision interval	
$TRES_j$	Time of arrival of the first input pulse after the start of a decision interval	
COUNT	Contents of the summation counter	
DTIME	Decision interval	
T_{max}	Maximum input period	
T_{min}	Minimum input period	
T_i	Period of the i th input line	
NPULSE	Number of pulses presented to the respective input counter over a decision interval	

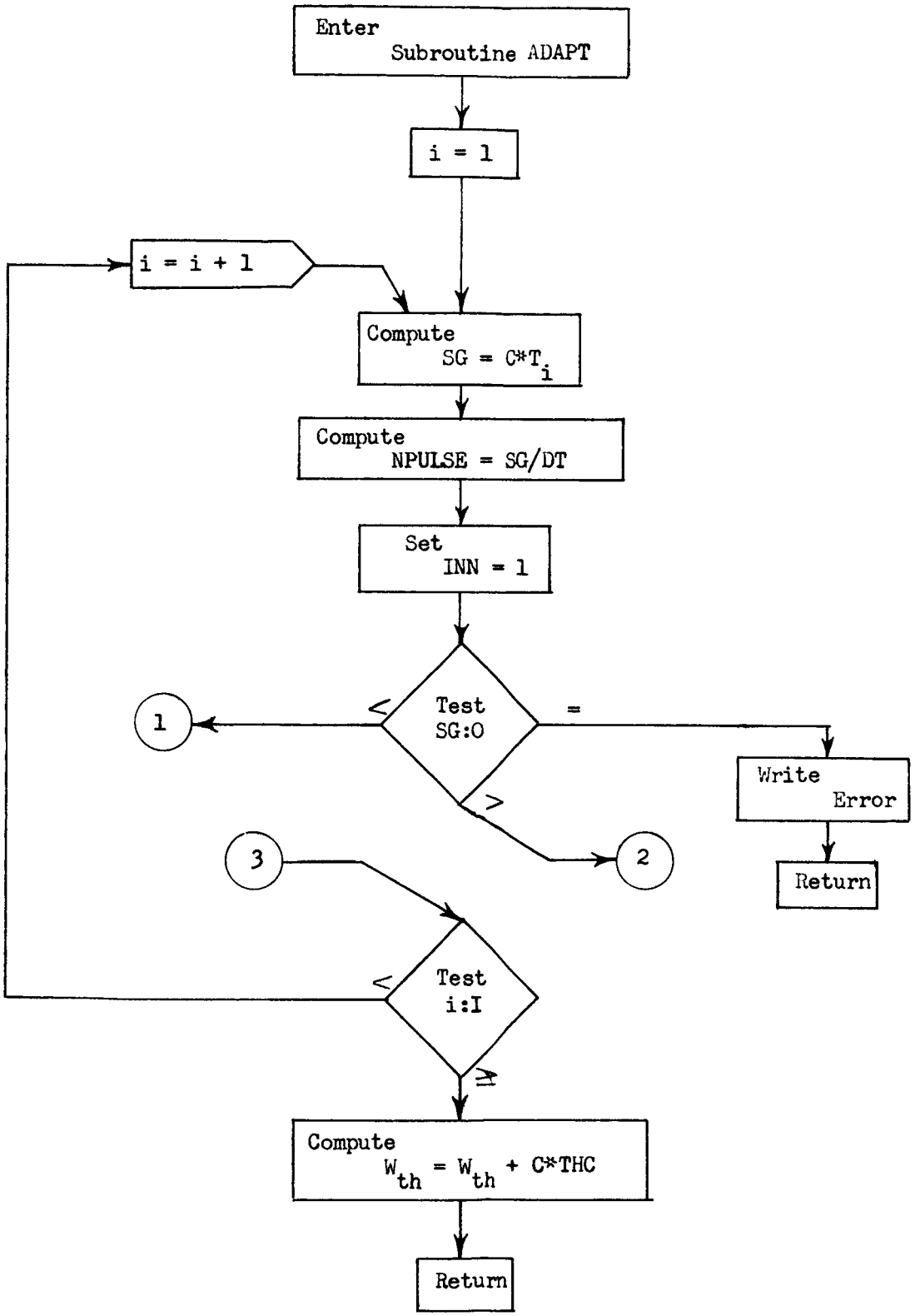

Start
Main Program
Read
Design Parameters
Training Parameters
Training Patterns
Initialize
$W_{n,i} = 0$ $PRES_i = 0$
$W_{th} = 0$ $TRES_i = 0$
n = 1 to N
i = 1 to I
Set
Operating Mode
to TRAIN
Compute
$T_{min} = 1/F_{max}$
$T_{max} = 1/F_{min}$
$DT = T_{max} - T_{min}$
$DTIME = NCI * T_{min}$
$H = T_{min}/I$
Write
Design Parameters
Training Parameters
m = 1
8
m = m = 1
Set
Adaption Counter = 0
1

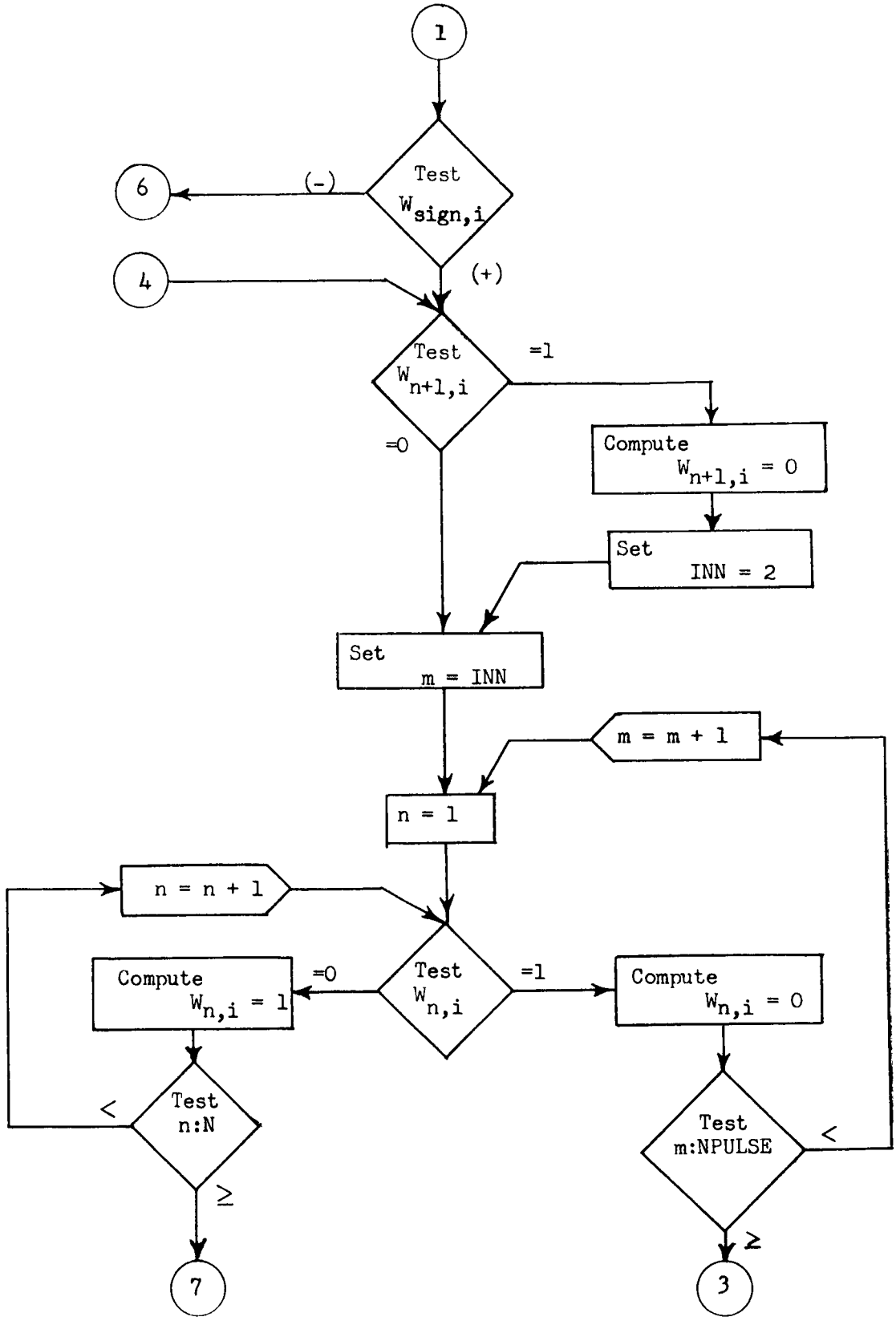

1
j = 1
7
j = j = 1
Set
COUNT = 0
Compute
$C_{n,i}$ = 0
T_i = $TSET_{i,j}$
n = 1 to N
i = 1 to I
CALL BRM
CALL CAT
Write
Results of Classification
Training Pattern j
correct
classification
Test
incorrect
classification (-C)
6
C = -C
incorrect
classification
(+C)
Advance
Adaption Counter
2

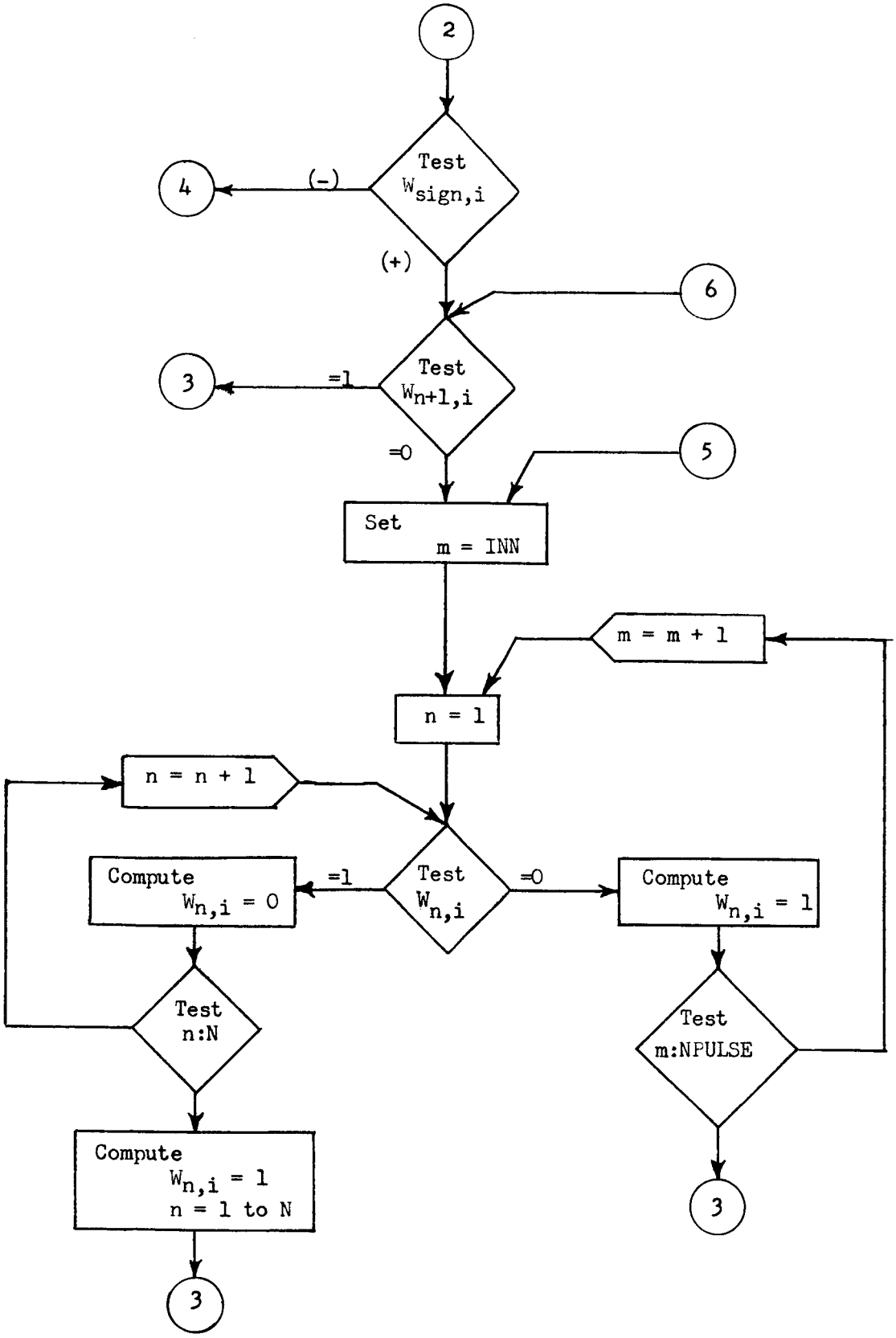

2
CALL ADAPT
Write
Results of Adaption
6
7
Test
j:J
<
≥
Write
Number of Adaptions
in Training Set
Iteration m
Test
Adaption Counter
:0
≤
>
Write
Entire Training Set
Correctly Classified
8
Test
m:MAX
<
≥
Write
Maximum Iterations
of the Training Set
Exceeded
3

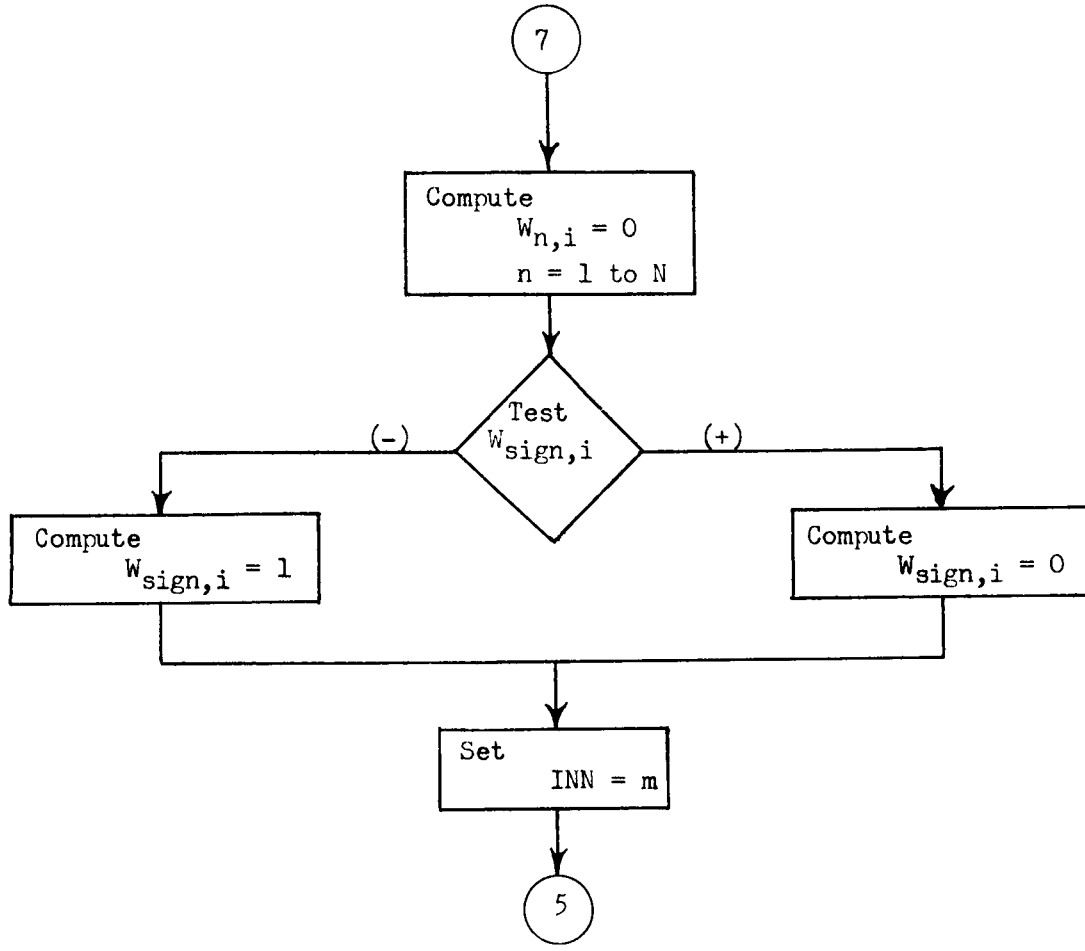

3
Set
Operating Mode
to CLASSIFY
Read
Patterns to be
Classified
Set
Classification Counters = 0
j = 1
9
j = j + 1
Compute
$T_i = CSET_{i,j}$
i = 1 to I
Set
COUNT = 0
CALL BRM
CALL CAT
correct
classification
Test
incorrect
classification
4
5

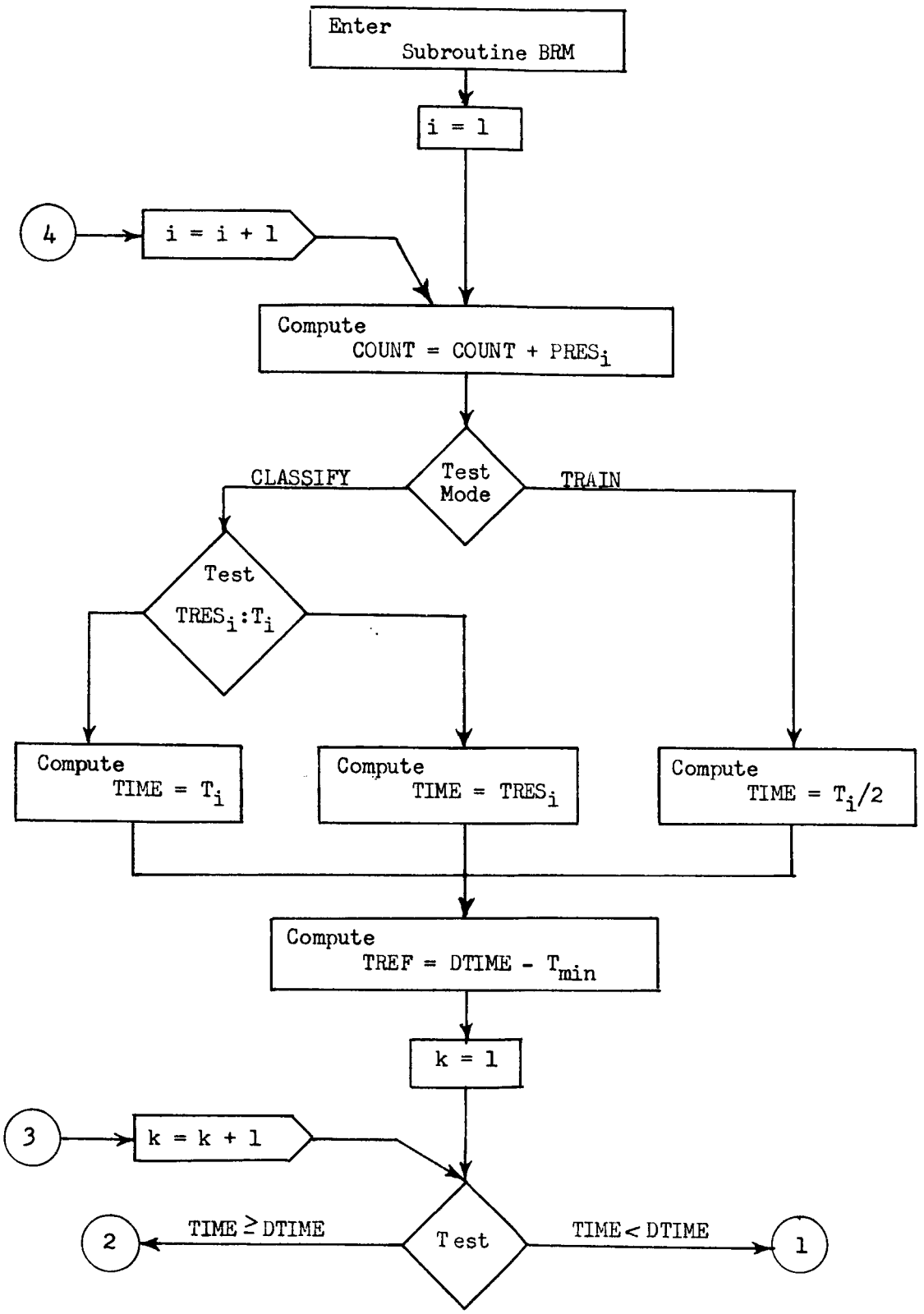

4
5
Write
Correct
Classification
Pattern j
Write
Incorrect
Classification
Pattern j
Advance
Correct
Classification
Counter
Advance
Incorrect
Classification
Counter
9
Test
j:J
Write
Number of Correct
and Incorrect
Classifications
End

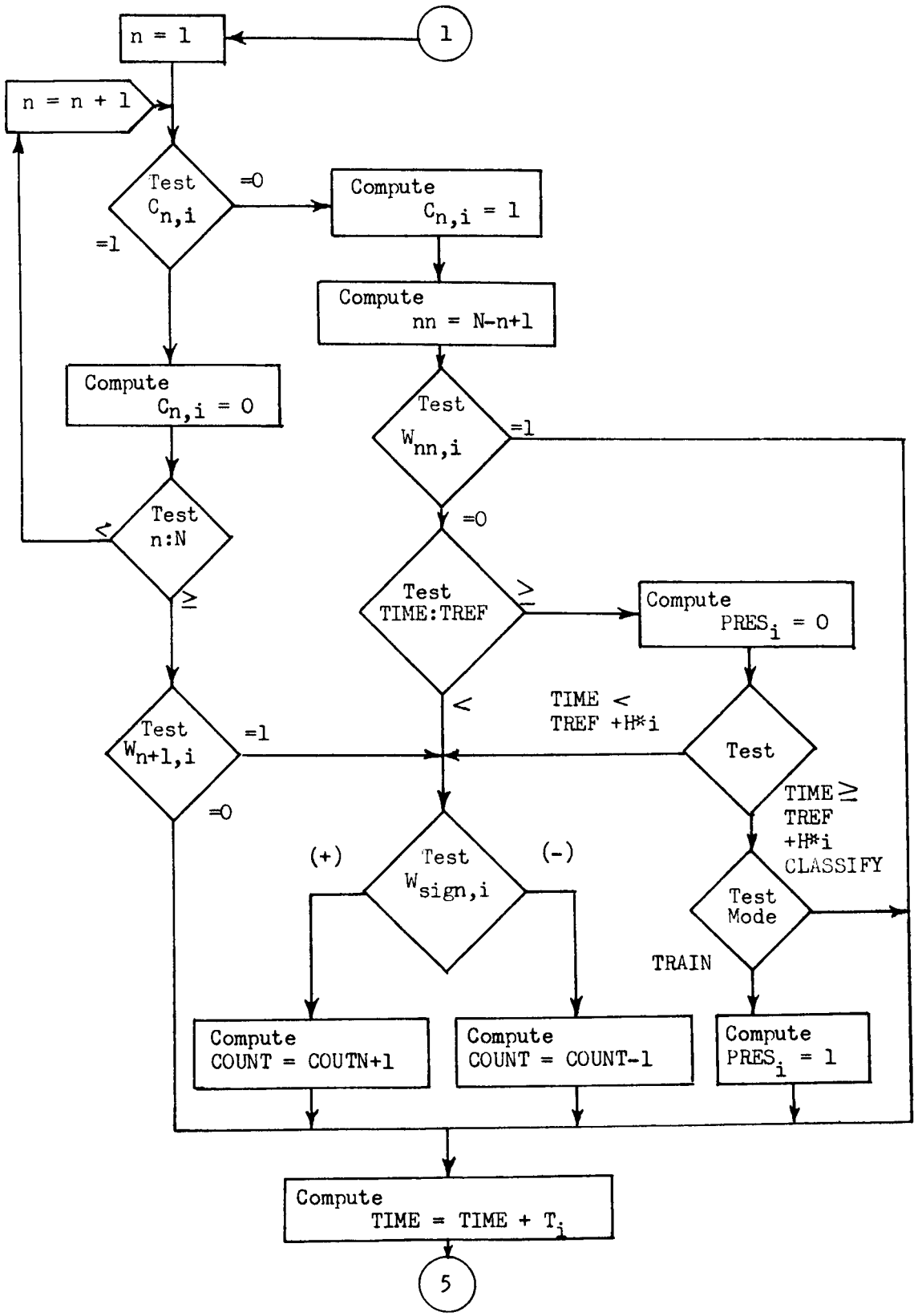

Enter
Subroutine ADAPT
i = 1
i = i + 1
Compute
SG = C*T_i
Compute
NPULSE = SG/DT
Set
INN = 1
Test
SG:O
<
=
>
1
Write
Error
Return
2
3
Test
i:I
<
≥
Compute
W_th = W_th + C*THC
Return

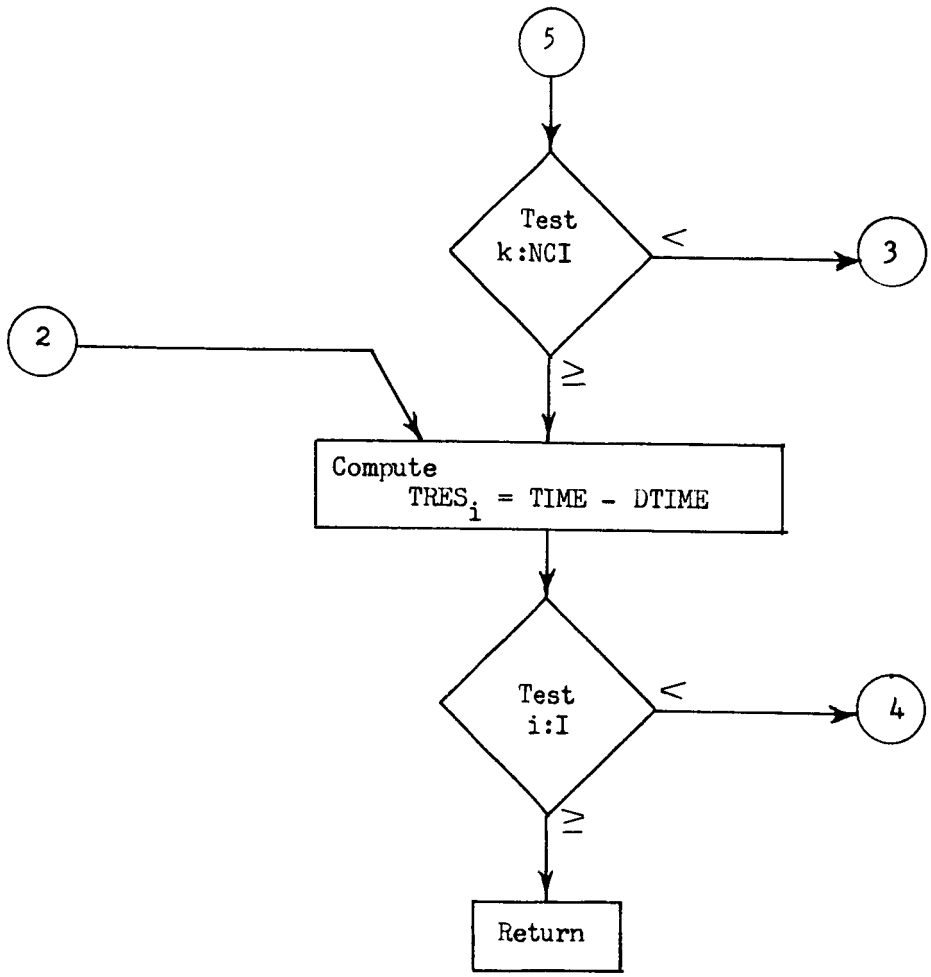

1
Test
$W_{sign,i}$
(-)
6
(+)
4
Test
$W_{n+1,i}$
=1
=0
Compute
$W_{n+1,i} = 0$
Set
INN = 2
Set
m = INN
m = m + 1
n = 1
n = n + 1
Compute
$W_{n,i} = 1$
=0
Test
$W_{n,i}$
=1
Compute
$W_{n,i} = 0$
Test
n:N
<
≥
Test
m:NPULSE
<
≥
7
3

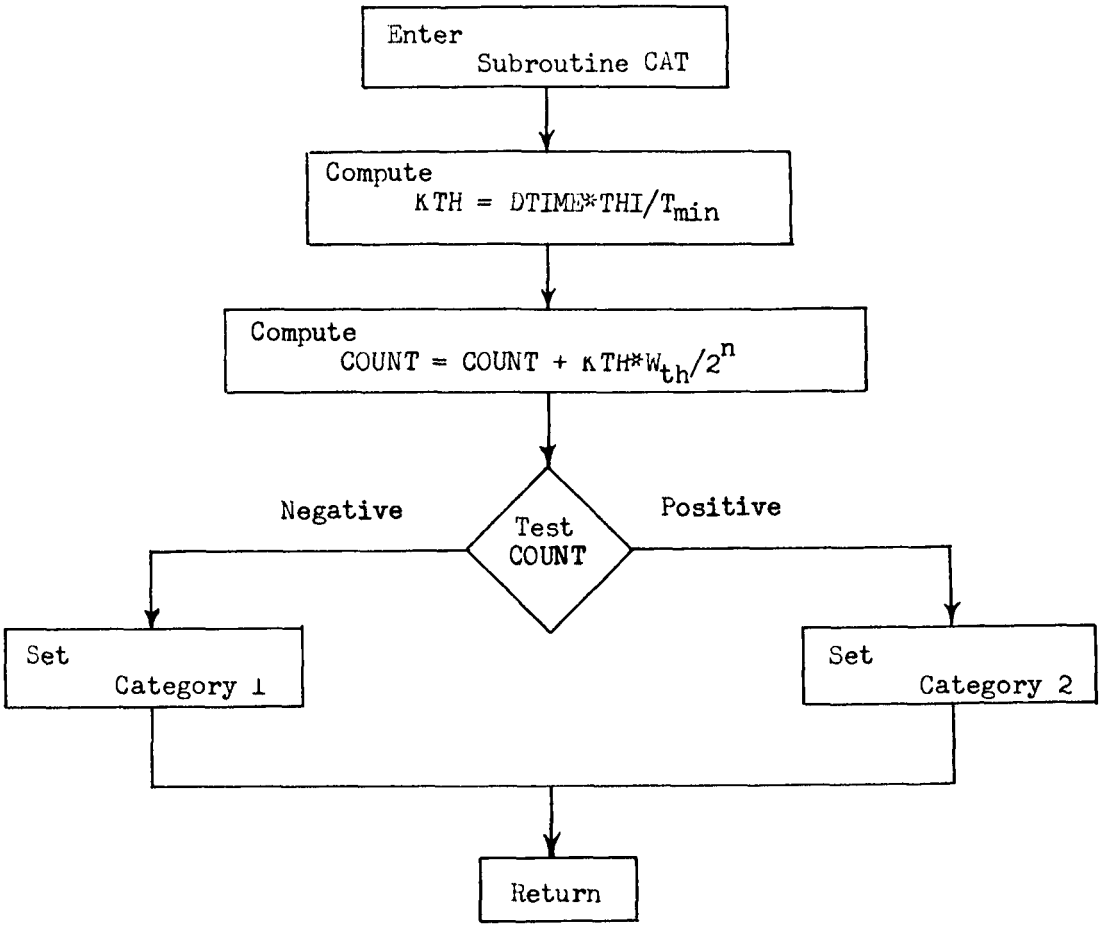

2
Test
$W_{sign,i}$
4
(-)
(+)
6
Test
$W_{n+1,i}$
3
=1
=0
5
Set
m = INN
m = m + 1
n = 1
n = n + 1
Compute
$W_{n,i}$ = 0
=1
Test
$W_{n,i}$
=0
Compute
$W_{n,i}$ = 1
Test
n:N
Test
m:NPULSE
Compute
$W_{n,i}$ = 1
n = 1 to N
3
3

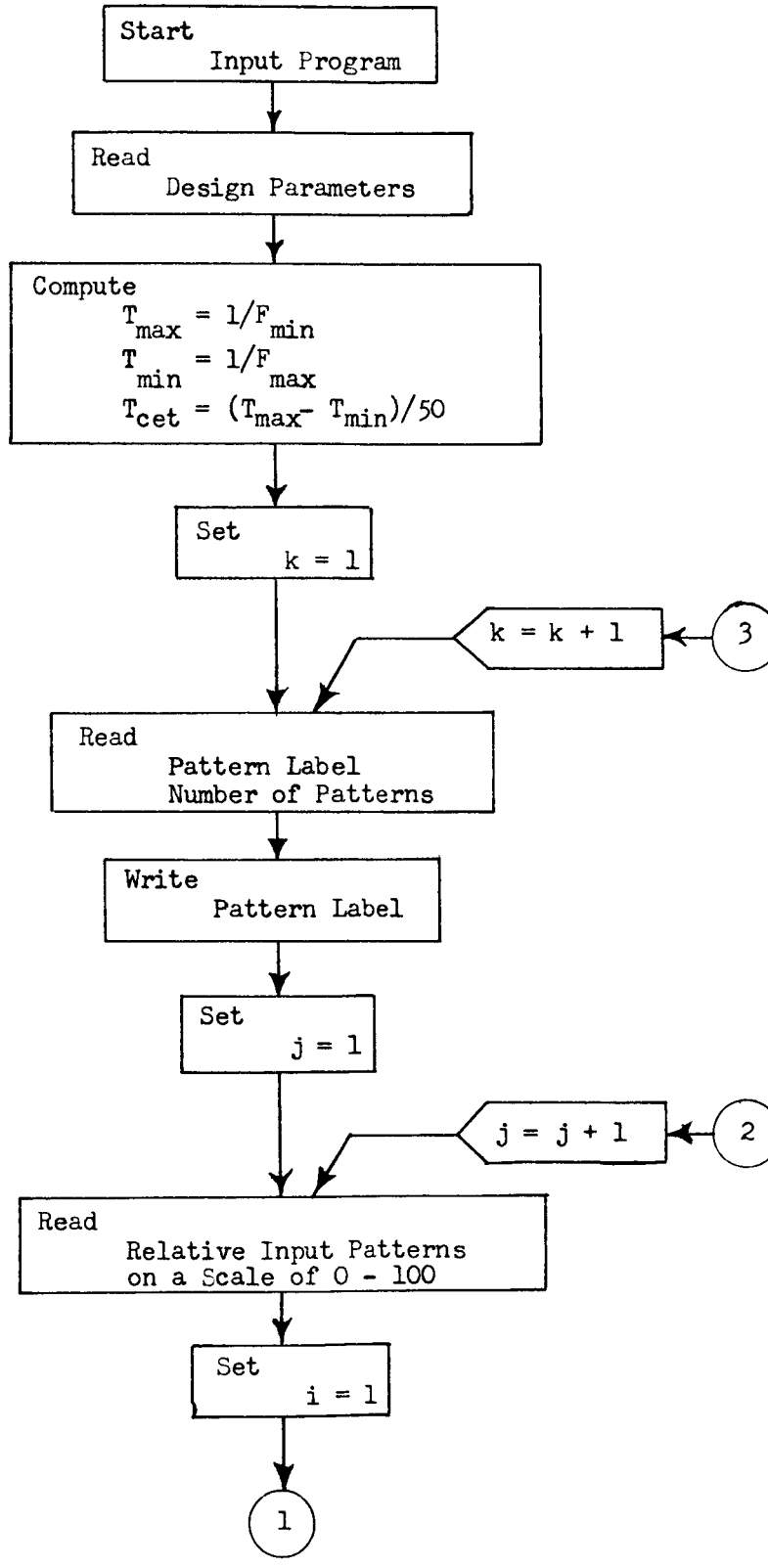

7
Compute
$W_{n,i} = 0$
n = 1 to N
Test
$W_{sign,i}$
(-)
(+)
Compute
$W_{sign,i} = 1$
Compute
$W_{sign,i} = 0$
Set
INN = m
5

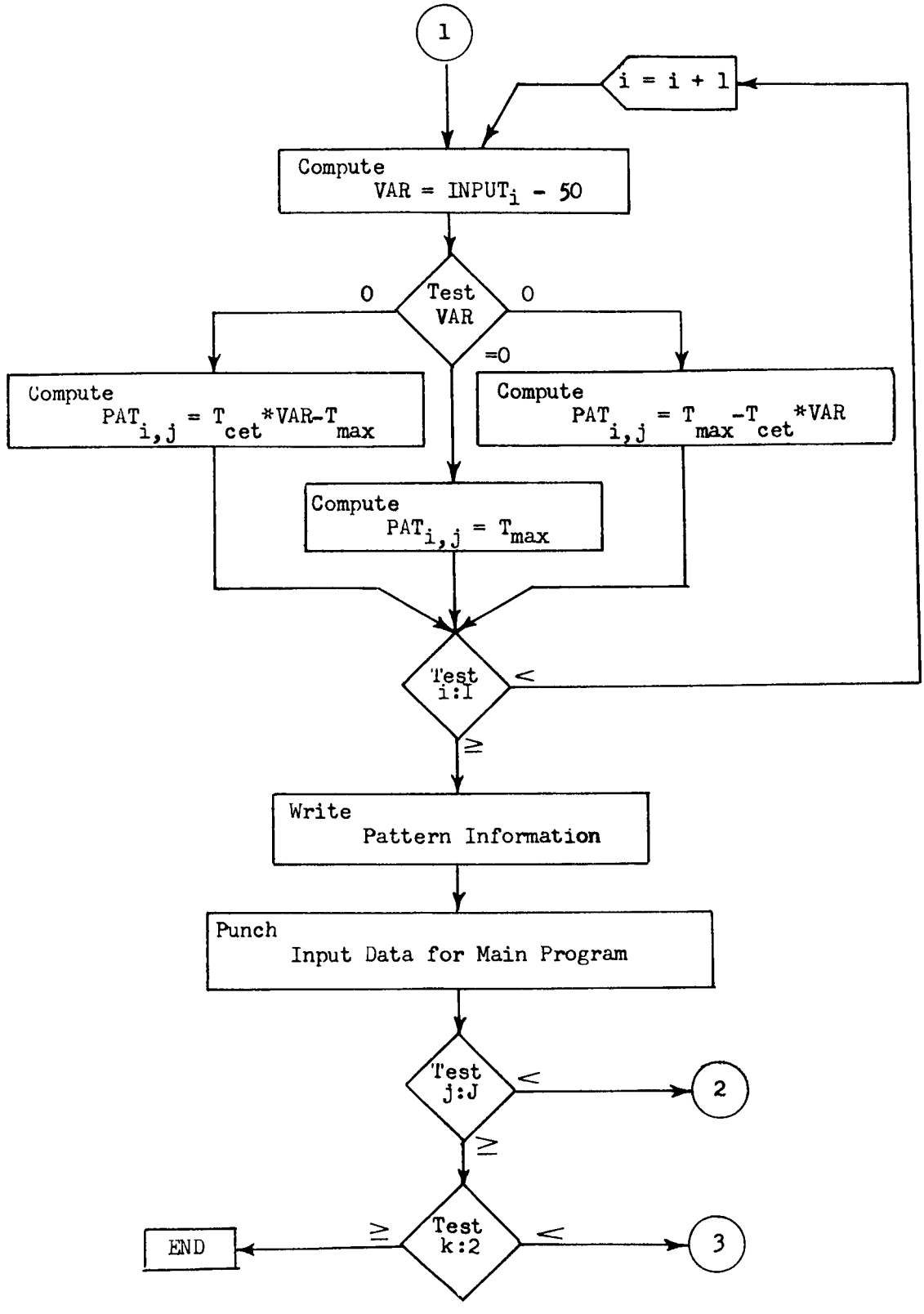

Enter
Subroutine BRM
i = 1
4
i = i + 1
Compute
COUNT = COUNT + PRES$_i$
Test
Mode
CLASSIFY
TRAIN
Test
TRES$_i$:T$_i$
Compute
TIME = T$_i$
Compute
TIME = TRES$_i$
Compute
TIME = T$_i$/2
Compute
TREF = DTIME - T$_{min}$
k = 1
3
k = k + 1
2
TIME ≥ DTIME
Test
TIME < DTIME
1


n = 1
1
n = n + 1
Test $C_{n,i}$
=0
=1
Compute $C_{n,i}$ = 1
Compute nn = N-n+1
Compute $C_{n,i}$ = 0
Test $W_{nn,i}$
=1
=0
Test n:N
<
≥
Test TIME:TREF
≥
<
Compute $PRES_i$ = 0
TIME < TREF +H*i
Test $W_{n+1,i}$
=1
=0
Test
TIME ≥ TREF +H*i
CLASSIFY
(+)
(-)
Test $W_{sign,i}$
Test Mode
TRAIN
Compute COUNT = COUTN+1
Compute COUNT = COUNT-1
Compute $PRES_i$ = 1
Compute TIME = TIME + T_i
5


5
Test
k:NCI
<
3
2
≥
Compute
TRES$_i$ = TIME - DTIME
Test
i:I
<
4
≥
Return


Enter
Subroutine CAT
Compute
KTH = DTIME*THI/T_{min}
Compute
COUNT = COUNT + KTH*$W_{th}/2^n$
Negative
Test
COUNT
Positive
Set
Category 1
Set
Category 2
Return


Start
Input Program
Read
Design Parameters
Compute
$T_{max} = 1/F_{min}$
$T_{min} = 1/F_{max}$
$T_{cet} = (T_{max} - T_{min})/50$
Set
k = 1
k = k + 1
3
Read
Pattern Label
Number of Patterns
Write
Pattern Label
Set
j = 1
j = j + 1
2
Read
Relative Input Patterns
on a Scale of 0 - 100
Set
i = 1
1



NASA-Langley, 1968 —— 8 CR-1035