DIGITAL AUTO-CORRELATOR

DIGITAL AUTO-CORRELATOR:
DESIGN AND CONSTRUCTION

by

Akira Masuko, B.Eng.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

May, 1969

MASTER OF ENGINEERING (1969) McMASTER UNIVERSITY
(Electrical Engineering) Hamilton, Ontario

TITLE: Digital Auto-Correlator: Design and Construction

AUTHOR: Akira Masuko, B.Eng. (Tokyo Metropolitan University)

SUPERVISOR: Professor R. Kitai

NUMBER OF PAGES: 95.

SCOPE AND CONTENTS:

This thesis describes a portable special-purpose digital instrument which uses digital-computer processes to provide decimal readouts of the autocorrelation function (and hence the mean square value too) of a fluctuating voltage, and also of the measuring time involved. The mathematical concepts set out in a paper by R. Kitai and F. Diest[1] are extended to provide the basic equations for the measurement of correlation function. The design and construction processes using integrated circuits are described in detail. The time difference between two samples can be varied at will from 10 μseconds downwards, with no lower limit. The measurement commences on a start signal and runs for a programmed time. Readouts are available immediately after the end of the measuring time. The upper frequency limit is several KHz. There are no low-frequency limits. Some familiar signals were measured and the results confirm that the system works satisfactorily.

ACKNOWLEDGEMENTS

I wish to thank Professor R. Kitai for his constant encouragement and assistance during the course of this work and in the preparation of this thesis.

The financial assistance provided by the Department of Electrical Engineering is gratefully acknowledged.

Also, finally, I would like to thank Mrs. Gibson for typing this thesis.

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF ILLUSTRATIONS	vii
LIST OF TABLES	x
LIST OF SYMBOLS	xi
CHAPTER I: INTRODUCTION	1
CHAPTER II: PRINCIPLE OF DIGITAL CORRELATOR	5
2.1 Correlation Function Using Amplitude and Time Quantization	5
2.2 Weighting Principle	7
2.3 Basic Essentials of Special Purpose Processor	7
2.4 Alternate Methods of Implementation	9
CHAPTER III: DESIGN OF SPECIAL PURPOSE COMPUTER	12
3.1 System blocks	12
3.2 Computing Logic	13
3.2.1 Computing Process	13
3.2.2 Addition Logic	21
3.3 Sampling Pulse Generator	27
3.4 Slave Clock Generator	28
3.5 Analog to Digital Converter	28
3.6 Register S	29
3.7 Register R	29

PAGE

3.8 τ_m-shift Register ---------- 31

3.9 τ_m-select Switch ---------- 31

3.10 Accumulator ---------- 34

3.11 Decimal Display Counter ---------- 34

3.12 C_o Counter ---------- 34

3.13 Control Logic ---------- 35

3.13.1 Master Control Flip-Flop F_{DO} ---------- 35

3.13.2 Slave Control Flip-Flop F_T ---------- 36

3.13.3 4-bit Control Counter ---------- 37

3.13.4 ADD.T and C_a.T ---------- 41

3.14 Explanation of Actual Processes ---------- 41

3.14.1 Power is ON ---------- 42

CHAPTER IV: CONSTRUCTION ---------- 48

4.1 Location of Board and Switch ---------- 48

4.2 Location of the I.C. Unit ---------- 48

4.2.1 Accumulator -A1- ---------- 52

4.2.2 Computing Logic (a) -A2- ---------- 53

4.2.3 Computing Logic (b) -A3- ---------- 54

4.2.4 Register R -A4- ---------- 56

4.2.5 τ_m-shift Register (a) -A5- ---------- 57

4.2.6 τ_m-shift Register (b) -A6- ---------- 58

4.2.7 τ_m-shift Register (c) -A7- ---------- 59

4.2.8 Switch Filter -B1- ---------- 60

4.2.9 Register S -B2- ---------- 61

PAGE

4.2.10 Control Logic (a) -B3- ---------- 62

4.2.11 Control Logic (b) -B4- ---------- 63

4.2.12 Control Logic (c) -B5- ---------- 64

4.3 τ_m-select Switch ---------- 67

CHAPTER V: PERFORMANCE TESTS ---------- 68

5.1 Check by d.c. Input ---------- 68

5.2 Measurements of some Typical Signals ---------- 69

CHAPTER VI: CONCLUSION ---------- 76

6.1 Obtaining Cross-Correlation Functions ---------- 78

6.2 Automatic Selection of Time Difference τ_m ---------- 78

6.3 Error Analysis ---------- 79

APPENDIX ---------- 82

1. Proof of Equation 2-2 ---------- 82

2. Proof of Equation 2-3 ---------- 83

3. SN74XXN Series ---------- 85

4. Circuit Configuration of NAND Gate ---------- 88

BIBLIOGRAPHY ---------- 95

LIST OF ILLUSTRATIONS

FIGURE		PAGE
2.1	Sampling and Quantization	8
2.2	Relation Between Actual Value and the Corresponging Level	8
2.3	Flow Chart for Basic Process	10
2.4	Process I	11
2.5	Process II	11
3.1	System Block Diagram	14
3.2	Block Diagram of Way (a)	16
3.3	Flow Chart of Way (a)	16
3.4	Register S	18
3.5	Block Diagram of Way (b)	18
3.6	Flow Chart of Way (b)	19
3.7	Complete Logic of Two Stage Parallel Adder	26
3.8	Timing of A/D Conversion	28
3.9	Performance of Control Pulse A	30
3.10	Register S and It's Input Logic	30
3.11	Register R and It's Input Logic	32
3.12	τ_m-shift Register	33
3.13	F_{DO} and the Input Gates	36
3.14	Slave Control Flip-Flop	37
3.15	Undesired Spikes Due to the Propagation Delay of Flip-Flop	39

FIGURE		PAGE
3.16	Control Counter	40
3.17	ADD.T and C_a.T	41
3.18	Flow Chart of the Complete Process	45
3.19	System Timing Chart	46
3.20	Computing Timing Chart	47
4.1	Location of Board and Switch	49
4.2	Board Connectors	50
4.3	Locations of I.C. Units on the Board	50
4.4	Overlapping Due to Propagation Delay	65
4.5	Switch Diagram	67
5.1	Auto-correlation Function of a Sine Wave (500 Hz)	70
5.2	Auto-correlation Function of a Sine Wave (1 KHz)	71
5.3	Auto-correlation Function of a Sine Wave (5 KHz)	72
5.4	Auto-correlation Function of a Rectangular Wave (1 KHz)	73
5.5	Auto-correlation Function of a Sine Wave (5 KHz) Plus Noise	74
5.6	Auto-correlation Function of a Rectangular Wave (5KHz) Plus Noise	75
6.1	Logic for Obtaining Cross-correlation Function	80
6.2	Information in the Memory	80
6.3	Automatic Selection of τ_m	81
Appendix		
1.	Joint Probability Distribution	82
2.	Geometrical Expression for the Example	83

FIGURE PAGE

Appendix

3. Dual 4-input NAND Buffer ---------- 85

4. 4-wide 2-input AND-OR-INVERT Gate ---------- 85

5. Dual 2-wide 2-input AND-OR-INVERT Gate ---------- 85

6. Quadruple 2-input NAND Gate ---------- 85

7. Dual 4-input Expander ---------- 86

8. Quadruple 2-input NOR Gate ---------- 86

9. Dual J-K Master-slave Flip-flop ---------- 86

10. Dual D-type Edge-triggered Flip-flop ---------- 86

11. Quadruple 2-input NAND Gate with Open Collector Output - 87

12. Dual 4-input NAND Gate ---------- 87

13. Triple 3-input NAND Gate ---------- 87

14. 8-input NAND Gate ---------- 87

15. Circuit Configuration of NAND Gate ---------- 89

LIST OF TABLES

TABLE		PAGE
3.1	Comparison Between Way (a) and Way (b) ----------	21
3.2	Truth Table of One Stage Parallel Addition ----------	23
3.3	Truth Table of Two Stage Parallel Addition ----------	24
4.1	Function of Board ----------	49

Appendix

1.	Connections From τ_m-select Switch to Board ----------	90
2.	Direct Decimal Display Counter Readout for d.c. Input --	91

LIST OF SYMBOLS

$P(X_{mr}, Y_{ms} : \tau_m)$ = Joint probability such that X(t) lies within the r^{th} interval at time nT and Y(t) lies within the s^{th} interval at τ_m seconds later.

T = Sampling interval.

X_{mr} and Y_{ms} = Mid-interval values of X(t) and Y(t) corresponding to levels r and s respectively.

N = Number of quantization levels

τ_m = Time difference between sample pair and is given, in the system described here, by $\tau_m = mT$, where m = 0 ----- 19.

$P_{r,s}$ = Joint probability that function X(t) exceeds r^{th} level at time nT and function Y(t) exceeds s^{th} level at τ_m seconds later.

C_o = Total number of sample pairs collected.

$C_{r,s}$ = Number of sample pairs which are above r^{th} level at time nT and also above s^{th} level at τ_m seconds later.

Δ = Time required for the addition of one word.

σ = Slave clock period.

CP = Slave clock pulse.

S_{2^P} = $(P + 1)^{th}$ bit from the least significant bit of register S and hence binary weighted by 2^P

$A_1, A_2 .. A_{13}$ = Bits of the accumulator, A_1 represents the least significant bit.

$R_1, R_2 .. R_{12}$ = Bits of register R, R_1 represents the least significant bit.

Acc = Accumulator

R(r) = Contents r in register R.

S(s)	= Contents s in register S.
A/D (r or s)	= Contents r or s in A/D converter.
F_{DO}	= Master control flip-flop.
F_T	= Slave control flip-flop.
F_1, F_2, F_4, F_8	= Nembers of control counter.
A, B, C, D	= Control pulses.
T_{14}	= Timing pulse.
ADD.T and C_a.T	= Timings for two-stage parallel addition.
LSB	= Least significant bit.
MSB	= Most significant bit.
T_m	= One measurement time.

CHAPTER I

INTRODUCTION

The object of this project was to devise a means of evaluating the auto-correlation function of a given time function by employing digital techniques. This could be achieved by using either a general purpose digital computer or a special purpose digital computer.

A general purpose computer is one in which the desired operations are performed in a sequence simply by reading a set of instructions. The sequence of operation can be varied at will by changing the set of instructions. Hence this type of computer may also be used to solve a wide variety of problems. Because of large flexibility the general purpose computer becomes quite expensive, bulky, and also needs specially trained staff for operation.

On the other hand, a special-purpose computer, as the name signifies, is meant for certain predetermined set of operations only. It does not have the flexibility of a general purpose computer, but, where only a certain definite type of operation is wanted, such a computer could be more advantageously used and be more economic. It is specially useful where factors such as speed, weight, and power consumption are of critical importance.

An instrument constructed recently in our laboratory[2,3] can be cited as a typical example of a special purpose computer. This machine measures the mean square value of a fluctuating voltage by sampling techniques. In this thesis the principle established by R. Kitai and

and F. Diest[1] for the instrument mentioned above has been extended by the author for the measurement of a correlation function.

Correlation function is quite important in a large number of applications such as biological and biomedical research, seismological or geological work, physical and chemical research and various engineering areas (vibration analysis, electrical system analysis). The auto-correlation function will be utilized in those cases where a completely unknown signal is to be detected in the presence of an extremely high noise level. Also, this function provides a valuable technique for performing noise measurements or analyzing noise which exists in a system.

The crosscorrelation function can be used to extract a signal from noise if a specific known waveform or frequency is being sought, to measure transmission times, or to measure degree of conformity and phase shift between two similar signals.

The situations where a very accurate and stable result is desired, and especially if long time delays are involved in the computation, the digital technique of obtaining correlation function proves to be superior to the analog method. Since in a digital machine the binary system is used for processes such as quantizing the input signal, storing, computing, etc., the major error lies in the number of quantization levels.

Six bit (sixty-four level) analog to digital conversion is applied to the system described in this thesis so that the error due to the quantization is reduced considerably. The general design specifications are as follows: -

1. Input voltage: ±5V max.
2. Input frequency range: D.C. to approximately 5KHz.
3. Range of delays: 0 sec. to 19 x T sec., where T is a sampling interval.
4. Maximum sampling rate: 100 KHz. (= 1/T).
5. Sample range: 10^4, 10^5 and 10^6. (Number of sample pairs collected for one measurement.

The salient features of this digital correlator can be summarized as follows: -

1. There is no low frequency limitation, i.e., it is capable of handling d.c. signals.
2. At the end of the measurement both correlation function and measurement time are immediately available in decimal form.
3. Although the readouts are decimal, the code can be changed at will by simply changing the readout display counters. The entire logic section remains unaltered. Hence it can be conveniently adapted to some on-line systems.
4. This correlator also obtains the mean square value of a voltage. This leads to two notable areas of application; one is in control system optimization where the voltage represents an error between actual and desired system response, and the system is optimized by minimizing the mean square error. The other application is in the measurement of the mean square value (and hence the r.m.s. value) of a slowly fluctuating voltage where conventional methods fail.
5. Low power consumption (12.8W) and small size (the main logic is condensed into a panel space 6" x 7").

Integrated circuits (Texas Instruments SN74N series) are used in the entire system because of the inexpensiveness, high reliability, low power consumption, and compactness.

This thesis first investigates mathematically the principle of a special purpose processor for obtaining a correlation function by digital techniques. Then the design and construction are described, and finally some familiar time-functions were measured for the performance tests.

The order of presentation is as follows: -

1. Principle of digital correlator.
2. Design of the special purpose processor.
3. Construction.
4. Performance tests.
5. Conclusion.

The system described in this thesis was originally intended to obtain an auto-correlation function only. However, it is easily extended for crosscorrelation function without any change of logic section and this is suggested in the last chapter.

CHAPTER II

PRINCIPLE OF DIGITAL CORRELATOR

In the paper "Digital Transfer Voltmeters"[1] R. Kitai and F. Deist describe a principle to obtain the mean square value of a voltage by a sampling technique. This chapter considers a development of this principle for the measurement of correlation function using the same technique.

The first two sections describe the mathematical investigation for the basic formula and in the last two sections the basic system is proposed.

2.1 CORRELATION FUNCTION USING AMPLITUDE AND TIME QUANTIZATION:

Suppose two continuous functions X(t) and Y(t) are sampled and quantized within N levels (see Figure 2.1). The time-averaged correlation function $R_{XY}(\tau_m)$ is expressed by:

$$R_{XY}(\tau_m) = \lim_{T_m \to \infty} \frac{1}{T_m} \int_0^{T_m} X(t)\, Y(t + \tau_m)\, dt \qquad (2\text{-}1A)$$

and the ensemble averaged correlation function by

$$R_{XY}(\tau_m) = \sum_{r,s=1}^{N} P(X_{mr}, Y_{ms} : \tau_m) X_{mr} . Y_{ms} \qquad (2\text{-}1B)$$

where there are N levels total for each variable. In practice X(t) and Y(t) are sampled and quantized, and the mid-interval values between adjacent quantization levels are used in the computing process. Let X_{mr} be the mid-interval quantized value of a particular sample X(t) and let Y_{ms} be the corresponding value of a particular sample of $Y(t + \tau_m)$,

taken τ_m secs later. Let $P(X_{mr}, Y_{ms}:\tau_m)$ be the joint probability of these events.

$P(X_{mr}, Y_{ms}:\tau_m)$ is expressed in terms of the joint probability $P_{r,s}$, where $P_{r,s}$ represents the probability that X(t) is above the rth level at time t = nT and Y(t) is above the sth level at τ_m seconds later. The relation between $P(X_{mr}, Y_{ms}:\tau_m)$ and $P_{r,s}$ is given by

$$P(X_{mr}, Y_{ms}:\tau_m) = P_{r-1,s-1} - P_{r,s-1} - P_{r-1,s} + P_{r,s} \qquad (2\text{-}2)$$

The proof of Eq. 2-2 is shown in Appendix 1.

The actual value corresponding to the level r is defined as the value at the mid interval of the r-1th and the rth level (see Figure 2.2). Hence we have $X_{mr} = (2r-1)/2N$ and $Y_{ms} = (2s-1)/2N$, where we assume X(t) and Y(t) are normalized in amplitude.

Therefore Eq. 2-1B becomes

$$R_{XY}(\tau_m) = \frac{1}{4N^2} \sum_{r,s=1}^{N} (P_{r-1,s-1} - P_{r,s-1} - P_{r-1,s} + P_{r,s})(2r-1)(2s-1)$$

This is also expressed by (see Appendix 2): -

$$R_{XY}(\tau_m) = \frac{1}{4N^2} \{P_{o,o} + 2\sum_{r=1}^{N-1} P_{r,o} + 2\sum_{s=1}^{N-1} P_{o,s} + 4\sum_{r=1}^{N-1}\sum_{s=1}^{N-1} P_{r,s}\} \qquad (2\text{-}3)$$

In practice, the input signal is superimposed on a d.c. voltage. We assume that X(t) and Y(t) are always positive. Therefore $P_{o,o} = 1$. $P_{r,s}$ is expressed in the sampling system as

$$P_{r,s} \cong \frac{C_{r,s}}{C_o}$$

where C_o is the total number of sample pairs required to obtain one correlation function for one particular value of τ_m and $C_{r,s}$ is the number of sampling combinations such that X(t) is above the rth level at time t = nT and Y(t) is above the sth level at τ_m seconds later.

Eq. 2-3 then becomes

$$R_{XY}(\tau_m) = \frac{1}{4N^2C_o}\left\{1+2\sum_{r=1}^{N-1} C_{r,o}+2\sum_{s=1}^{N-1} C_{o,s}+4\sum_{r=1}^{N-1}\sum_{s=1}^{N-1} C_{r,s}\right\} \qquad (2\text{-}4)$$

Since C_o is large the first term in Eq. 2-4 is usually negligible. Thus we obtain the final expression,

$$R_{XY}(\tau_m) = \frac{1}{2N^2C_o}\left\{\sum_{r=1}^{N-1} C_{r,o}+\sum_{s=1}^{N-1} C_{o,s}+2\sum_{r=1}^{N-1}\sum_{s=1}^{N-1} C_{r,s}\right\} \qquad (2\text{-}5)$$

2.2 WEIGHTING PRINCIPLE APPLIED TO EQUATION 2-5:

The design of the instrument is such that all processing is carried out at the same time as sampling, so that no unnecessary storage is involved. It is shown in Appendix 2A that if a particular sample pair has values r and s respectively, then the contribution of this pair to $R_{XY}(\tau_m)$ as given in Equation 2-5 is: -

$$r + s + 2rs \qquad (2\text{-}6)$$

This quantity is called the "weighting number" to be fed to an accumulator register.

2.3 BASIC ESSENTIALS OF SPECIAL PURPOSE PROCESSOR:

To obtain a correlation function $R_{XY}(\tau_m)$ for a certain value of time difference τ_m we sample the function X(t) and obtain level r (say) and τ_m seconds later sample the function Y(t) and obtain level s (say). Then accumulate the corresponding weighting number shown in Eq. 2-6 and

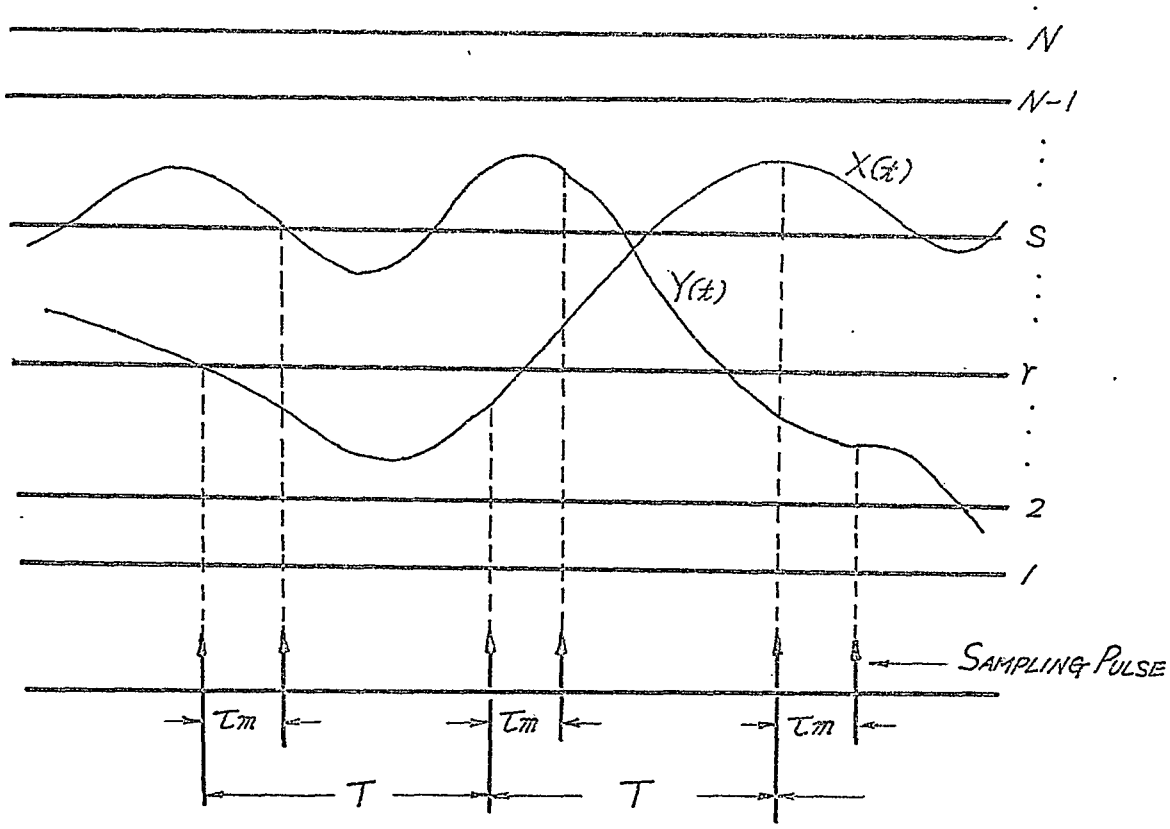


FIGURE 2.1 Sampling and Quantization

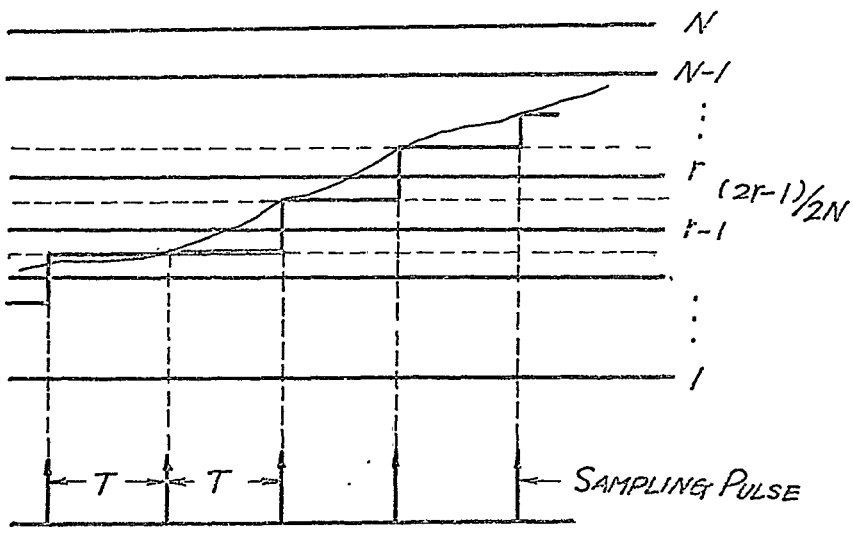


FIGURE 2.2 Relation between Actual Value and the Corresponding Level

repeat this process C_o times. The flow chart for this process is shown in Figure 2.3.

2.4 ALTERNATE METHODS OF IMPLEMENTATION:

Two processes are proposed for the actual system and they are denoted Process I and Process II respectively. Process I has memories to provide time difference τ_m. Hence the computation $r + s + 2rs$ is carried on immediately after the first sample is taken (see Figure 2.4). Instead of having no memory, Process II has a large counter to produce the timing pulses according to the selected τ_m and sampling interval T (see Figure 2.5). The computation for each sample pair has to wait until both samples are taken.

The flow charts are also shown in Figure 2.4 and Figure 2.5 respectively.

Process I was chosen in our system for the reasons mentioned below: -

1. Process I is faster than Process II.
2. Shift registers are quite convenient for use as memories and are now available cheaply in integrated-circuit form.

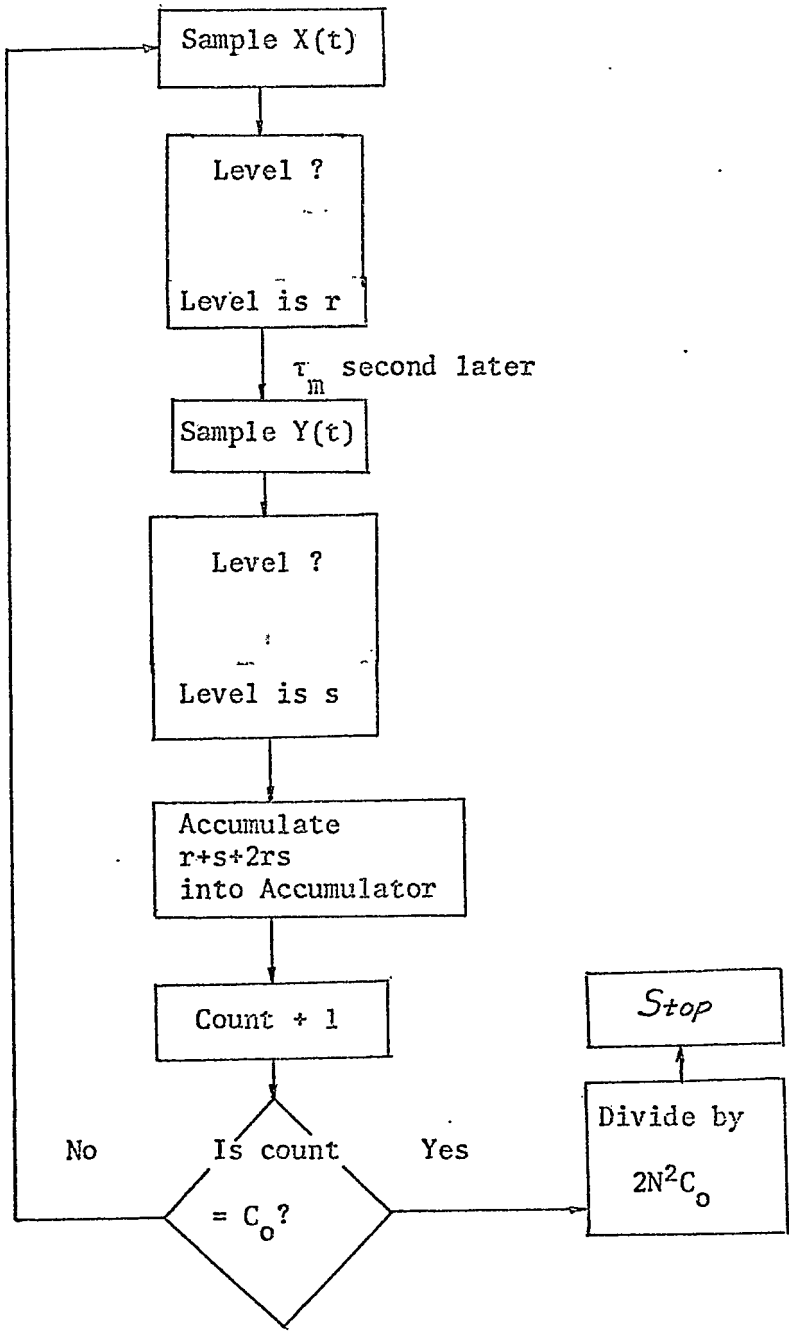
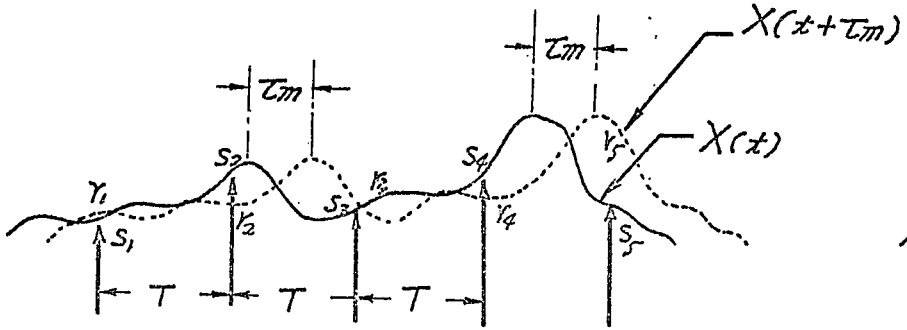
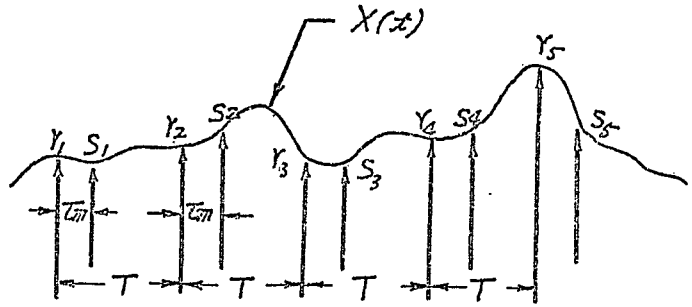


FIGURE 2.3 Flow Chart for Basic Process



Time delayed informations r_1, r_2 ... etc., are stored in a memory having 18-positions with 6 bits/position in the instrument built.



There is no memory. Each pair, r and s, has to be taken in sequence. Then one computes for each pair.

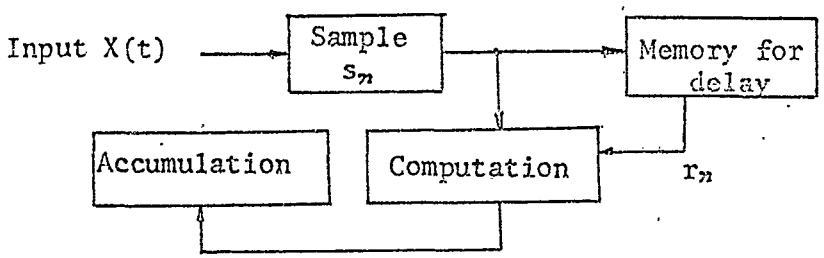


FIGURE 2.4 Process I

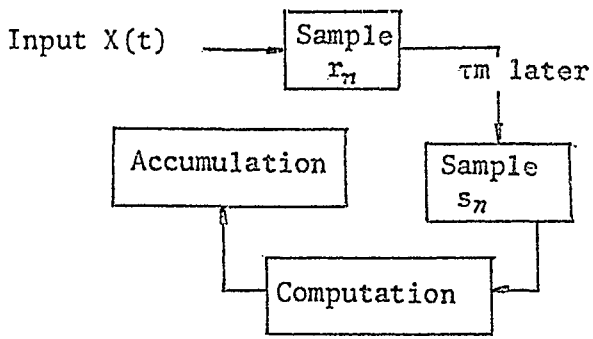


FIGURE 2.5 Process II

CHAPTER III

DESIGN OF SPECIAL PURPOSE COMPUTER

This chapter describes the complete design of the system. After the general process is explained in the first section a description of each block in the system is described in the succeeding sections. In the section 3.2 computing processes and addition logic are mathematically investigated in general form. The last section (3.14) explains the actual processes in order, starting from the power switch and continuing until the completion of one measurement for one correlation function.

3.1 SYSTEM BLOCKS:

Figure 3.1 shows the system block diagram. The input signal is quantized by an A/D converter into 6 binary bits by successive approximation and then fed into register S and τ_m-shift register. For convenience the content in register S is called "Sample s" and the one in τ_m-shift register or in register R is called "Sample r". The sample r is shifted at every sampling instant to provide the time difference τ_m from sample s. The τ_m-select switch connects the m^{th} stage of the shift register to register R where m should be set beforehand manually.

The time difference τ_m between r and s is designed to satisfy the following condition:

$$\tau_m = mT$$

$$m = 0 - 19$$

$$T = \text{Sampling interval}$$

When the process is ON (the definition of this is given in section 3.13) the samples r and s are fed to the computing logic and the result is accumulated in the accumulator. At the same time counter C_o counts the number of sample pairs and when it reaches some large number (powers of ten) it commands the control logic to stop the process.

The decimal display counter counts the carry from the most significant bit of the accumulator (hence the division by $2N^2 = 2^{13}$ is automatically accomplished since the accumulator consists of 13 bits) and displays the answer in decimal form.

3.2 COMPUTING LOGIC:

In this section we shall first examine the different logic techniques for the required computing process [{r + s + 2rs}→ Accumulator] and then investigate the various type of addition logic.

3.2.1 Computing Process

The basic processes are:

1. Take sample s from register S and sample r from register R.
2. Feed them to the computing logic.
3. Accumulate the result in the accumulator.

These are modified for more practical purposes as follows: -

[{r + s + 2rs} → Accumulator] = [{r + (2r + 1)s} → Accumulator]

Hence we perform the following steps: -

Step 1. R(r) + Acc regardless of sample s

Step 2. Convert from R(r) to R(2r + 1)

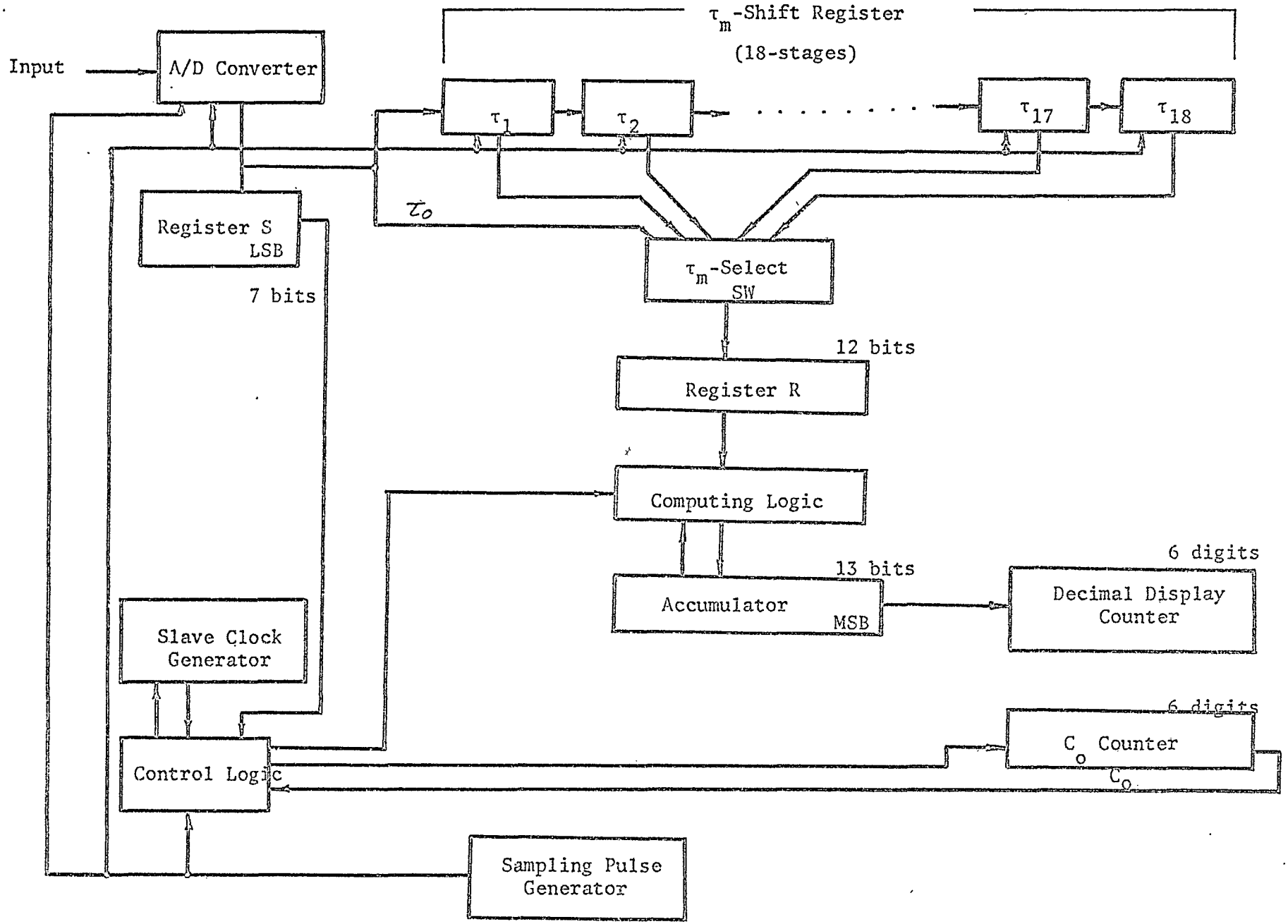


FIGURE 3.1 System Block Diagram

Step 3. R(2r + 1) → Acc depending on sample s.

Where R(r): sample r in register R

Acc : Accumulator

Step 1 is simple addition between register R and the accumulator and Step 2 is explained in detail in section 3.6.

We now concentrate on the investigation of step 3. For simplicity let 2r + 1 = r' and consider the process {r's → Acc}. This can be achieved in one of the two following ways:

(a) One way is to repeat the addition {r' → Acc} s times, i.e., every time after the addition s is subtracted by one and the adding process is repeated until s becomes zero. A down-counter is used for register S. It also acts as a normal register and while the computing is going on it counts the last timing pulse downward to provide the same effect of subtraction by one. The block diagram and the flow chart including the step 1 and 2 are shown in Figure 3.2 and Figure 3.3 respectively. In Figure 3.2, gates 1 and 2 accomplish steps 3 and 1, respectively, and control pulse C is used for step 2.

(b) The other possible way is more elegant than the first and saves computing time. The reason for this is explained in the last part of this section.

Suppose register S consists of n bits and each bit is named S_{2^0}, S_{2^1}, S_{2^2}, ..., etc., according to the binary weights; 2^0, 2^1, 2^2, ..., respectively (see Figure 3.4). The multiplication {r's} is rearranged as follows: -

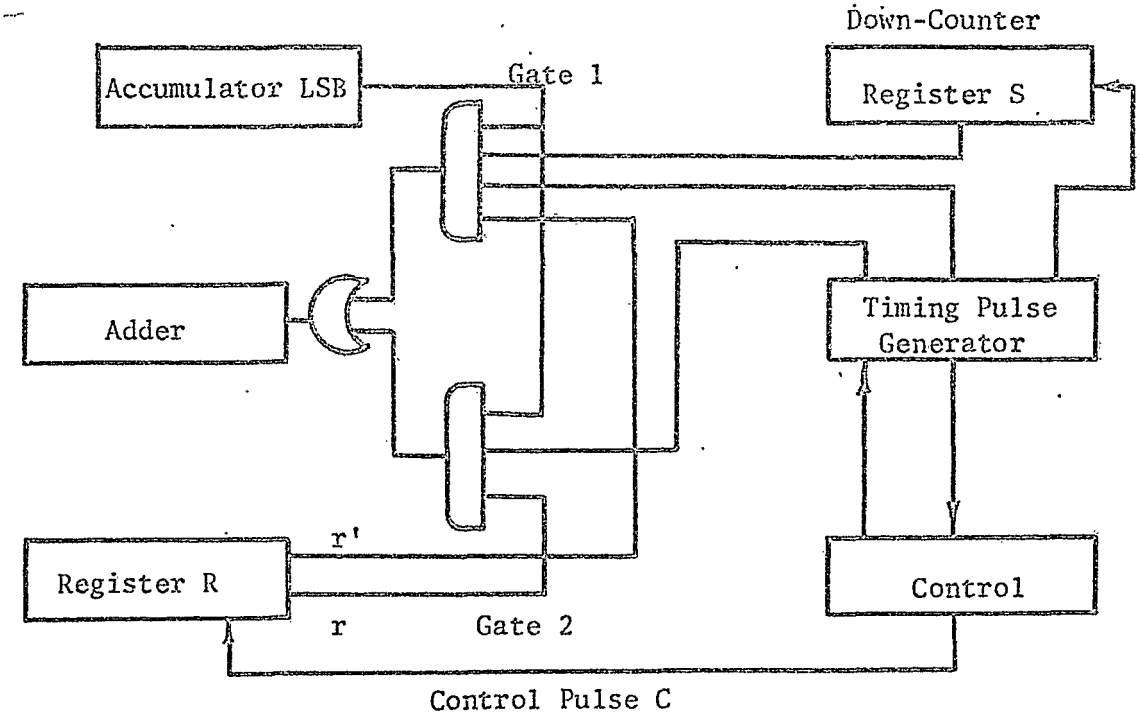


FIGURE 3.2 Block Diagram of Way (a)

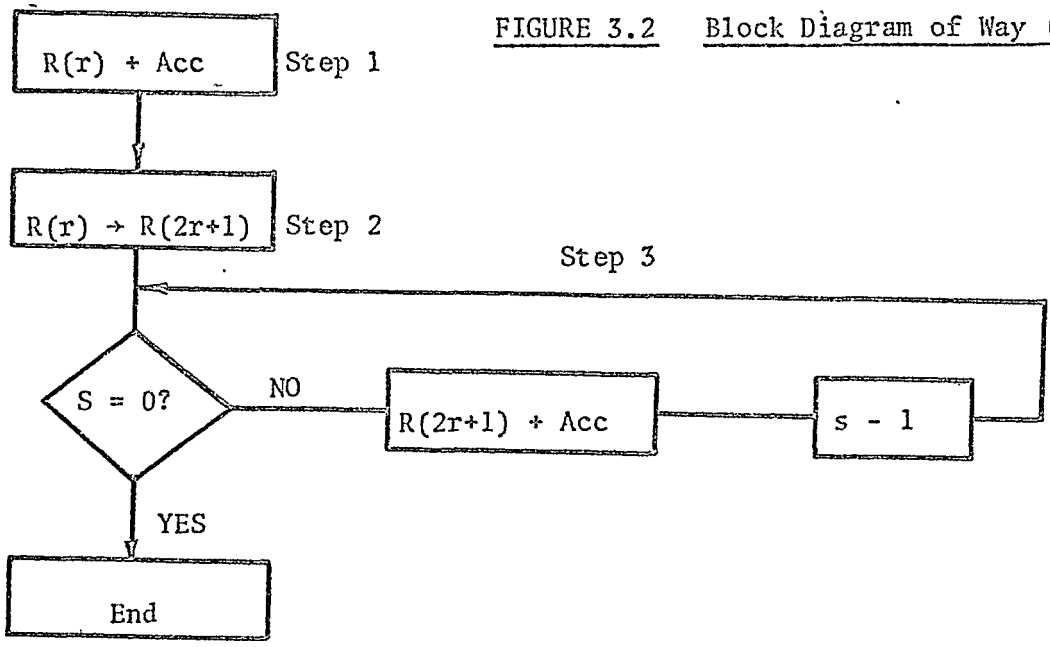


FIGURE 3.3 Flow Chart of Way (a)

$$r's = r'(S_{2^0} + S_{2^1} + S_{2^2} + \ldots + S_{2^P} + \ldots + S_{2^{n-1}})$$

$$= r' S_{2^0} + r'S_{2^1} + \ldots + r' S_{2^P} + \ldots + r'S_{2^{n-1}}$$

$$= r' \sum_{P=0}^{n-1} S_{2^P} \qquad (3\text{-}1)$$

where S_{2^P} is (P + 1)th bit from the least significant bit of register S and binary weighted by 2^P. Therefore each multiplication in Eq. 3-1 is analyzed as follows: -

If S_{2^P} = "1" then $r'S_{2^P} = r'.2^P$

If S_{2^P} = "0" then $r'S_{2^P} = 0$

The process $r'.2^P$ is easily performed by simply shifting r' in register R P bits to the upper significant bit.

Now the process $\{r's \rightarrow Acc\}$ becomes the process $\{r' \sum_{P=0}^{n-1} S_{2^P} \rightarrow Acc\}$. Suppose s in register S is also shifted to the lesser significant bit by the same shift pulse of register R and we watch only the output of the least significant bit (LSB) of register S, then S_{2^P} appears at the LSB and r' in register R becomes $r'.2^P$ at the Pth shift pulse. Hence what we have to do is simply perform the addition $\{R(r'.2^P)+Acc\}$ if only LSB = "1", i.e., in this method only the LSB controls the addition between register R and the accumulator.

In Figure 3.5 of the block diagram, gates 1 and 2 accomplish steps 1 and 3, respectively, and control pulse C contributes to step 2.

Let us now examine mathematically the relative requirements of bits and computing time of the two possible methods (a) and (b).

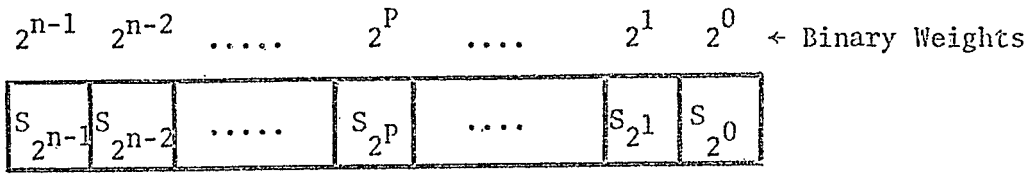


FIGURE 3.4 Register S

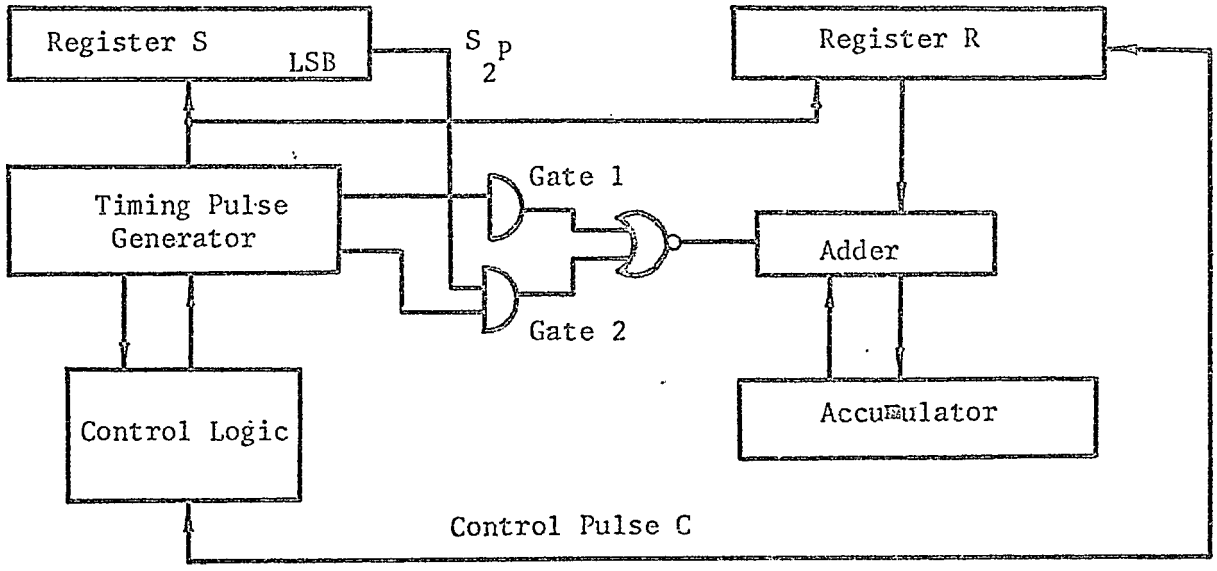


FIGURE 3.5 Block Diagram of Way (b)

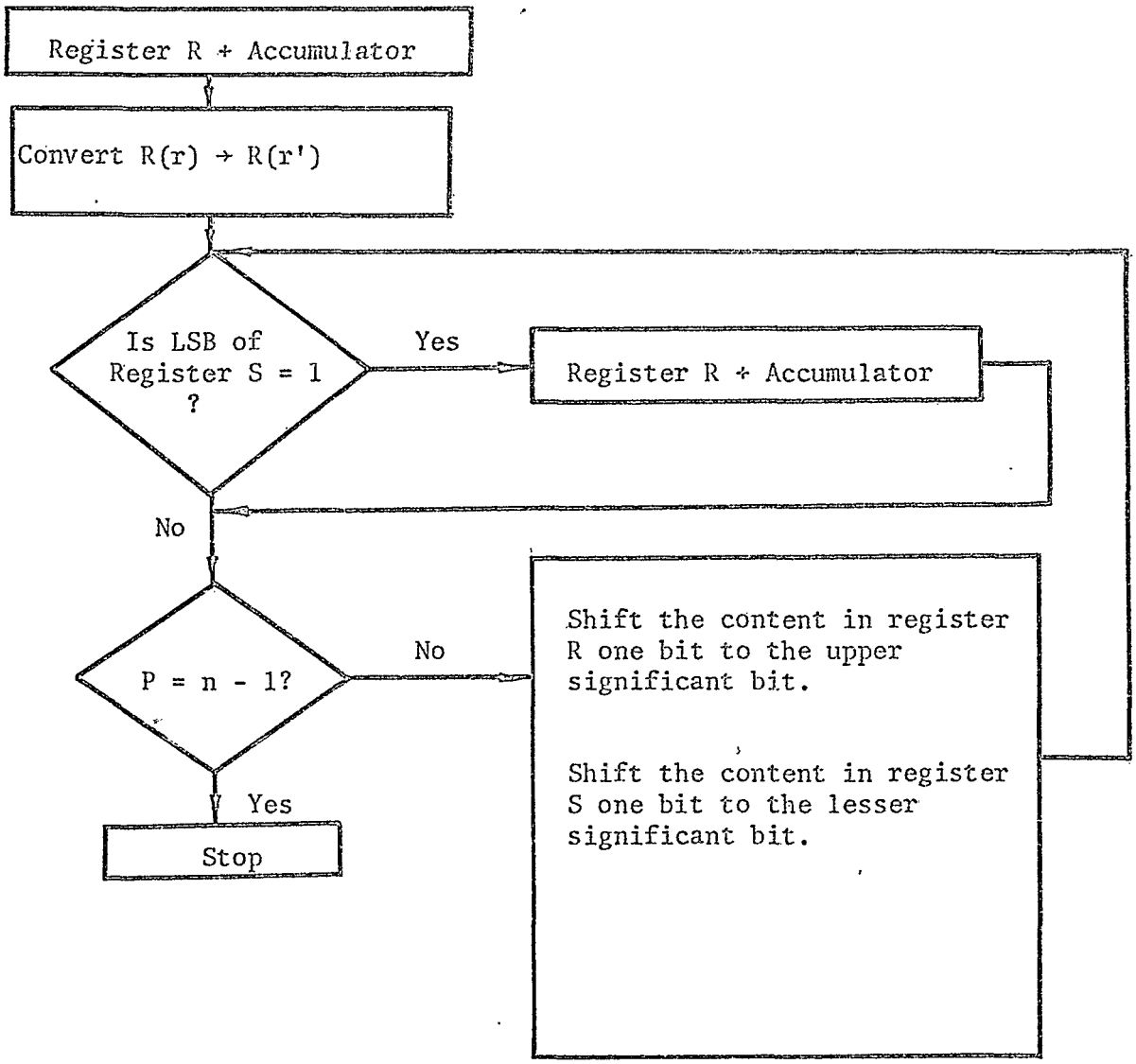


FIGURE 3.6 Flow Chart of Way (b)

Suppose N is the number of level quantization. The maximum number which register S contains is

$$\text{Max } S = N - 1 < N$$

Hence $\log_2 N = n$ bits are required for register S. While the number which register R contains at a certain moment is $2r + 1$ in way (a) and $(2r + 1)S_{2^P}$ in way (b), hence

$$\text{Max } (2r + 1) = 2(N - 1) + 1 = 2^{n+1} - 1 < 2^{n+1}$$

$$\text{Max } (2r+1)S_{2^P} = \{2(N - 1) + 1\}2^{n-1} < 2^{2n}$$

Therefore $(n + 1)$ bits are required in way (a) and $2n$ bits in way (b), respectively, for register R.

Suppose the addition takes Δ seconds. The first way takes roughly $(N - 1)\Delta$ seconds to accomplish one computation $\{r + 2 + 2rs\}$ because the process is a repetition of addition.

Since the second way applies the shifting technique and register S is $n(= \log_2 N)$ bits it takes $n\Delta$ seconds.

The major difference lies in these two points mentioned above and summarized in Table 3.1.

If the system requires minimum cost rather than a high speed process the first method is convenient. We used the second method in our system because of the following reasons: -

1. A high speed process is required.
2. A shift register is not expensive using integrated circuit.

TABLE 3.1 Comparison Between Way (a) and Way (b)

where $n = \log_2 N$, N is the number of level quantization

	Bits requirement to register S	Bits requirement for register R	Computing time of {r÷s÷2rs}
Way (a)	n	n ÷ 1	(N-1) Δ
Way (b)	n	2n	$(\log_2 N)\Delta$

3.2.2 Addition Logic

Two possible types of adders could be used, viz.,

1. Series adder
2. Parallel adder { One stage / Two stage

The feasibility of using either for our system will be investigated.

(a) Series adder -

This type of adder is the most commonly used. The addition is performed one bit by one bit with some synchronous clock pulse. Suppose one word consists of n bits and the clock period is σ seconds, nσ seconds are required to complete the addition of one word. Since we applied way (b), one word consists of 2n bits and there are seven words (one is from step 1 and six are from step 3). Hence, at least 14 nσ seconds are required to compute {r ÷ s ÷ 2rs}. For example, n = 6 bits, σ = 1 μsecond, then 84 μseconds are required, where we neglect some unknown extra time. This is too long and not applicable to our system.

(b) Parallel adder - one stage -

This performs the addition of one word within one clock period. We consider the process {R(r) + Acc} → Acc. In other words, add the content of register R to the content of the accumulator, then store the result in the accumulator. In this method the accumulator performs two things: one is the addition with register R and another is the accumulation of the result.

Let A_N: Nth bit of accumulator.

R_N: Nth bit of register R.

C_N: Carry from the lesser significant bits.

The new state of A_N(New A_N) depends on R_N, C_N and A_N itself. All the possible combinations for A_N to be changed into New A_N are given in Table 3.2.

From Table 3.2 the signal which should change the state of A_N complementarily is

$$\text{Comp } A_N \text{ (say)} = R_N \oplus C_N$$

and C_{N+1} is expressed by

$$C_{N+1} = \bar{R}_N A_N C_N + R_N \bar{A}_N C_N + R_N A_N \bar{C}_N + R_N A_N C_N$$
$$= (R_N \oplus A_N) C_N + R_N A_N$$

where $N = 1, 2, 3, \ldots$

$C_1 = 0$

and $R_N \oplus A_N = R_N \bar{A}_N + \bar{R}_N A_N$

Thus this type of adder is constructed from the combinations of exclusive OR gates (Half Adders).

TABLE 3.2 Truth Table of One Stage Parallel Addition

where C_{N+1} is the carry due to A_N, R_N and C_N and it affects the more significant bits, A_{N+1} A_{N+2} ...., etc.

R_N	A_N	C_N	New A_N	C_{N+1}
0	0	0	0	0
0	(0)	1	(1)	0
0	1	0	1	0
0	(1)	1	(0)	[1]
1	(0)	0	(1)	0
1	0	1	0	[1]
1	(1)	0	(0)	[1]
1	1	1	1	[1]

(c) Parallel adder - two stage -

This is described in the Logic Handbook by Digital Equipment Corporation[4]. Suppose we divide the process of one stage parallel addition into two stages such as

1. Half-addition timing (ADD.T)
2. Carry timing (C_a.T)

In other words, A_N is changed complementarily if R_N = 1 at half-addition timing (ADD.T) and again complemented by C_N if C_N = 1 at carry timing (C_a.T). Then the logic which complements A_N becomes

$$\text{Comp } A_N = R_N.(ADD.T) + C_N.(C_a.T)$$

The truth table for this case is shown in Table 3.3. A_N' is the state of A_N after ADD.T and New A_N is the result of A_N caused by C_N and it is the final answer for one parallel addition.

Table 3.3 tells us: -

1. A_N should be changed complementarily by $R_N = 1$ at ADD.T, i.e. $A_N \rightarrow A_N'$ by $R_N = 1$ at ADD.T.
2. A_N' should be changed complementarily by $C_N = 1$ at C_a.T, i.e. $A_N' \rightarrow$ New A_N by $C_N = 1$ at C_a.T.

TABLE 3.3 Truth Table of Two Stage Parallel Addition

R_N	A_N	A_N'	C_N	New A_N	C_{N+1}
0	0	0	0	0	0
0	0	0 ←	1	1	0
0	0	1	0	1	0
0	1	1 ←	1	0	1
1 →	0 →	1	0	1	0
1 →	0 →	1 ←	1	0	1
1 →	1 →	0	0	0	1
1 →	1 →	0 ←	1	1	1
← ADD.T →			← C_a.T →		

Care must be taken to obtain C_{N+1} which has effects on A_{N+1}, A_{N+2}, ... C_{N+1} is dependent on C_N, A_N' and R_N and not on A_N because A_N is changed to A_N' already at ADD.T.

Now we can derive C_{N+1} from Table 3.3,

$$C_{N+1} = \bar{R}_N A_N' C_N + R_N A_N' C_N + \bar{A}_N R_N$$

$$= A_N' C_N + \bar{A}_N' R_N$$

Since A_N' is just the notation of the state of A_N after ADD.T, C_{N+1} is rewritten in actual logic system such as;

$$C_{N+1} = A_N C_N + \bar{A}_N R_N$$

Hence Comp A_N becomes

$$\text{Comp } A_N = R_N(\text{ADD.T}) + C_N(C_a.T)$$

$$= R_N(\text{ADD.T}) + C_{N-1} \cdot A_{N-1} (C_a.T) + A_N + R_{N-1} (C_a.T)$$

$$= \overline{R_N \cdot (\text{ADD.T})} \cdot \overline{C_{N-1} A_{N-1} \cdot (C_a.T)} \cdot \overline{\bar{A}_{N-1} \cdot R_{N-1} (C_a.T)}$$

where

$$N = 2, 3, 4, \ldots.$$

$$C_1 = 0$$

In the logic diagram of Figure 3.7, the process {(register R + Accumulator) → Accumulator} is assumed. Symbols R_1, R_2, ... and A_1, A_2, ... are the bits of register R and the accumulator respectively.

We have seen that the two stage parallel adder takes a time twice that of the similtaneous type and yet we apply it to our system for the following reasons: -

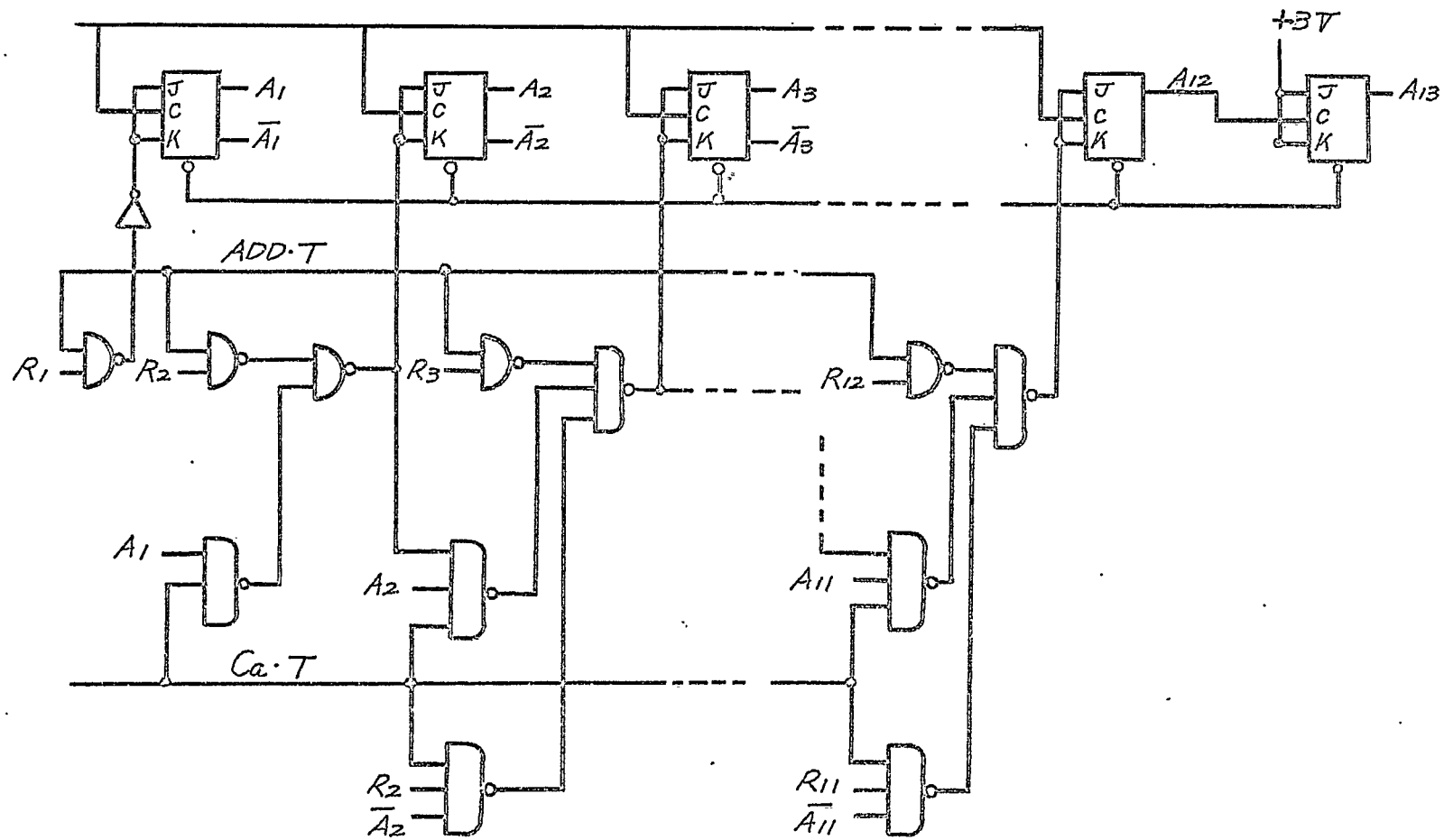


FIGURE 3.7 Complete Logic of Two Stage Parallel Adder, where slave clock pulse is controlled by Control Logic and ADD.T, $C_a \cdot T$ come from Control Logic.

1. The type of one stage parallel adder which is now commercially available does not have exactly the same logic as described in 3.2.2 (b) and needs another register (answer register).

2. Even in integrated circuit form the one-stage parallel adder is still expensive, considering the fact that we assemble the two stage parallel adder from simple NAND gates.

(d) Frequency limitation of slave clock pulse due to parallel addition -

The minimum usable period of slave clock pulses which carry on the parallel addition (one stage or two stage) is restricted by the overall propagation delay of C_N. Suppose one gate (including flip-flop) has P seconds delay and C_N goes through a maximum number of q gates. Then the maximum clock frequency is 1/pq Hz. For example, Texas Instruments Type SN74N series has 13 ηseconds propagation delay norminal (hence $p = 13 \times 10^{-9}$) and one word is 12 bits in our adding system (see Section 3.7). Hence maximum frequency of slave clock pulse is 1/pq = 6.4 MHz.

3.3 SAMPLING PULSE GENERATOR:

This unit generates sampling pulses to initiate A/D conversion and it also sets the time difference τ_m in the τ_m-shift register.

The author designed a simple monostable multivibrator using open-collector NAND gates over the frequency range 1 KHz - 50 KHz (see Section 4.2.12). However, its temperature stability is less than desirable and so it is necessary to measure the frequency each time before the system is operated. The system also accepts externally generated pulses if desired for the sampling pulses.

3.4 SLAVE CLOCK GENERATOR:

A Digital Equipment Corporation Type M401 unit is used for this purpose. A level change from high to low with fall time less than 400 nseconds is required to trigger the generator. As seen in the system timing chart (Figure 3.19) or in the computing timing chart (see Figure 3.20) if the process is ON, 1 MHz clock pulses are generated by the slave control flip-flop $\bar{F}_T$ (see Section 3.13.2) to compute {r+s+2rs}.

3.5 A/D CONVERSION:

A Digital Equipment Corporation Type A801 A/D converter is used for this purpose. It generates a done pulse after the conversion (see Figure 3.8).

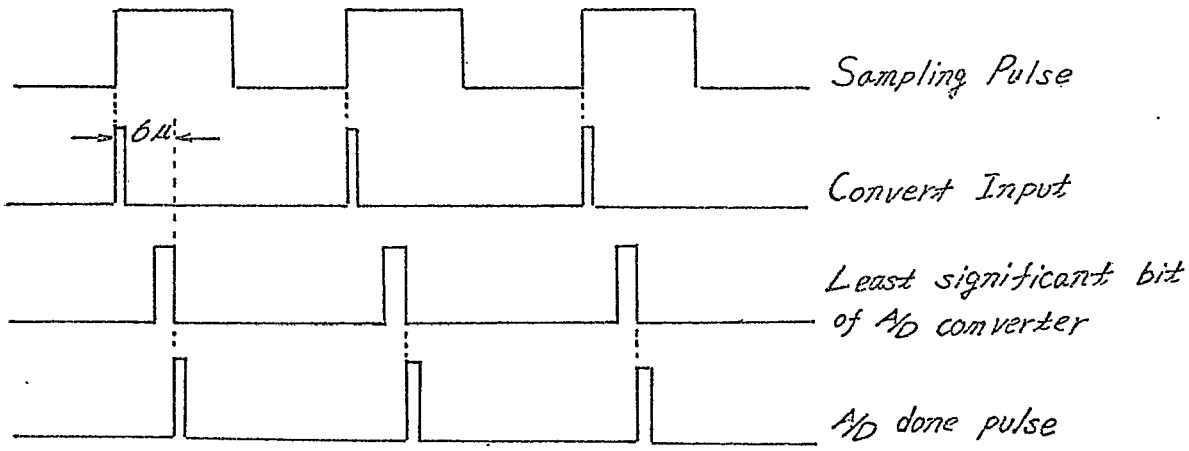


FIGURE 3.8 Timing of A/D Conversion

Since the A/D convert input is a pulse of narrow width and sharp rise time (pulse width; min 100 nseconds, max 500 nsecond, rise time; 20 nseconds norminal). A one-shot circuit using open collector NAND gates is used (see Section 4.2.12).

3.6 REGISTER S:

Register S performs two operations:

1. Receive the information (sample s) from A/D converter.
2. Shift the information (sample s) toward the lesser significant bit to control the computation $(2r + 1)s$.

Since both operations 1 and 2 are performed by the same shift pulse F_1 which is generated from the control logic, control pulse A (say) is used to distinguish the operations such as (see Figure 3.9),

(a) If A = 1 operate 1

(b) if A = 0 operate 2.

Normally 6 bits are required for this register but because of practical reasons one dummy bit is added to the least significant bit. The purpose of this is described in Section 3.14. Figure 3.10 shows the block diagram and the complete logic diagram is shown in Section 4.2.9.

3.7 REGISTER R:

Register R also performs two operations:

1. Receive the information (sample r) from τ_m-selecting switch.
2. Shift the information (sample r) to the upper significant bit.

The control pulse A associated with register S is also used here to distinguish the two operations. The required forming of $(2r+1)$ described in Section 3.2 is achieved by shifting the sample r one bit to the upper significant bit and at the same time feeding binary "1" to the least significant bit. Control pulse C is used for this purpose. The maximum number in register R is $(2r+1)S_{25}$ (see Section 3.2 of

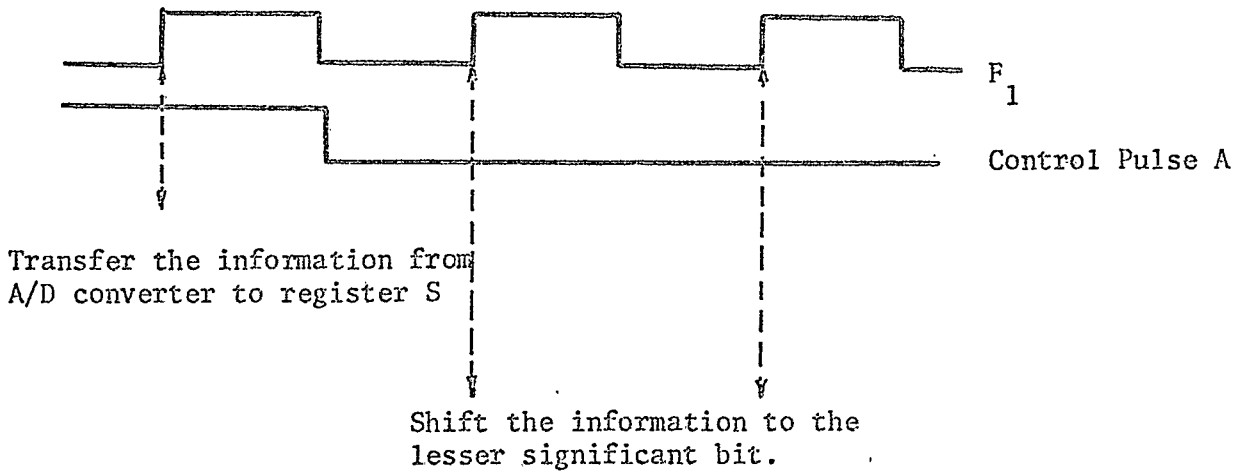


FIGURE 3.9 Performance of Control Pulse A

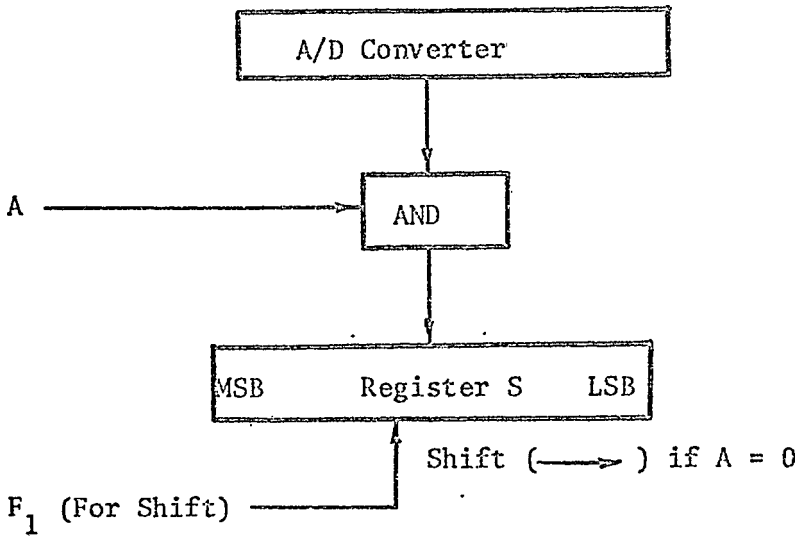


FIGURE 3.10 Register S and its Input Logic

of computing process) is 4064. Hence 12 bits are required for the register. Figure 3.11 shows the block diagram and the complete logic diagram is shown in section 4.2.4.

3.8 τ_m-SHIFT REGISTER:

Use is made of D-type flip-flops (see Appendix 3). If power is ON shifting is continuously operated since the sampling pulse generator runs continuously and hence there is no need to clear the registers before any measurement.

18 stages of shift register are constructed. Each stage consists of 6 bits. Information for $\tau_m = 0$ comes directly from the A/D converter to the τ_m-select switch. Hence the total number of τ_m coordinates are 19 in our system.

The first few stages are shown in Figure 3.12. The buffer in Figure 3.12 can drive up to 30 units loads and input C of each D-type flip-flop is equivalent to 2 units loads. For 18 stages the total load becomes 216 units loads. Therefore, approximately 7 buffers are required to drive all the stages.

3.9 τ_m-SELECT SWITCH:

A six section-twenty position switch is used. Each section corresponds to binary weighted numbers 32, 16, 8, 4, 2, 1, respectively, and m^{th} position (m = 1 - 19) corresponds to the m^{th} stage of the τ_m-shift register. A detailed diagram is given in Section 4.3. The 20^{th} position is the wiper arm and is connected to the input gates of register R.

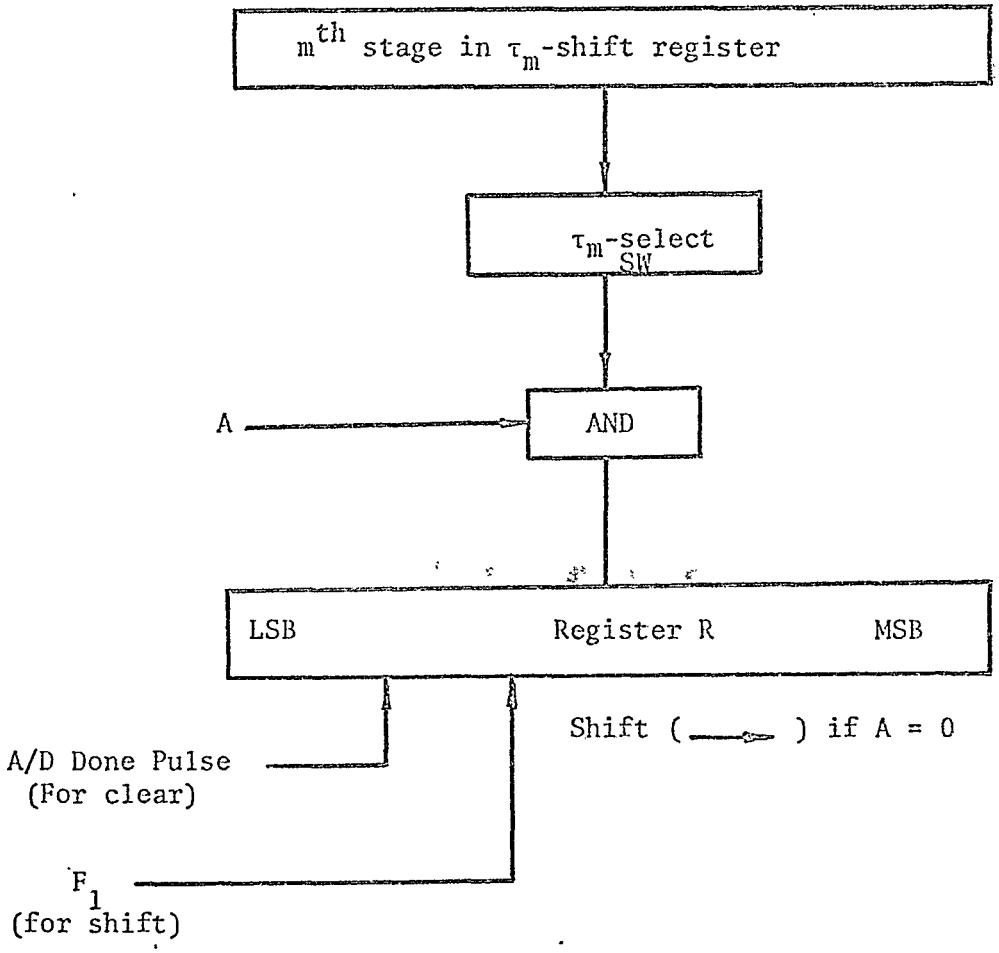


FIGURE 3.11 Register R and it's Input Logic

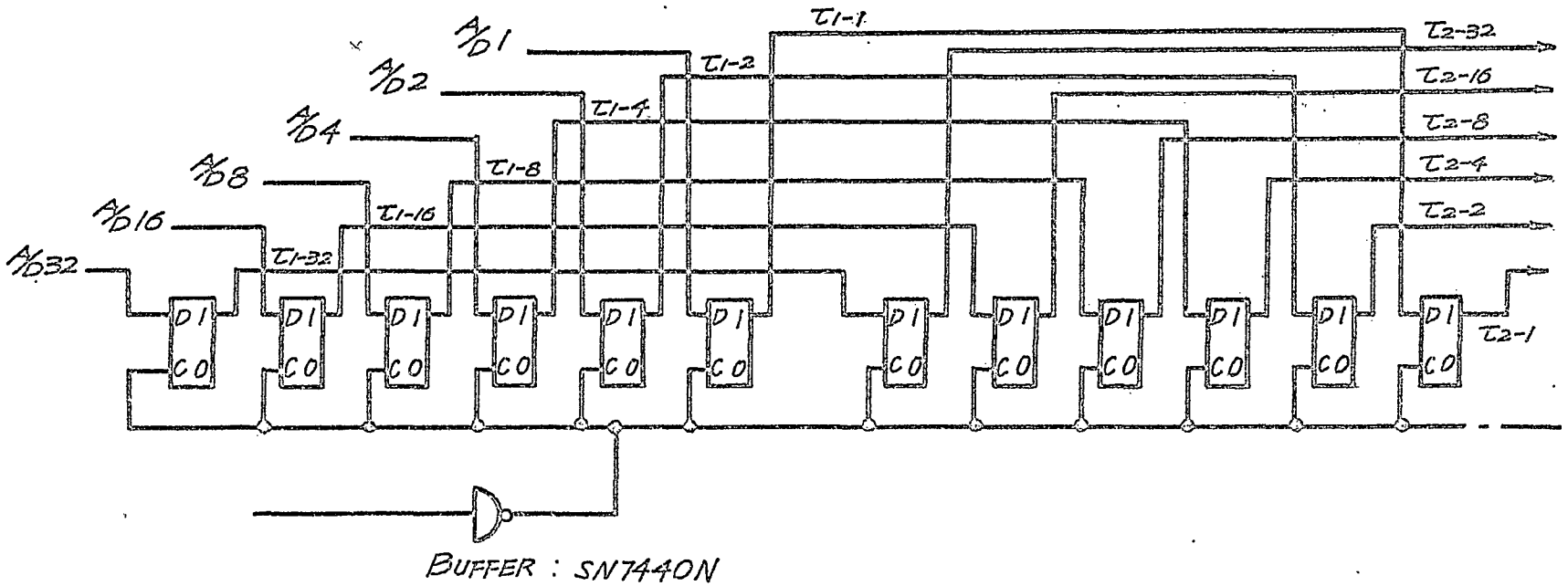


FIGURE 3.12 τ_m-Shift Register, where A/D 32, for example, represents the output from A/D converter and it is binary weighted by 32.

3.10 ACCUMULATOR:

The accumulator performs the following operations: -

1. Takes part in the computing logic.
2. Accumulates the number {r + s + 2rs}. C_o.
3. Divides by $2N^2$ (N = 64).
4. Feeds the most significant bit to the display counter.

In our system $2N^2 = 2^{13}$, hence 13 bits are required for the accumulator to perform operation 3. Since operations 1 and 2 are involved in the computing logic (section 3.2) the logic diagram is shown in Figure 3.7.

3.11 DECIMAL DISPLAY COUNTER:

A six decimal digit counter (H.P., Type 5216A) is used. In normal state it counts the output pulse from the most significant bit of the accumulator to provide the direct answer. If the frequency button (say) is ON it measures the frequency of the sampling pulse (i.e. it measures the sampling interval T).

3.12 C_o COUNTER:

A six decimal digit counter (DAWE Type 990 A/D) is used here. It counts the output pulse of the control flip-flop F_T (i.e. the number of sample pairs) and produces a positive-going pulse when it reaches C_o. This pulse is used to reset the master control flip-flop F_{DO} (see 3.13.1).

Due to the type of flip-flop used in the master control flip-flop a narrow pulse is required. Since the output pulse of C_o is not

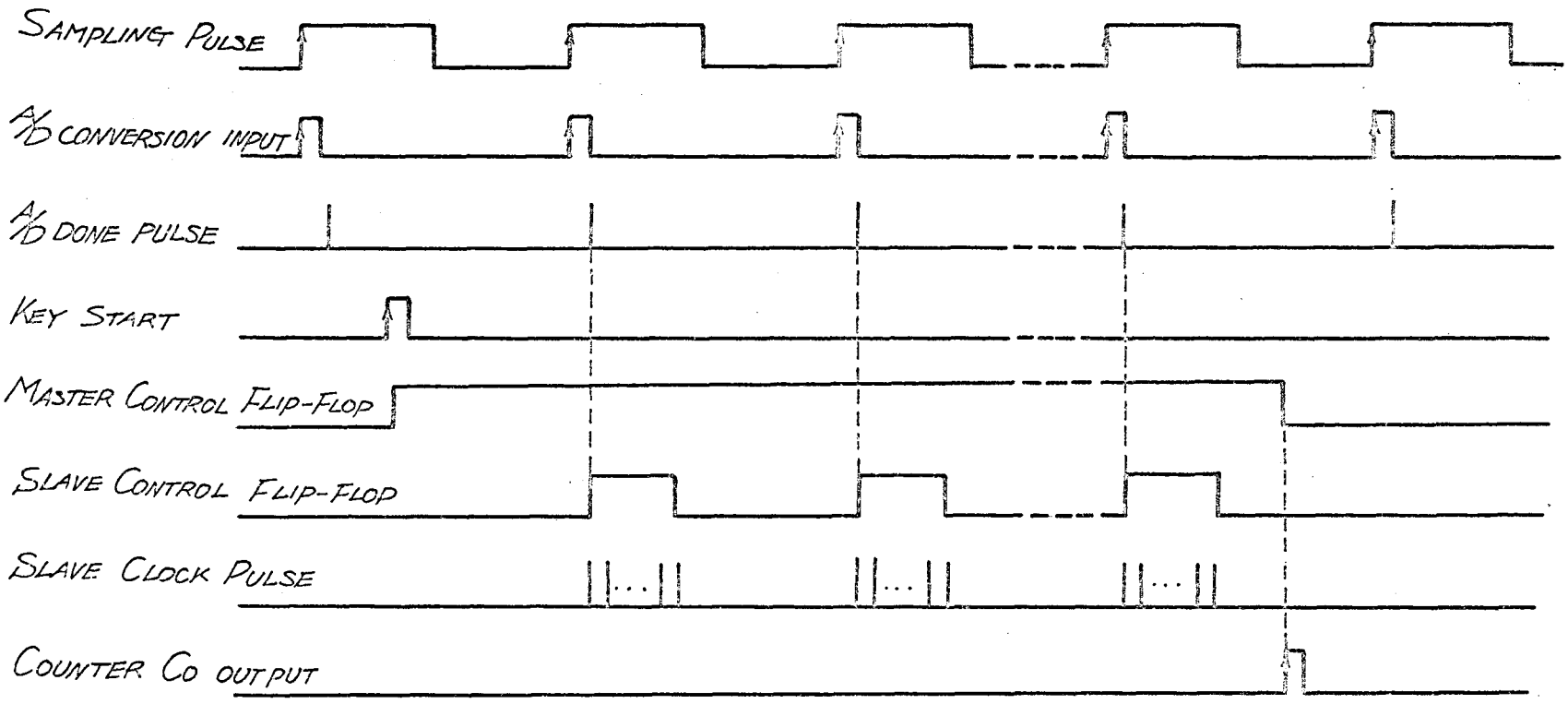


FIGURE 3.19 System Timing Chart

sufficiently narrow a one-shot circuit is connected to the output of C_o counter to produce the required narrow pulse when C_o operates.

3.13 CONTROL LOGIC:

The control section consists of: -

1. Master control flip-flop F_{DO}.
2. Slave control flip-flop F_T.
3. 4 bit-control counter F_1, F_2, F_4, F_8.
4. Gates to produce various timing pulses such as ADD.T, C_a.T, control pulses A, B, C, etc.

3.13.1 Master Control Flip-Flop F_{DO}

While F_{DO} = "1" we say that the process ON and all the required processes to obtain a correlation function for a selected value of τ_m are performed.

F_{DO} is set by only a key-start pulse and normally reset to "0" by pulse C_o. If the process is set to take an infinite number of samples instead of the finite number C_o, i.e., if Key-run = "1", pulse C_o is inhibited to reset F_{DO} and a Key-stop pulse resets F_{DO} to "0". Hence the input logic of F_{DO} becomes:

$$\text{Set } F_{DO} = \overline{\text{Key-start}}$$

$$\text{Reset } F_{DO} = \text{Key-clear} + \text{Key-stop} + C_o.\overline{\text{Key-run}}$$

Figure 3.13 shows the complete logic diagram of F_{DO} and its input gates.

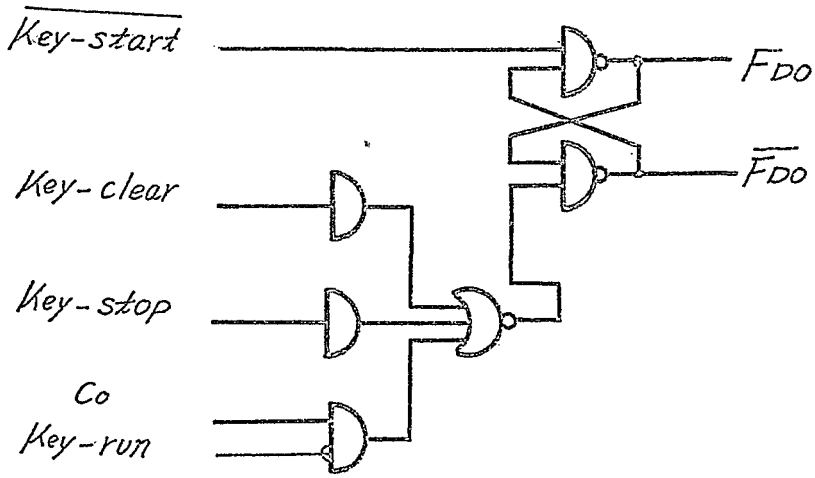


FIGURE 3.13 F_{DO} and the Input Gates

3.13.2 Slave Control Flip-Flop F_T

F_T has two objectives: -

(a) It initiates the slave clock generator for the computation {r+s+2rs}.

(b) It is fed to the C_o counter for counting the number of computations (i.e., the number of sample pairs).

Since each computation has to start after the A/D conversion has settled down, F_T is set by the AND gate F_{DO}.A/D done pulse. As seen in Figure 3.20 of computing timing chart F_T is reset to "0" when one computation is completed. T_{14} (say) which comes from the control counter is used for this purpose.

Each step of computing process (see the computing timing chart, Figure 3.20) is carried on by the trailing edge of slave clock pulse. Therefore F_T must also be reset by the trailing edge of T_{14}. Hence a J-K flip-flop (see Appendix 3) is used for F_T.

Suppose F_T is initially cleared and F_{DO}.A/D done pulse(a) comes first and T_{14}(b) comes next; then the OR gate of (a) and (b) will produce the same effect as setting F_T by (a) and resetting F_T by (b). The logic diagram is shown in Figure 3.14.

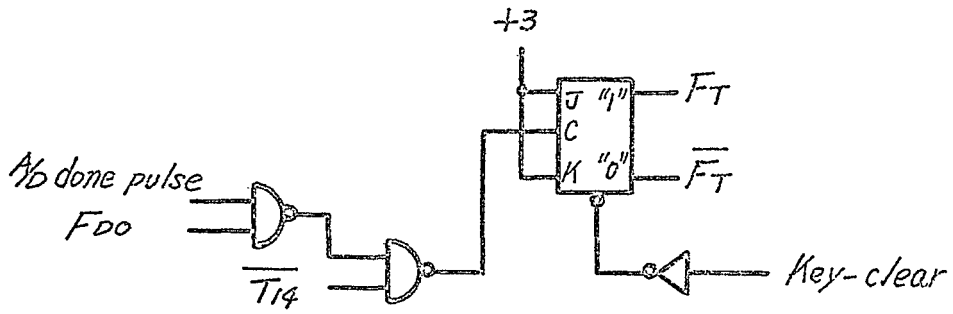


FIGURE 3.14 Slave Control Flip-Flop

3.13.3 4-Bit Control Counter

This consists of 4 flip-flops; F_1, F_2, F_4, F_8. The operations associated with this counter are listed below:

1. Starts counting the inverted slave clock pulse $\overline{CP}$.
2. Stops counting and is reset by slave control flip-flop $\bar{F}_T$.
3. Provides the control pulse A, B, C, and T_{14}.
4. The first flip-flop F_1 is used as shift pulse of registers S and R respectively.
5. F_1 is also used as the timing pulse of ADD.T and C_a.T.

Two types of counters are considered:

(a) One is a synchronous type and the other is an asynchronous type. Each bit of the first type is exactly synchronized with clock pulse. Hence the overall propagation delay is equal to a single propagation delay of a flip-flop.

(b) For the second type, the flip-flop are connected in cascade and the clock pulse is fed to the first stage only. Therefore the overall propagation delay increases linearly with the number of flip-flops connected. The timing pulses recorded from this type of counter are accompanied with some undesired spikes and causes trouble, especially when the system requires many timing pulses.

The second type of counter is used in our system because only four timing pulses viz., control A, B, C and T_{14} are used.

The possibility of interference due to the spikes are investigated now. For convenience let all the possible timing pulses from the counter be named T_0, T_1, T_2, ..., etc. (see Figure 3.15). From the system requirement (see Figure 3.20) the control pulses can be decoded as follows:

$$\text{Control pulse A} = T_0 + T_1 = \bar{F}_1\bar{F}_2\bar{F}_4\bar{F}_8 + F_1\bar{F}_2\bar{F}_4\bar{F}_8 = \bar{F}_2\bar{F}_4\bar{F}_8$$

$$\text{Control pulse B} = T_0 + T_1 + T_2 = \bar{F}_2.\bar{F}_4.\bar{F}_8 + \bar{F}_1F_2\bar{F}_4\bar{F}_8$$

$$\text{Control pulse C} = T_2 + T_3 = \bar{F}_1F_2\bar{F}_4\bar{F}_8 + F_1F_2\bar{F}_4\bar{F}_8 = F_2\bar{F}_4\bar{F}_8$$

$$T_{14} = F_2.F_4.F_8.CP$$

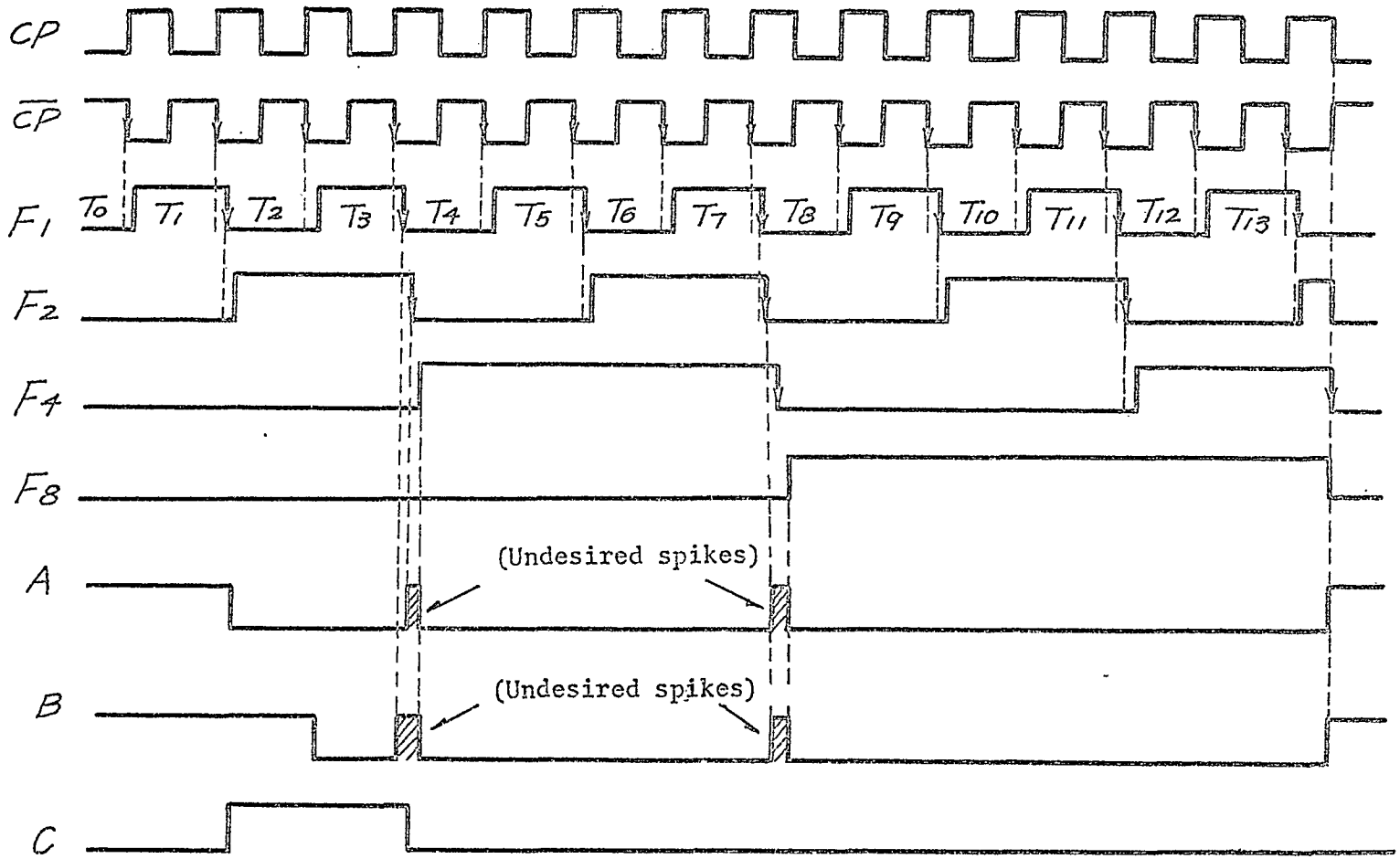


FIGURE 3.15 Undesired Spikes Due to the Propagation Delay of Flip-Flop

Actual wave forms due to the propagation delay are shown in Figure 3.15. Note that the computing timing chart of Figure 3.20 is an idealized diagram.

Control pulse A is accompanied by two spikes, but as described in Section 3.6, A is significant only at the instant of leading edge of F_1 in the practical process. Hence these spikes do not affect the process.

Control pulse B has also two spikes (one is between T_3 and T_4 and other is between T_7 and T_8). B is used to distinguish the processes {r → Acc} and {{2r+1)s → Acc} and each computing step is performed by the trailing edge of slave clock pulse CP (see the computing timing chart of Figure 3.20). Therefore these spikes are still far away from the trailing edge of CP and have no effect on the process. Control pulse C has no spike at all because of the logic combination.

Thus we can use the asynchronous counter quite safely in our system. Figure 3.16 shows the logic diagram of this counter.

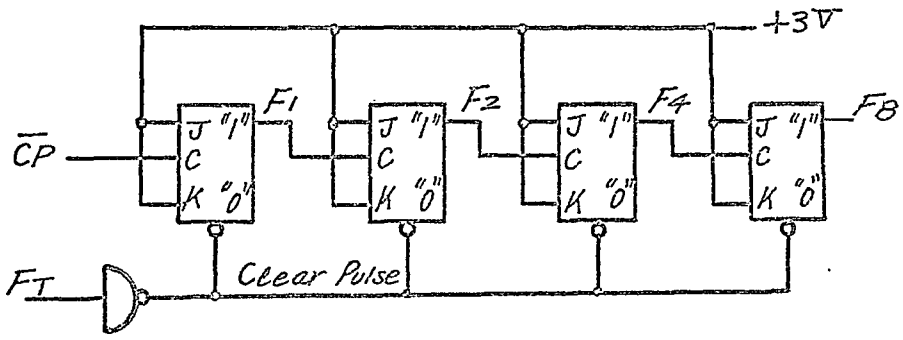


FIGURE 3.16 Control Counter (Asynchronous Type)

3.13.4 ADD.T and C_a.T

As mentioned in the previous section F_1 and $\bar{F}_1$ are used for timing of ADD.T and C_a.T respectively. Control pulse B distinguishes step 1 and step 3 described in Section 3.2.1. Hence we define the performance of B as follows:

(a) if B = "1" perform step 1.

(b) If B = "0" perform step 3.

Therefore, the logics of ADD.T and C_a.T becomes

$$ADD.T = \bar{B}.S.F_1 + B.F_1 = \overline{\overline{\bar{B}.S + B} + \bar{F}_1}$$

$$C_a.T = \bar{B}.S.\bar{F}_1 + B.\bar{F}_1 = \overline{\overline{\bar{B}.S.} + B} + F_1$$

Figure 3.17 shows the logic diagram.

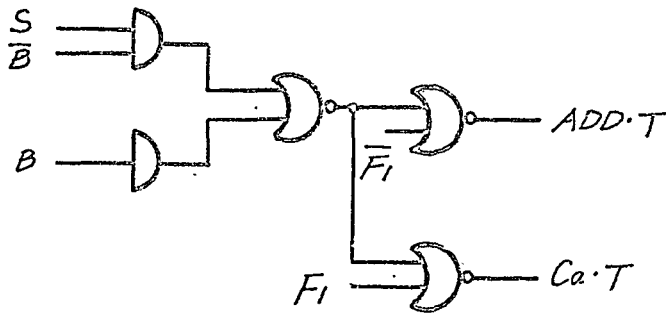


FIGURE 3.17 ADD.T and C_a.T

3.14 EXPLANATION OF ACTUAL PROCESS:

We have seen how each block operates and how it is designed. This section describes a complete process in detail to obtain one correlation function for a selected time difference τ_m. Figure 3.18 shows a complete flow of the whole process. System timing chart in Fig. 3.19

and computing timing chart in Figure 3.20 help to understand how each logic is correlated to another during the computation.

3.14.1 Power Is ON

Sampling pulses are generated and A/D conversion is carried on at every leading edge of sampling pulse. At the same time the previously quantized information (sample r) in A/D converter is transferred to the τ_m-shift register and shifted toward the m^{th} stage to make time difference $\tau_m \simeq mT$.

3.14.2 Set the sampling interval T at the desired length.

3.14.3 Set the τ_m-select switch at the desired m.

3.14.4 Push Key-clear button;

All the blocks such as the accumulator, control flip-flops, display counters, counter C_o, etc. are cleared and become ready to start.

3.14.5 Push Key-start button;

This starts the following sequency of operations: -

(a) F_{DO} is set to "1" (i.e., process is ON).

(b) F_T is set to "1" at the trailing edge of AND gate F_{DO}.A/D done pulse.

(c) Slave clock generator is initiated by the negative going pulse of $\bar{F}_T$ and produces 14 slave clock pulses.

(d) Control counter which was cleared by $\bar{F}_T$ starts counting the inverted slave clock pulse $\overline{CP}$.

(e) Since control pulse A is "1" the quantized information in A/D converter is transferred to register S (this information is called s) and at the same time another information (sample r) in the m^{th} stage

of τ_m-shift register is transferred to register R at the leading edge of the first F_1.

(f) Since control pulse B is "1" the 1st and 2nd CP carry on the addition {R(r) + Acc} with the aid of the first F_1 and $\bar{F}_1$. Note that the arrows in Figures 3.19 and 3.20 show the exact timing of each operation.

(g) When control pulse A = "0" and C = "1" the positive going pulse of 2nd F_1 shifts the sample s in register S to the lesser significant bit and sample r in register R to the upper significant bit. Hence S_{2^0} appears at the least significant bit of register S and r becomes (2r + 1) in register R, where S_{2^0} represents the binary signal which is weighted by $2^0 = 1$.

(h) When control pulse B is "0" the 3rd and 4th CP carry the addition,

$$R(2r+1).S_{2^0} + Acc = \begin{cases} R(2r+1) + Acc & \text{if } S_{2^0} = \text{"1"} \\ 0 + Acc & \text{if } S_{2^0} = \text{"0"} \end{cases}$$

(i) Since still A = "0" the positive going pulse of 3rd F_1 shifts sample s in register S to the lesser significant bit and also shifts the number (2r+1) in register R to the upper significant bit. Therefore, S_{2^1} appears at the least significant bit (LSB) of register S and the number (2r+1) in register R becomes (2r+1).2, where S_{2^1} represents a binary signal weighted by $2^1 = 2$.

(j) Since still B = "0" the 5th and 6th CP perform the addition,

$$R\{(2r+1).2\}.S_{2^1} + Acc = \begin{cases} R\{(2r+1).2\} + Acc & \text{if } S_{2^1} = \text{"1"} \\ 0 + Acc & \text{if } S_{2^1} = \text{"0"} \end{cases}$$

(k) Perform the same processes as (h) to (j) for the further additions;

$$R\{(2r+1).4\}.S_{2^2} + Acc$$

$$R\{(2r+1).8\}.S_{2^3} + Acc$$

$$\vdots$$

etc.

until the final addition $R\{(2r+1).32\}.S_{2^5} + Acc$ is completed.

(l) Since the trailing edge of the 14th CP is the exact time of the completion of $\{r + s + 2rs\}$. T_{14} is generated and it resets F_T by the trailing edge.

(m) Feed F_T to the C_o counter.

(n) Repeat (b) to (m) until counter C_o reaches C_o.

(o) F_{DO} is reset to "0" and the whole process is finished. Therefore the correlation function for a selected value of τ_m is now displayed in the display counter.

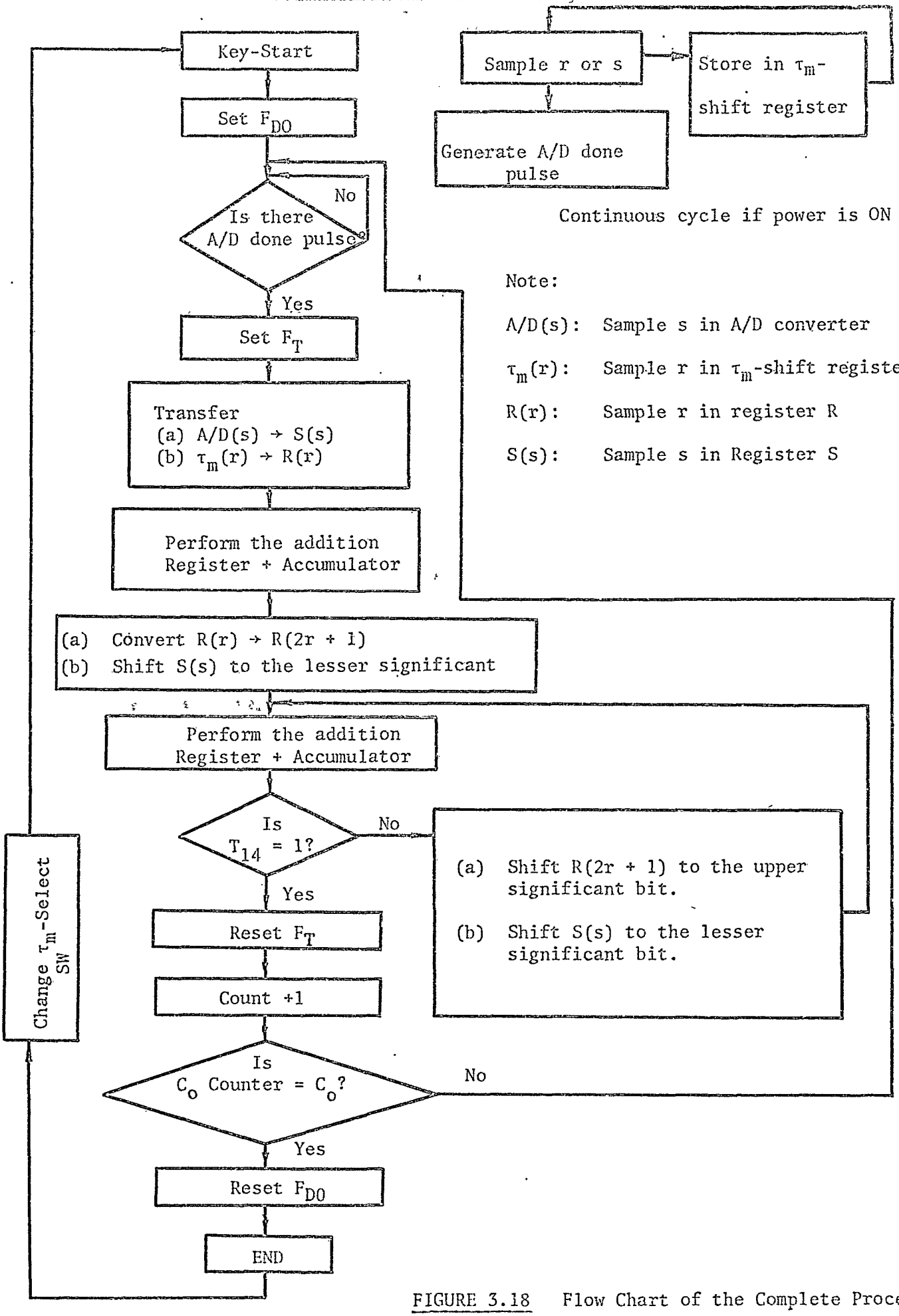


FIGURE 3.18 Flow Chart of the Complete Processes

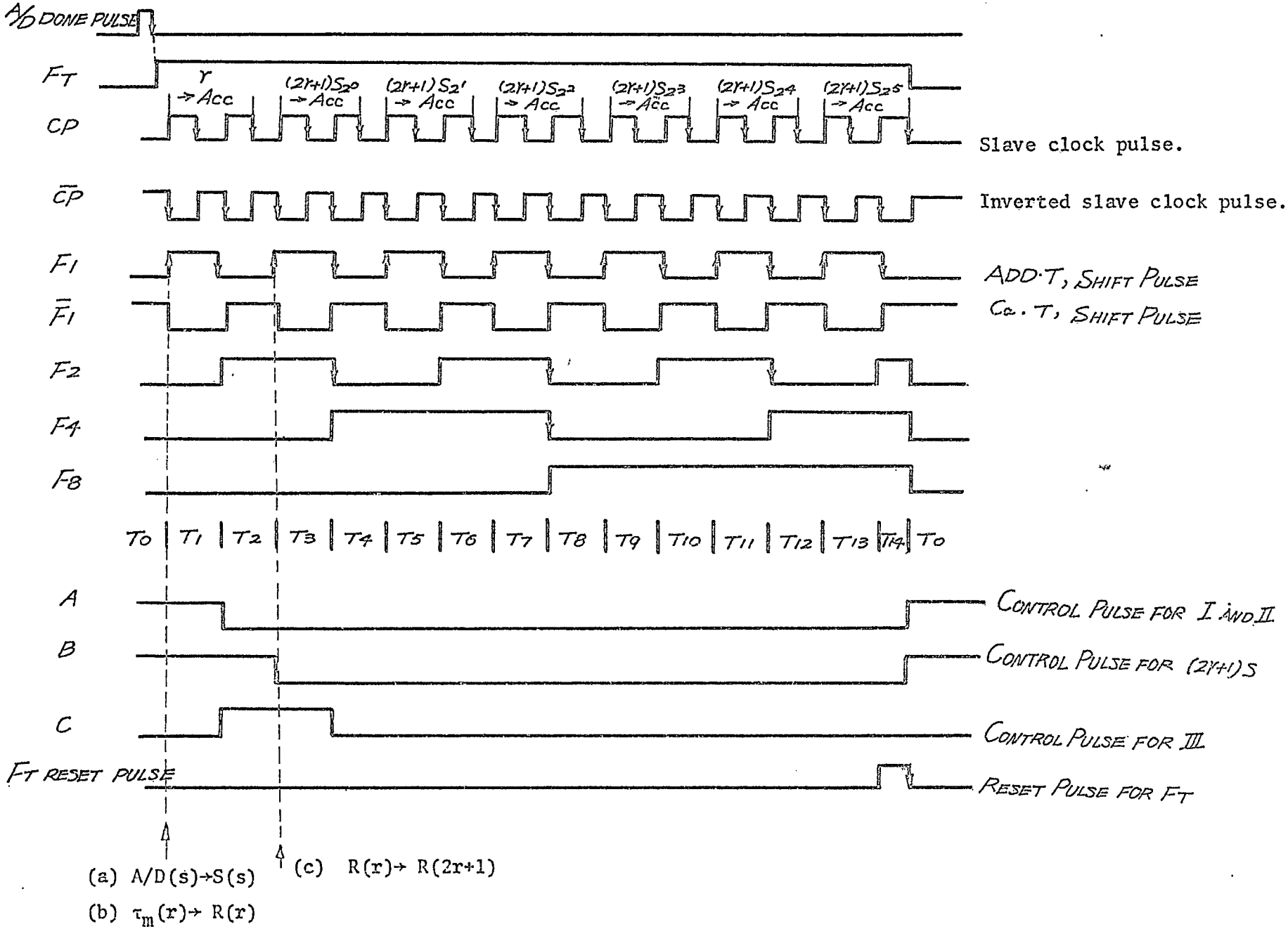


FIGURE 3.20 Computing Timing Chart

CHAPTER IV

CONSTRUCTION

This chapter describes the location of the I.C. units. In the first section the location of each board in the rack is classified and the major functinn associated with the board is given. The second section describes the locations of the units on the board and the input or output positions of the signals on the board connector.

The type of the integrated-circuits are summarized in Appendix 3.

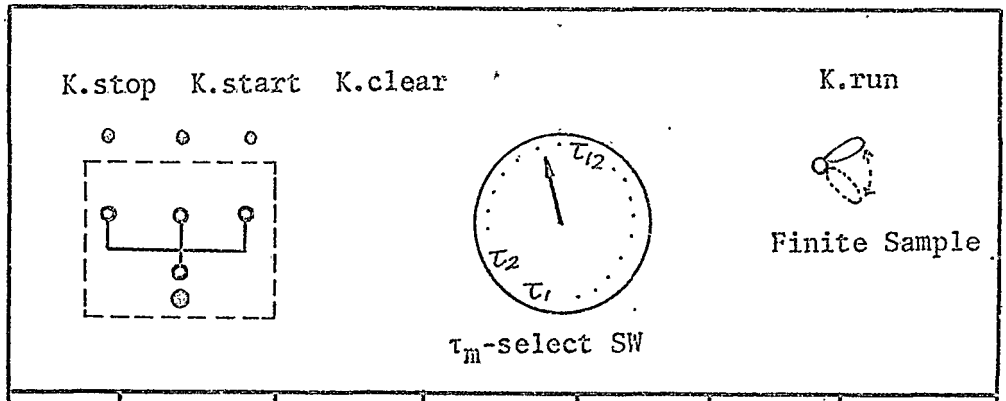
4.1 LOCATION OF BOARD AND SWITCH:

Figure 4.1 shows the location of the board by a letter and a number. For example B4 represents the 4^{th} board from the left side in the middle section. The positions of the connector pins are given after the above number such as B4 - C or B4 - 12 (see Figure 4.2).

The functions of the boards are tabulated in Table 4.1. A dotted line on the panel in Figure 4.1 shows an external connection for the desired range of sampling frequency in case the sampling pulse is taken from the internal generator. The internal sampling generator has a range 1 KHz to 50 KHz. Therefore if a wider range is required an external generator must be connected directly to B5 - 9.

4.2 LOCATION OF THE I.C UNIT:

Two types of boards are used. One has the capacity to mount up to twenty units and the other up to nine units. In Figure 4.3 the letter



A1	A2	A3	A4	A5	A6	A7
B1	B2	B3	B4	B5		
C1	C2	+5V Power Supply				

FIGURE 4.1 Location of Board and Switch

A1	Accumulator	B1	Switch Filter
A2	Computing Logic (a)	B2	Register S
A3	Computing Logic (b)	B3	Control Logic (a)
A4	Register R	B4	Control Logic (b)
A5	τ_m-shift Register (a)	B5	Control Logic (c)
A6	τ_m-shift Register (b)	C1	A/D Converter
A7	τ_m-shift Register (c)	C2	Slave Clock Generator

TABLE 4.1 Function of Board

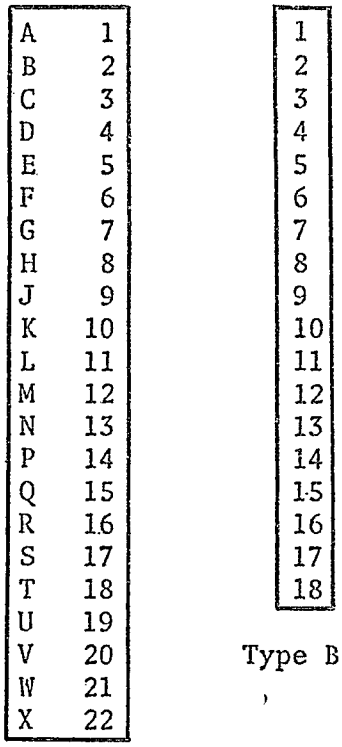
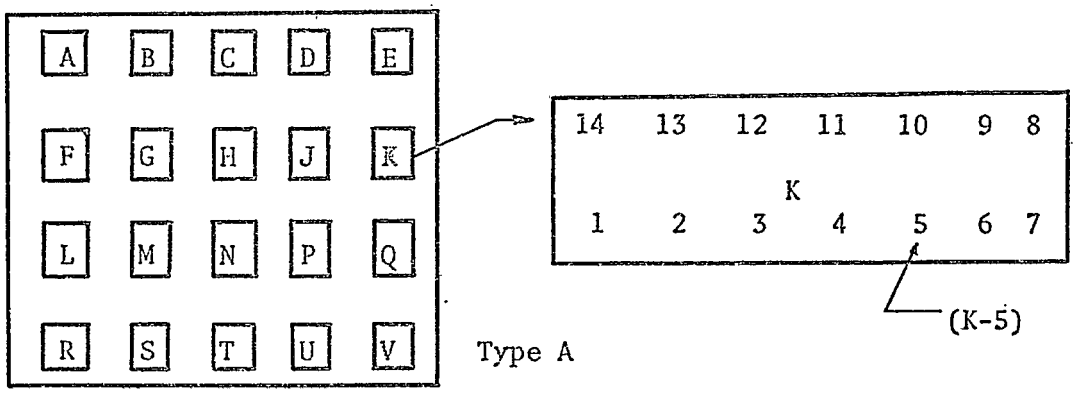


FIGURE 4.2 Board Connectors, Type A and Type B, where power supplies are connected to the following pins:

Type A: +5V pin A
G_{ND} pin X

Type B: +5V pin 1
G_{ND} pin 18



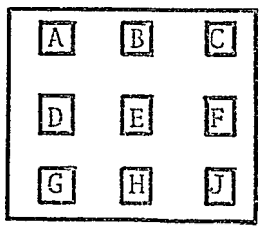
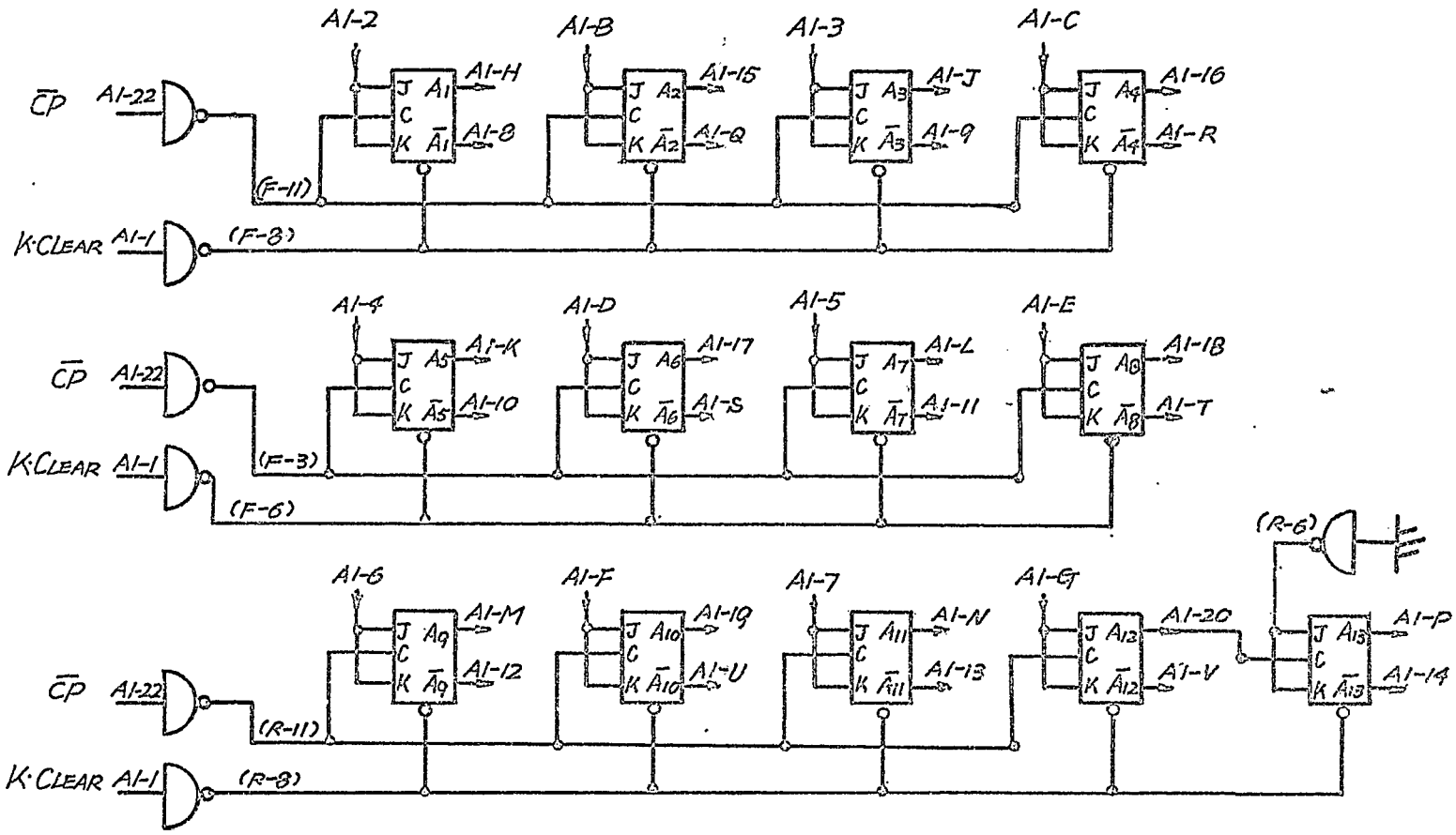


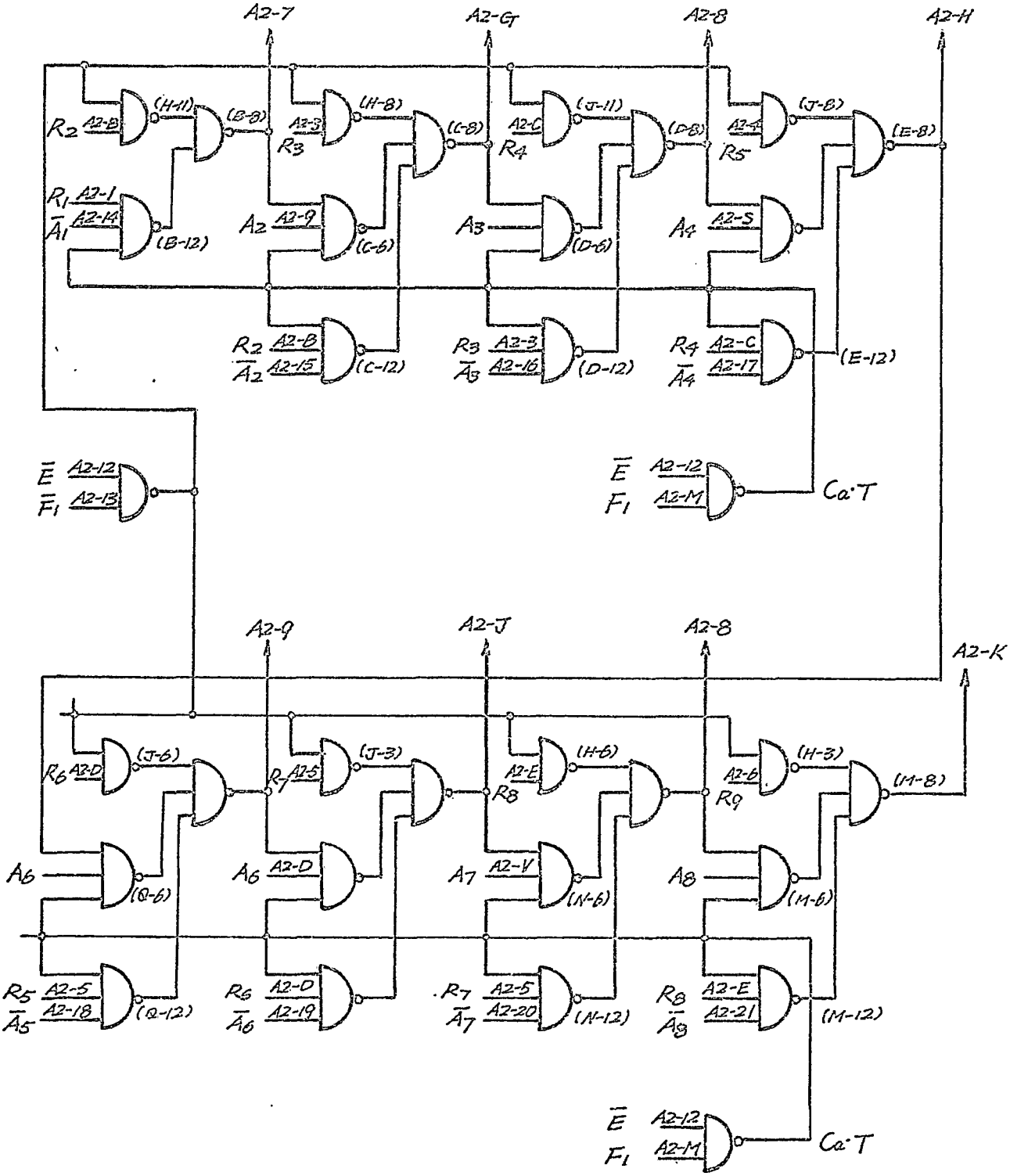
FIGURE 4.3 Locations of I.C Units on the Board

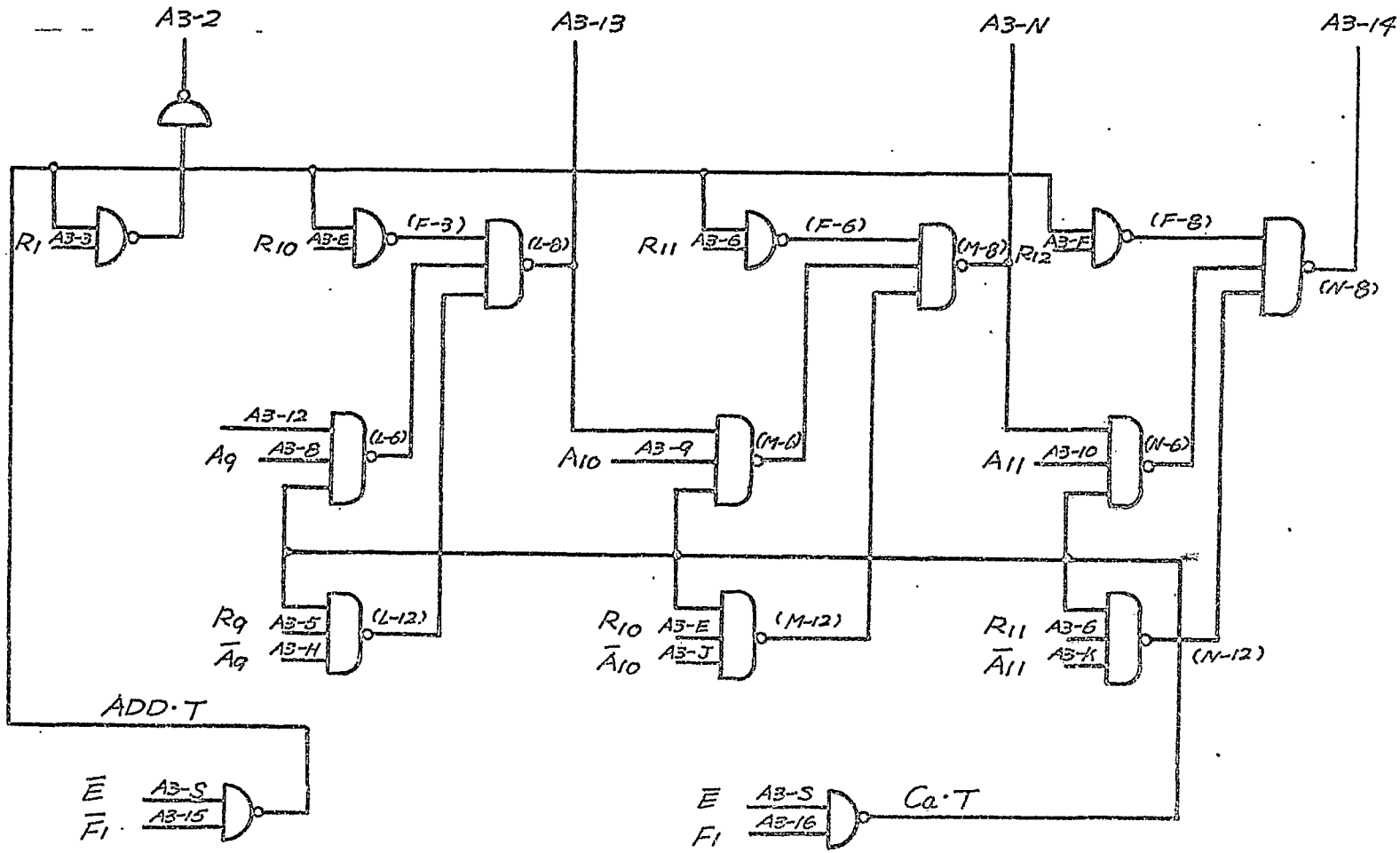
represents the location of the unit on the board and the number shows the terminal. Both of them are always accompanied by parenthesis to distinguish from the board number described in the previous section. For example, (F-9) means that the I.C unit is located at position F on the board and the terminal number is 9 (see Figure 4.3).



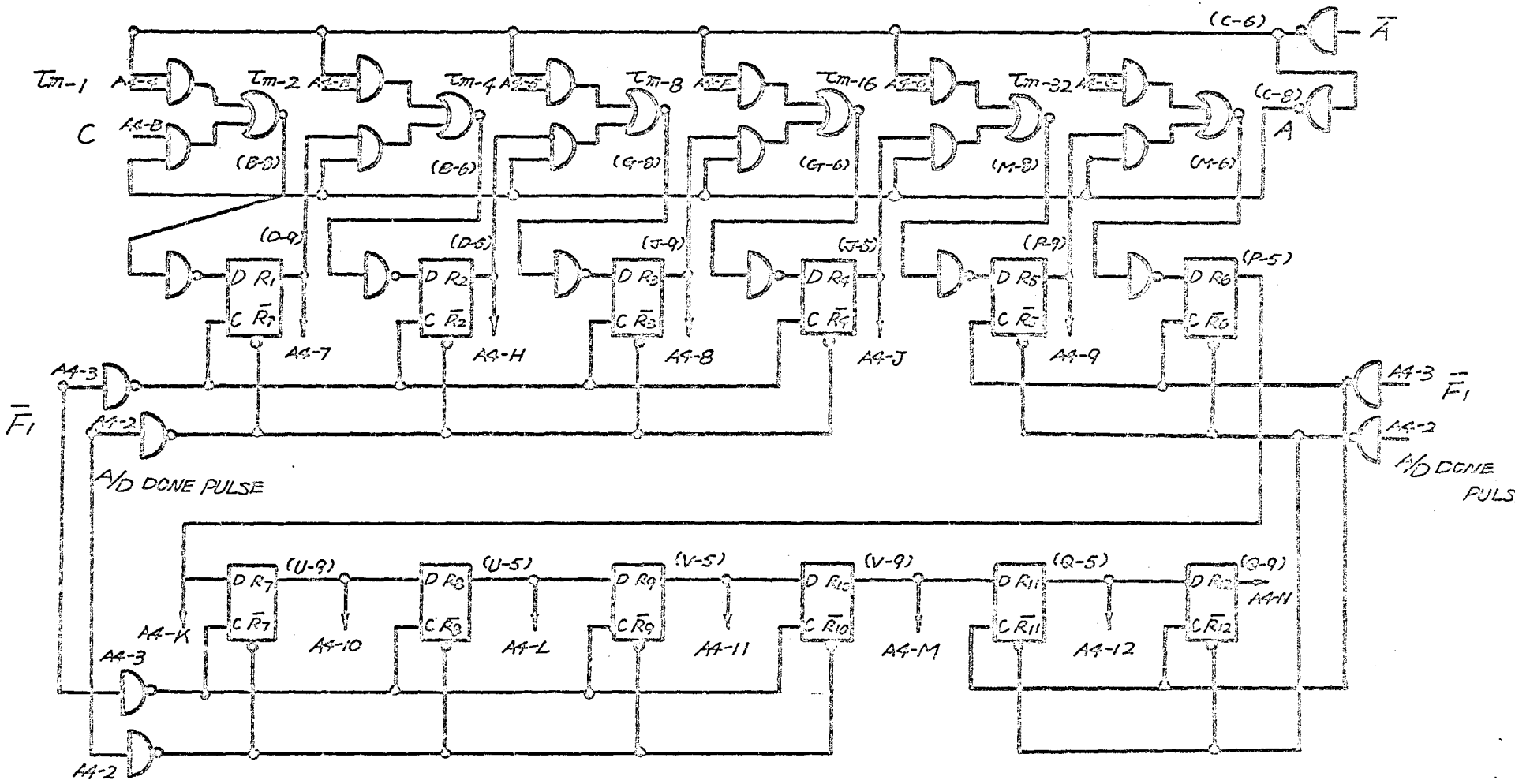
4.2.1 ACCUMULATOR -A1-

4.2.2 COMPUTING LOGIC (a) - A2 -



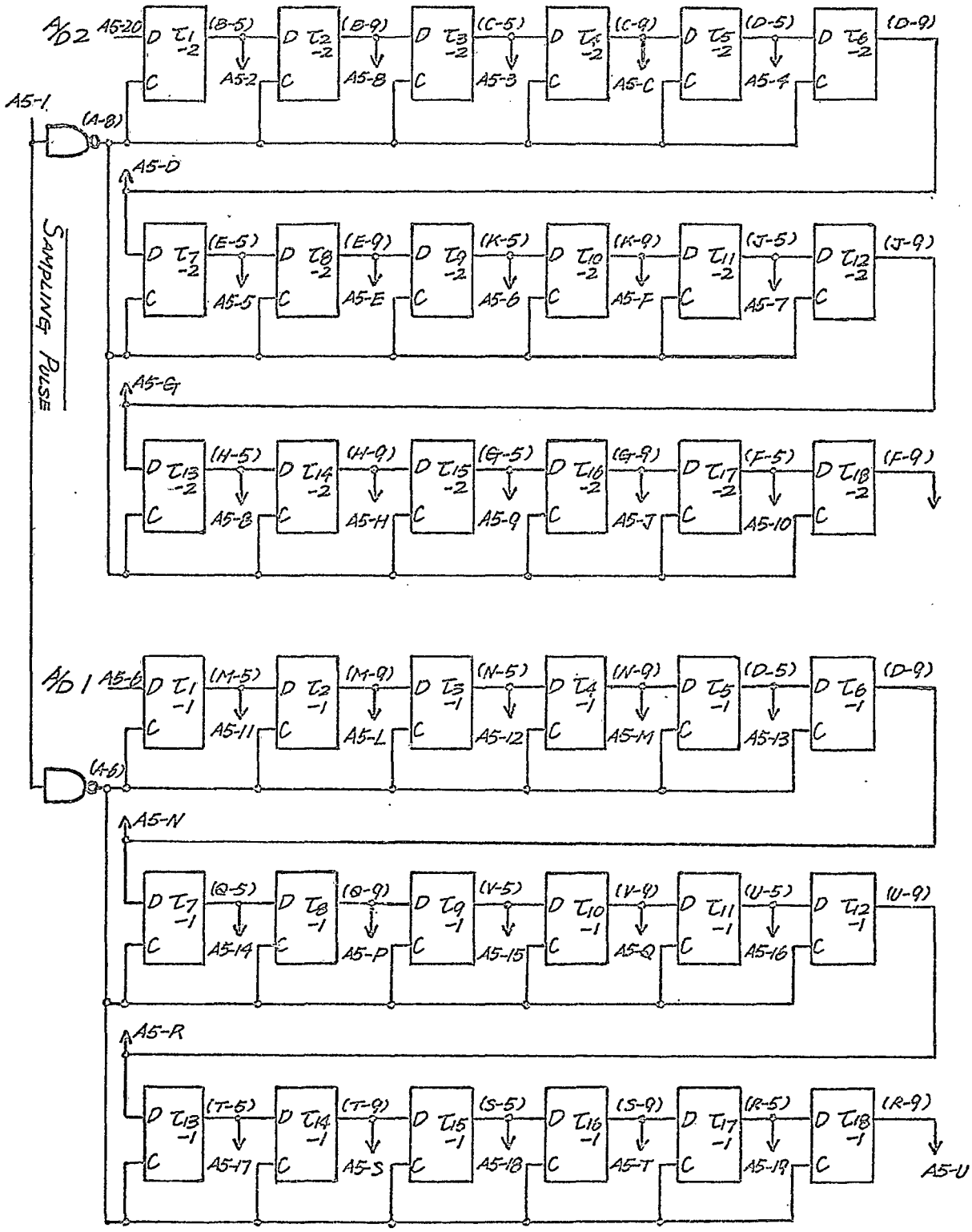


4.2.3 COMPUTING LOGIC (b) - A3 -

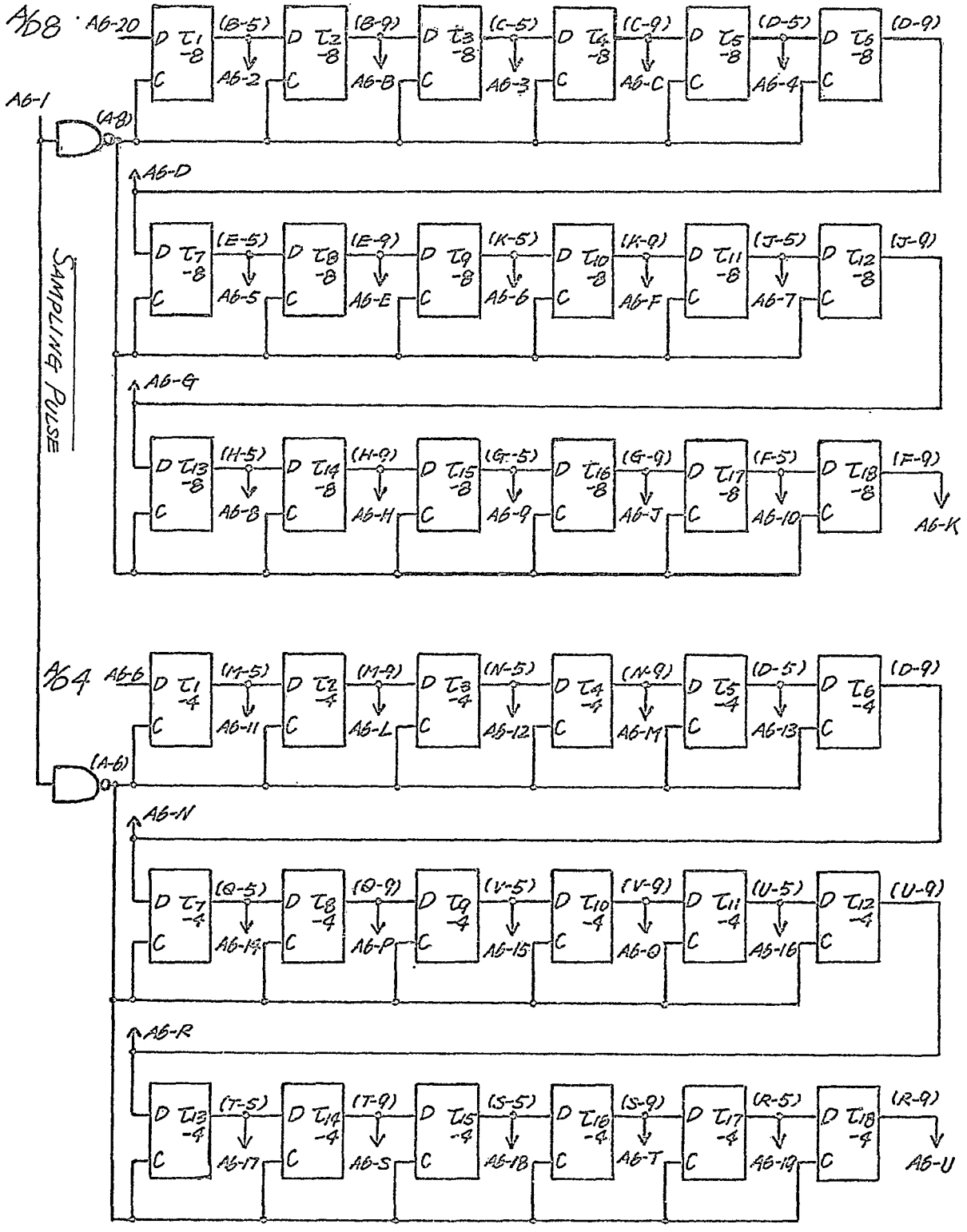


4.2.4 REGISTER R - A4 -

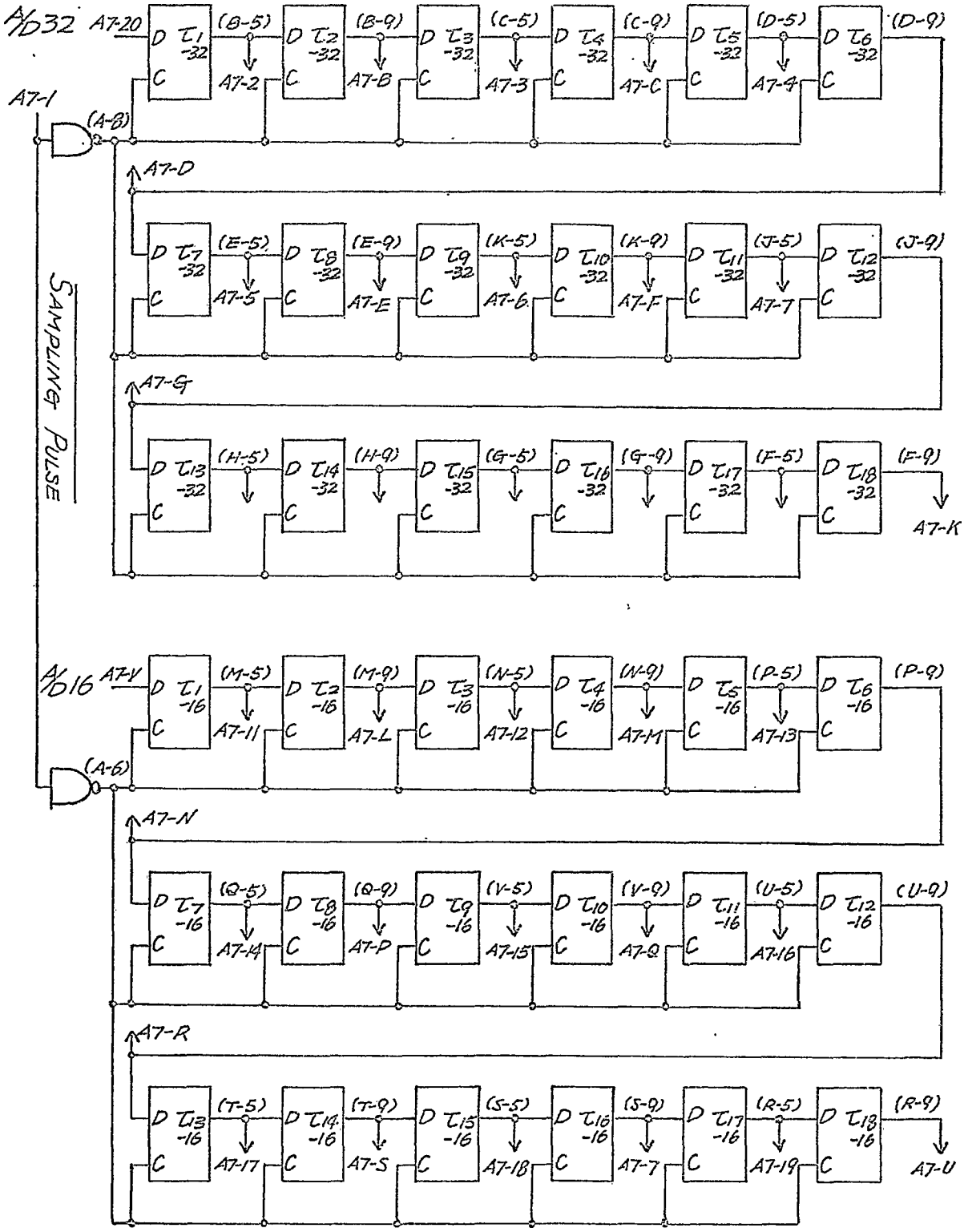
4.2.5 τ_m-shift Register (a) - A5 -



4.2.6 τ_m-SHIFT REGISTER (b) -A6-

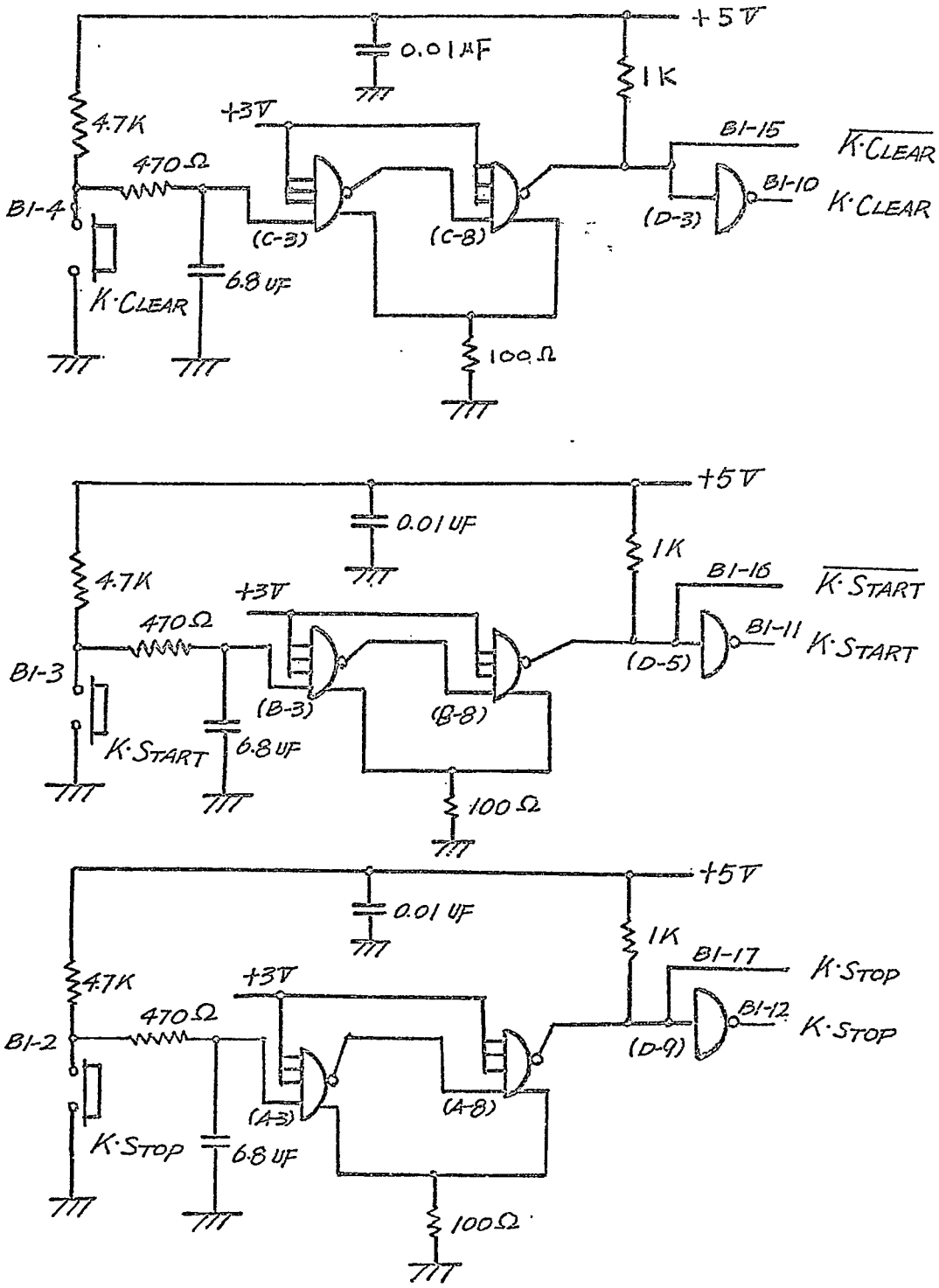


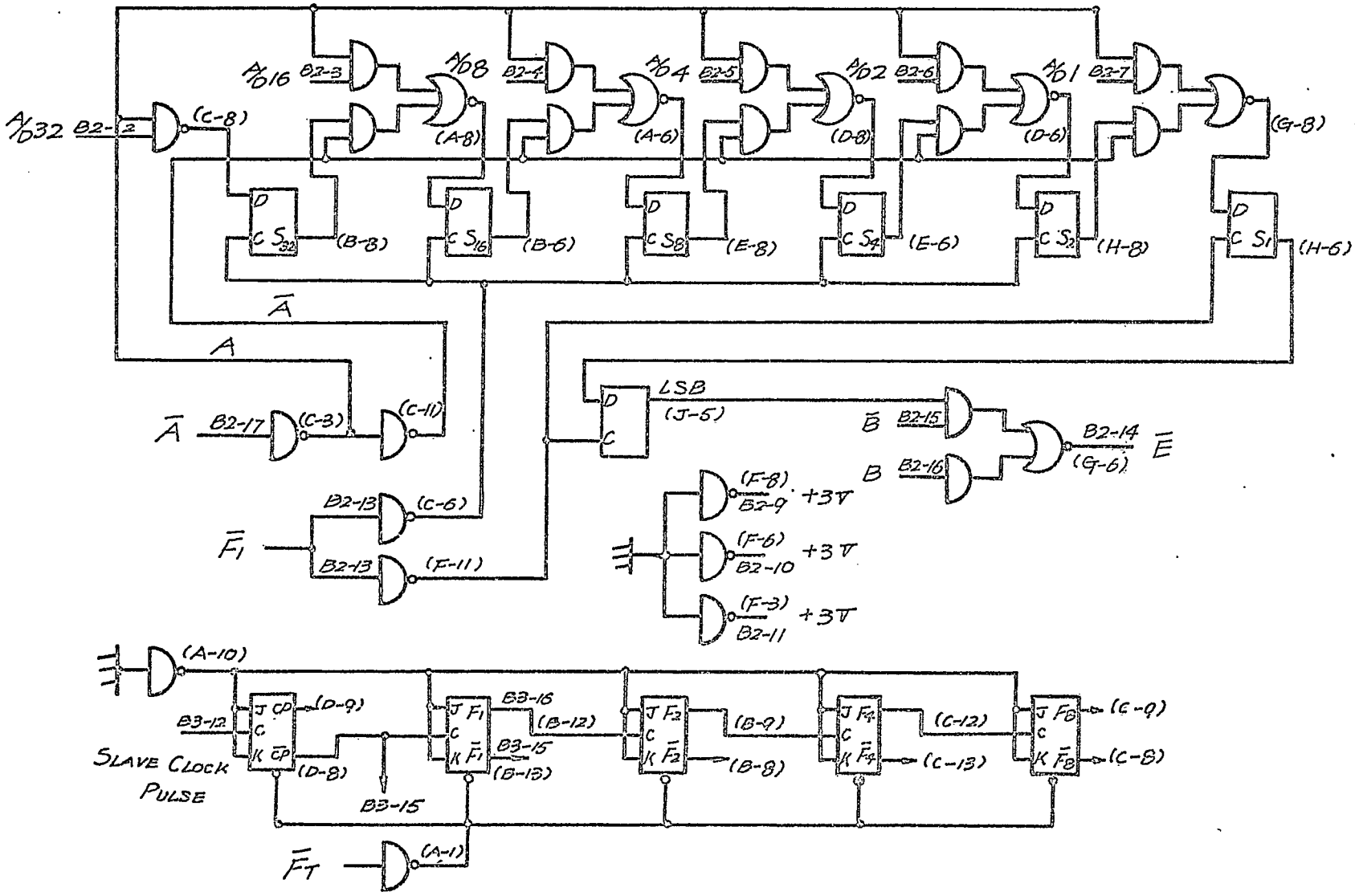
4.2.7 τ_m-SHIFT REGISTER (c) - A7 -



4.2.8 SWITCH FILTER -B1 -

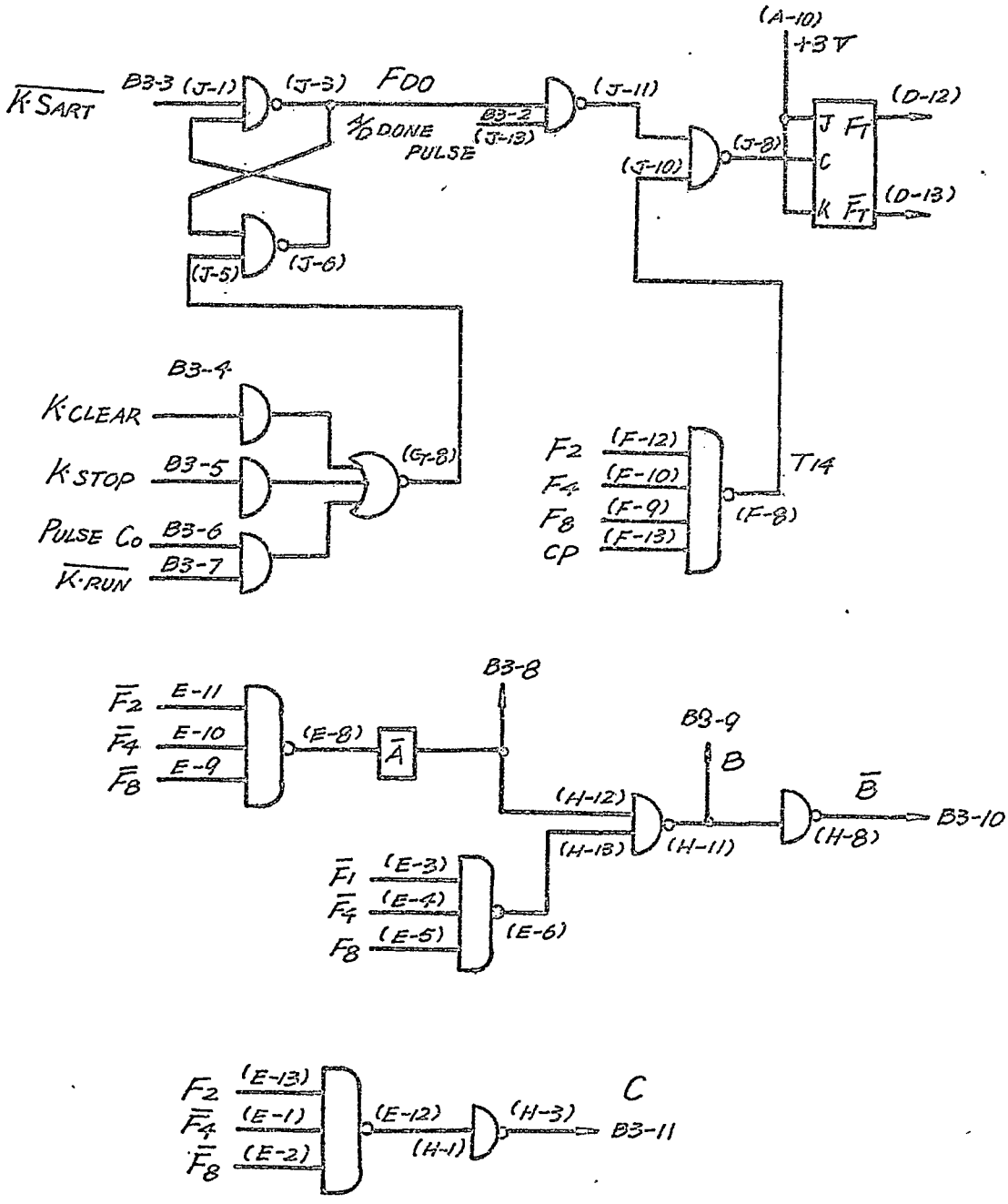
The purpose of this filter is to absorb the jitter due to the mechanical operation of the switch and to decrease the rise and fall time.





4.2.9 REGISTER S - B2 -

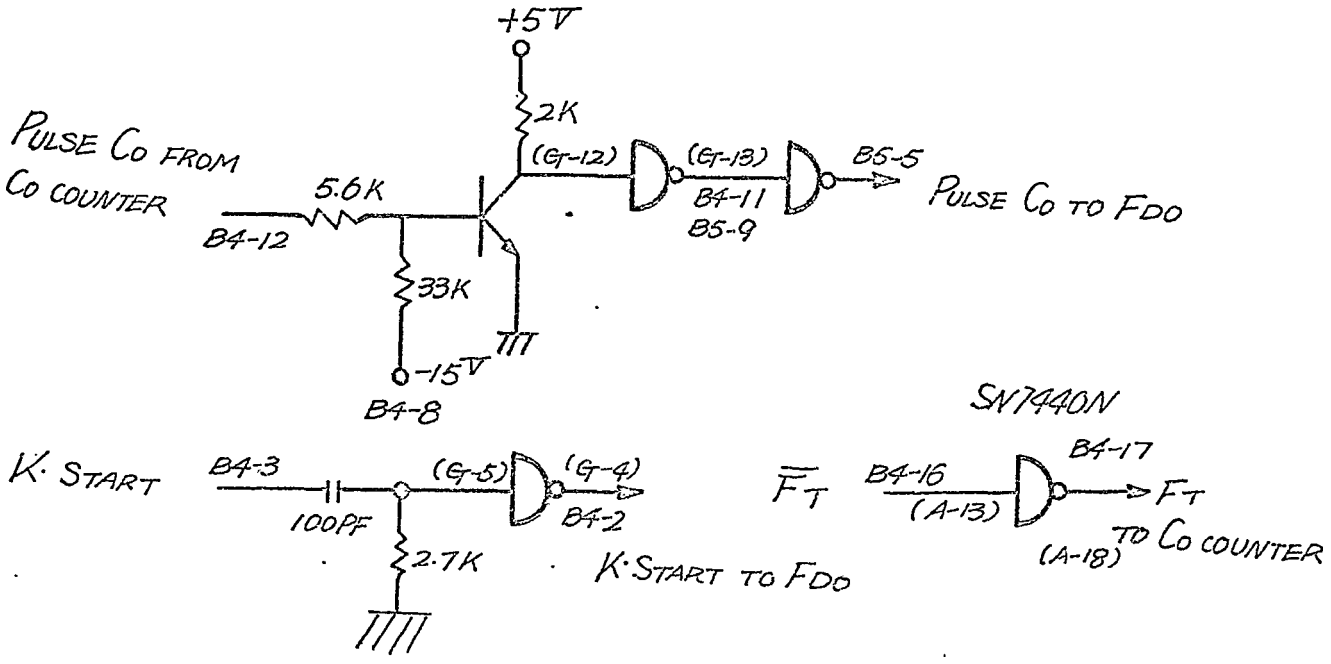
4.2.10 CONTROL LOGIC (a) - B3 -



4.2.11 CONTROL LOGIC (b) - B4 -

Three operations are performed on this board:

1. F_T is buffered by type SN7440N before going into the C_o counter.
2. Output of the C_o counter (DAWE Type 990 A/D) changes from 1.0V to 6V. Therefore before going into master control flip-flop F_{DO} it is converted into the pulse with level change from 0V to +3V by a NPN transistor type GE 2N2924.
3. Because of the type of flip-flop used for F_{DO} K-start pulse must always be narrow regardless of the time of pushing the Key. A one-shot circuit using open collector NAND gate (SN7401N) is used for this purpose.



4.2.12 CONTROL LOGIC (c) -B5 -

This board has three major units as follows:

1. A one-shot circuit to produce a narrow pulse from the sampling pulse (see Section 3.5).
2. A sampling pulse generator (see Section 3.3).
3. A delay circuit to delay slave clock pulse CP before it goes to the accumulator to perform the computation.

All these units are constructed by using the open collector NAND gates type SN7401N. The purpose of the delay circuit is explained below.

As seen in Figure 3.20, odd numbered CP's (CP-ODD, say) carry on the one stage of addition according to ADD.T and the even numbered CP's (CP-EVEN, say) perform the other stage of addition according to C_a.T. The trouble is that if there is some length (roughly more than 30 nseconds) of overlapped time between CP-ODD and C_a.T = "1", or CP-EVEN and ADD.T = "1" (see Figure 4.4) then CP-ODD carry on the computation regardless of ADD.T and so does CP-EVEN regardless of C_a.T. Actually this trouble occurred in our system due to the propagation delay of ADD.T and C_a.T. Therefore CP is delayed slightly before going to the accumulator.

With the value of RC coupling shown in the next page a delay time of 100 nseconds is obtained, which assures computation to be performed safely.

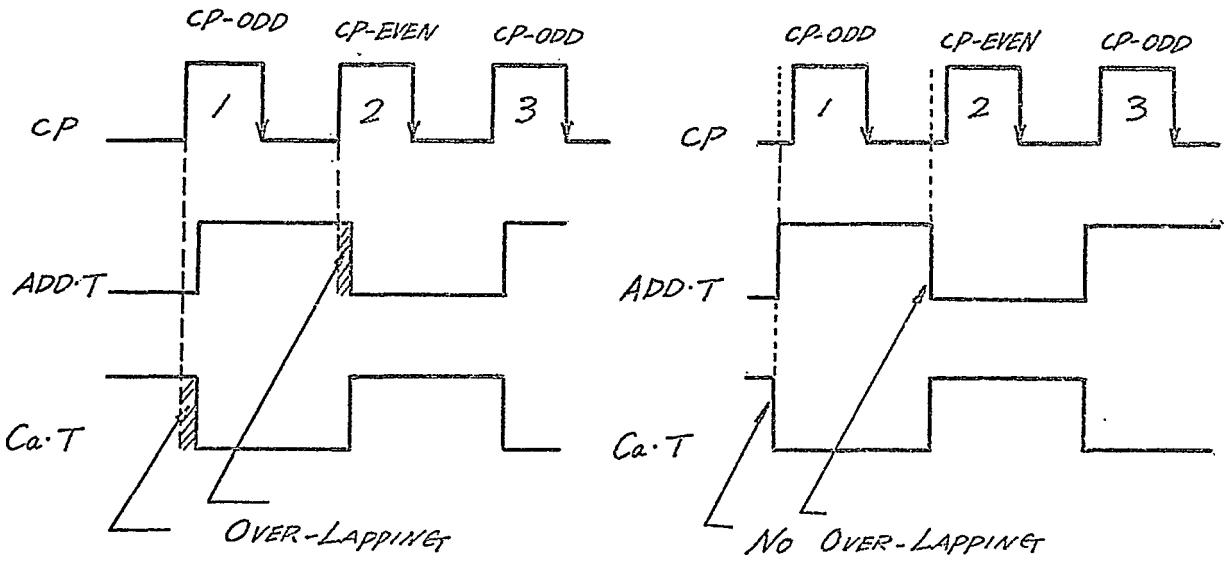
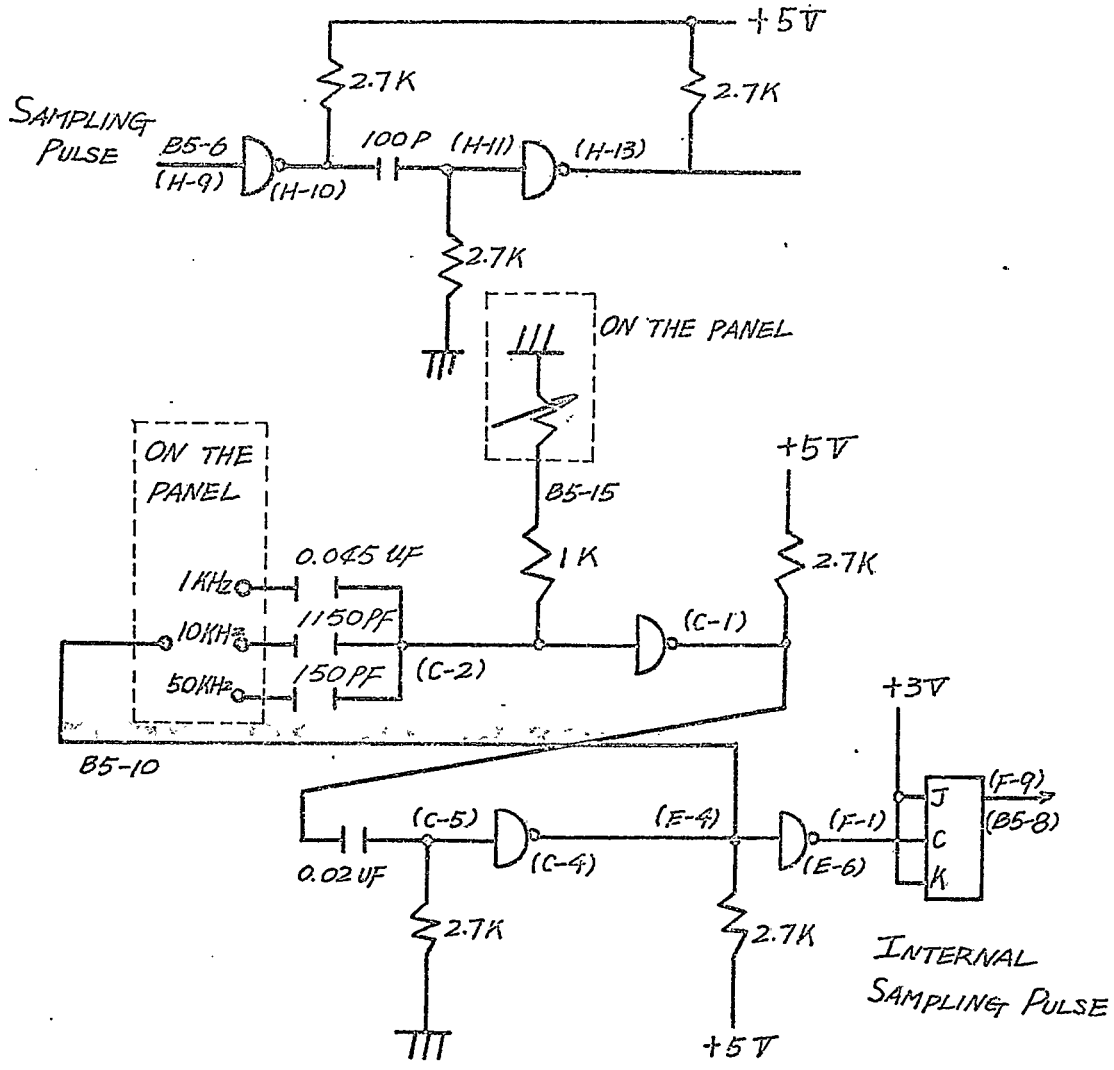
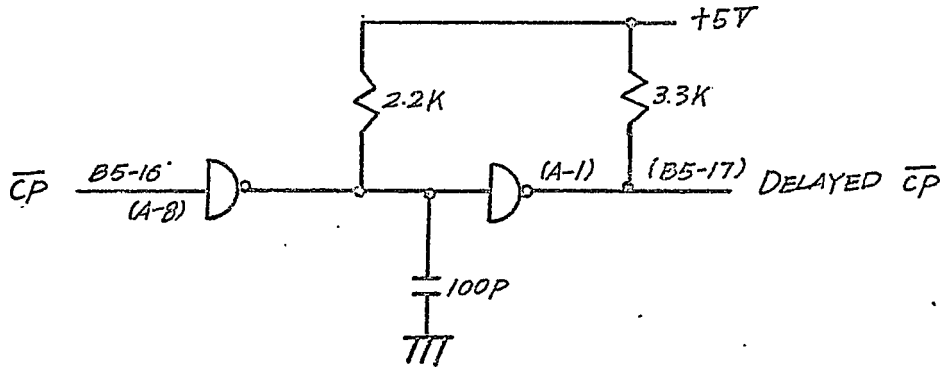


FIGURE 4.4 Overlapping Due to Propogation Delay


+5V
2.7K
2.7K
SAMPLING PULSE
B5-6
(H-9)
(H-10)
100P
(H-11)
(H-13)
2.7K
ON THE PANEL
ON THE PANEL
+5V
B5-15
1K
2.7K
0.045 UF
1KHz
1150PF
10KHz
(C-1)
150PF
50KHz
(C-2)
B5-10
+3V
(F-9)
J
C
K
(B5-8)
(C-5)
(E-4)
(F-1)
(C-4)
(E-6)
0.02 UF
2.7K
2.7K
+5V
INTERNAL SAMPLING PULSE


+5V
2.2K
3.3K
CP
B5-16
(A-8)
(A-1)
(B5-17)
DELAYED CP
100P

4.3 τ_m-SELECT SWITCH

Position 20 of each section is a master arm and goes to register R. Position 1 is connected directly from the A/D converter to supply the information with zero time delay. The rest of the positions are connected from the τ_m-shift register. Figure 4.5 shows the diagram and the detailed connection table is given in Table 1 of Appendix 5. The

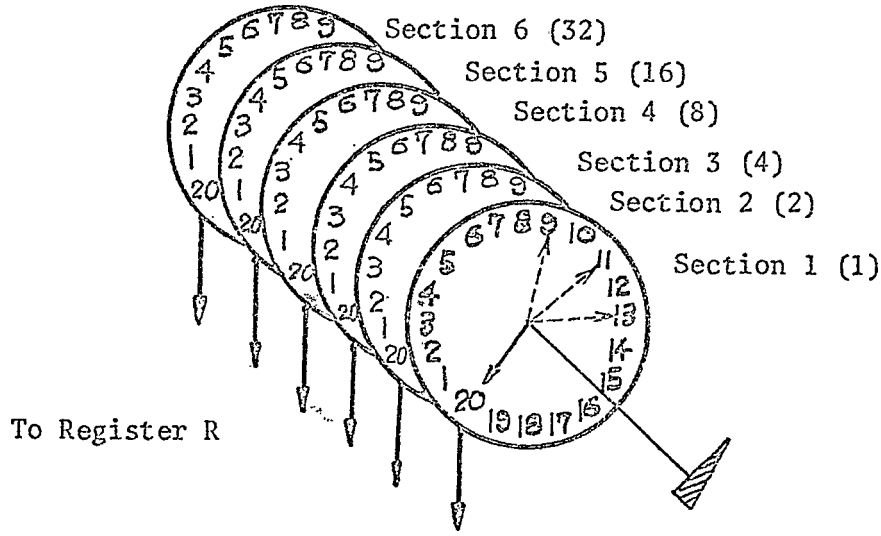


FIGURE 4.5 Switch Diagram, where section 4, for example, is assigned the binary weight 2^4.

symbol τ_{m-n} in the table represents a function, where the subscripts m and n stand for:

m: Time difference, for example, m = 4 means that the time difference is 4T, where T is the sampling interval.

n: Binary weight.

CHAPTER V

PERFORMANCE TESTS

The first section describes how to check the system performance. In the second section some results of practical measurements for the typical input signals are presented.

5.1 CHECK BY D.C. INPUT:

The performance of the instrument can be easily checked by using a d.c. input. The input is adjusted so that the voltage lies within a certain level interval. The result, in this case, will be the mean square value of the d.c. input.

In Equation 2-6, s is equal to r for a d.c. input and the resulting weighted number for a level r becomes

$$\frac{r(r+1)}{N^2 C_o} = \frac{r(r+1)}{64^2 \times 100{,}000} \quad \text{for } N = 64,\ C_o = 100{,}000$$

The decimal display counter reads out $r(r+1)/4096$ and the number $C_o = 100{,}000$ is displayed in the C_o counter. The formulas expressed in Equation 2-5 and Equation 2-6 are for the case when the input voltage is normalized. However, the actual voltage at the highest level is +10V. Hence the final answer is given by

$$\left(\frac{\text{decimal display counter readout}}{C_o \text{ counter readout } C_o = 100{,}000} \right) \times 100$$

Table 2 in Appendix 6 lists the correct display counter readout values for given d.c. input voltages.

The accuracy of each level setting in the A/D converter is checked by measuring the level at which the decimal counter changes reading.

5.2 MEASUREMENTS OF SOME TYPICAL SIGNALS:

When a square wave is autocorrelated the resulting function is a triangular wave. In those cases where a sinusoidal type of waveform is autocorrelated, the resulting function is in the form of a cosine wave. When noise is autocorrelated, the resulting waveform will depend on the center frequency and bandwidth of the noise. The autocorrelation function for general noise is given by:

$$R(\tau_m) = N\ell^{-\alpha\tau_m} \cos \beta\tau_m$$

where N is the mean square value of noise, β is the center frequency and α indicates the bandwidth.

For the tests of the entire system, autocorrelation function of sinusoidal signals and a rectangular wave were measured with a fixed sample size C_o = 10,000. Figures 5.1 - 5.3 show the results for the various frequencies of sinusoidal inputs. Figure 5.4 shows the case of a rectangular input with a frequency of 1 KHz.

The autocorrelation functions of signals superimposed by high noise were also measured. Figures 5.5 and 5.6 are the cases of a rectangular wave (5 KHz) plus noise and a sinusoidal wave plus noise, respectively. In both cases the original frequencies of 5 KHz are completely detected.

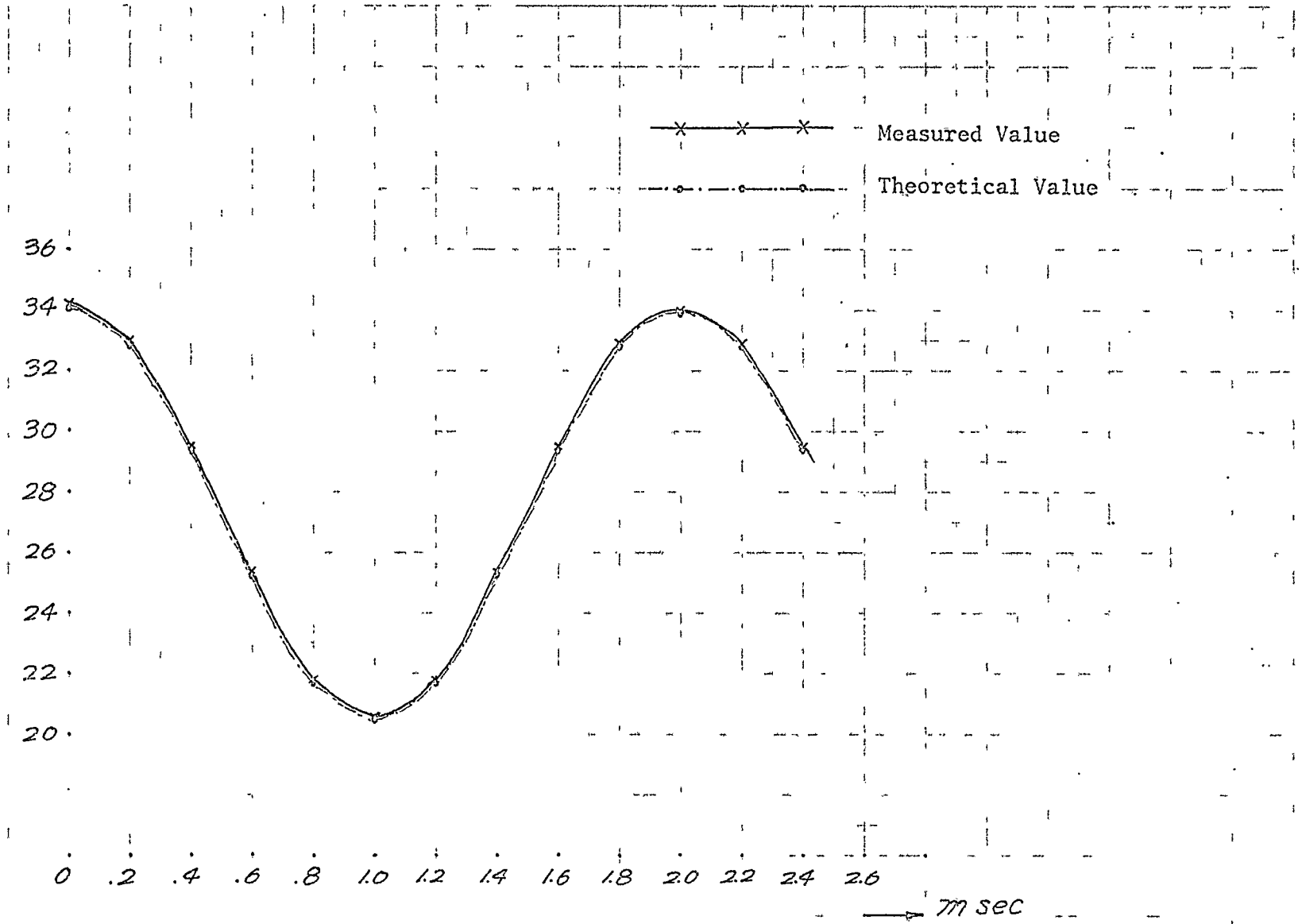


FIGURE 5.1 Auto-Correlation Functinn of Sine Wave (500 Hz)

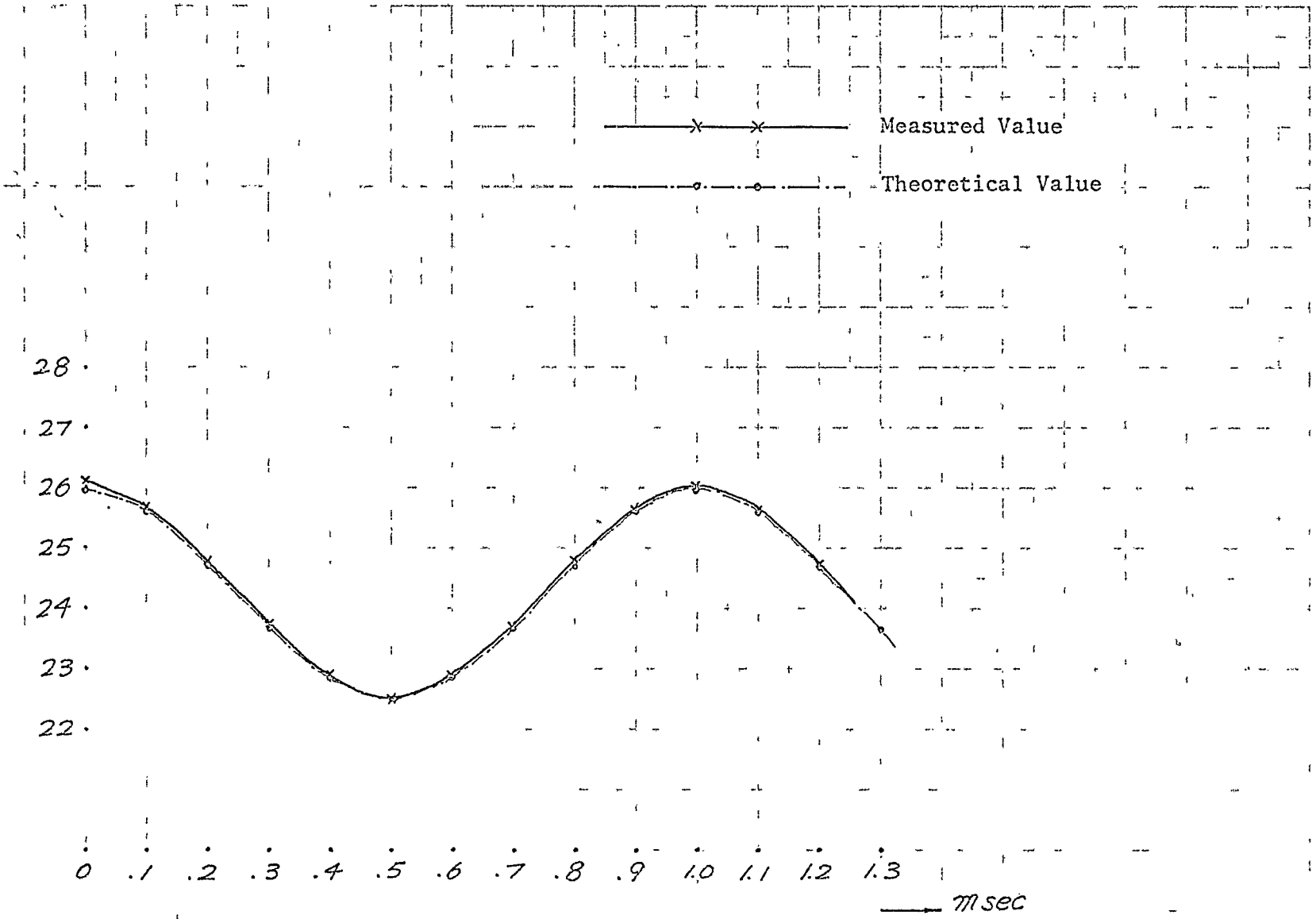


FIGURE 5.2 Auto-Correlation Function of Sine Wave (1 KHz)

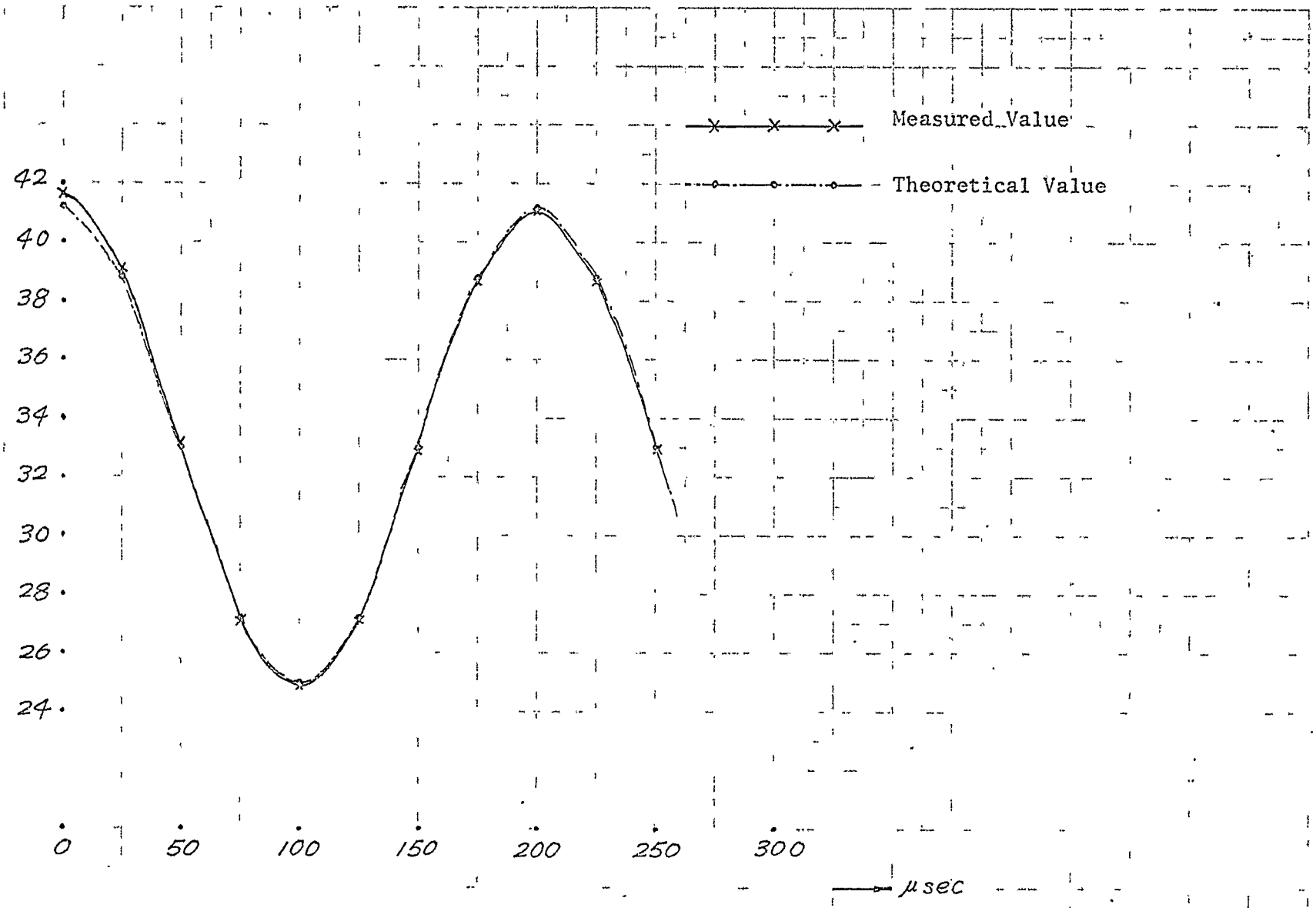


FIGURE 5.3 Auto-Correlation Function of Sine Wave (5 KHz)

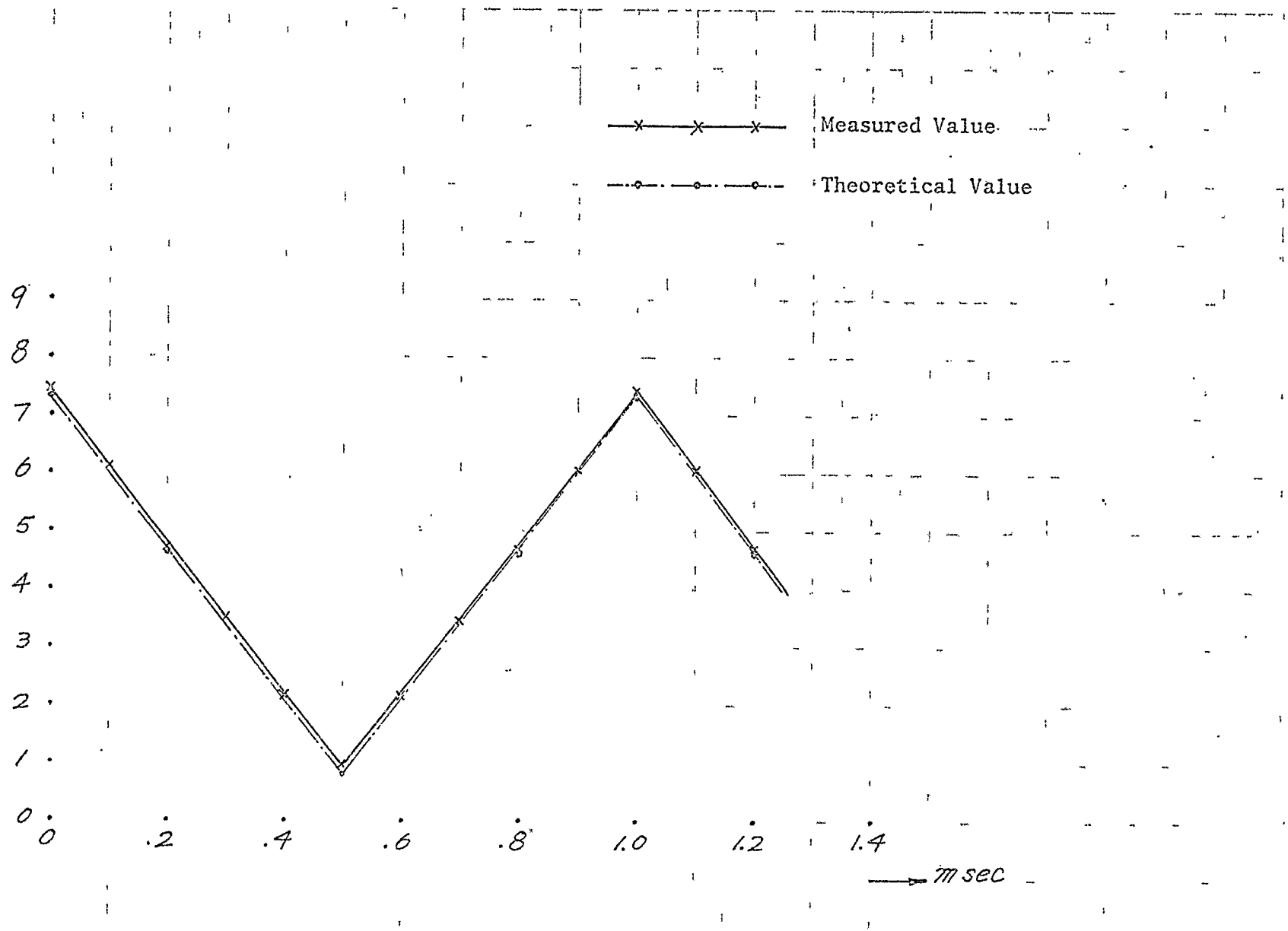


FIGURE 5.4 Auto-Correlation Function of Rectangular Wave (1 KHz)

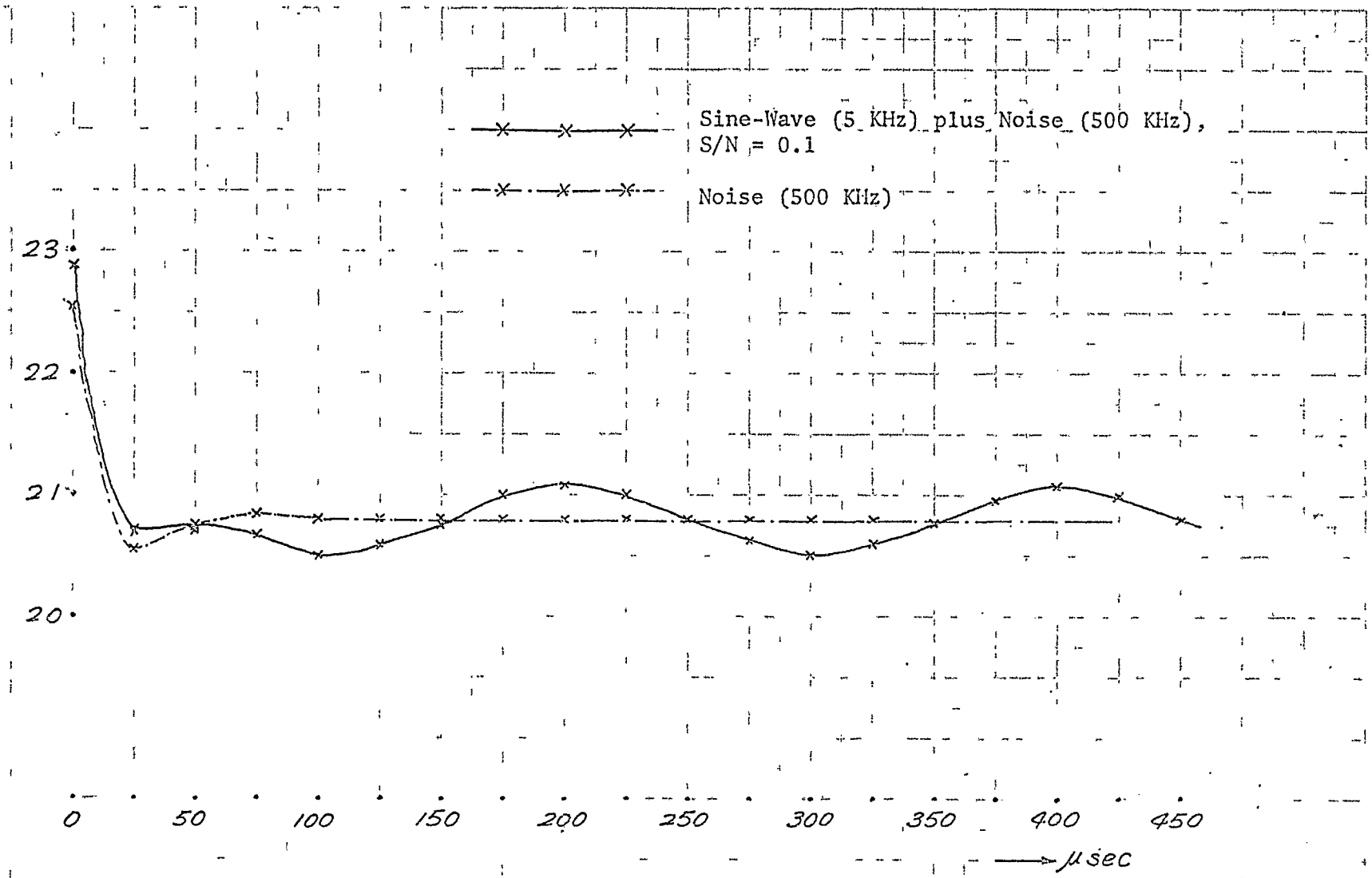


FIGURE 5.5 Auto-Correlation Functions of Sine Wave Plus Noise and Noise

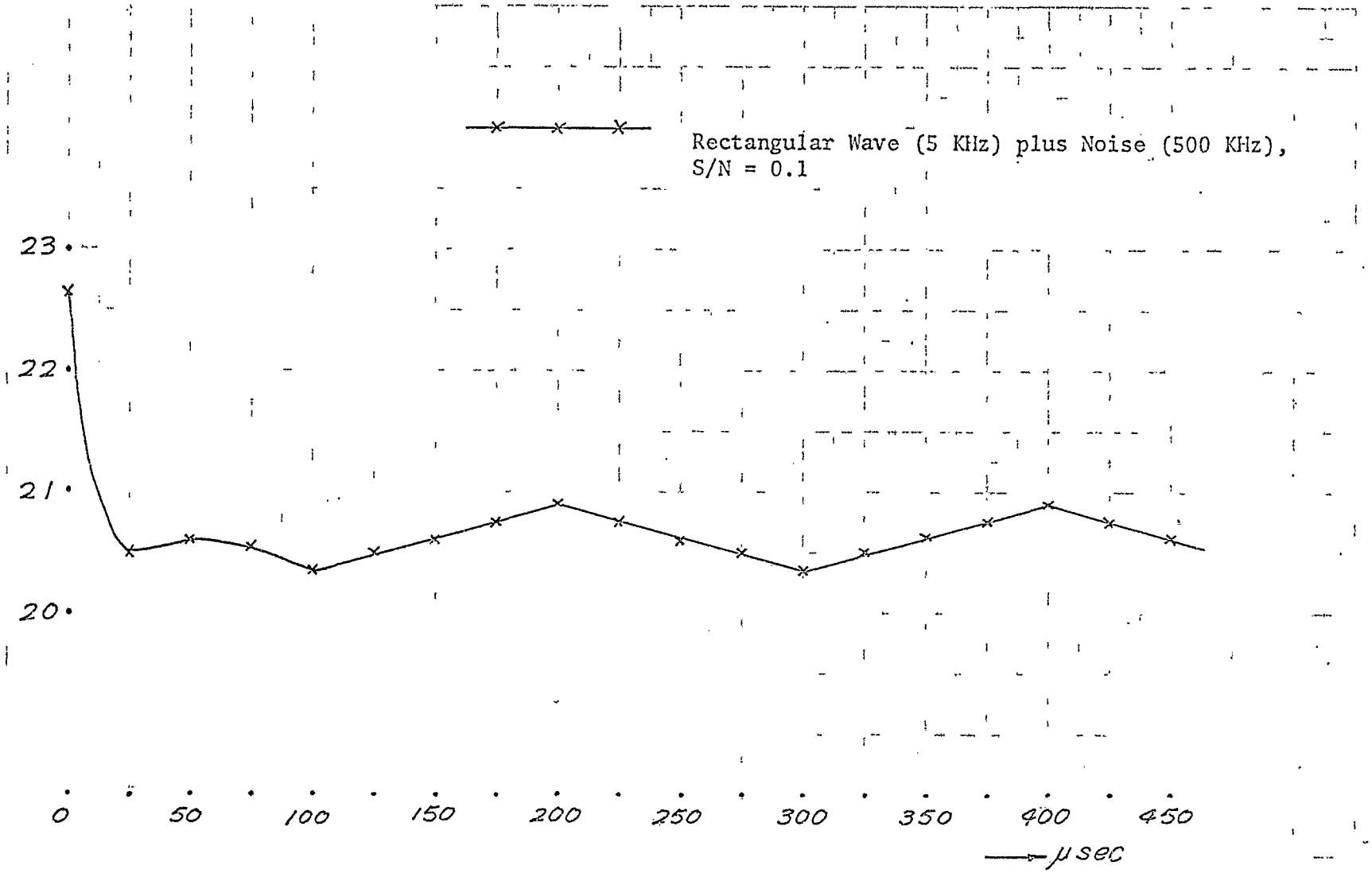


FIGURE 5.6 Auto-Correlation Function of Rectangular Wave Plus Noise

CHAPTER VI

CONCLUSION

The object of this thesis was to design and build a digital auto-correlator. An instrument for this purpose has been built and described in detail in the earlier chapters of this thesis. The salient features of this particular design can be summarized as follows:

1. There are no low-frequency limitations.
2. At the end of the measuring time, both correlation function and measuring time are immediately available in digital form on two separate counters so that the instrument can be incorporated in real-time process control.
3. The code of the digital read-outs (decimal, octal, binary, etc.) can be changed at will simply by changing the two counters for the answer and the measuring time. The entire logic section of the instrument remains unaltered.
4. Correlation function over a long range of time difference can be easily obtained without worrying about the distortion of input signal due to storing.
5. The mean square value of a time function is also obtainable.

From the view of looking at the correlator as a special-purpose computer, it can be generally said that time has been efficiently utilized for computation. For example, one of the bits in the control counter is used not only as a simple member of the counter but also as

shift pulses of registers R and S and also it is used as timing pulses of two-stage adding process. Therefore, many operations are performed simultaneously in each instance. Such a process is not normal in a general purpose computer.

Because of the introduction of integrated circuits in the correlator it is expected to be highly reliable since; -

1. The complex circuits are integrated into a single package, hence the failure rate of parts becomes quite small compared to the case of separate components.
2. The total number of soldering or wiring joints is reduced.

It is notable that in the correlator described in this thesis about three hundred gate circuits have been condensed in an area of less than 6" x 7". The power requirement is only 12.8 watts. The above points lead to the possibility of very conveniently adapting the correlator for some kind of on line optimizing control systems. This is particularly important for systems having restricted space and power requirements.

Sometimes we have to be careful in using I.C. logic systems since a noise spike which is 20 - 30 nseconds in width and more than 1 volt in amplitude becomes sufficiently effective. Such a pulse would, on the other hand, have negligibly small effect on the RC-coupled triggering method of the conventional DTL system.

We have seen in the previous chapter that the system works satisfactorily. Several suggestions for the further investigations and modifications are described in the following sections.

6.1 OBTAINING CROSS-CORRELATION FUNCTION:

Usually two A/D converters are required for obtaining the cross-correlation function. However it is possible to obtain the function with the system described in this thesis by the addition of one electronic switch and one J-K type flip-flop. This saves the use of an additional A/D converter or any other major circuit change. The logic diagram will be the one shown in Figure 6.1. Here A/D done pulse might be the best timing for the switching.

In this method the time difference τ_m is given by τ_m = T, 3T, 5T, ... , in other words, the capacity of the memory becomes halved because the quantized informations X and Y are memorized in the τ_m-shift register in such a sequence as X-Y-X-Y-X ... (see Figure 6.2).

6.2 AUTOMATIC SELECTION OF TIME DIFFERENCE τ_m:

It would be convenient if τ_m is automatically changed and each result for the particular value of τ_m is recorded in the externally connected recorder so that the system completes the measurement for the over-all range of time difference with only one start pulse.

In this case a counter should be introduced instead of the τ_m-select switch. The counter counts the number of changes of F_{DO}, where the existing logics for the input of F_{DO} remain unaltered. The decoder from the counter selects each stage in the τ_m-shift register sequentially. FF, say, is set to "1" by Key-start and while it is "1" the whole process is ON and computing processes are repeated until the counter reaches

the last stage. Figure 6.3 shows the logic diagram. The circuits described in Sections 4.2.11 and 4.2.12 are used respectively for the one-shot and the delay.

6.3 ERROR ANALYSIS:

This will be the most important part in the future work. The experimental results of Chapter V have showed that the maximum error is within 2% in overall measurements for up to 5KHz input signal. However, because of limited time it was not possible for the author to analyse it theoretically. The major errors to be considered are: -

1. Error due to quantization levels.
2. Error due to finite time of measurement.
3. Error due to sampling frequency.
4. Error due to aperture time of A/D conversion.

Solodovnikov[6] suggested that a reasonable lower limit of sampling frequency is twenty times the highest frequency component of the input signal. As for error 2, he investigated this by using the mean square error criterion and gave an example of a case that the auto-correlation function is expressed by $Ne^{-\alpha|\tau|}\cos\beta\tau$. This function is frequently encountered in random gaussian processes. He proposed that the finite time of measurement should be more than $K(\frac{1}{\alpha}+\frac{\alpha}{\alpha^2+\beta^2})$, where K is a constant and depends upon the tolerance of error percentage. Another interesting article on error 2 can be found in the paper by B. N. Kutin[7].

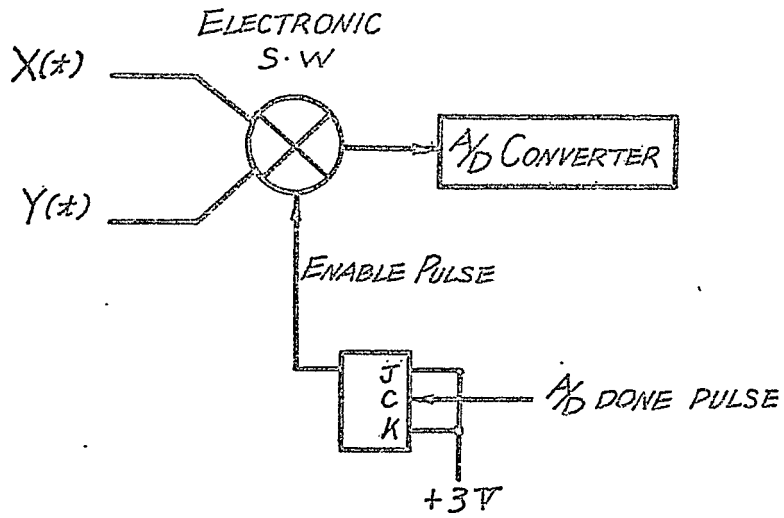


FIGURE 6.1 Logic for Obtaining Cross-Correlation

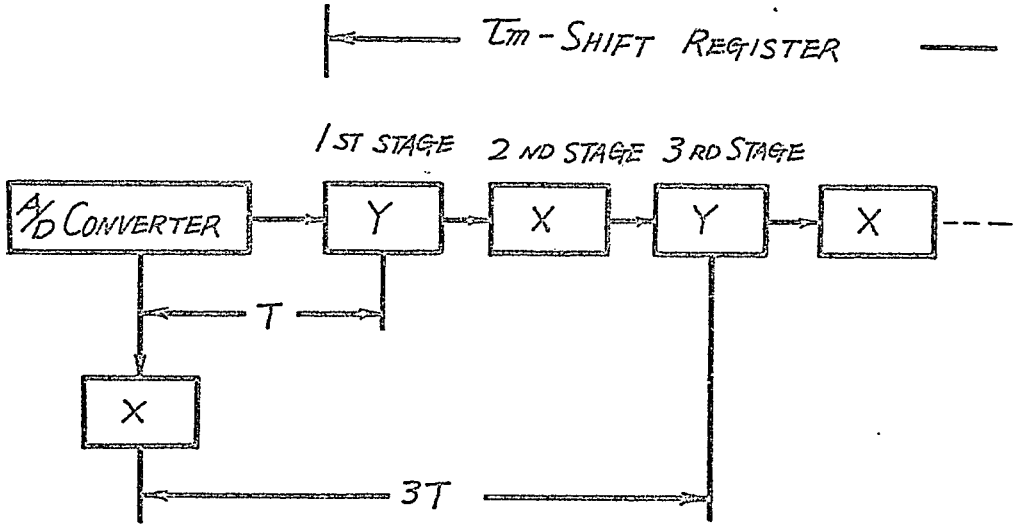


FIGURE 6.2 Information in the Memory

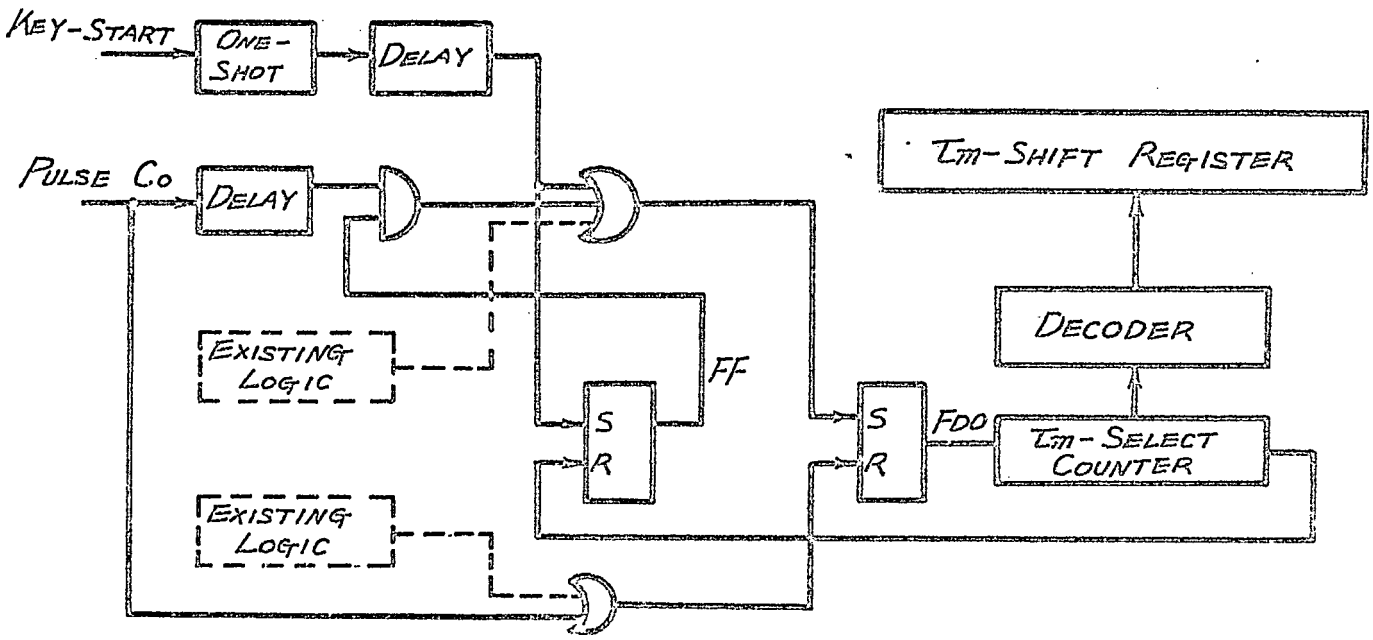


FIGURE 6.3 Automatic Selection of τ_m

APPENDIX

1. PROOF OF EQUATION 2-2:

The probability $P(X_{mr}, Y_{ms}; \tau_m)$ can be represented geometrically by the amplitude shown in Figure 1, $P(X_{m3}, Y_{m3})$ for example, is expressed by (see Figure 2).

$$P(X_{m3}, Y_{m3}) = \text{Sum of amplitudes in (a)} - \text{Sum of amplitudes in (b)} - \text{Sum of amplitudes in (c)} + \text{Sum of amplitudes in (d)}$$

$$= P_{2,2} - P_{3,2} - P_{2,3} + P_{3,3}$$

Hence, in general, $P(X_{mr}, Y_{ms}; \tau_m)$ is expressed by

$$P(X_{mr}, Y_{ms}; \tau_m) = P_{r-1,s-1} - P_{r,s-1} - P_{r-1,s} + P_{r,s}$$

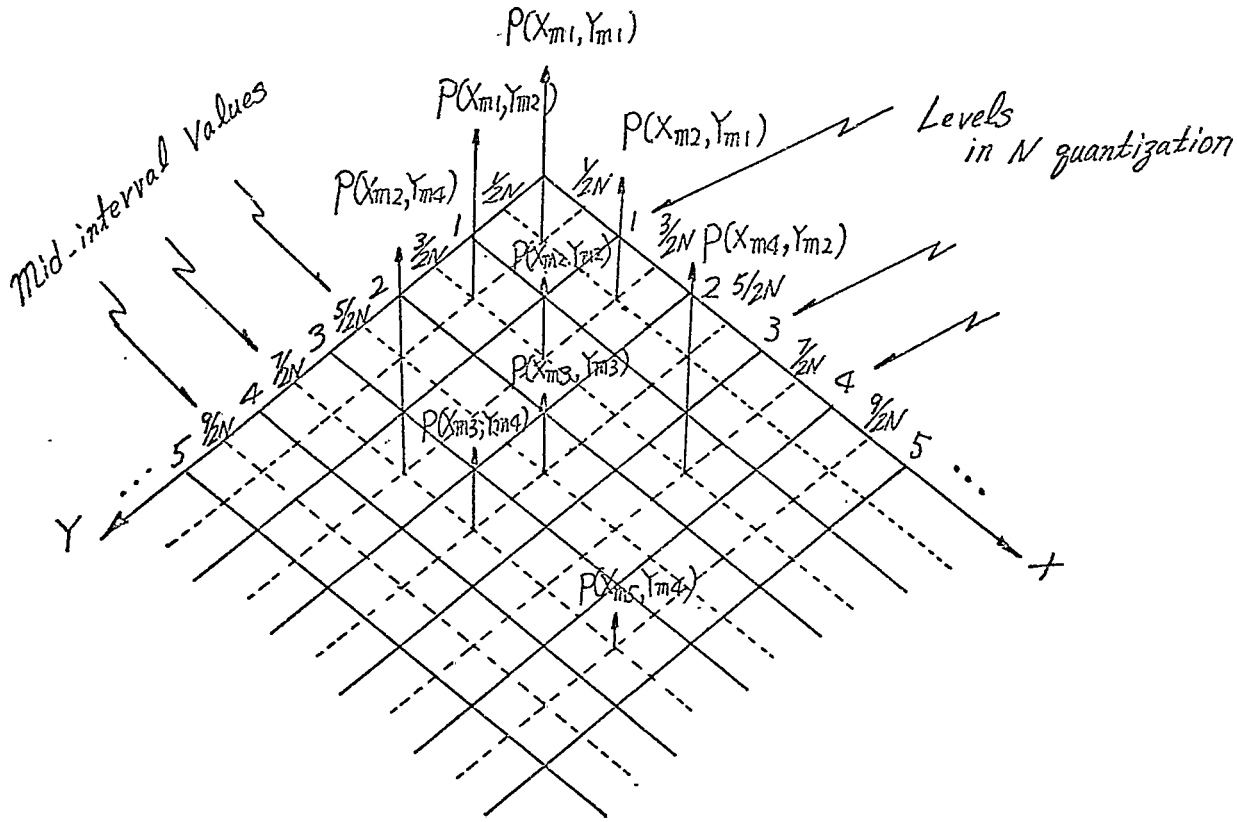


FIGURE 1 Joint Probability Distribution

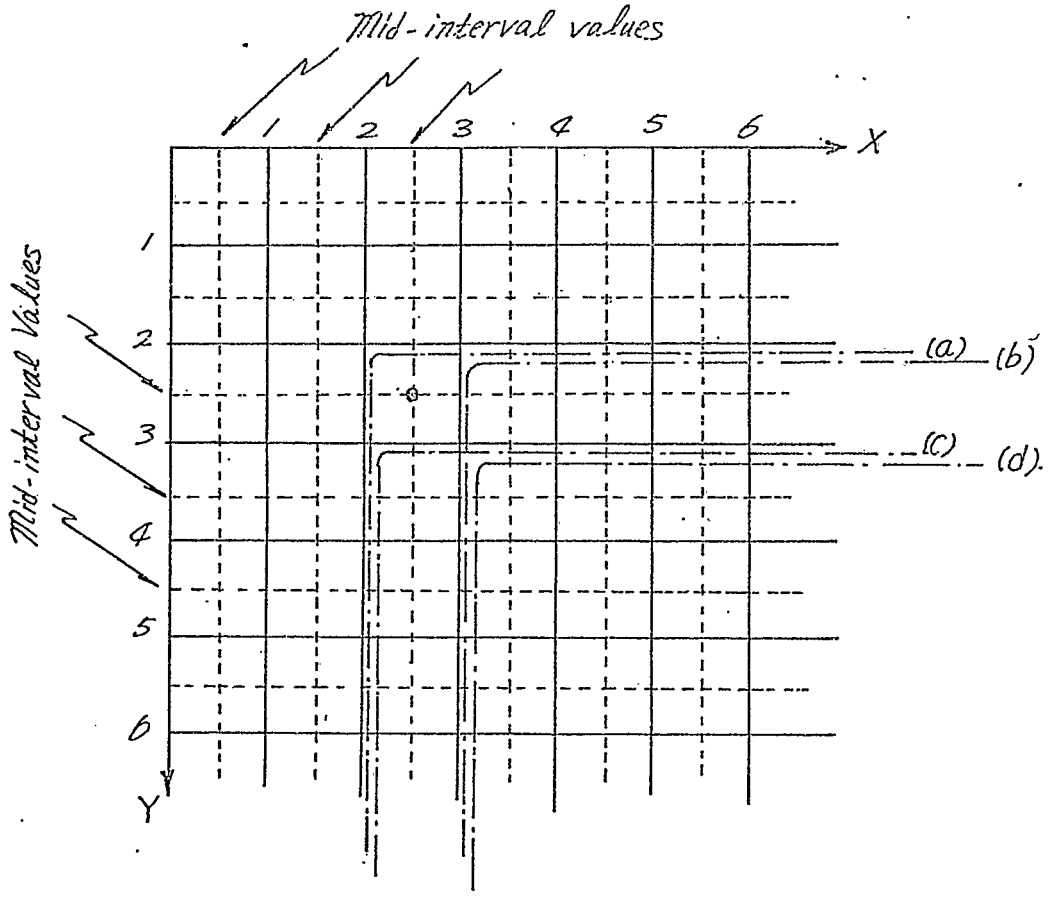


FIGURE 2 Geometrical Expression for the Example

2. PROOF OF EQUATION 2-3:

$$R_{XY}(\tau_m) = \frac{1}{4N^2} \sum_{r,s=1}^{N} \{P_{r-1,s-1} - P_{r,s-1} - P_{r-1,s} + P_{r,s}\}(2r-1)(2s-1)$$

$$= \frac{1}{4N^2} \sum_{r=1}^{N} \begin{vmatrix} \{P_{r-1,0} - P_{r,0} + P_{r,1} - P_{r-1,1}\}(2r-1).1 \\ + \{P_{r-1,1} - P_{r,1} + P_{r,2} - P_{r-1,2}\}(2r-1).3 \\ \vdots \\ + \{P_{r-1,N-1} - P_{r,N-1} + P_{r,N} - P_{r-1,N}\}(2r-1)(2N-1) \end{vmatrix}$$

Since the voltage does not exceed level N,

$$P_{r,N} = P_{r-1,N} = 0$$

Hence

$$4N^2 R_{XY}(\tau_m) = \sum_{r=1}^{N} \{P_{r-1,0} - P_{r,0}\}(2r-1).1 + \sum_{r=1}^{N} \{P_{r-1,1} - P_{r,1}\}(2r-1).2$$

$$+ \sum_{r=1}^{N} \{P_{r-1,2} - P_{r,2}\}(2r-1).2 \ \ldots + \sum_{r=1}^{N} \{P_{r-1,N-1} - P_{r,N-1}\}(2r-1).2$$

While

$$\sum_{r=1}^{N} \{P_{r-1,0} - P_{r,0}\}(2r-1)$$

$$= (P_{0,0} - P_{1,0}).1 + (P_{1,0} - P_{2,0}).3 + \ldots + (P_{N-1,0} - P_{N,0})(2N-1)$$

$$= P_{0,0} + 2 \sum_{r=1}^{N-1} P_{r,0} \Bigg| \sum_{r=1}^{N} \{P_{r-1,1} - P_{r,1}\}(2r-1).2$$

$$= (P_{0,1} - P_{1,1})1.2 + (P_{1,1} - P_{2,1})3.2 + \ldots + (P_{N-1,1} - P_{N,1})(2N-1)2$$

$$= 2P_{0,1} + 4 \sum_{r=1}^{N-1} P_{r,1}$$

Similarly in general case

$$\sum_{r=1}^{N} \{P_{r-1,N-1} - P_{r,N-1}\}(2r-1).2 = 2P_{0,N-1} + 4 \sum_{r=1}^{N-1} P_{r,N-1}$$

Therefore we obtain

$$R_{XY}(\tau_m) = \frac{1}{4N^2} \{P_{0,0} + 2P_{0,1} + 2P_{0,2} + \ldots + 2P_{0,N-1} + 2 \sum_{r=1}^{N-1} P_{r,0}$$

$$+ 4 \sum_{r=1}^{N-1} P_{r,1} + 4 \sum_{r=1}^{N-1} P_{r,2} + \ldots + 4 \sum_{r=1}^{N-1} P_{r,N-1}\}$$

$$= \frac{1}{4N^2} \{P_{0,0} + 2 \sum_{r=1}^{N-1} P_{r,0} + 2 \sum_{r=1}^{N-1} P_{0,s} + 4 \sum_{r=1}^{N-1} \sum_{s=1}^{N-1} P_{r,s}\}$$

2A. WEIGHTING NUMBER:

When we sample the function X(t) at time nT and get the highest level exceeded as r^{th} and τ_m seconds later we sample the other function Y(t) and get level exceeded as s^{th}. After these two sampling combinations, we must feed some weighting number to the accumulator. From Equation 2-5 the following computations are carried out: -

$$\frac{1}{2N^2C_o} \left\{ \sum_{i=1}^{r} C_{i,0} + \sum_{j=1}^{s} C_{0,j} + 2 \sum_{i=1}^{r} \sum_{j=1}^{s} C_{i,j} \right\}$$

$$= \frac{1}{2N^2C_o} \{ \underbrace{C_{1,0} + C_{2,0} + \ldots + C_{r,0}}_{r} + \underbrace{C_{0,1} + C_{0,2} + \ldots + C_{0,s}}_{s} + 2 \sum_{i=1}^{r} \{ \underbrace{C_{i,1} + C_{i,2} + \ldots + C_{i,s}}_{s} \} \}$$

$$= \frac{1}{2N^2C_o} \{ r + s + 2s \{ \underbrace{C_{1,s} + C_{2,s} + \ldots + C_{r,s}}_{r} \} \}$$

$$= \frac{1}{2N^2C_o} \{ r + s + 2rs \}$$

3. SN74XXN SERIES:

This is referred from the Texas Instruments TTL Integrated Circuits Catalogue[5]. Logic diagrams and the corresponding specifications are given from Figure 3 to Figure 14.

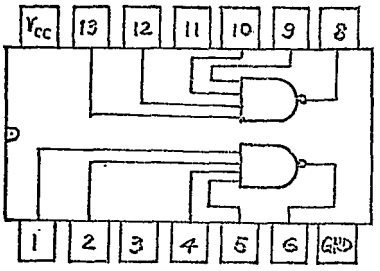


FIGURE 3 SN7440N

Dual 4-input NAND buffer
Propagation delay - 13 nsec
Power dissipation - 50 mW total for two gates (25 mW per gate)
Fan out - 30 Series 54/74 loads

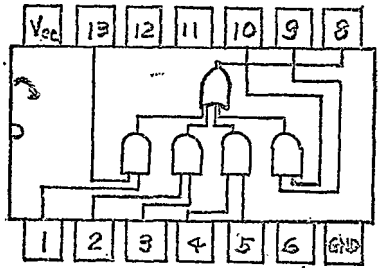


FIGURE 4 SN7454N

4-wide 2-input AND-OR-INVERT gate
Propagation delay - 13 nsec
Power dissipation - 22 mW
Fan out - 10 Series 54/74 loads

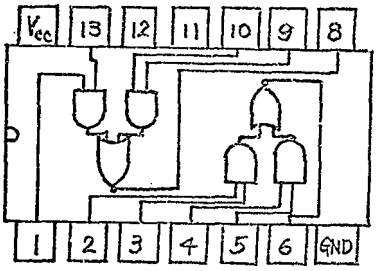


FIGURE 5 SN145IN

Dual 2-wide 2-input AND-OR-INVERT gate
Propagation delay - 13 nsec
Power dissipation - 28 mW total for two gates (14 mW per gate)
Fan out - 10 Series 54/74 loads

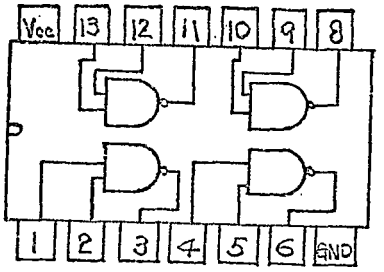


FIGURE 6 SN7400N

Quadruple 2-input NAND gate
Propagation delay - 13 nsec
Power dissipation - 40 mW total for four gates (10 mW per gate)
Fan out - 10 Series 54/74 loads

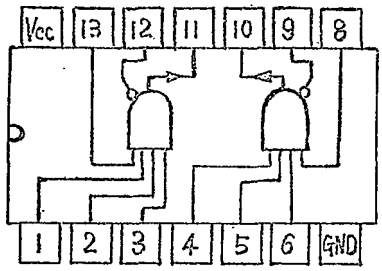


FIGURE 7 SN7460N

Dual 4-input expander
Propagation delay - 5nsec
Power dissipation - 8 mW total for two expander (4 mW per expander)
Fan out (max number of expanders connected to SN5450N or SN5453N)-4

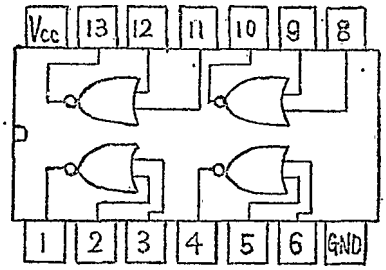


FIGURE 8 SN7402N

Quadruple 2-input NOR gate
Propagation delay - 13 nsec
Power dissipation - 48 mW total for four gates (12 mW per gate)
Fan out - 10 Series 54/74 loads

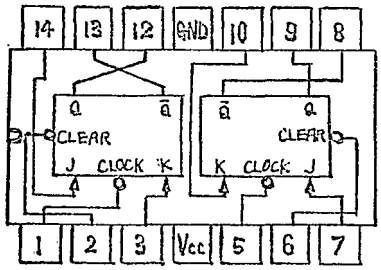


FIGURE 9 SN7473N

Dual J-K master-slave flip-flop
Propagation delay - 30 nsec
Power dissipation - 80 mW total for two flip-flops (40 mW per flip-flop)
Fan out - 10 Series 54/74 loads

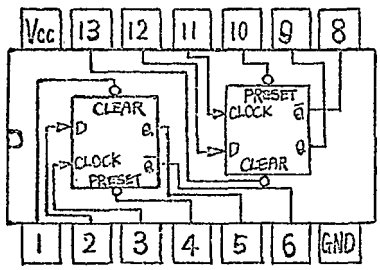


FIGURE 10 SN7474N

Dual D-type edge-triggered flip-flop
Propagation delay - 24 nsec
Power dissipation - 84 mW total for two flip-flops (24 mW per flip-flop)
Fan out - 10 Series 54/74 loads

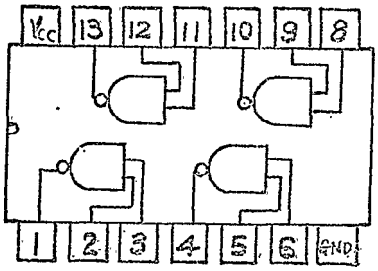


FIGURE 11 SN7401N

Quadruple 2-input NAND gate with
open collector output
Propagation delay - 35 nsec
Power dissipation - 88 mW total for
four gates (22 mW per gate)
Fan out - 10 Series 54/74 loads

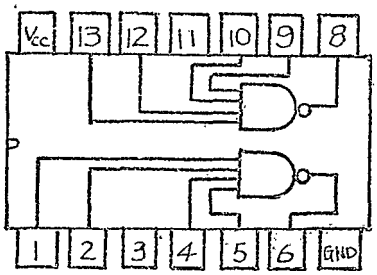


FIGURE 12 SN7420N

Dual 4-input NAND gate
Propagation delay - 13 nsec
Power dissipation - 20 mW total for
two gates (10 mW per gate)
Fan out - 10 Series 54/74 loads

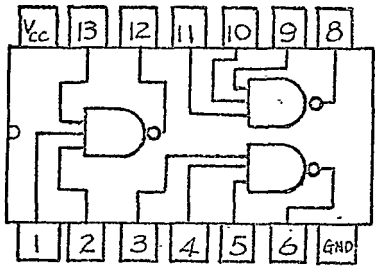


FIGURE 13 SN7410N

Triple 3-input NAND gate
Propagation delay - 13 nsec
Power dissipation - 30 mW total for
three gates (10 mW per gate)
Fan out - 10 Series 54/74 loads

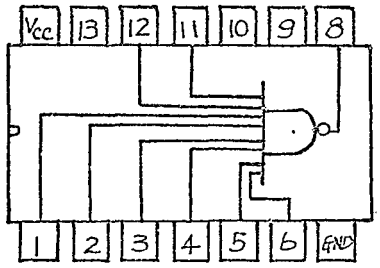


FIGURE 14 SN7430N

8-input NAND gate
Propagation delay - 13 nsec
Power dissipation - 10 mW
Fan out - 10 Series 54/74 loads

4. CIRCUIT CONFIGURATION OF NAND GATE:

Figure 15 shows the equivalent circuit of NAND gate (SN7400N). The operation is briefly explained as follows: -

(a) With a logic "0" at the inputs, the B_1-E_1 junction becomes conductive and one ma (nominal) flows through it. The base of Q_1 sits at approximately +1V, but this potential is not enough to conduct Q_2 and Q_3. Hence Q_2 and Q_3 are turned off and logic "1" appears at the output.

(b) With a logic "1" at the inputs, the B_1-E_1 junction becomes reverse biased and the majority current flow through Q_1 is from B_1 to C_1. At this time the base of Q_1 sits at approximately +2.25V and junctions B_2-E_2 and B_3-E_3 become forward biased, turning on Q_2 and Q_3. B_4 sits at approximately +1V, keeping Q_4 turned off, as +1.5V are needed to forward bais the series string of the B_4-E_4 junction plus the diode. Hence logic "0" appears at the output.

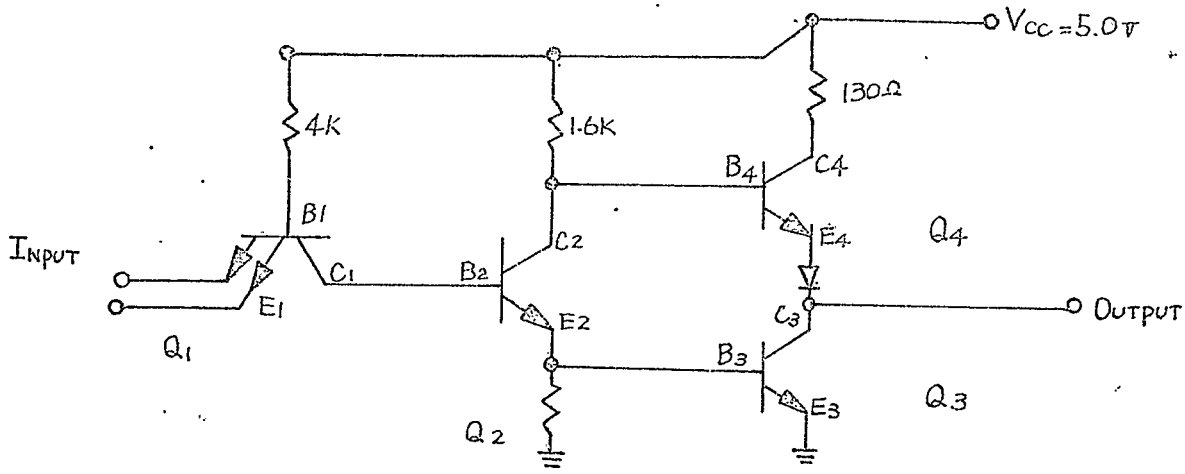


FIGURE 15 Circuit Configuration of NAND gate

TABLE 1 Connections from τ_m-Select Switch to Boards

Position No in Switch	Section 1		Section 2		Section 3		Section 4		Section 5		Section 6	
from	function	to	function	to	function	to	function	to	function	to	function	to
1	τ_{0-1}	B2-7	τ_{0-2}	B2-6	τ_{0-4}	B2-5	τ_{0-8}	B2-4	τ_{0-16}	B2-3	τ_{0-32}	B2-2
2	τ_{1-1}	A5-11	τ_{1-2}	A5-2	τ_{1-4}	A6-11	τ_{1-8}	A6-2	τ_{1-16}	A7-11	τ_{1-32}	A7-2
3	τ_{2-1}	A5-L	τ_{2-2}	A5-B	τ_{2-4}	A6-L	τ_{2-8}	A6-B	τ_{2-16}	A7-L	τ_{2-32}	A7-B
4	τ_{3-1}	A5-12	τ_{3-2}	A5-3	τ_{3-4}	A6-12	τ_{3-8}	A6-3	τ_{3-16}	A7-12	τ_{3-32}	A7-3
5	τ_{4-1}	A5-M	τ_{4-2}	A5-C	τ_{4-4}	A6-M	τ_{4-8}	A6-C	τ_{4-16}	A7-M	τ_{4-32}	A7-C
6	τ_{5-1}	A5-13	τ_{5-2}	A5-4	τ_{5-4}	A6-13	τ_{5-8}	A6-4	τ_{5-16}	A7-13	τ_{5-32}	A7-4
7	τ_{6-1}	A5-N	τ_{6-2}	A5-D	τ_{6-4}	A6-N	τ_{6-8}	A6-D	τ_{6-16}	A7-N	τ_{6-32}	A7-D
8	τ_{7-1}	A5-14	τ_{7-2}	A5-5	τ_{7-4}	A6-14	τ_{7-8}	A6-5	τ_{7-16}	A7-14	τ_{7-32}	A6-5
9	τ_{8-1}	A5-P	τ_{8-2}	A5-E	τ_{8-4}	A6-P	τ_{8-8}	A6-E	τ_{8-16}	A7-P	τ_{8-32}	A7-E
10	τ_{9-1}	A5-15	τ_{9-2}	A5-6	τ_{9-4}	A6-15	τ_{9-8}	A6-6	τ_{9-16}	A7-15	τ_{9-32}	A7-6
11	τ_{10-1}	A5-Q	τ_{10-2}	A5-F	τ_{10-4}	A6-Q	τ_{10-8}	A6-F	τ_{10-16}	A7-Q	τ_{10-32}	A7-F
12	τ_{11-1}	A5-16	τ_{11-2}	A5-7	τ_{11-4}	A6-16	τ_{11-8}	A6-7	τ_{11-16}	A7-16	τ_{11-32}	A7-7
13	τ_{12-1}	A5-R	τ_{12-2}	A5-GT	τ_{12-4}	A6-R	τ_{12-8}	A6-GT	τ_{12-16}	A7-R	τ_{12-32}	A7-GT
14	τ_{13-1}	A5-17	τ_{13-2}	A5-8	τ_{13-4}	A6-17	τ_{13-8}	A6-8	τ_{13-16}	A7-17	τ_{13-32}	A7-8
15	τ_{14-1}	A5-S	τ_{14-2}	A5-H	τ_{14-4}	A6-S	τ_{14-8}	A6-H	τ_{14-16}	A7-S	τ_{14-32}	A7-H
16	τ_{15-1}	A5-18	τ_{15-2}	A5-9	τ_{15-4}	A6-18	τ_{15-8}	A6-9	τ_{15-16}	A7-18	τ_{15-32}	A7-9
17	τ_{16-1}	A5-T	τ_{16-2}	A5-J	τ_{16-4}	A6-T	τ_{16-8}	A6-J	τ_{16-16}	A7-T	τ_{16-32}	A7-J
18	τ_{17-1}	A5-19	τ_{17-2}	A5-10	τ_{17-4}	A6-19	τ_{17-8}	A6-10	τ_{17-16}	A7-19	τ_{17-32}	A7-10
19	τ_{18-1}	A5-U	τ_{18-2}	A5-K	τ_{18-4}	A6-U	τ_{18-8}	A6-K	τ_{18-16}	A7-U	τ_{18-32}	A7-K
20		A4-4		A4-D		A4-5		A4-E		A4-6		A4-F

TABLE 2

Direct Decimal Display Counter Readout

D.C Input Voltage Volts	Display Counter Readout
0. - .15625	0
.15625 - .31250	48
.31250 - .46875	146
.46875 - .62500	292
.62500 - .78125	488
.78125 - .93750	732
.93750 - 1.09375	1025
1.09375 - 1.25000	1367
1.25000 - 1.40625	1757
1.40625 - 1.56250	2197
1.56250 - 1.71875	2685
1.71875 - 1.87500	3222
1.87500 - 2.03125	3808
2.03125 - 2.18750	4443
2.18750 - 2.34375	5126
2.34375 - 2.50000	5859
2.50000 - 2.65625	6640
2.65625 - 2.81250	7470
2.81250 - 2.96875	8349
2.96875 - 3.12500	9277

TABLE 2 (Continued)

Direct Decimal Display Counter Readout

D.C. Input Voltage Volts	Display Counter Readout
3.12500 - 3.28125	10253
3.28125 - 3.43750	11279
3.43750 - 3.59375	12353
3.59375 - 3.75000	13476
3.75000 - 3.90625	14648
3.90625 - 4.06250	15869
4.06250 - 4.21875	17138
4.21875 - 4.37500	18457
4.37500 - 4.53125	19824
4.53125 - 4.68750	21240
4.68750 - 4.84375	22705
4.84375 - 5.00000	24218
5.00000 - 5.15625	25781
5.15625 - 5.31250	27392
5.31250 - 5.46875	29052
5.46875 - 5.62500	30761
5.62500 - 5.78125	32519
5.78125 - 5.93750	34326
5.93750 - 6.09375	36181
6.09375 - 6.25000	38085

TABLE 2 (Continued)

Direct Decimal Display Counter Readout

D.C. Input Voltage Volts	Display Counter Readout
6.25000 - 6.40625	40039
6.40625 - 6.56250	42041
6.56250 - 6.71875	44091
6.71875 - 6.87500	46191
6.87500 - 7.03125	48339
7.03125 - 7.18750	50537
7.18750 - 7.34375	52783
7.34375 - 7.50000	55078
7.50000 - 7.65625	57421
7.65625 - 7.81250	59814
7.81250 - 7.96875	62255
7.96875 - 8.12500	64746
8.12500 - 8.28125	67285
8.28125 - 8.43750	69873
8.43750 - 8.59375	72509
8.59375 - 8.75000	75195
8.75000 - 8.90625	77929
8.90625 - 9.06250	80712

TABLE 2 (Continued)

Direct Decimal Display Counter Readout

D.C. Input Voltage Volts	Display Counter Readout
9.06250 - 9.21875	83544
9.21985 - 9.37500	86425
9.37500 - 9.53125	89355
9.53125 - 9.68750	92334
9.68750 - 9.84375	95361
9.84375 -10.00000	98437

BIBLIOGRAPHY

1. Deist, F., and Kitai, R., "Digital Transfer Voltmeters: Principles and Error Characteristics", Proc. I.E.E., Vol. 110, No. 10, pp. 1887 - 1904, October 1963.

2. Braithwaite, D., "Ultra Low-frequency Digital Analyser: Design and Construction", M.Eng. Thesis, McMaster University, 1967.

3. Majithia, J., "Digital Instrumentation for the Time Integral-squared of a Voltage and its Error Characteristics", M.Eng. Thesis, McMaster University, 1968.

4. Digital Equipment Corporation, Digital Logic Handbook, 1968 Edition.

5. Texas Instruments Incorporated, TTL Integrated Circuits Catalogue, Series Sn 54/74.

6. Solodovnikov, V., "Statistical Dynamics of Linear Automatic Control Systems", Van Nostrand, 1965, Princeton, N. J.

7. Kutin, B., "On the Calculation of the Correlation Function of a Stationary Random Process From Experimental Data", Automation and Remote Control, Vol. 18, pp. 221 - 246, 1957.