1975

A graphics vector generator for a computer display system.

Appanna. Chottera
University of Windsor

A GRAPHICS VECTOR GENERATOR FOR A COMPUTER DISPLAY SYSTEM

by

APPANNA CHOTTERA

A Thesis

Submitted to the Faculty of Graduate Studies
through the Department of Electrical Engineering
in Partial Fulfillment of the Requirement for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario

1974

ABSTRACT

The design and implementation of a vector generating sub-system is described in this thesis.

Binary rate multiplier techniques are used in the design of the vector generator. It generates vectors of any length and at any angle during a constant ramp-generation time. It also functions as a position generator to place visible or invisible dots in a required position on the screen.

The digital logic functions were constructed with D.E.C. flip-flop modules.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. P.A.V. Thomas for his supervision and suggestions.

Acknowledgement also goes to the other members of the faculty who have taken an interest to accomplish this project.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

I. INTRODUCTION

In recent years a need for faster and more convenient methods of man-computer communication interface have arisen. The use of the cathode ray tube as a read out device allows computer data to be available instantly compared to the time delay involved in use of an electromechanical printer.

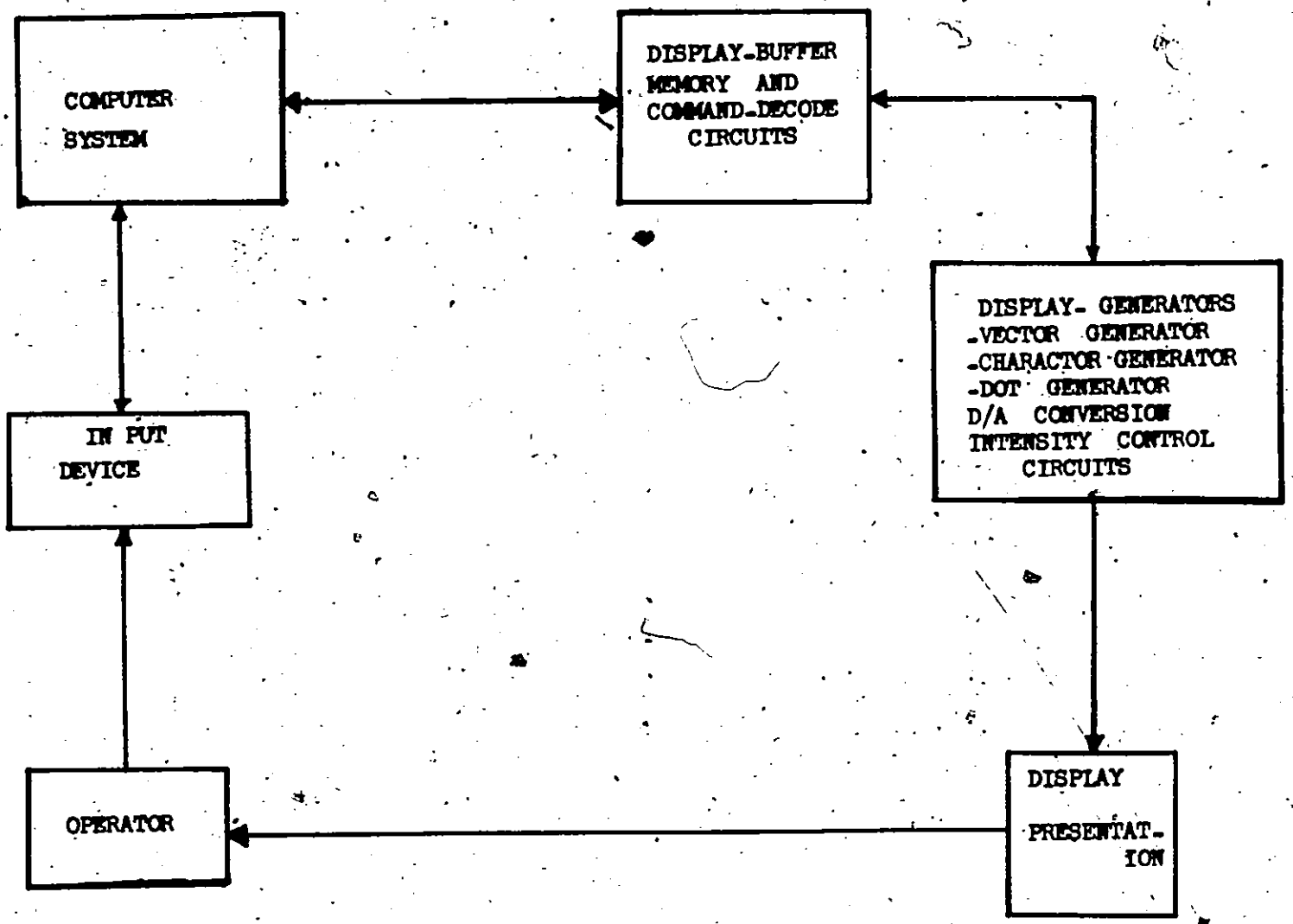
Information may be displayed on the screen of a cathode ray tube, as alphanumeric and symbolic characters, vectors, points, continuous curves, grid lines, etc., and may be randomly located or arranged in orderly fashion on the screen. All display information is supplied from an external computer instruction file for easy graphic interpretation. Flexibility in input is maintained by use of various knobs and switches, as well as light pen device, which the operator can use to indicate, to the computer, areas of interest in the display so that it may branch to a routine to modify the file to be displayed.

II. A GENERALIZED DISPLAY SYSTEM

There are various techniques of information display, but a generalized functional block diagram shown in fig. 1.1 indicates the most common parts of the system. Because the system contains some display generation hardware, the external computer need supply the information to the command decoding circuits only once, and thereafter the buffer memory can store this information and recycle it to the display generating circuits often enough to avoid any flicker in the display. The command decoding circuits decode the instruction and channel the data into proper display generators to display corresponding graphic symbols. Thus, the computer is left to perform less routine operations and will interrupt the external peripheral hardware only to update earlier information or to issue new commands as a result of information from the manual controls or other automatic input control.

III. DISPLAY SYSTEM WITH CRT DISPLAY:

As mentioned earlier, the display read out device of major importance in visual computer data display systems is the cathode ray tube. There are three main categories of C.R.T. displays, alpha-numeric, graphic, and situation displays. The alphanumeric provides a formatted typewriter-like presentation. In graphic display, lines may be



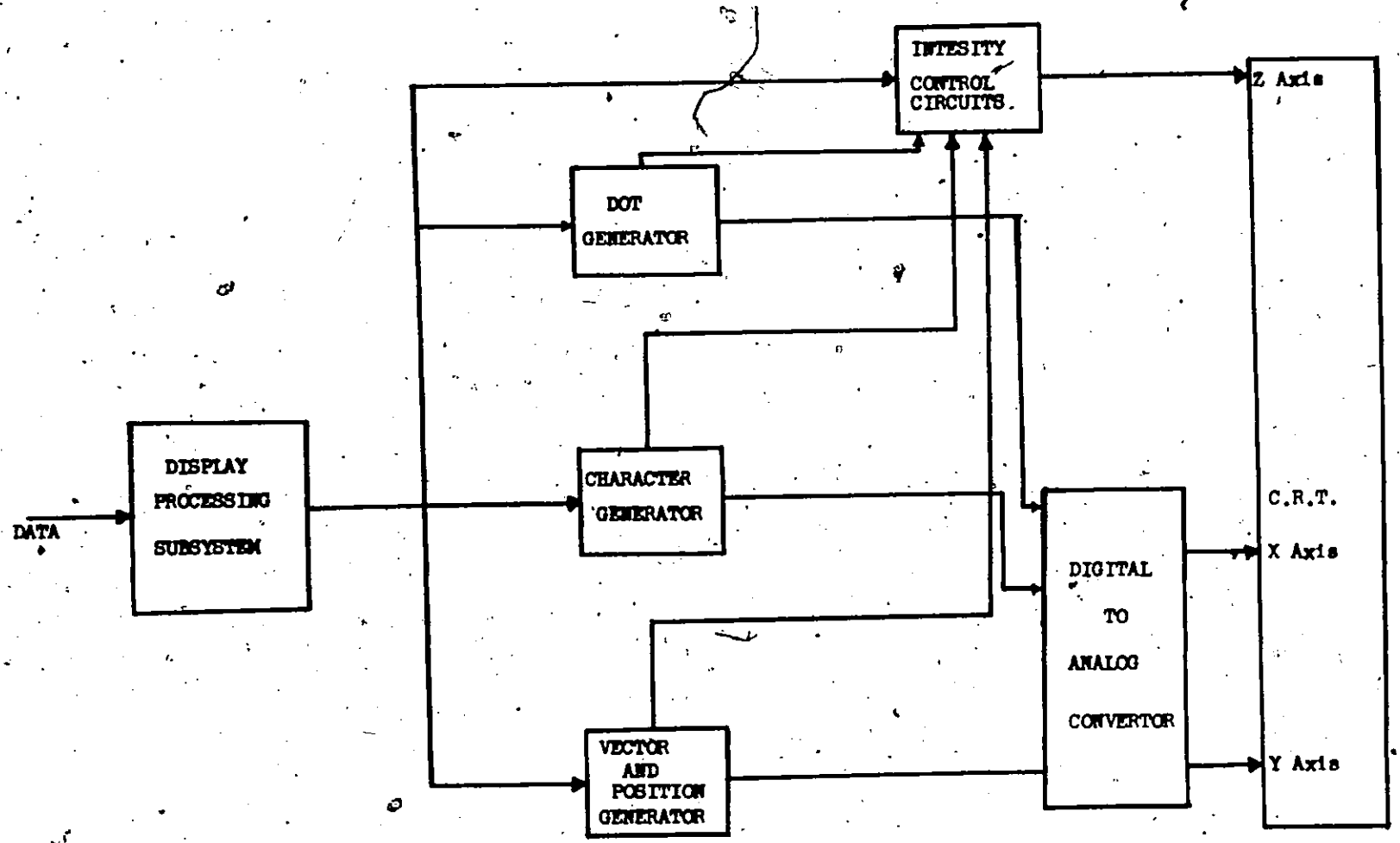
GENERALIZED FUNCTIONAL BLOCK DIAGRAM OF THE DISPLAY SYSTEM.

FIGURE 1.1

presented along with randomly placed alphanumerics and symbols. A situation display contains alphanumeric symbols, and lines, along with some form of background information.

All three C.R.T. display categories are usually used in specific applications. Alphanumeric displays are used as electronic typewriters and provide a message, composition and editing capability along with data presentation. Graphic displays provide a means for generating geometric figures plus alphanumeric data and are used in computer-aided design applications. Situation displays are used in command and control systems and the background can be radar, sonar, or map information.

The figure 1.2 shows a graphic display system with C.R.T. display. This is typical of the display system at the Electrical Engineering Department of the University of Windsor. The external computer used is a PDP-8 data processor. The display processing sub-system consists of a high speed refresh memory[1] and data flow controller which has been designed and implemented. It also has alphanumeric[2] and dot generators[3]. It also had a vector generator[4] which used an analog method of vector generation, but this has been replaced by a vector generator which uses a 'binary rate multiplier technique' to generate vectors which is the work of this thesis.



DISPLAY SYSTEM WITH C.R.T. DISPLAY.

FIGURE 1.2

The vector generator gives the C.R.T. the capacity to draw blank, solid or dashed lines of any length at any angle. This gives the desirable ability to draw graphs, plots or other shapes.

The C.R.T. used is a Hewlett-Packard X-Y Display Model 1300A[5]. It has also intensity control circuitry associated with it, which helps to overcome the problem of inconsistent brightness levels in the display.

IV. IMPLEMENTATION:

The circuitry was constructed from Digital Equipment Corporation, (D.E.C.) Modules[6]. The logic was implemented using 'R' series logic modules which can operate up to a frequency of 2 M.HZ. These modules use negative 'NAND' logic in which 0 volts represent a '0' state and -3 volts represents a '1' state.

CHAPTER II

VECTOR GENERATOR

I. INTRODUCTION:

The vector generator provides the signals required to draw a line vector on the cathode ray tube screen. The signals that are used to generate vectors could be obtained by various methods. They are, the analog method, the digital or combination of the two methods.

The data from the computer, for vector generation, is transmitted to the display, in one of the two formats[7]. In one format the co-ordinates of the end points of the vector are specified. In the other format the X and Y-increments of the vector are specified. For this vector generator subsystem, the latter method was adopted. Thus, to generate a vector, the computer needs to supply information, regarding the starting point of the vector and then the X and Y-increments of the vector.

The process of vector generation involves the generation of positive or negative ramps of the same time duration. These ramp signals are fed into the

X and Y deflection circuitry of the cathode ray tube and then a vector is drawn on the screen, due to the 90° relationship between the X and Y-axis of the deflection subsystem. The length of the vector drawn is proportional to $\sqrt{\Delta X^2 + \Delta Y^2}$, where ΔX is the X-vector increment and ΔY is the Y-vector increment. The angle which the vector makes with the horizontal is equal to $\tan^{-1} (\Delta Y/\Delta X)$. Since ΔX or ΔY can be either positive or negative, the direction of the vector depends on the polarity associated with the X and Y ramp signals.

As mentioned earlier, to generate a vector, the computer specifies the direction and length of X and Y vector increment. One bit of the data could be used to specify the direction of the increment in the data. The length of the increments could be specified by any number of bits, the greater the number of bits, the greater is the accuracy of the generated vector. After a vector instruction is received, all the following output from memory is considered as data (ΔX and ΔY). This process of transferring data to the vector generator continues until a new instruction is encountered.

A 10-bit register was used for the digital to analog (D/A) converter. Thus there are 1024 distinct positions along each axis. A 8 x 10-inch

INSTRUCTION WORD:-

1	0	0	0	0		F	TYPE		L	INT
Bit:- 0	1	2	3	4	5	6	7	8	9	10	11

F: 0- No frame Synch.

1- Frame Synch. on

L: 0- Light pen insensitive

1- Light pen sensitive

TYPE:-

Bit 7	Bit 8	VECTOR TYPE
0	0	POSITION
0	1	TIP
1	0	DASH LINE
1	1	SOLID LINE

INT:-

BIT 10	Bit 11	Intensity
0	0	NORMAL
0	1	DIM
1	0	BRIGHT
1	1	BLANKED

DATA WORD:-

0	Z				10 - BIT X DATA					
Bit:- 0	1	2	3	4	5	6	7	8	9	10	11
0	B				10 - BIT Y DATA					

B: 0- Blink is off.

1- Blink is on

Z:-

Bit	OVER RIDE OPTION
0	NORMAL
1	BLANKED

INSTRUCTION AND DATA WORD CONFIGURATION

FIGURE 2.1

screen[5] was used, the raster unit being about 0.01 inch. The word formats are shown in figure 2.1. There are four types of vectors that could be generated. They are (a) POSITION (b) TIP (The Intensified Position) (c) DASH LINE (d) SOLID LINE.

Thus in the POSITION (POS) mode, the computer sends out the X and Y-components, and the electron beam is positioned as per the received data, but it is blanked during this mode. TIP mode is the same as the POS mode, except that the electron beam is unblanked during this mode.

In the line mode (DASH or SOLID), the initial starting point data is received from the computer. Then the ΔX and ΔY-increments are received which are then used to generate the proper ramp signals. These are then fed into the X and Y deflection circuitry of the cathode ray tube. While the vector is thus being generated, the Z-axis of the CRT is unblanked, thus producing a visible line on the screen.

II. ANALOG VECTOR GENERATOR:

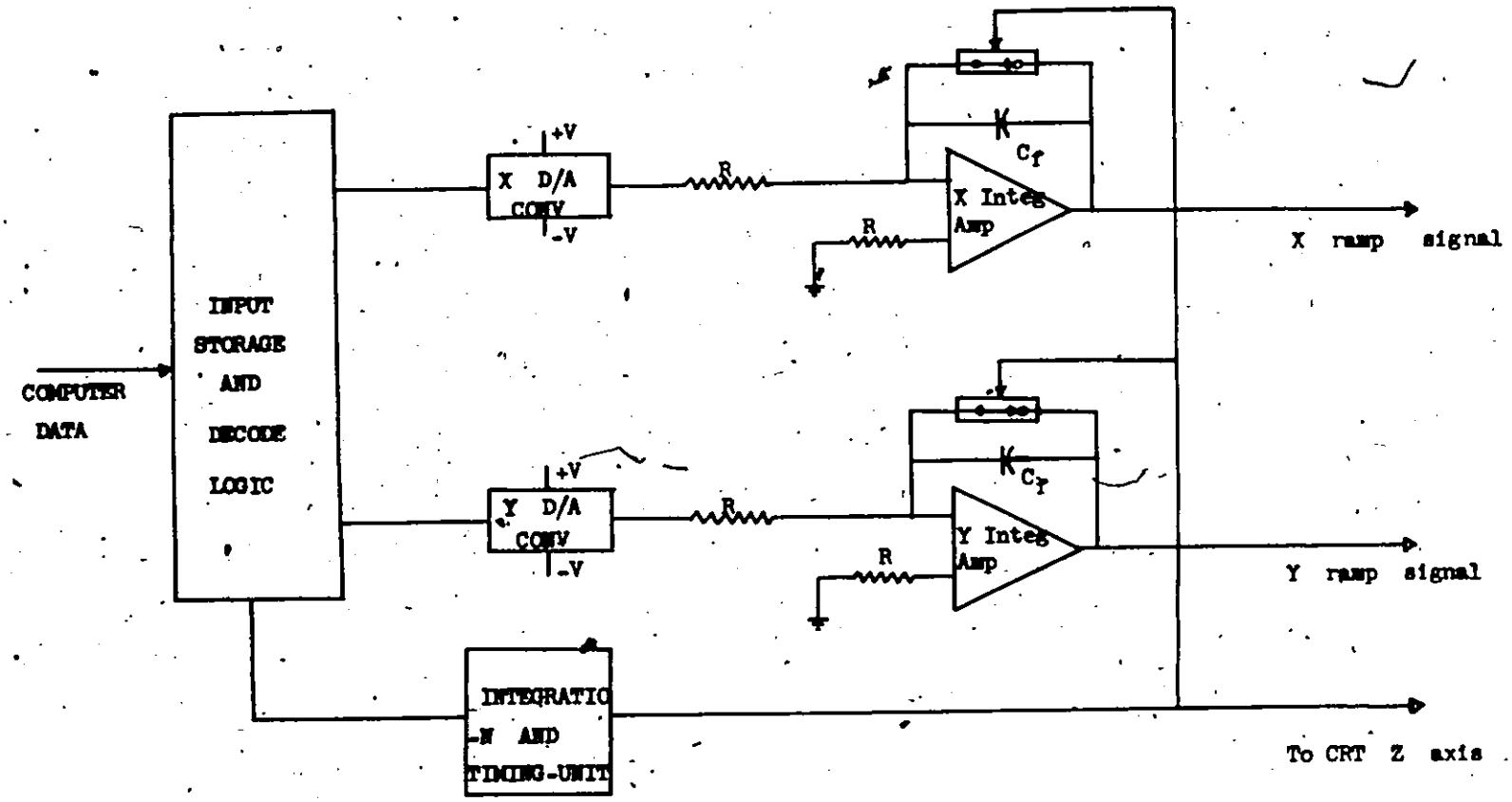
The analog method of vector generation[8] shown in fig. 2.2, employs an X and Y integrating amplifier to produce the required ramp signals. The amplitude and direction of the ramp signals are controlled by the computer data.

Each digital to analog converter in fig. 2.2 receives computer data via the input storage and decode logic. If all the data bits are true, the output of the digital to analog converter will be '+V' volts. If all the bits are false, the output will be '-V' volts. Since the most significant bit (M.S.B.) of data is true when output is positive and false when output is negative, the M.S.B. is considered as the sign (direction) bit. All other bits can be considered to be either ΔX or ΔY-length bits.

The outputs of the D/A converters are used to control the slope of the ramp signals coming from X and Y integrating amplifiers. When the D/A output is negative, a positive ramp will be generated.

The actual generation of the ramp signals is controlled by the switch (shown as manual switch for ease of explanation) placed across the feedback capacitor of the integrating amplifiers.

After the D/A converter output has been applied to the integrating amplifier, and the switch is opened, the amplifier output will start moving linearly. The rate at which the output moves will be determined by the D/A converter output voltage, amplifier gain, and the gain factor RC_f. For a fixed integration time, the output of the integrating amplifier will be a ramp whose direction and amplitude will depend on the D/A converter output polarity and amplitude respectively.



ANALOG VECTOR GENERATION USING X AND Y VECTOR COMPONENT DATA.

FIGURE 2.2

One of the problems associated with the above circuitry of analog vector generation, is the drift[9] of the integrating amplifiers. The drift does not take place when integration is not taking place. This is because the amplifier gain is zero while the switch is closed. The drift could be minimized by making the integration time small and also by adding compensating circuits to the integrating amplifier.

The problem mentioned above gives rise to an error in the slope of the vector drawn. In addition to this, another problem encountered with this form of line generation is the disturbance, or ringing, which occurs at the start of the ramp generation. The disturbance is caused by the transient response of the integrating amplifier. Therefore, the transient response must be controlled by some means of amplifier compensation.

The Z-axis gating signal is obtained from the integrating timing unit which is also used to control the feedback switches. Thus the ramp generation time for both the integrating amplifiers is synchronized with the Z-axis signal so that the required line is displayed on the cathode ray tube. Even though the vector generated using this technique is straight line in pattern (Note - in digital technique it is staircase in pattern), there is always an error involved in

the slope of the vector drawn, due to problems mentioned earlier. Because of these problems of drift with the analogue technique, a digital vector generation technique was investigated.

III. DIGITAL VECTOR GENERATOR:

The digital vector generator[10] makes use of the binary rate multiplier[10][11] (BRM) technique, which is a simple means of manipulating digital signals (i.e. pulse trains) to represent the ΔX and ΔY-increments of the vector.

(a) Binary Rate Multiplier:

A binary rate multiplier is a digital circuit which accepts an input pulse rate and presents a decreased pulse rate at its output. The output pulse rate is the product of the input pulse rate and the binary fraction held in the multiplier register, i.e., the input data register.

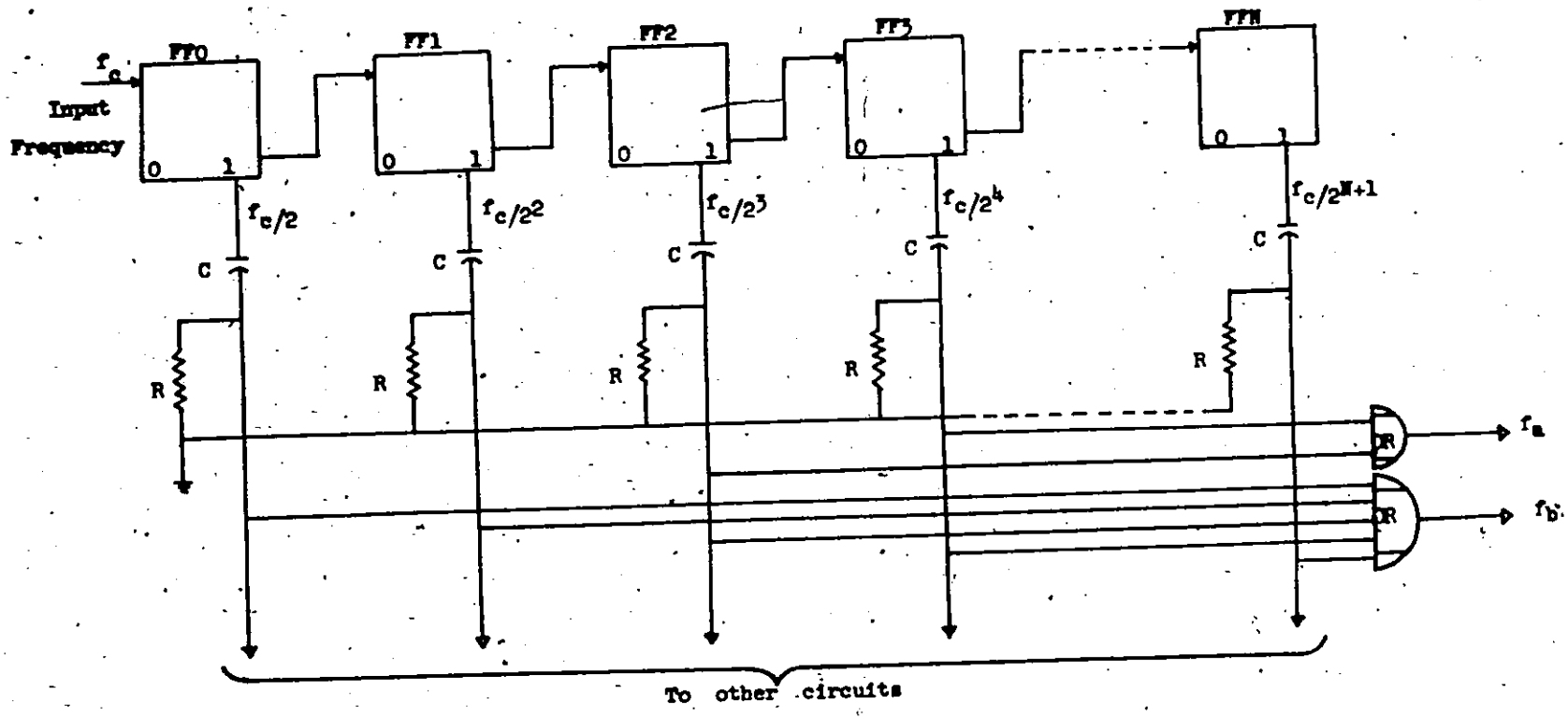
Consider an ordinary n-bit unidirectional counter with the input pulse rate of frequency, f_c, as shown in figure 2.3. The '1' output response of the first four flip-flops in the binary chain is as shown in the figure 2.4(a). In this figure we notice: (a) for any input pulse, except the first one, there is one or more than one 1 → 0 state transitions, (b) for any input pulse there is only one 0 → 1 state transition. It is the later part, i.e., 0 → 1 state transition

which is of importance and, this can be detected by the arrangement shown in fig. 2.3. The '1' outputs of the flip-flops are made to drive a resistor-capacitor differentiating circuit, which in turn provides a pulse corresponding to 0 - 1 transition of the flip-flops. Thus, once the 0 - 1 state transition is detected, then there is a pulse on only one line in a specific clock interval. Then some means of selecting the particular lines could be achieved by having the lines connected to 'OR' gates as shown in fig. 2.3. Thus, any desired pulse rate can be obtained. To achieve this, the outputs of another n-bit register can be compared with the outputs of the main counter. The following meaning could be assigned to these register outputs: a true level indicates a selection of the frequency to which this line is compared; and a 'false' level indicates an omission of the frequency to which it is compared. The selection or omission can be achieved using the simple AND gates as shown in figure 2.5. The output rate will then be given by

$$R = f_c \sum_{i=1}^{N} a_i/2^i$$

where the a_i are either '1' or '0' level of the i^{th} flip-flop in the data register.

For the four bit BRM shown in fig. 2.5, the number in the data register is $(1011)_2$. Thus,

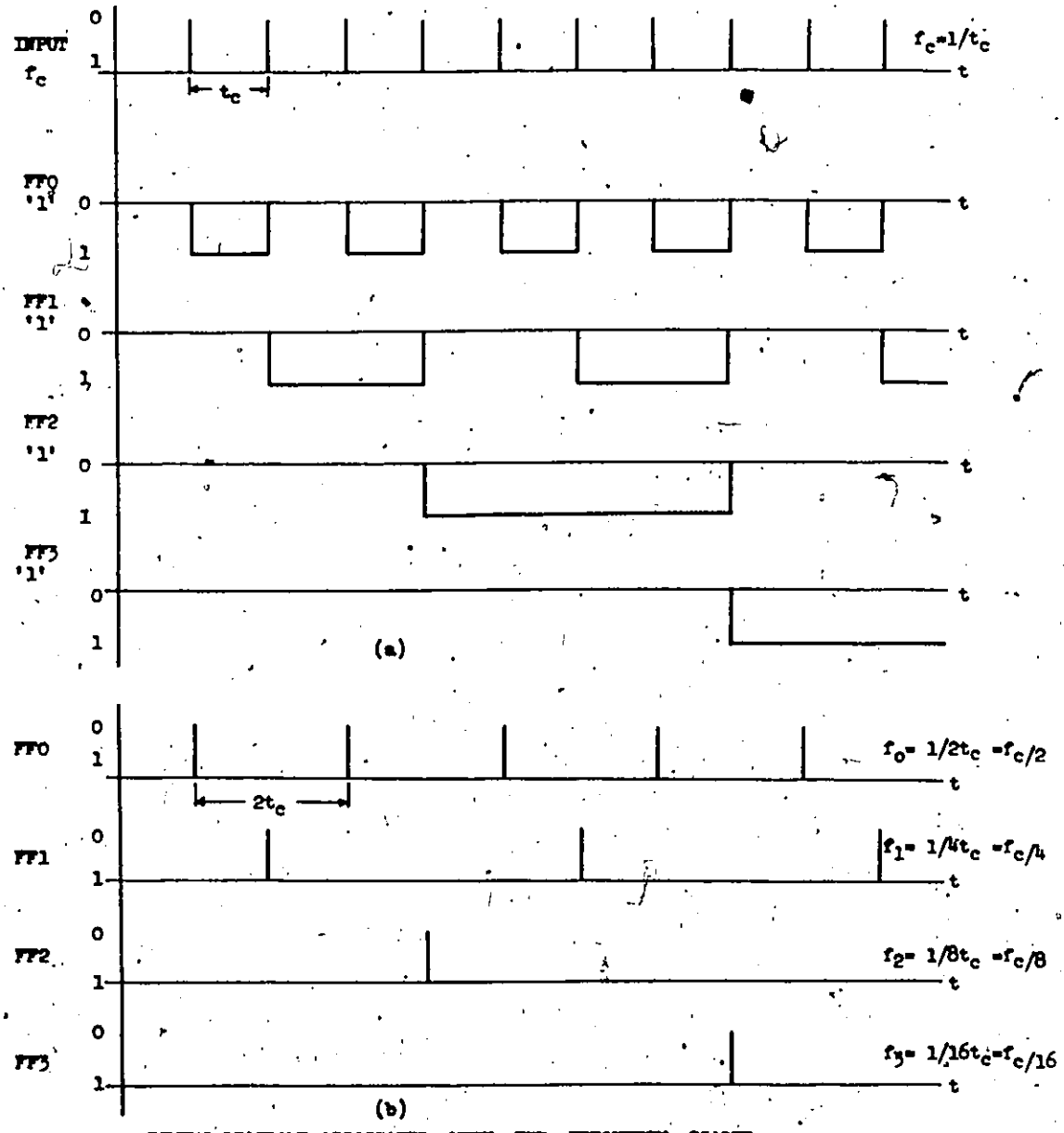


$f_a = f_c/2^3 + f_c/2^4 = 3f_c/16$

$f_b = f_c/2 + f_c/4 + f_c/8 + f_c/16 + f_c/2^{n+1}$

CONSTANT FREQUENCY SELECTION

FIGURE 2.3



TIMING DIAGRAMS ASSOCIATED WITH THE FREQUENCY SCALER

(a) indicates the counter FF changes on the ''1'' output, and
(b) indicates pulse train formed from 0→1 transitions of the FF

FIGURE 2.4

as $a_4 = 1$, the output line of FFO is selected whose output pulse rate is $f_c/2$, $a_3 = 0$, hence output pulse rate $f_c/4$ of FF1 is omitted, but the rest of data $a_2 = 1$ and $a_1 = 1$ and hence the corresponding output rates $f_c/8$ and $f_c/16$ are selected. Thus, the output rate is $(11/16)f_c$. In other words, if we think of the number stored in the data register, as the binary fraction $(0.1011)_2$ which is equal to $(11/16)_{10}$, then the output rate is the product of this binary fraction and the input pulse rate f_c, i.e. output pulse rate is $(11/16)_{10} * f_c$. Thus, as shown in figure 2.6, for 16 input pulses the output will consist of 11 pulses.

This output rate from BRM can then be used to increment or decrement an accumulating (another n-bit, up-down counter) whose contents represent the instantaneous value of the parameter involved.

Thus the basic idea of the incremental digital technique, then, is the horizontal and vertical increments of a vector are represented by pulse rates from a pair of BRMs which feed into summing or accumulating registers, the outputs of which are used to position the electron beam of a cathode ray tube in discrete steps with time. A general arrangement is shown in the figure 2.7. X-data is the X-increment of vector and Y-data is the Y-increment of the vector. These are loaded into X and Y-data registers by the

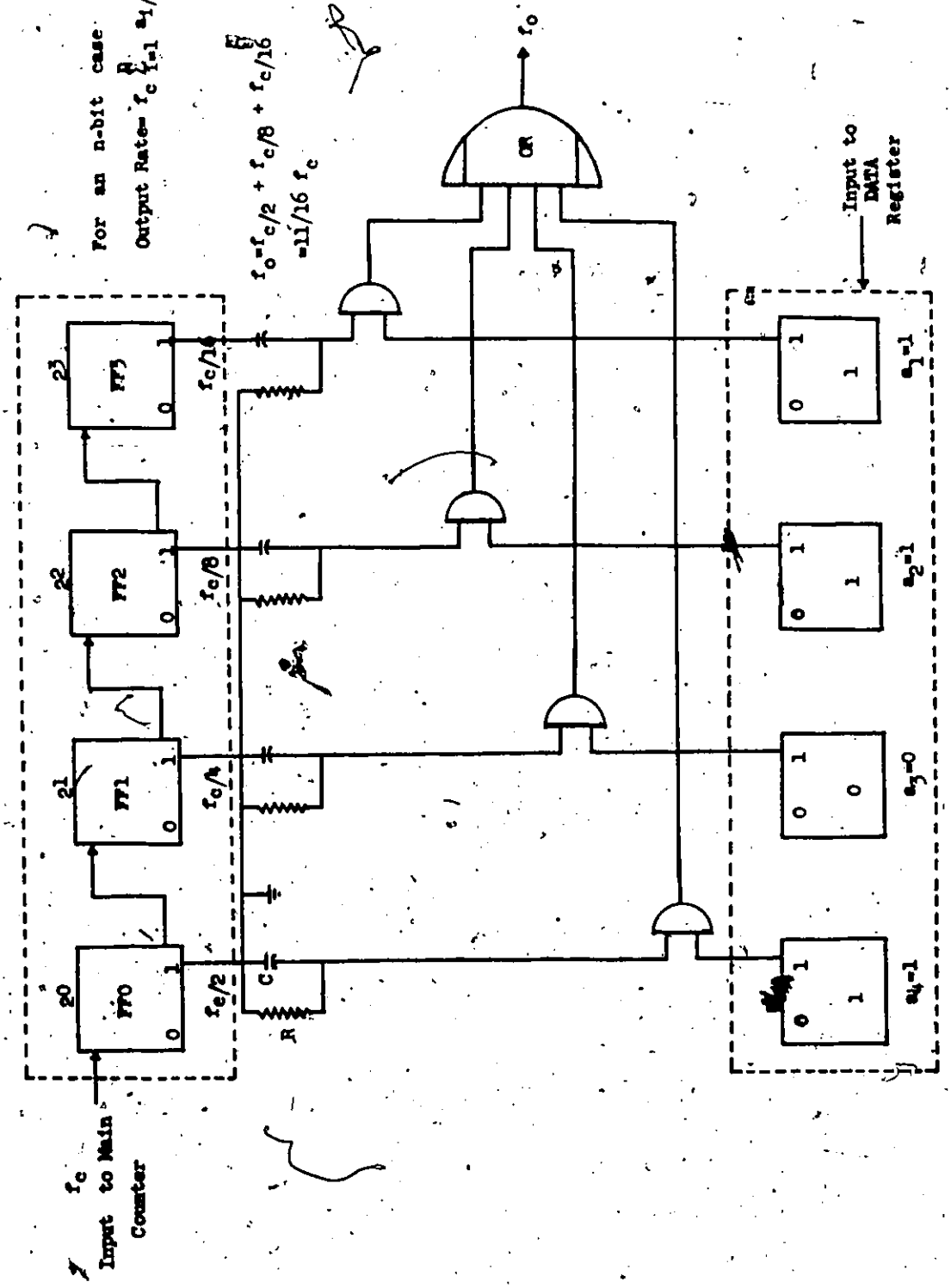
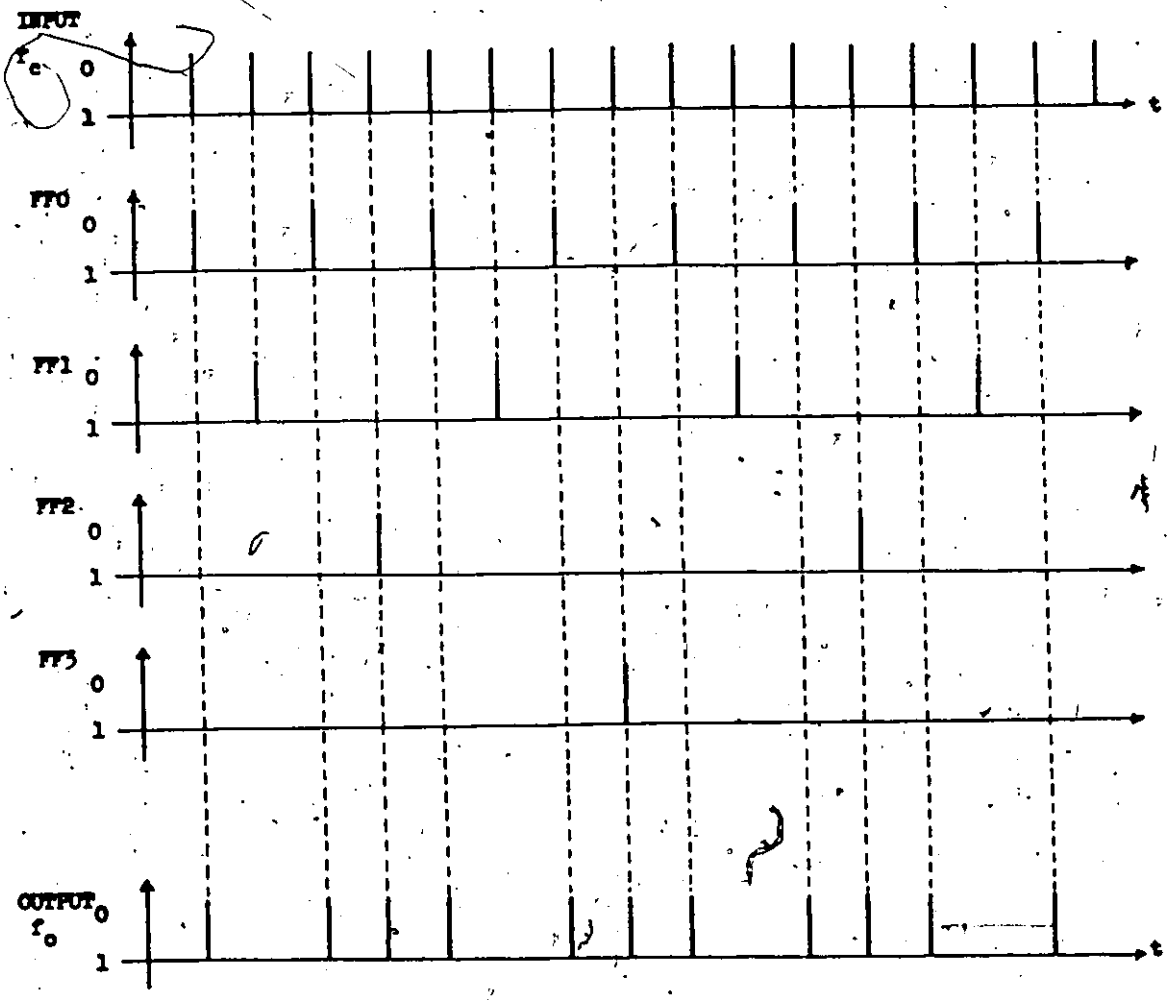

For an n-bit case
Output Rate= $f_c \sum_{i=1}^{n} a_i/2^i$
$f_o = f_c/2 + f_c/8 + f_c/16$
$= 11/16\ f_c$
f_o
OR
Input to DATA Register
FF3
FF2
FF1
FF0
2^3
2^2
2^1
2^0
$f_c/16$
$f_c/8$
$f_c/4$
$f_c/2$
R
C
a_1-1
a_2-1
a_3-0
a_4-1
f_c
Input to Main Counter
Four-bit DDM Example

FIGURE 2.5



Timing diagram and output pulse rate of BCH example.

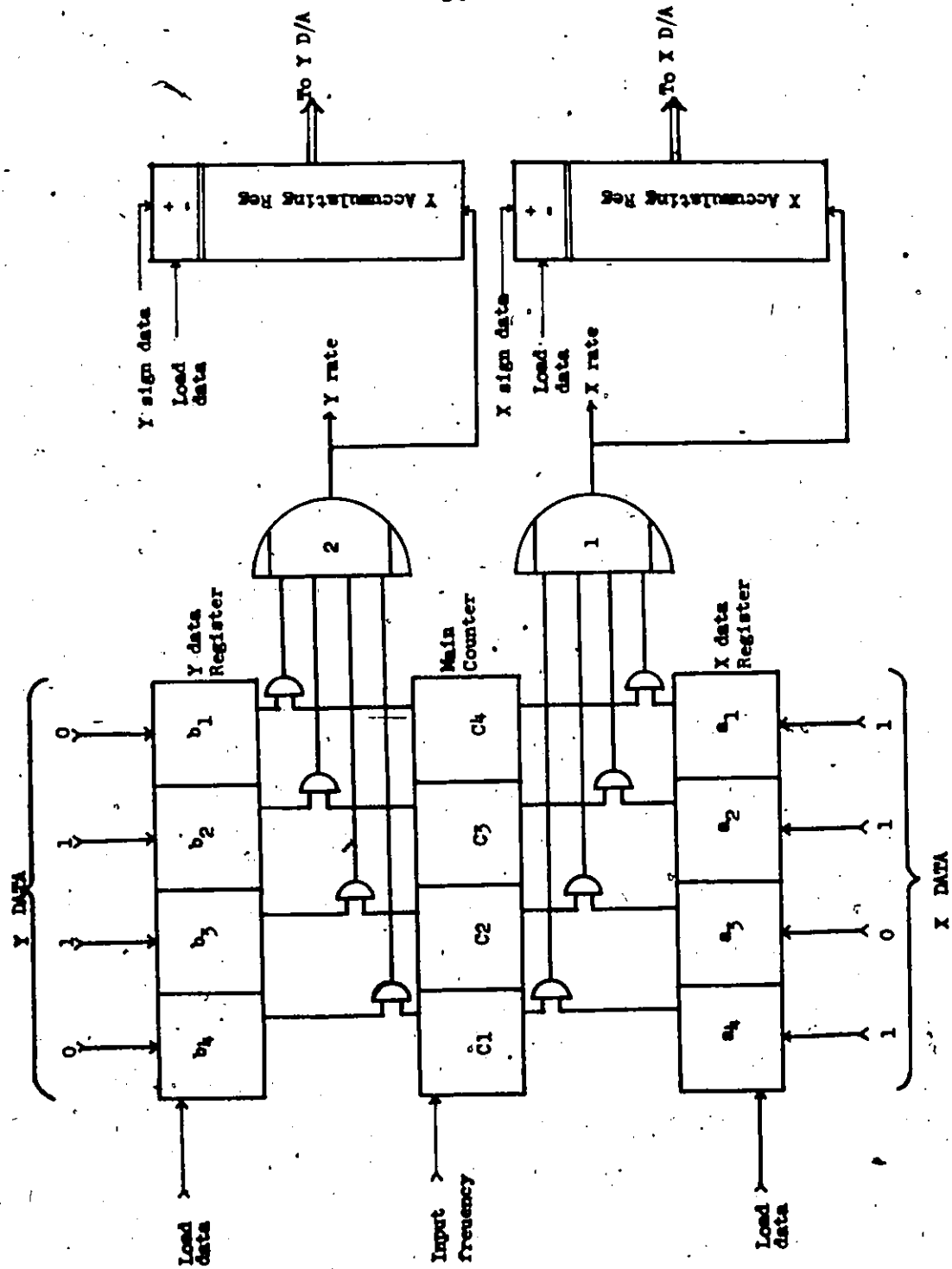
FIGURE 2.6

load data pulses. b_4, b_3, etc., and a_4, a_3,...etc., represent the contents of each flip-flop of the data registers Y and X respectively. C1, C2, C3, and C4 are the flip-flops of the main counter whose output lines are ANDed with the output lines of data registers. Thus the output of OR gate 1 is the X-rate and that of the OR gate 2 is the Y-rate. These are then fed into the accumulating registers which would count up or down depending on the X or Y sign. The outputs of the X/A are then used to position the electron beam. Obviously the outputs of the D/A's are staircase in shape. Therefore, the vector generated is staircase in pattern. But by making the increments of the resulting staircase pattern small enough, the desired continuous line can be represented with a good degree of resolution. It is important to indicate at this point, that if the data register of a BRM is also made to count by using the outputs of some BRM's as input to others, more complex functions than straight lines can be generated. In particular, second order curves can be formed from a pair of BRM's.

It is worthwhile to point out some of the errors[11] associated with this method of vector generation. The output rate of BRM is exact only if the number of input pulses equals 2^n (for an n-bit counter). For less than 2^n counts the output rate of the BRM is only approximately the one desired,

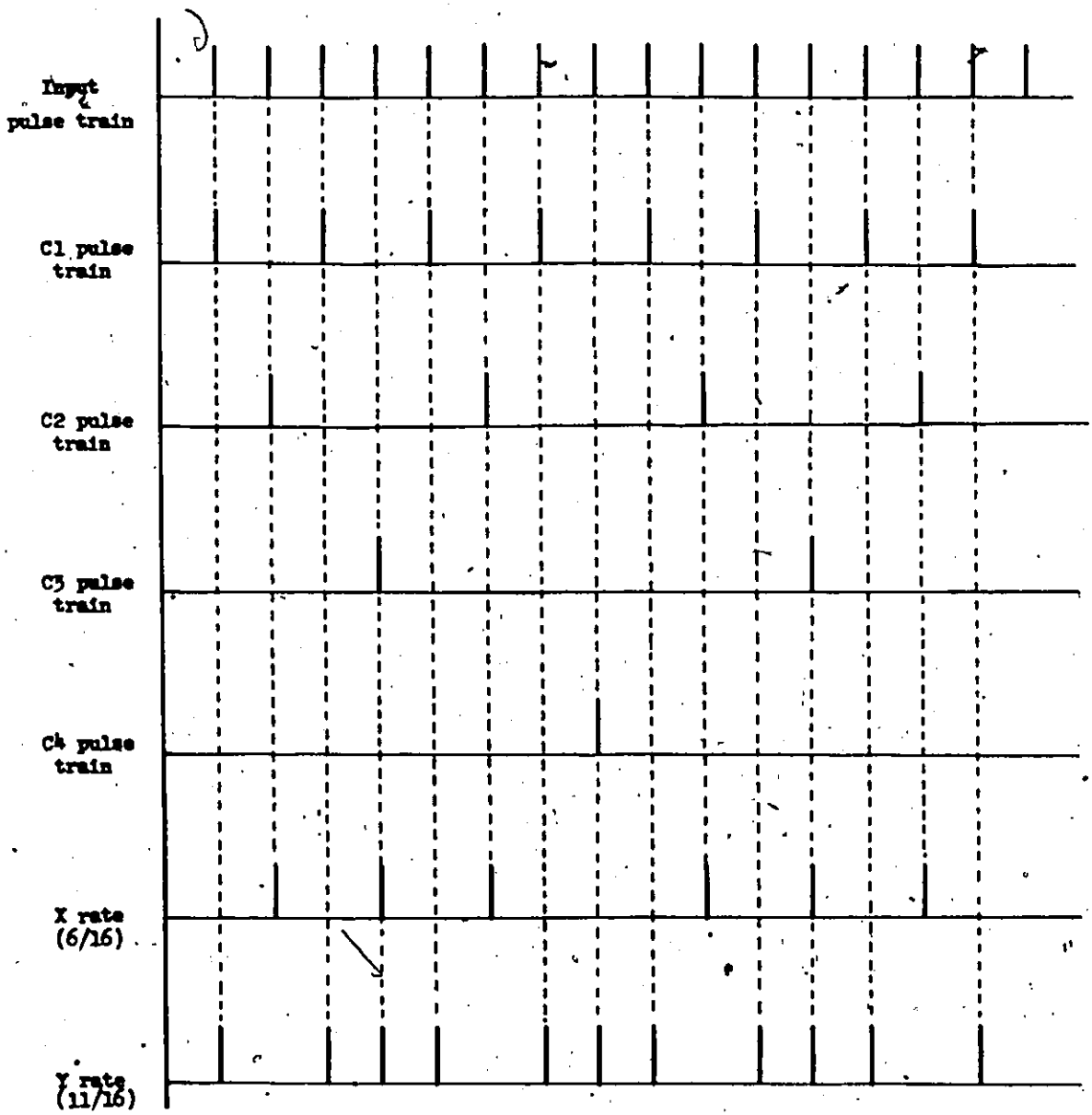
and the deviation from true rate decreases with an increasing number of input pulses, and is zero only for multiples of 2^n counts. In practice, with appropriate scaling, the error is almost negligible. This round-off error becomes very difficult to evaluate because the number that is stored in the data register is variable, thus the error then becomes a dynamic one.

The second type of error[11] associated with this method could be seen in the output pulse rate shown in Fig. 2.8. If now, the X-data register contained a number 0110 and Y-data register contained a number 1011 as shown in figure 2.7, then the number in the X-data register could be, as shown in figure 2.7, thought of as binary fraction 0.0110_2, i.e. equal to $(6/16)_{10}$. Thus, the X pulse rate would contain 6 pulses & Y-rate contains 11 pulses corresponding to 16 input clock pulses into the main counter. It is clearly seen that the output pulses of the BRM are unevenly spaced in time. This is the second drawback of the BRM technique. Due to the uneven spacing of output pulses of BRM, the corresponding D/A output will have an uneven series of steps as shown in figure 2.9. It shows the Y pulse rate $(11/16)_{10}$ and the corresponding Y ramp, i.e. output of Y D/A which has 11 uneven steps and also X pulse rate $(6/16)_{10}$ and the corresponding X D/A output which has 6 uneven



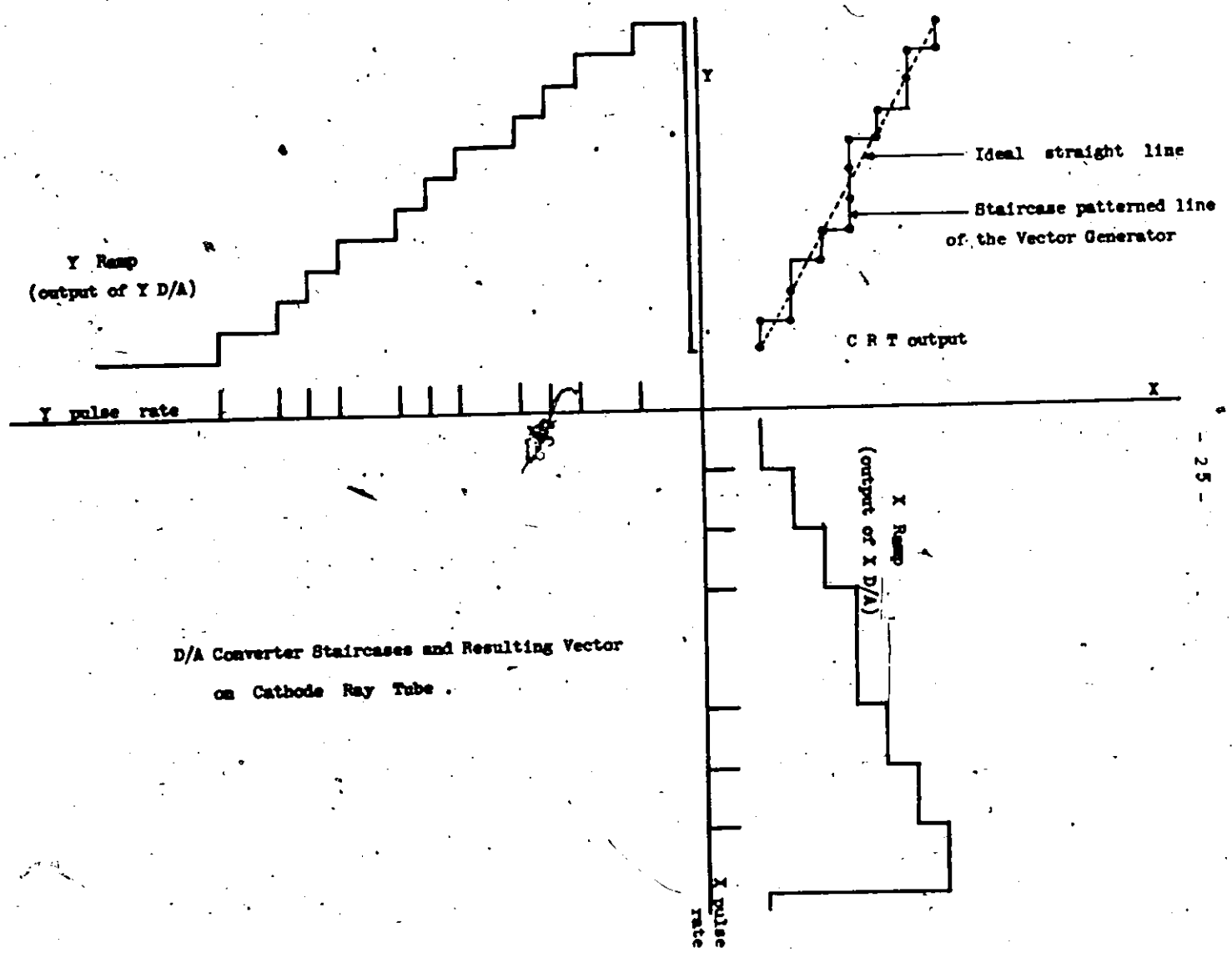
Arrangement of two Four-bit B R Ms to obtain X and Y rates

FIGURE 2.7



X and Y rates corresponding to datas stored in X and Y registers of the two 4-bit B R Ms .

FIGURE 2.8



D/A Converter Staircases and Resulting Vector on Cathode Ray Tube .

FIGURE 2.9

steps. Thus this uneveness causes irregular error when generating straight line functions. The resulting staircase pattern generated by the multiplier technique is thus irregularly distributed around the ideal straight line because at certain points of the path, the error or deviation from the ideal is considerable as seen in fig. 2.9, and at others the actual path is accurate. However, with appropriate scaling the excursion from the desired path can be kept within allowable tolerances; similarly if the steps generated are small enough, the path generated by the multiplier technique is acceptable.

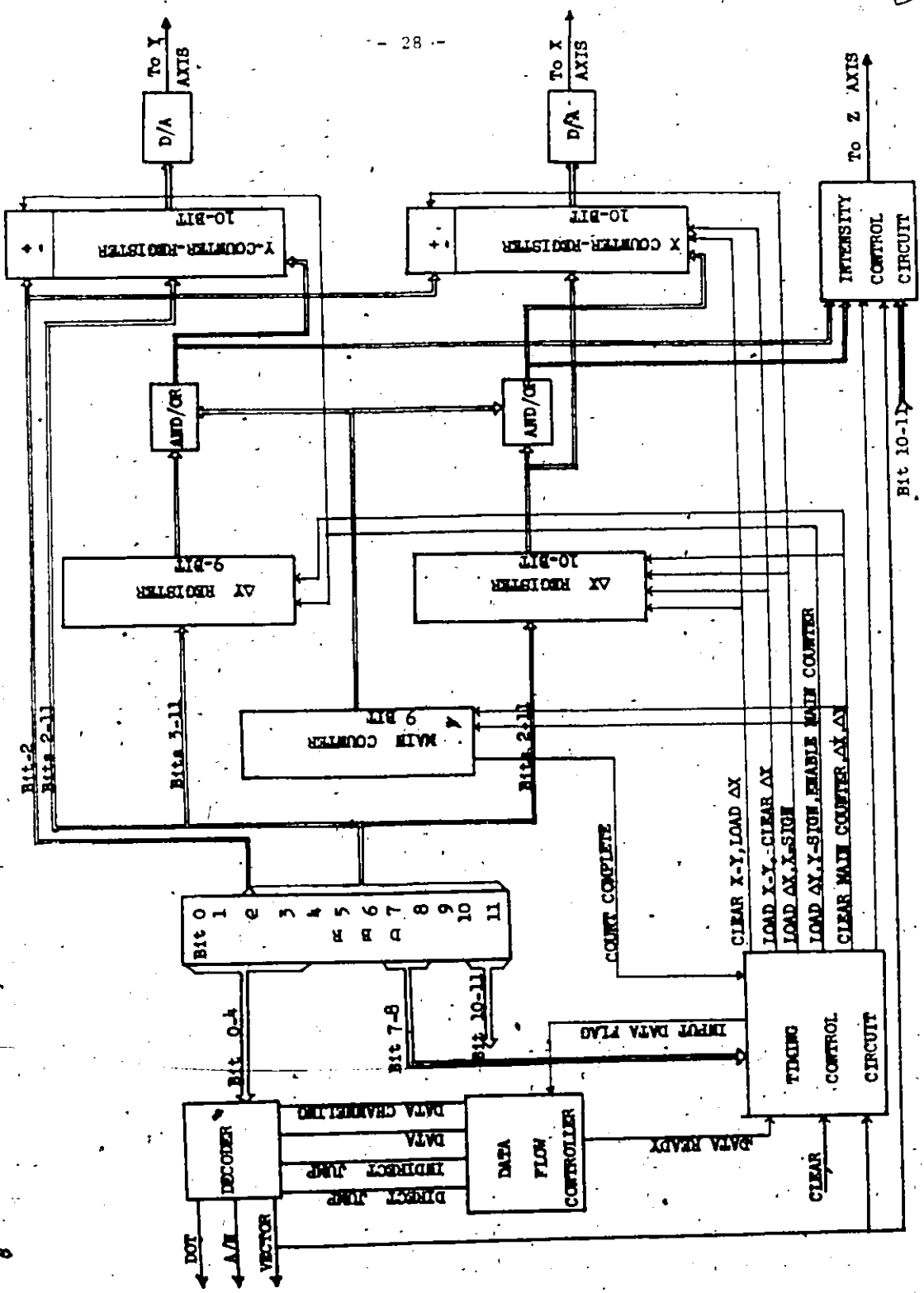
(b) Hardware Design:

The complete vector generator system is shown in the block diagram representation of figure 2.10. The ΔX-register is the X-increment data register and ΔY is the Y-increment data register. As mentioned earlier, the X or Y-increment data are 10-bit each (bits 2-11 of DBR) but the 0^{th} bit is the sign bit which is loaded into the sign bit flip-flop of the counter register. Thus the actual data is 9-bits each (bits 3-11) and hence ΔY-register is a 9-bit register. ΔX-register is a 10-bit register because it is made use of during Position Mode, during which a 10-bit X-data is temporarily stored in ΔX-register before it is being loaded into the X-counter register (the

accumulating register). During line mode, i.e., vector mode the 10^{th} bit of the data is inhibited from loading into 10^{th} bit of ΔX-register and thus ΔX-register acts as 10-bit data register while in Position Mode, but as a 9-bit data register while in vector mode.

The ΔX-register, ΔY-register with their associated AND/OR gate circuitry and the MAIN counter constitutes the two BRM's for producing the X and Y rate pulses which are fed into the X and Y-counter-registers (accumulating registers). The main counter is a 9-bit unidirectional counter which along with the timing control circuit, controls the ramp generation time.

A data ready pulse is sent by the Data Flow Controller (DFC) and Decoder circuits, soon after the data is loaded into the Display Buffer Register (DBR). As mentioned earlier, bits 7-8 are used to indicate the type of the vector and these are loaded into the decoder circuit of the timing control unit. Bits 10 → 11 are loaded into the intensity control circuit. As soon as the data ready pulse (DRP) is received by the timing control unit, it starts to process the data and simultaneously sets the Input Data Flag (IDF) to '0' state indicating to the DFC that no more data is required. Soon after it has processed the data, the timing control



Block Diagram of Digital Vector Generation System

FIGURE 2.10

now sets the IDF to '1' state indicating to the DFC that more data is required.

The functions of the BRM, main counter, timing control circuit, etc., are presented in detail in the following sections.

(i) Main Counter:

The main counter is the heart of the binary rate multiplier. It is shown in figure 2.11. The lower part of the circuitry is a part of the timing control circuit.

In this case the extraction of the pulse corresponding to 0 → 1 transition of the flip-flops is carried out in a manner that is different from that shown in figures 2.3 or 2.5. The circuits in figure 2.3 or 2.5 employ resistor-capacitor differentiating circuits which are connected to the '1' side or the true side of the flip-flops and these provide the pulses to the AND/OR circuitry. The problem encountered with these circuits, is that there is ringing or disturbance at the output, due to the transient response of the capacitor and the output of the circuit contains more than a clock pulse which may disable the operation of the rest of the circuitry connected to the main counter.

In order to avoid the above problems, the circuitry shown in fig. 2.11 was designed which

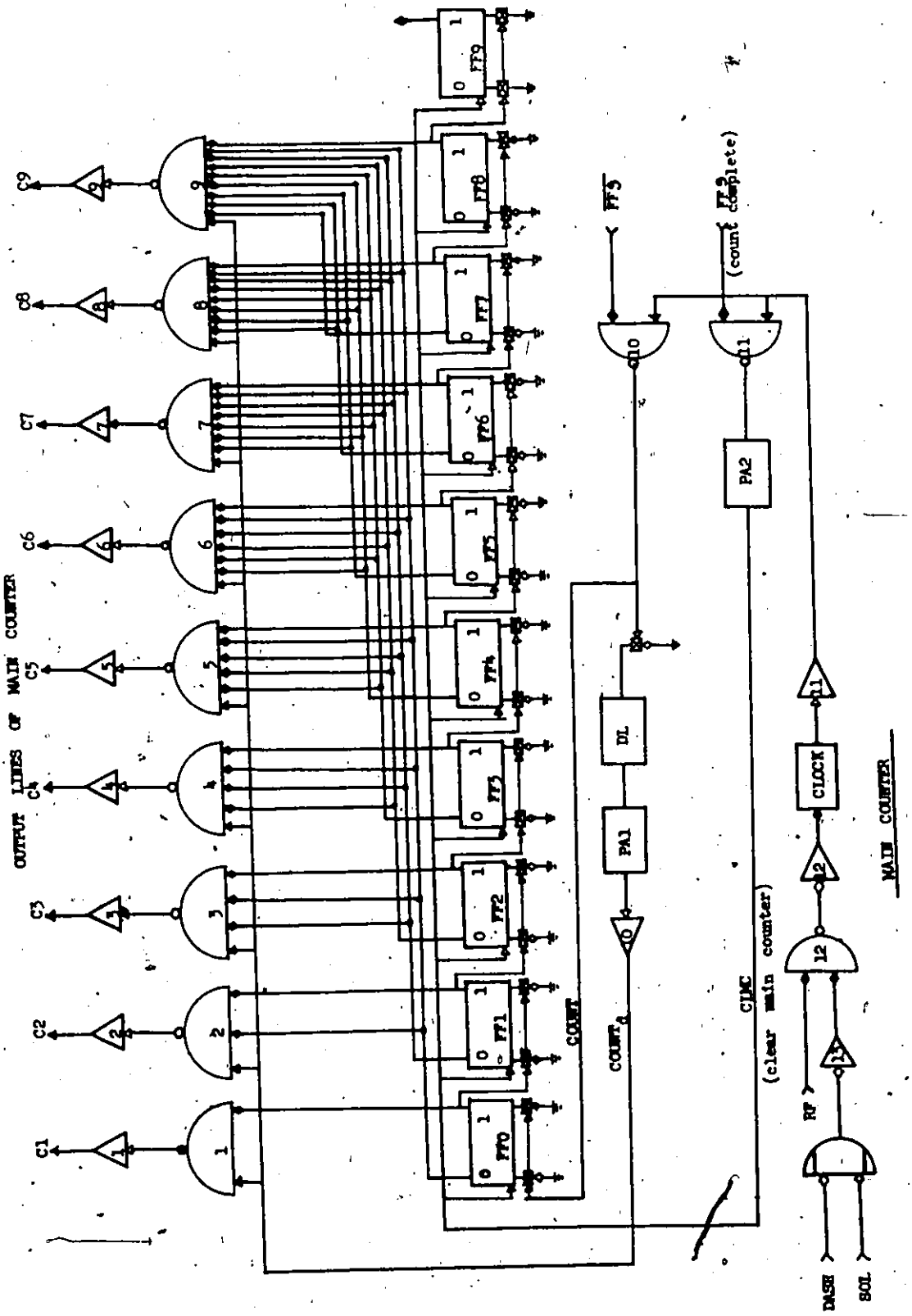

OUTPUT LINES OF MAIN COUNTER
C1
C2
C3
C4
C5
C6
C7
C8
C9
FF0
FF1
FF2
FF3
FF4
FF5
FF6
FF7
FF8
FF9
COUNT
COUNT_d
DL
PA1
PA2
FF5
(count complete)
CLMC
(clear main counter)
CLOCK
RF
DASH
SOL
MAIN COUNTER

FIGURE 2.11

makes use of digital components, to achieve the same purpose (extraction of pulse corresponding to 0 → 1 transition of the flip-flops).

In order to clearly see this, let us assume there is a continuous flow of COUNT pulses to the counter. The first COUNT pulse that is transmitted from NAND gate 10 sets FFO to '1' state. Thus, NAND gate '1' is enabled, but the rest of the NAND gates (2 → 9) are disabled as $\overline{FFO}$ is gated to them. So the $COUNT_d$ pulse which is the delayed COUNT pulse, is now transmitted through NAND gate 1 and through INVERTER '1' thus indicating a 0 → 1 transition of the FFO flip-flop of the counter. The second count pulse would set the FF1 flip-flop to '1' state, thus enabling the NAND gate 2 while the rest is disabled, and hence the second $COUNT_d$ pulse is transmitted through NAND gate '2' and INVERTER '2'. This pulse would then correspond to 0 → 1 transition of 'FF1' and so on for the remainder of the count pulses.

Boolean expressions for the pulse trains C1, C2, C3,.....C9 could be written as below.

$C1 = FFO \cdot COUNT_d$ pulse train

$C2 = \overline{FFO} \cdot FF1 \cdot COUNT_d$ pulse train

$C3 = \overline{FFO} \cdot \overline{FF1} \cdot FF2 \cdot COUNT_d$ pulse train

.

.

.

$C9 = \overline{FFO} \cdot \overline{FF1} \cdot \overline{FF2} \cdot \ldots \cdot \overline{FF7} \cdot FF8 \cdot COUNT_d$ pulse train

The '.' represents AND operation.

Thus, from the above discussion, it is clear that for any one COUNT pulse, there would be a pulse output from only one of the nine inverters at the output lines of the MAIN counter corresponding to the 0 → 1 transition of a particular flip-flop of the counter.

(ii) AND/OR Gate Circuitry:

The essence of the binary rate multiplier technique is in the selection of the output lines of the main counter to form the pulse rate representing the desired data. The basic element of selection circuit used is the NAND gate. One input to each of the NAND gate is provided by the main counter and the other input to each of the NAND gate is provided by the data register. It should be noted that the most significant bit of the data register is compared with least significant bit of the main counter. By comparing in this manner, i.e. by NANDing them, (and similarly for the remaining nine successive bits of both registers) we can detect any 0 → 1 transition by NANDing the NAND gate output as shown in the figure 2.12. Thus the combination acts as an AND/OR circuitry.

The boolean expression for the pulses P1, P2,....,etc., are

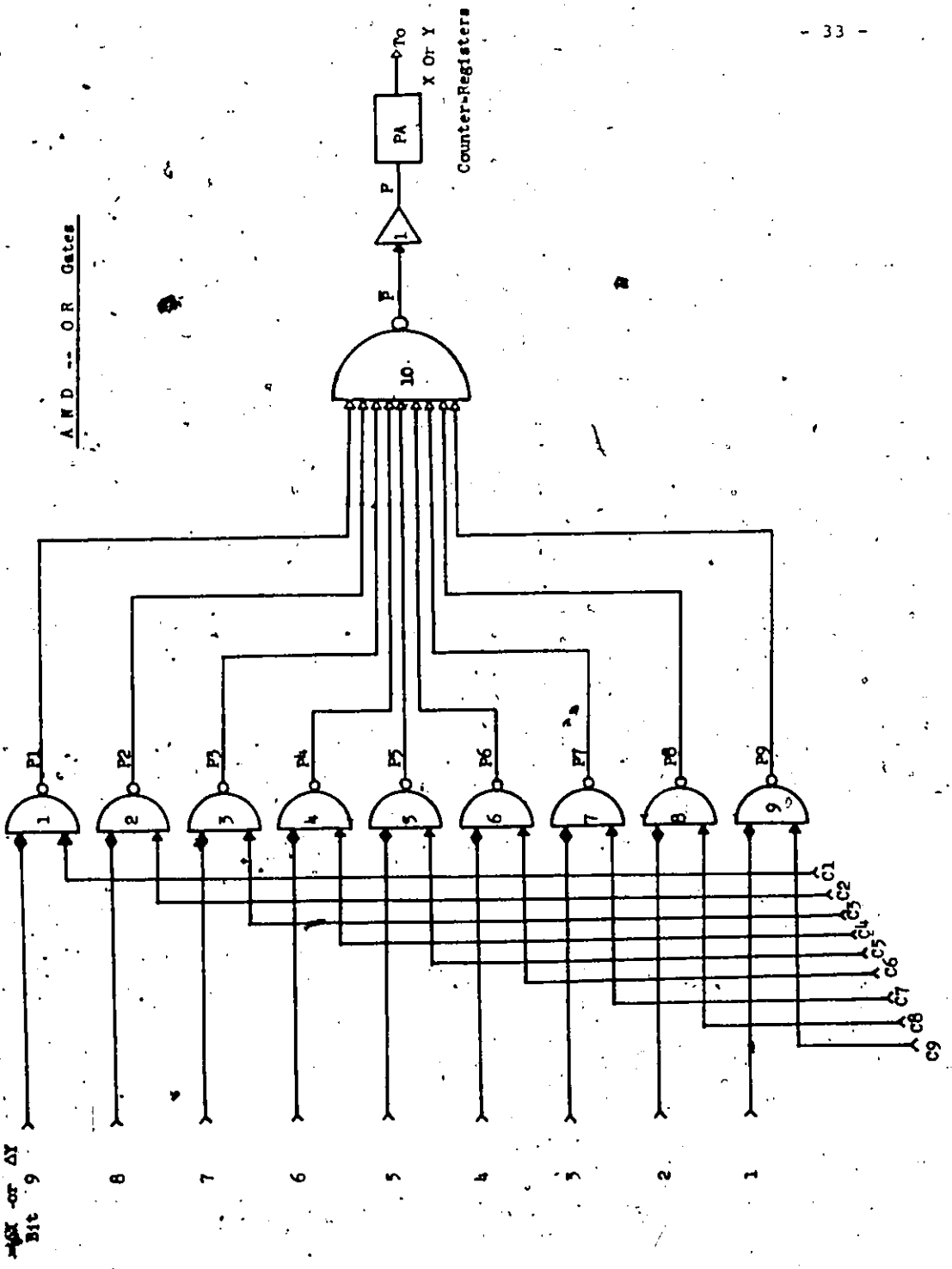

AND -- OR Gates
To
X Or Y
Counter-Registers
PA
P
1
F
10
P1
P2
P3
P4
P5
P6
P7
P8
P9
1
2
3
4
5
6
7
8
9
C1
C2
C3
C4
C5
C6
C7
C8
C9
Bit
9
8
7
6
5
4
3
2
1

FIGURE 2.12

$$P1 = \overline{C1 \cdot \Delta 9}$$

$$P2 = \overline{C2 \cdot \Delta 8}$$

$$P3 = \overline{C3 \cdot \Delta 7}$$

$$\vdots$$

$$P9 = \overline{C9 \cdot \Delta 1}$$

These pulse trains are positive going and the output pulse train $\overline{P}$ of NAND gate 10 is given by

$$\overline{P} = \overline{P_1 + P_2 + P_3 \ldots + P_9}$$

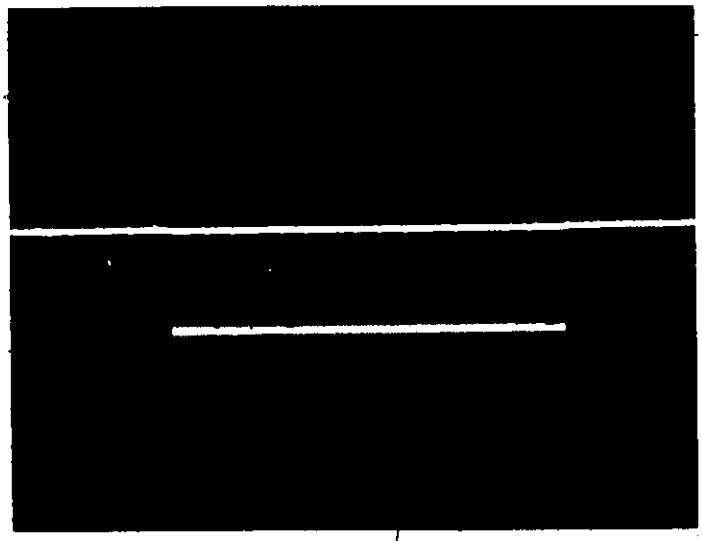
Since the negative NAND acts as a positive NOR gate for positive going signals, the pulse train at the output of the inverter is

$$P = P_1 + P_2 + P_3 + \ldots + P_9$$

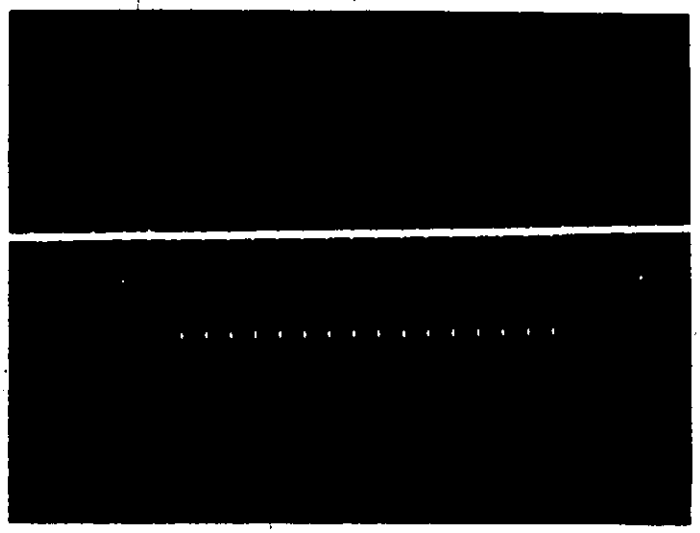
This pulse train, now corresponds to the desired data and is transmitted to the accumulating registers, i.e. COUNTER-REGISTERS. Two such configurations are provided, one for the horizontal increment and the other for the vertical increment of the vector.

Figures 2.12A shows the output pulses of X and Y BRM's and the corresponding ramps of vector, which has a large X-increment and a small Y-increment.

Output of the X BRM



Output of the Y BRM



Corresponding X and Y Ramps

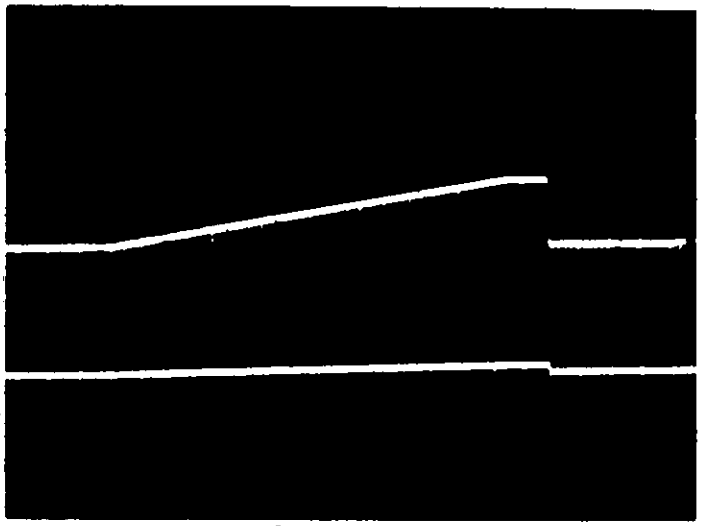


FIGURE 2.12A

(iii) TIMING CONTROL CIRCUIT:

The DFC and Decoder Circuits[1] shown in figure 2.10 generates mode pulses such as Dot, A/N, vector and data ready pulses. Besides these there are several other signals that should be generated for vector generator subsystem. This is carried out by the timing control circuit. It directs the loading of data from DBR into the ΔX, ΔY-data registers, X and Y counter-registers and sign bits and also control of ramp generating time.

The timing control circuit is divided into two parts. The first part shown in figure 2.13 controls the loading of data from DBR into the various registers and the second that controls the time of the generation of the ramp signals for the vector display. The second part is associated with the main counter shown in the lower part of the figure 2.11.

Before a mode selection pulse (Dot Mode, Vector Mode or A/N Mode) is sent a clear instruction pulse (CLI) is sent out by the core memory. It will clear all the flip-flops in the figure 2.13 except the vector data flip-flop (VDFF), i.e. initializes them to '0' state at the true side of the flip-flop.

When a vector instruction is encountered a set vector mode pulse (SVM) is sent from DFC and

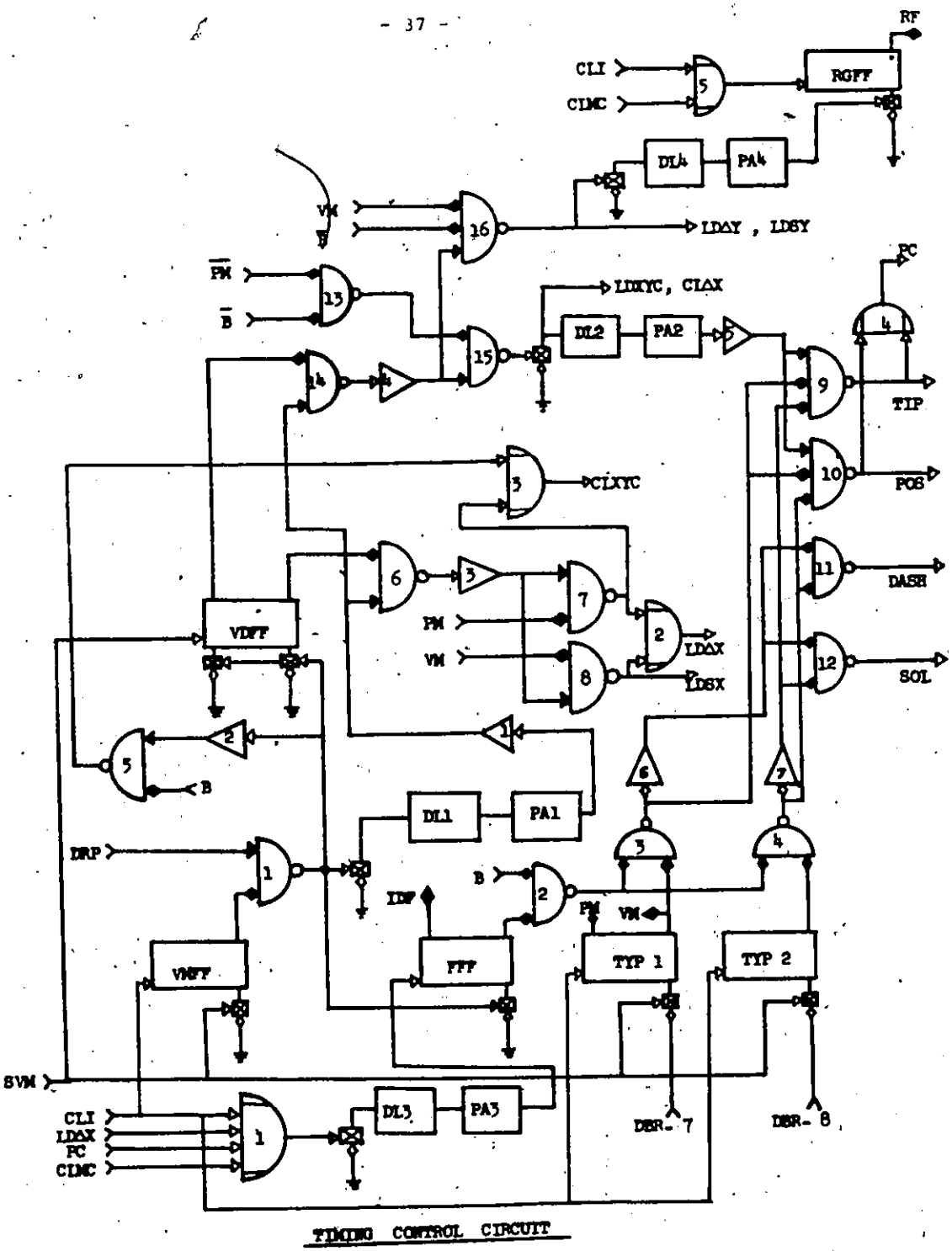

RF
CLI
CLMC
5
RGFF
DL4
PA4
LDΔY , LDSY
16
PM
B
13
LDXYC, CLΔX
DL2
PA2
15
14
PC
4
9
TIP
10
POS
3
CLXYC
6
11
DASH
7
2
LDΔX
LDSX
12
SOL
VDFF
8
1
5
B
DL1
PA1
DRP
1
IDF
B
2
PM
VM
3
4
VMFF
FFF
TYP 1
TYP 2
SVM
CLI
LDΔX
PC
CLMC
1
DL3
PA3
DBR- 7
DBR- 8
TIMING CONTROL CIRCUIT

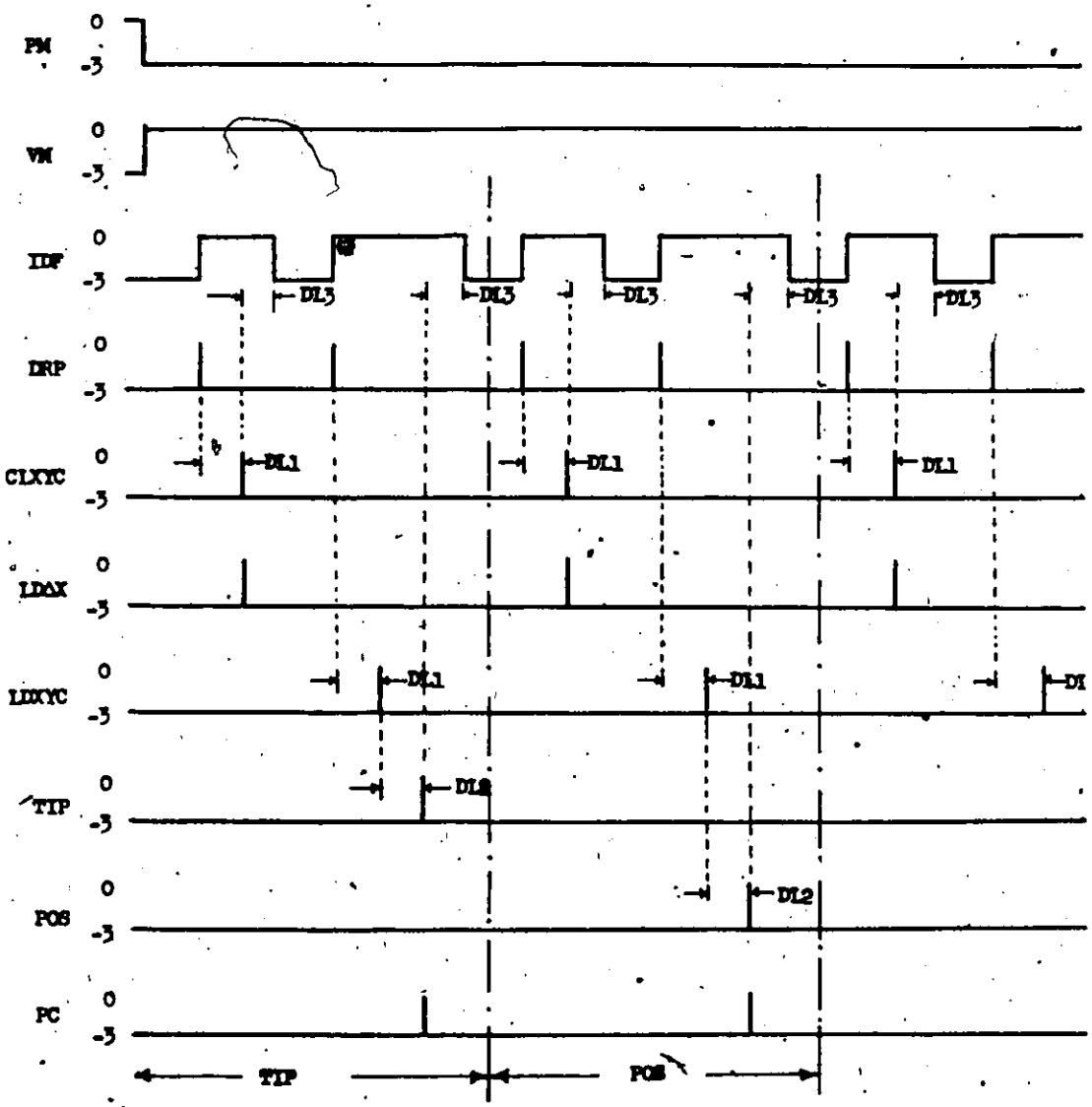
FIGURE 2.13

decoder circuits and this pulse sets VMFF to '1' state and VDFF to '0' state. At the same time, bits 7 and 8 of display buffer register is enabled to transmit into decoder flip-flops TYP1 and TYP2. As mentioned earlier, there are four possible types of vectors to be generated, namely, POS, TIP, DASH and SOL.

Assuming a POS or TIP as the instruction that is received, TYP1 is in '0' state. Therefore, PM is at -3 volts, i.e. '1' state. Approximately 400 nano seconds later a data ready pulse (DRP) is received and transmitted through NAND gate 1 to set the VDFF to '1' state and FFF also to '1' state. So the input data flag (IDF) is at '0' level which indicates to the data flow controller that no more data is required. The true side of the flip-flop FFF is NANDed with B which is used for the over-ride option. (This is discussed in the INTENSITY CIRCUITRY.) Usually when over-ride option is not used 'B' is in '0' state. Therefore, the output of the NAND gate '2' is at -3 volts, i.e., at '1' state which enables the decoder to generate POS, TIP, DASH and SOLID vector display. The same pulse which sets VDFF to '1' state is delayed until VDFF is set and then transmitted through NAND gate 6 and then through NAND gate 7 and then again through OR gate 2 to produce the pulse

(load ΔX-register) LDΔX, to load the ΔX-register and at the same time it is transmitted through OR gate 3 to send out CLXYC, i.e. to clear X and Y counter-registers. The LDΔX pulse is then transmitted through OR gate '1' and after a delay it clears the FFF and thus sets IDF to '1' state which indicates to D.F.C. that vector generator is ready to process more data. The DL3 delays the LDΔX pulse (before it clears FFF), thus providing sufficient time for ΔX-register to be loaded properly.

About 8 μ sec. later, i.e., after the FFF has been cleared a second DRP pulse is sent by the DFC. As before, this sets FFF to '1' state and IDF to '0' state indicating to the DFC that no more data is required. The same pulse again compliments the VDFF flip-flop and this time the delayed pulse is transmitted through NAND gate 14, then through NAND gate 15 and then through a pulse amplifier PA4 to produce the pulse LDXYC (load X-Y counter-registers) which loads X-Y counter-registers and at the same time clears the ΔX-register. The LDXYC pulse is then delayed and passed through an amplifier PA2 and then transmitted to NAND gates 9 and 10 of the 'vector type' decoder circuit. The output pulses TIP and POS are 'OR'ed and a position-mode complete (PC) is obtained through OR gate 4 and this pulse is then transmitted through 'OR' gate 1



Timing Diagram for TIP and POS cycles

FIGURE 2.14

and delayed for a while, and then used to clear FFF which sets IDF to '1' state again. The following description would become clear from the timing diagram shown for TIP and POS modes in figure 2.14. TIP signal is also sent into the intensity circuitry, and this unblanks the Z-axis after the beam has been positioned and thus producing an intensified spot on the screen.

When a DASH or SOL instruction is encountered the process of loading data into ΔX and ΔY-registers are almost similar to that of TIP or POS mode. But in this case TYP1 is in '1' state and PM is at ground level, i.e., VM is at '1' state. Thus, the output of NAND gates 11 or 12 will be at '0' level depending on whether the TYP2 flip-flop is at '0; or '1' state, i.e., whether the instruction is DASH or SOL respectively.

Now, as before, the SVM pulse will set the VMFF to '1' state and the first DRP is transmitted through NAND gate '1' and this first sets VDFF to '1' state and then after a delay passes through NAND gate '8', since the instruction is in DASH or SOL, i.e. VM is in '1' state. Thus there is again a LDΔX pulse which loads ΔX-register with the X increment data from DBR and LDSX loads the X-sign flip-flop associated with the X counter-register.

Again, as before, LDΔX after a delay clears FFF and thús sets IDF to '1' state indicating to DFC that more data is required. After about 8 μ secs another DRP follows which complements VDFF. Then the delayed pulse would now be transmitted through NAND gate 14 and then through NAND gate 16, producing LDΔY pulse which is then used to load the ΔY-register with Y-increment data and and Y-sign bit of Y-counter-register with Y-sign data from the DBR. The same pulse after a short delay and pulse amplification would set ramp generation flip-flop (RGFF) and thus RF to '1' state. This enables start of ramp generation for the vector to be drawn.

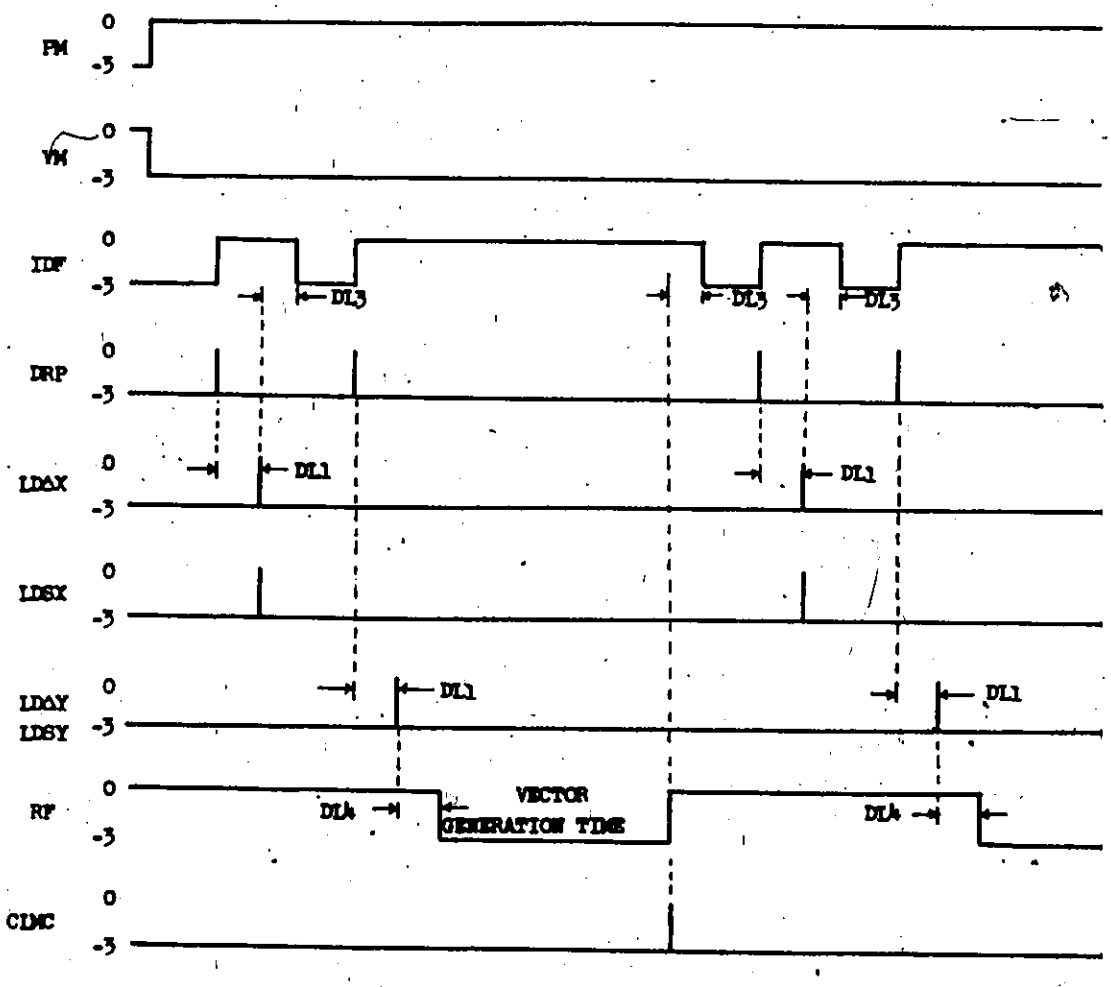
Referring back to figure 2.11, it is seen that as soon as RF is set to '1' state, the NAND gate 12 is enabled (it should be noted that the other input that enables NAND gate 12 is the output of inverter 13, which would be at '1' state, i.e., at -3 volts if either DASH or SOL is the type of vector to be drawn). Thus this makes the output of inverter 12 to go to -3 volts, thus enabling the clock. The clock then sends out pulses at a rate of 1.5 mHz. Since $\overline{FF9}$ is at '1' state, the NAND gate '10' enables the clock pulses and so they are transmitted through NAND gate 10 to count up the main counter. For the 512^{th} clock pulse FF9 is set to '1' state. (Note that for a 9-bit main

counter, the number of input pulses should be equal to $2^9 = 512$ in order that there is no error in the output rates of BRM). So the 513^{th} clock pulse is transmitted through NAND gate 11 producing the clear main counter (CLMC) pulse which clears the main counter and sets RGFF flip-flop to '0' state. This in turn disables the clock and thus terminates the generation of the X and Y-ramps. The ramp generation time is equal to the product of number of input pulses to counter and (1/clock frequency). This value is approximately equal to 341 μ sec.

The CLMC pulse is also transmitted through OR gate '1' of figure 2.13 and after a delay clears FFF, setting IDF to '1' state, which indicates to the DFC the completion of vector generation. Timing diagram for DASH and SOL is shown in figure 2.15.

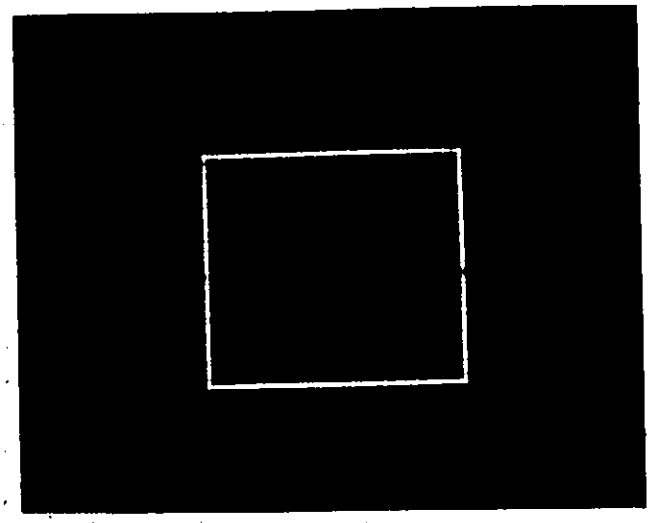
Generally, a vector is drawn by the following procedure. First a POS or TIP instruction is used. Then the X and Y-data are sent to set the starting point from where the required vector is to be drawn. Secondly, a SOL instruction is used and the X and Y-components are loaded. Then the required vector is drawn on the screen of the cathode ray tube.

The figure 2.16 shows a square generated by the vector generator and the associated X and Y-ramps.



Timing Diagram for DASH or SOL cycles

FIGURE 2.15



A simple square generated by the Vector Generator.

X and Y ramps corresponding to the above figure.

FIGURE 2.16

To draw this, first the initial starting corner of the square is set up using the TIP or POS mode, then a SOL instruction is sent in and following this the X and Y data corresponding to four sides of the square are sent out to the vector generator and then a square is drawn on the screen.

CHAPTER 3

THE INTENSITY CIRCUITRY

I. INTRODUCTION:

The data presentation on CRT is accomplished by controlling the electron beam in three axis. The deflection of the beam of CRT could be controlled in the X and Y-axis. The intensity of the electron beam is controlled in the Z-axis.

II. INTENSITY GENERATION:

(i) TIP Intensity Generation:

During TIP mode, after the beam has been positioned, the Z-axis has to be unblanked, in order to intensify the spot. This is achieved by using the pulse that is transmitted from NAND gate '9' in the figure 2.13. This pulse is sent to a pulse amplifier for standardizing the pulse in amplitude and width. A 100 n.sec. pulse is produced. A delay unit is used to control the pulse width which can be widened up to a desirable brightness of the dot. The TIP intensity generation is shown in figure 3.1.

The output of the intensity circuit IPT (intensity pulse train) are sent to the intensity control circuitry.

(ii) Vector Intensity Generation:

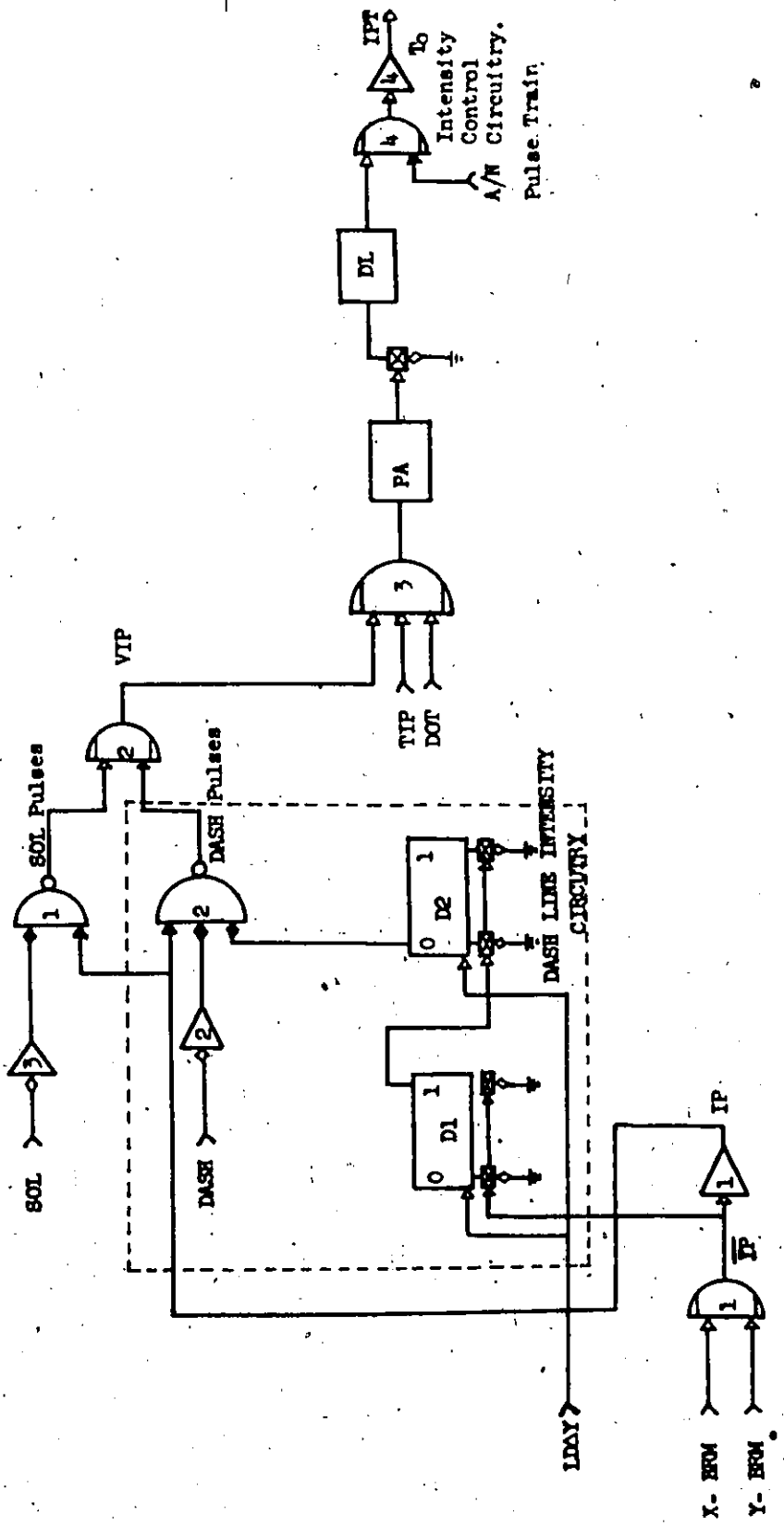
(a) SOL Intensity Generation:

Intensity pulses (IP) during vector generation is obtained by 'OR'ing the X and Y BRM pulses as shown in figure 3.1. The output of 'OR' gate 1 is inverted and it is transmitted to NAND gates 1 and 2. If the vector generation is in SOL (solid line) mode the NAND gate 1 is enabled. Thus the IP pulse train is transmitted through NAND gate 1, the output of which is the SOL intensity pulses. This is the passed through OR gate 2 to obtain the vector intensity pulses (VIP).

(b) DASH Intensity Generation:

During DASH (dash line mode) output of the inverter 2 is at (-3 volts) '1' state. Then the VIP pulses are solely due to DASH intensity pulses, transmitted through NAND gate 2. The transmission of the pulses through NAND gate 2 is controlled by the status of the flip-flop D2.

The function of this dash line circuitry is as follows. Initially, before the generation of ramps, the LDΔY pulse clears two flip-flops D1 and D2. Therefore, $\overline{D2}$ is at '1' state, i.e., -3 volts.



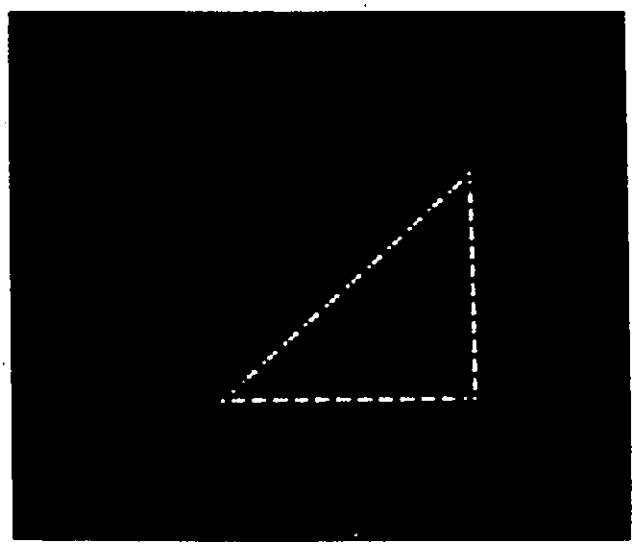
Intensity Generation for TIP, DASH and SOL

FIGURE 3.1

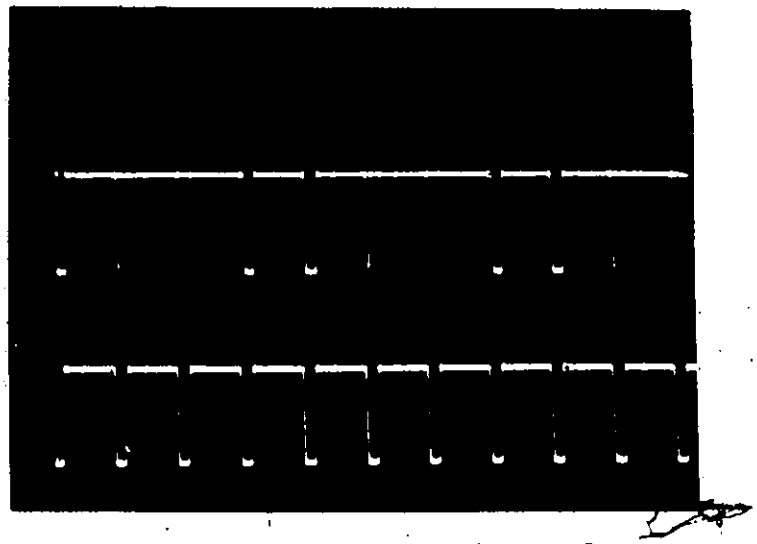
So the first $\overline{IP}$ pulse that arrives through 'OR' gate '1' sets D1 to '1' state and it is inverted and then transmitted through NAND gate 2 and 'OR' gate 2 to produce a vector intensity pulse. The next $\overline{IP}$ pulse sets 'D1' to '0' state and 'D2' to '1' state. But the IP pulse is transmitted through NAND gate '2' before the D2 changes its state. Thus a second VIP pulse is produced. The third IP pulse is inhibited because D2 is at '1' state. The fourth $\overline{IP}$ pulse sets D1 and D2 to '0' state and IP pulse is transmitted. through NAND gate '2'. Thus this process continues. So the pulse train for DASH line, will have its first two intensity pulses transmitted, then the third intensity pulse is inhibited and then on every fourth pulse is inhibited. So in the display of dash line three dots are intensified but fourth is blanked. Thus a DASH line or vector is created.

Figure 3.2(i) shows the dash lines at different angles: figure 3.2(ii) shows the input pulse train and output pulse train of the dash line intensity pulse generation circuitry of figure 3.1 (shown inside dotted lines). It could be seen that in the output pulse train the third pulse is inhibited and thereafter every fourth pulse is inhibited by the dash line circuitry.

Figure 3.3 shows the complete input and output pulse train of dash line intensity circuitry



(i) DASH lines generated by the Vector Generator.

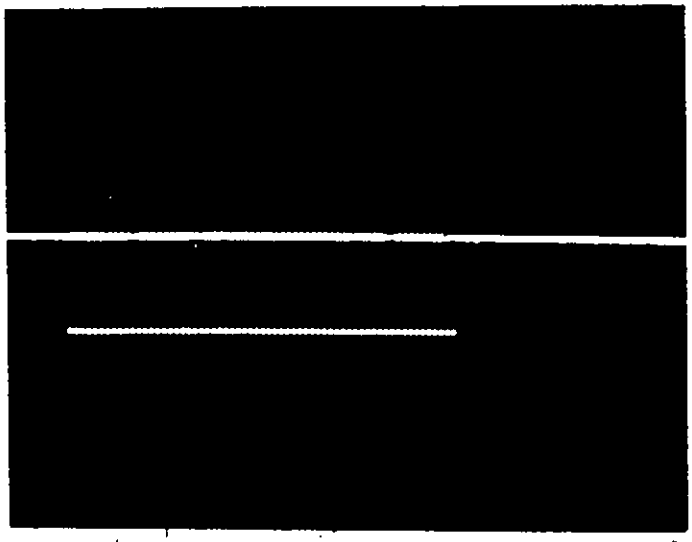


(ii) Enhanced view of intensity pulses to, and from, the intensity circuitry for DASH line generation.

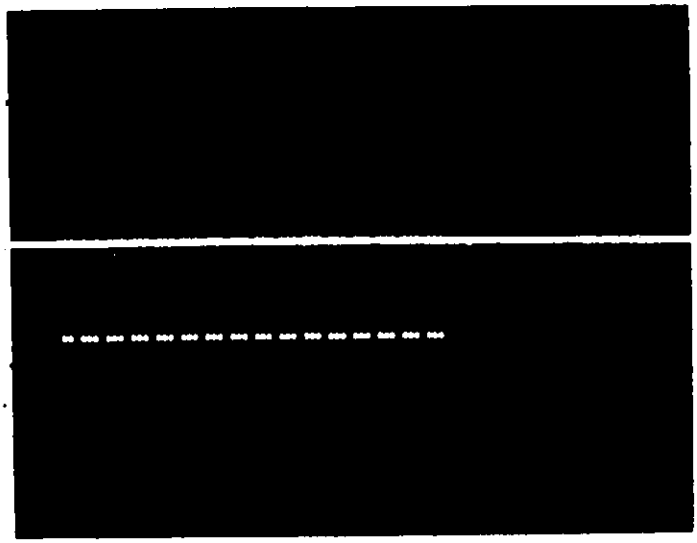
(a) output pulses of intensity circuitry
(b) input pulses to the intensity circuitry

FIGURE 3.2

Complete input pulse train to the intensity circuitry corresponding to a horizontal DASH line.



Complete Output Pulse Corresponding to the above



The corresponding X ramp

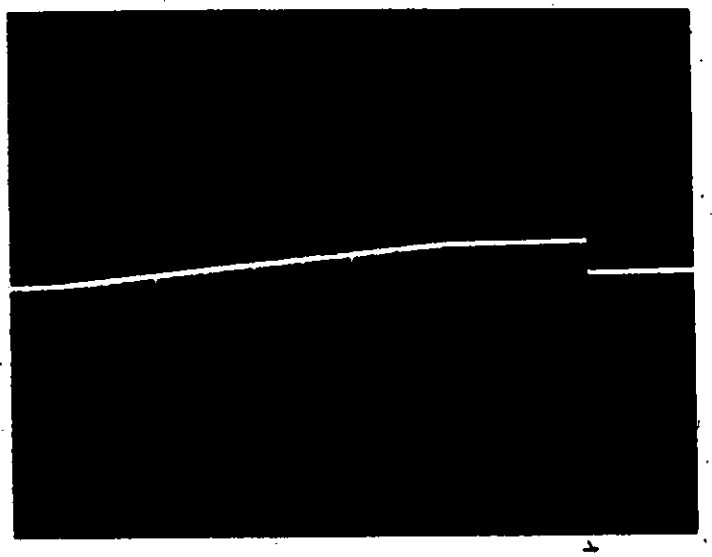


FIGURE 3.3

and the corresponding X-ramp for a horizontal dash line.

(c) Intensity Control Circuitry:

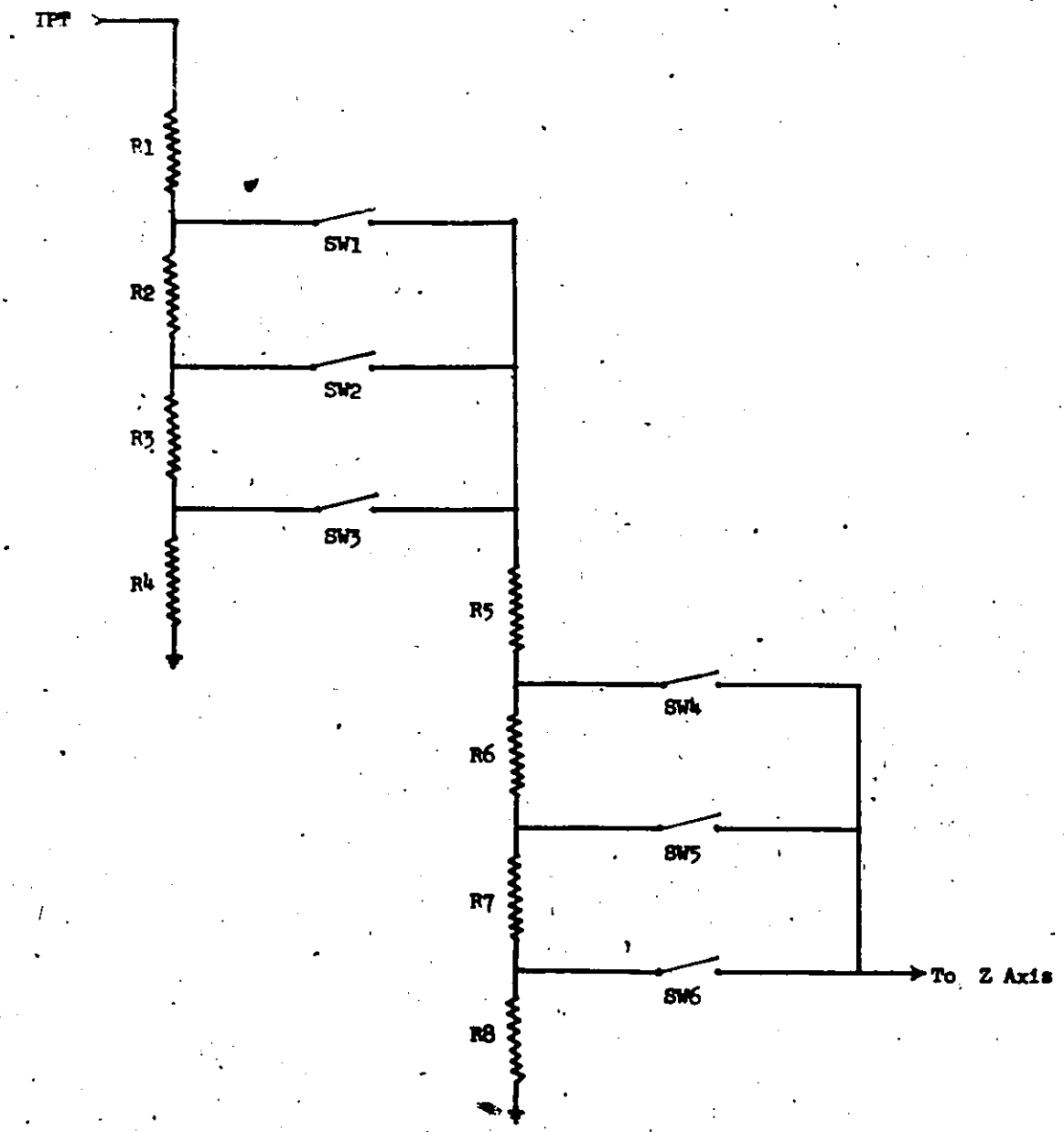
The output of the intensity generation circuit shown in figure 3.1 gives output pulses which have a constant amplitude. This output cannot be used for intensification, especially during A/N mode because there is an inconsistent brightness of the different sizes of characters due to the packing density of the dots. The use of the intensity control circuitry shown in figure 3.4 overcomes this problem. The whole circuitry represents a stepped potentiometer arrangement. The circuitry can be split into two parts. The upper part provides an output voltage with three different amplitudes. This output is then transmitted to the lower part, which attenuates this input further, and provides output voltage with three different amplitudes.

As mentioned above, during the A/N mode the brightness of characters varies with the size. The upper part of the control circuitry is used to overcome this problem. The potential divider creates three different amplitudes for small size, normal size and capital size characters. They are controlled by switches SW1, SW2, and SW3. The signals to these switches are provided by the decoder logic that is associated with the intensity

control circuitry. The decoder logic, one used exclusively during A/N mode and second which is used for variation of intensity level of display, have already been designed and implemented and reference[4] should be made to previous research carried out in the Department of Electrical Engineering at the University of Windsor. Thus, each time a character is sent in, bit 2 and 3 of buffer register is also transmitted into the decoder logic, and then, decoded to generate proper signal to turn on the corresponding switch. For instance, when a capital size character is written on the screen, bit 2 and 3 are transmitted into the decoder logic used in A/N mode, and this decoder logic, then sends out a signal to close switch SW1. This causes the dots composing the letter to be brighter than normal, thus compensating for the fact that the dots are spread farther apart than normal.

The dots that are generated during vector or dot mode are treated as bright as normal size character and thus switch SW2 is closed during these modes.

The brightness of the display could be varied over four distinct levels. This is performed by the lower part of the intensity control circuitry, which again generates three distinct amplitudes



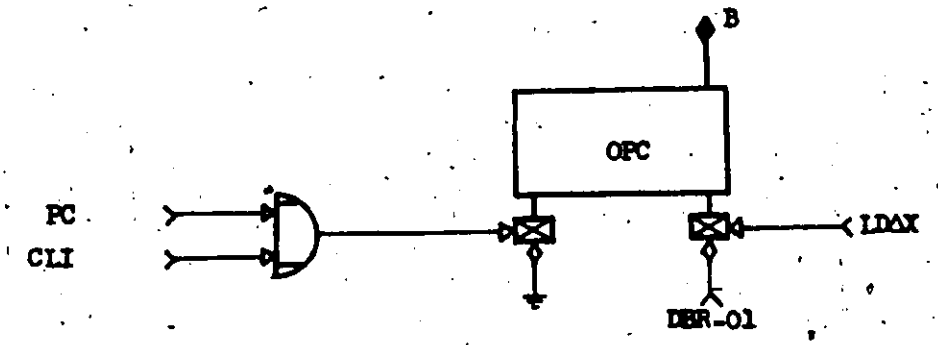
Intensity Control Circuitry

FIGURE 3.4

Character Type	Bit 2	Bit 3	SW1	SW2	SW3
NORMAL	0	0	0	1	0
CAPITAL	0	1	1	0	0
SuperScript (SMALL)	1	0	0	0	1
Sub Script (SMALL)	1	1	0	0	1

Intensity	Bit 10	Bit 11	SW4	SW5	SW6
NORMAL	0	0	0	1	0
DIM	0	1	0	0	1
BRIGHT	1	0	1	0	0
BLANKED	1	1	0	0	0

(a) Truth table of switches



(b) Override option Flip Flop

FIGURE 3.5

corresponding to bright, normal and dim controlled by switches, SW4, SW5 and SW6. When one of the display instruction is sent in, then the corresponding mode pulse (set vector mode, set dot mode, set A/N) enables the instruction word bit 10 and 11 to be transmitted into the decoder logic, from display buffer register. The decoder then produces a proper signal to turn on the proper switch. It should be noted that when a BLANKED instruction is sent in, then all the three switches SW4, SW5, SW7, are open, because the decoder does not provide any signal for this instruction and thus the electron beam of CRT is blanked. The truth table associated with the switches are shown in figure 3.5(a).

Another feature of the intensity circuitry is the override option which is available only when the vector generator is used. This option allows one to position a blanked starting point from where a vector will be drawn. This option allows one to draw any vector from a new starting point without giving a new instruction. This option is incorporated for the convenience of the programmer and this feature saves core space. Bit 1 of the first data word for the vector generator controls the use of this option. The control flip-flop OPC is as shown in figure 3.5(b). Initially this flip-flop is

cleared by CLI pulse. Then at the same time when the first data is loaded into ΔX-register the loading pulse LDΔX is also used to load the OPC flip-flop with bit 1 of the data. Thus 'B' is set to 1 state. Then 'B' is ANDed with '1' output of FFF as shown in figure 2.13, which then makes the outputs of NAND gates 3 and 4 to go to '1' state regardless of what type of vector which is indicated in the instruction word.

The LDΔX pulse, after a short dealy, clears FFF and so, a short time later the second DRP pulse arrives, this pulse is immediately passed through NAND gate 5 and then through OR gate 3 to produce the CLXYC pulse which clears the X and Y-counter registers. The same DRP compliments VDFF and after it is delayed by delay DL1, it is transmitted through NAND gate 14 and to inverter #4. The output of the inverter is disabled by NAND gate 16 because $\overline{B}$ is now at '0' state, but enabled by NAND gate 15, and thus a LDXYC pulse is transmitted which loads X and Y-counter registers with the new data and after a delay it clears the OPC flip-flop. Thus a full cylce of POS is achieved.

CHAPTER 4

CONCLUSIONS

As mentioned before, the output rate of the BRM is exact only if the number of input pulses to the main counter is 2^n where n is the number of bits which is also equal to number of data bits. In this case, the data are 9-bits each. So the number of input pulses to the counter should be equal to $2^9 = 512$. But in the actual case the input pulses were reduced to 128 pulses. So the vector generation time was reduced to approximately 80 μ sec. compared to the exact case (i.e. input pulses = 512) in which it took 341 μ sec.

One disadvantage of reducing the input pulses was that the length of the vector displayed was ¼th of the actual length of the vector, specified by the X and Y data. But this was overcome, by transmitting the BRM pulses to the 4^{th} count position of the X and Y counter-register (accumulating registers). In this manner a 1:1 scale was achieved.

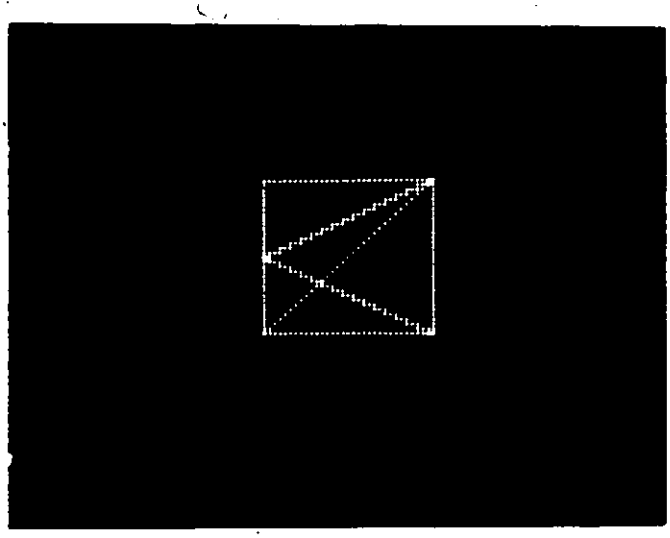
As indicated above, there is error involved in the slope of the vector displayed, since the input

pulses are equal to $2^7 = 128$ less than the required for a 9-bit main counter. This error is difficult to evaluate as the number in the data register is a variable and so the error involved is a dynamic one.

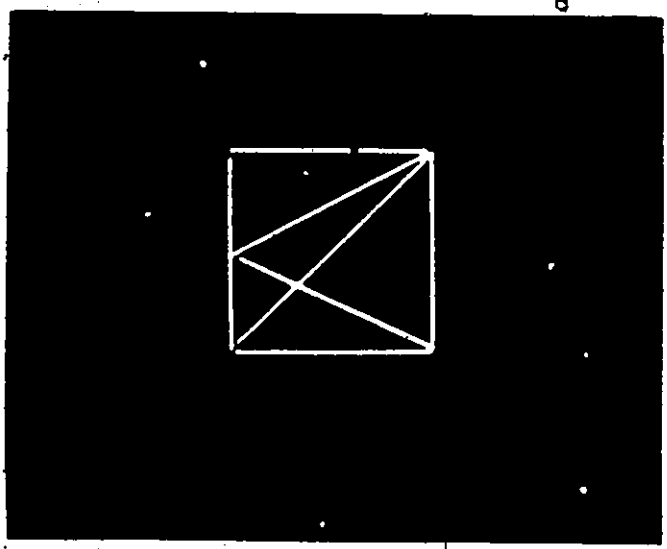
On the whole, the quality of the vector display by the incremental technique was good for horizontal, vertical and diagonal lines which are the most common for many displays. At other directions the vectors drawn were staircase in pattern. This could be seen in figure 4.3.

Even though the vectors drawn by digital technique are staircase in pattern, a comparison of the outputs of the digital vector generator and analog vector generator shown in figures 4.1(a) and (b), reveals some of the advantages of the digital vector generator. It can be seen that there is no mismatch at vertices of the square displayed and also the slopes of the vectors drawn are quite accurate. Figure 4.2 shows the X and Y-ramps associated with the display shown in figure 4.1(a).

Another advantage of the digital technique was that the dashed vectors could be drawn with consistent length of dash. This is shown in figure 4.4. In the case of the analog vector generator[4], it was able to generate dashed lines

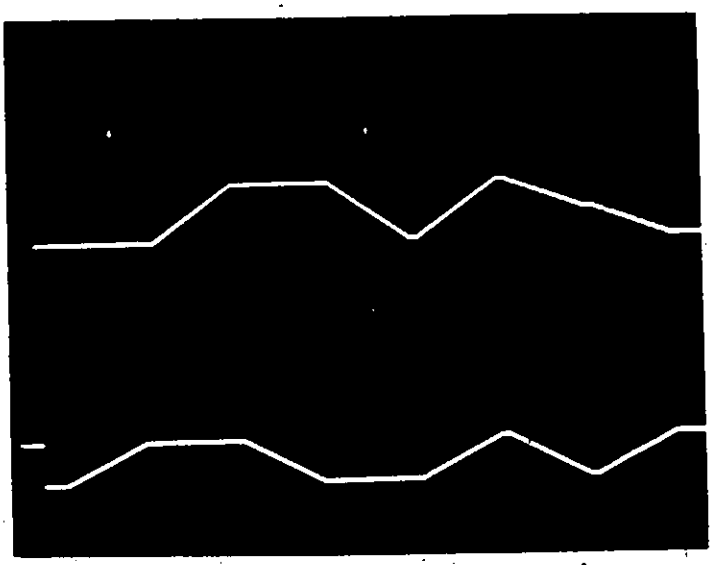


(a) Output of Digital Vector Generator.



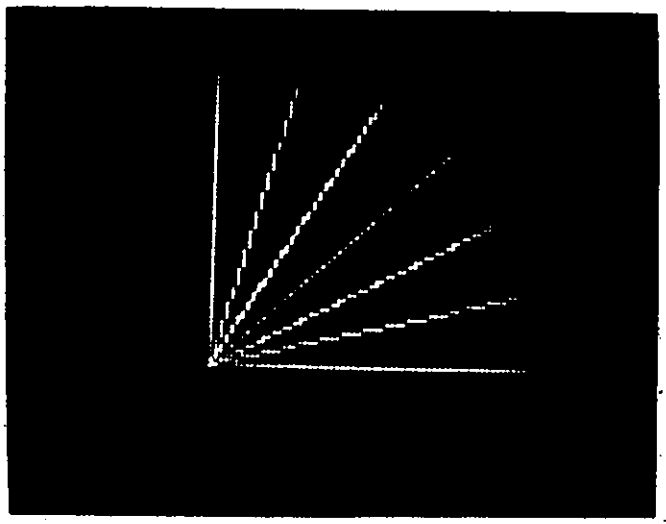
(b) Output of Analog Vector Generator.

FIGURE 4.1



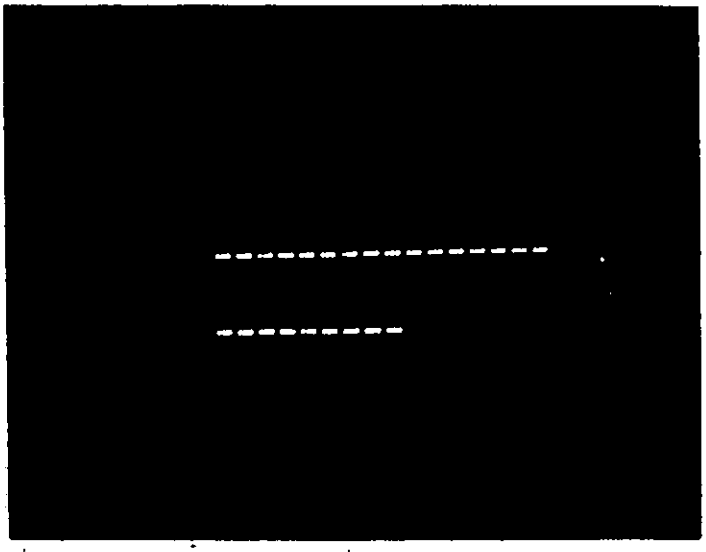
X and Y ramps corresponding to the output of the Digital Vector Generator.

FIGURE 4.2



Vectors with different slopes.

FIGURE 4.3



DASH lines of two different lengths with consistent length of dash.

FIGURE 4.4

but the length of dash varied with length of the dashed vector drawn.

Finally, the timing control unit was checked and it was found to operate as expected. It directed the loading of X-Y counter-registers, ΔX and ΔY registers and also controlled the duration of ramp generation. The BRM circuit was also checked and it was also found to operate as expected.

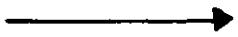
Thus, it is concluded that the vector generator sub-system was designed and implemented successfully.

APPENDIX

LOGIC NOTATION*

(1) Preferred Pulse Input:

Positive Pulse 

Negative Pulse 

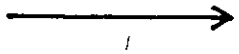
(2) Preferred Level Input:

Positive Level 

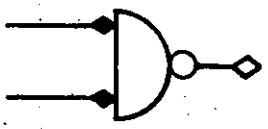
Negative Level 

(3) Interconnections:

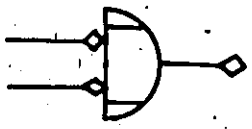
Input 

Output 

(4) NAND Gate:



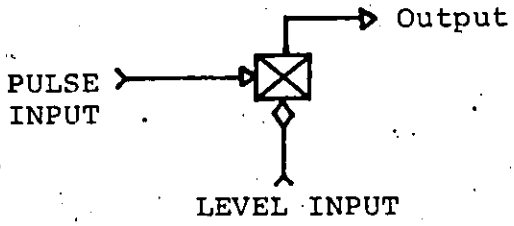
(5) OR Gate:



(6) Inverter:

(7) Pulse Amplifier:

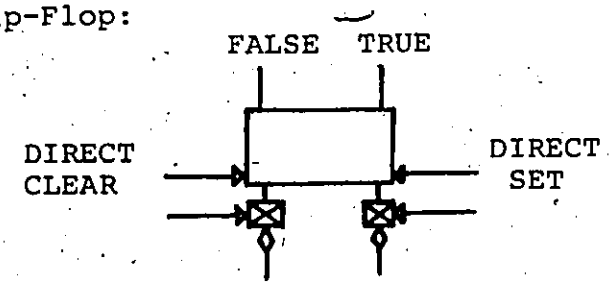
(8) DCD Gate:



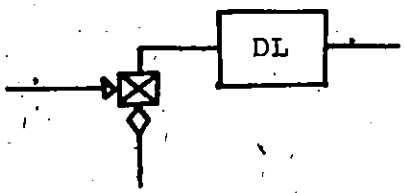
Level must be present at least 400 n sec. before gate is pulsed.

This gate acts as an AND gate. It provides a logical delay which is essential for sampling flip-flops at the time they are changing.

(9) Flip-Flop:



(10) Delay:



(11) Clock:

A is an enable level which allows the clock to operate. Output is a series of 100 n sec. pulses, the frequency of which may be varied.

(12) Switch:

When the control signal is GROUND level the switch is opened. When the control signal is held as -3 volts the switch is closed.

REFEPENCES

1. Murrey, D.L.; A Data Flow Controller and Refresh Memory for a Computer Display System; A Thesis Submitted to the Department of Electrical Engineering at the University of Windsor (1970).

2. Mennie, W.E.; An Alphanumeric Text Generator for a Computer Display; A Thesis Submitted to the Department of Electrical Engineering at the University of Windsor (1970).

3. Bristow, F.W.; A Graphics Terminal for a Digital Computer; A Thesis Submitted to the Department of Electrical Engineering at the University of Windsor (1970).

4. Houang, T.H.; A Vector Generator for a Computer Display System; A Thesis Submitted to the Department of Electrical Engineering at the University of Windsor (1972).

5. Staff of Hewlett-Packard Co.; Operating and Service Manual X-Y Display; Hewlett-Packard Co., Colorado Springs, Colorado (1968).

6. Staff of Digital Equipment Corporation; Logic Handbook; Digital Equipment Corporation, Maynard, Massachusetts (1968).

7. Siders, R.A.; Computer Graphics; American Management Association, New York, N.Y. (1966).

8. Davis, Samuel; Computer Data Displays; Prentice-Hall, Inc., Englewood Cliffs, N.J. (1969).

9. Fifer, Stanley; Analogue Computation; Volume-I, McGraw-Hill, New York, N.Y. (1961).

10. Polzen, K.P.; Line Generator Using Binary-Rate Multiplier Techniques; N.R.C. of Canada. Radio and Electrical Engineering Division, Ottawa, Ontario, September 1966.

11. Arnstein, W., Mergler, H.W. and B. Singer; Digital Linear Interpolation and Binary Rate Multiplier. Control Engineering. June 1954, pp. 79-83.

12. Millman, Jacod and Herbert Taub; Pulse Digital and Switching Waveforms; McGraw-Hill, New York, N.Y. (1965).

VITA AUCTORIS

1951 Born on December 7, Coorg, Mysore, India.

1966 Completed high school at St. Anne's Higher Secondary School. Coorg, Mysore, India.

1967 Completed the Pre-University course at the University of Madras, Madras, India.

1972 Graduated from University of Madras, Madras, India, with the degree of Bachelor of Engineering in Electronics and Communications.

1974 Candidate for the degree of M.A.Sc. in Electrical Engineering at the University of Windsor, Windsor, Ontario.