

Image Processing Laboratory

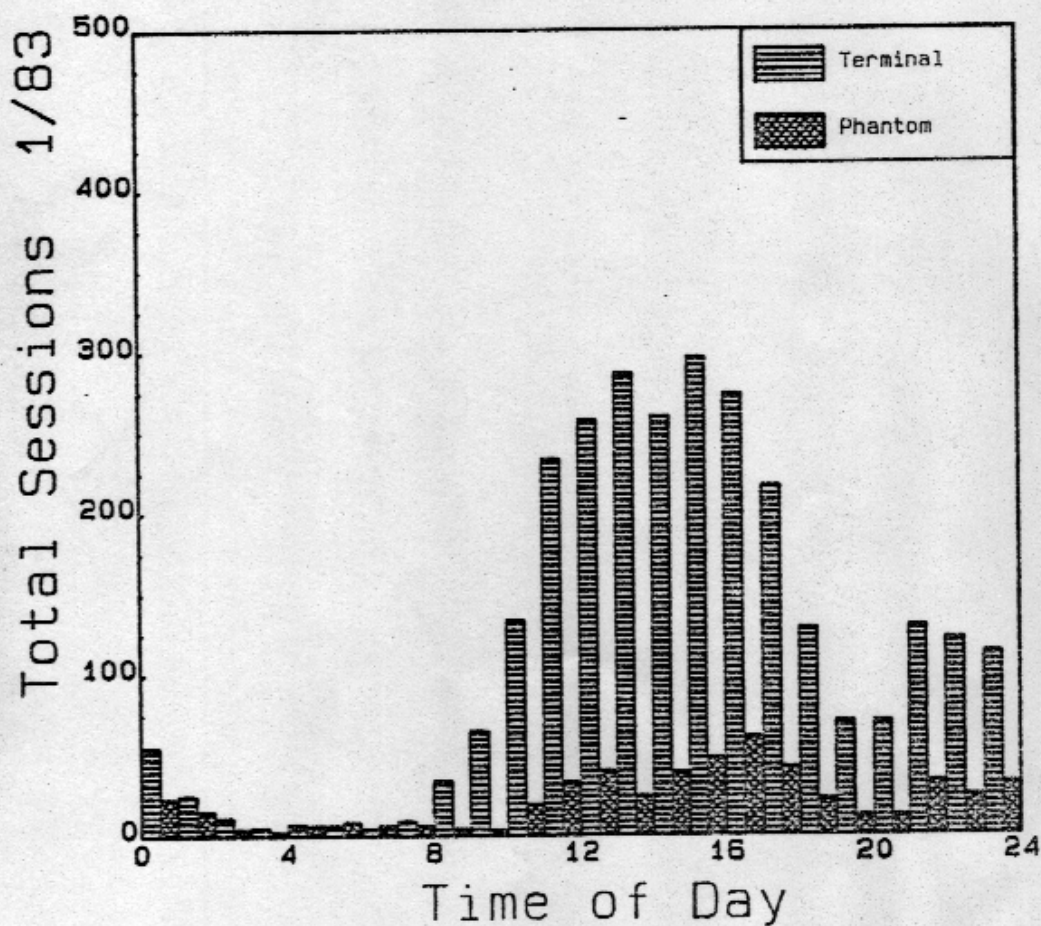
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, New York 12181



The IPL Newsletter

February 1983

Number 1



MULTIPLEXED IMAGE TRACKING:

A Simple Method of Encoding 3-dimensional objects

Douglas Lyons

Abstract

Presented here is a simple, efficient and reliable method for encoding 3-dimensional information. Using video cameras, lasers and a networked system of small and large computers 3-dimensional collection display and storage can be implemented in real time.

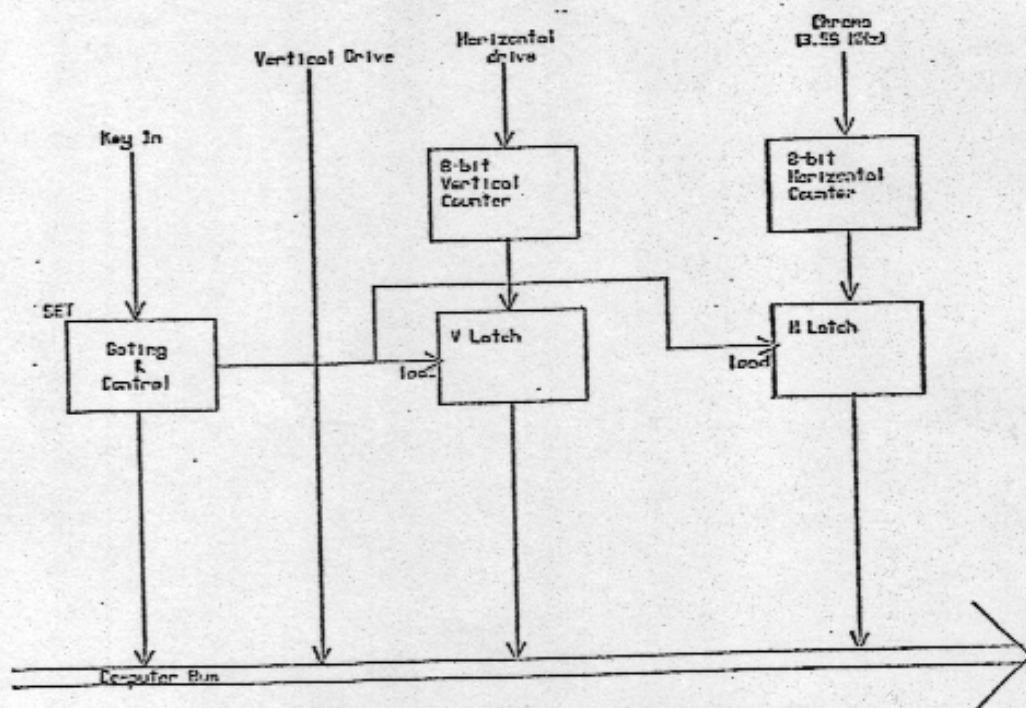
For further information contact Douglas Lyon, the author, at 1 Wрпи Plaza, Troy NY 12181; or phone: (518)-270-6248.

Three Dimensional Pantomation

Background:

A black and white television camera scans an image in much the same way an electron gun scans a television screen. 525 lines are scanned in 66 milliseconds or one line every 63.5 microseconds may be obtained. Positional information may be derived by clocking a pair of digital counters (see fig. 1) with horizontal sync and subcarrier.

The output of a black and white television camera shows a linear relationship between amplitude of waveform and intensity of image. Utilization of this fact allows easy implementation of Pantomation. Pantomation is a process by which the position of a bright spot in the video field may be derived. Say a light emitting diode is powered by a comparator which is set to pulse at a certain brightness from a video waveform. The light emitting diode can be used to trigger a light pen which in turn records the positional information associated with a bright spot in the video field.



Based on a diagram by Philip Edelstien
March 1977 of a Key Positioner block diagram

fig. 1

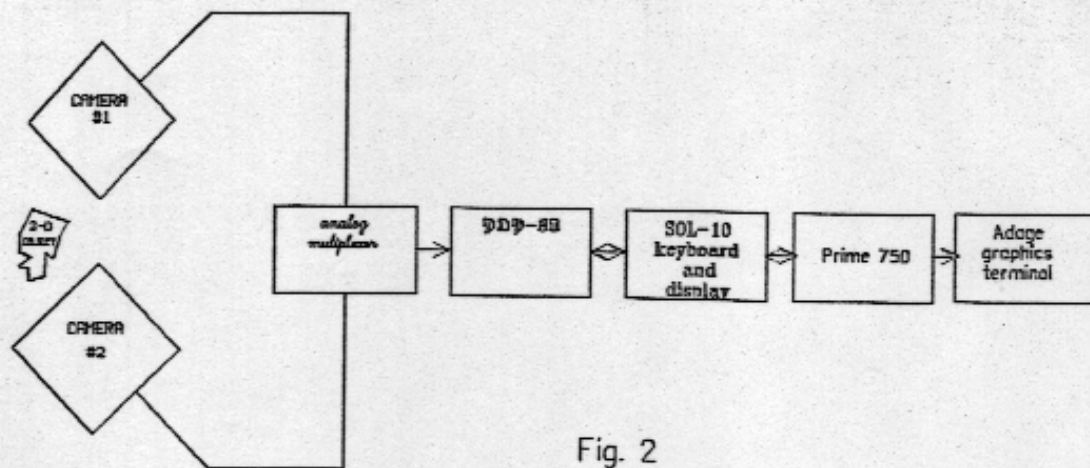
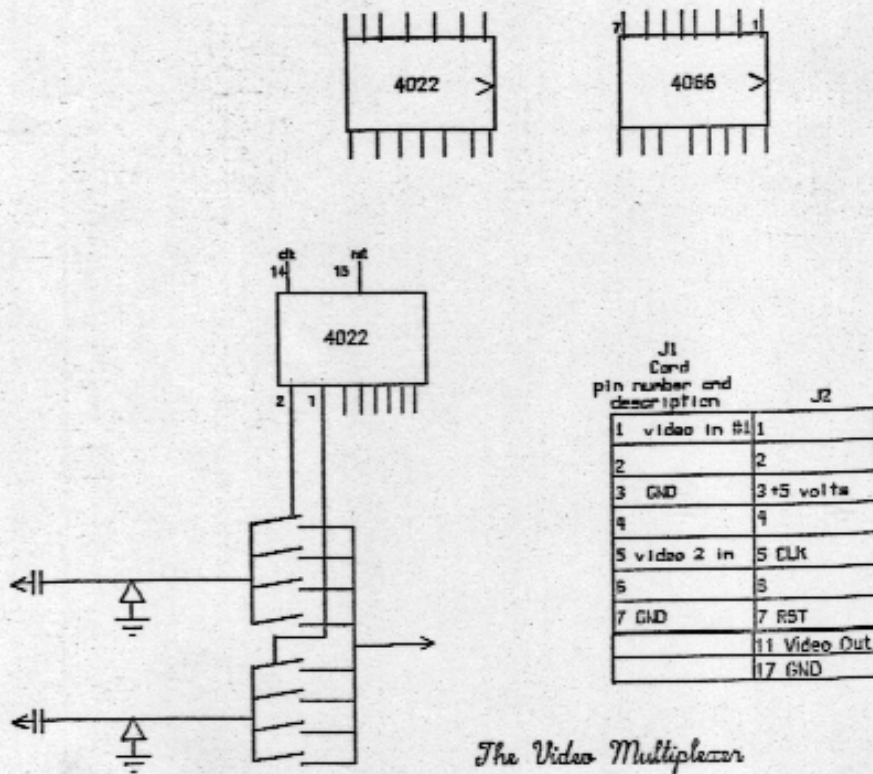
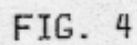


Fig. 2



The Video Multiplexer

Fig. #3



Pantomation in the Third Dimension

If two video cameras are multiplexed together pantomation should give us two sets of X-Y coordinates with a label for each camera (see Fig. 2). With a little thought it is easy to see that there exists an axis of redundancy, from which a Z axis can be derived.

A Video Multiplexer

Composite video signals range in voltage from about -.3 of a volt to about .7 of a volt. In order to use the multiplexer without a bipolar supply we must condition the signal such that it ranges from 0 to 1 volts, here a clamp is called for. In order to multiplex several cammeras into one output, quad analog switches are used. Since the impedance of the cable used to transfer the video signal from the cammera is about 75 ohms I hook 4 switches in parallel thus reducing the effective impedance to a scant 20 ohms. The effect of decreasing resistance is to match more precisely the impedance of the cable which minimizes the standing wave ratio. It also increases the noise immunity of the switcher, a critical factor since the board must coexist with a computer. The switching logic takes the form of a simple walking ring counter since the computer software was originally designed to drive a sequencer (see Fig. 3)

The Total Encoding Process

There are eight corners to a cube. This implies that a box with eight cameras and eight computer deflected lasers could scan an object within the box in real time. All information needed to reconstruct a graphic image of the object is then collected in real time.

Data bandwidth out of the computer doing the laser scan upon the object within the box is calculated as follows: 8 cameras x 2 coordinates x 30 times a second to allow for a reasonably fast scan rate gives 480 coordinates per second. There are 525 lines in a television picture in the vertical direction and 525 divisions in the horizontal. Since a coordinate has 525 division in the Y direction and about 1,024 in the X direction we can use two ten bit words for the definition of a coordinate. This will require a transmission speed of about 4,800 bps + 90 bps for the 8 camera labels. A dedicated microcomputer is ideal for this type of data collection.

Getting Started

A Cold Start from the SOL-10 Terminal Computer.

Assuming the Adage is unassigned and that the SOL-10 and PDP-8L are on.

Hit the "mode select" key, and a ">" prompt should appear.

Hit the "uppercase" key and an LED in the uppercase key should light.

Type "TE" and the SOL should be in terminal mode. You may now proceed to login in the usual mannner.

Type "DELAY 10", this should cause the characters on the screen to be printed at a slow enough rate for the SOL-10 to accept them without losing a character.

Type "AB 'DEWITT A>ABFILE' -list". This should implement the abbreviations file and give you a list of all the instructions at your disposal.

Typing

"ZIP HELLER"

"R *LOADSOL BGETKEY40"/1/

will run a program which should download the SOL-10. After successfully downloading the SOL-10 type:

"A DEWITT A"

"ZIP HELLER"

"ZIP GOODSTUFF"

"ASS AGT"..... Marks the beginning of a warm start.

"GLOAD I.DRAW5"

"SEG #PANT012"

The Commands Are:

M - toggles move/draw mode.

N - new origin for move

D - Draw one line in move mode.

C - Clear (prompts for constants) scale = 24.

Q - Quit.

X rotate commands

Y

Z

Summary of Commands:

from the SOL-10

"<mode select>"

"TE"...(login)

"A DEWITT A"

"AB 'DEWITT A>ABFILE' -LIST"

"ZIP HELLER"

"ZIP SOL"

"R *LOADSOL BGETKEY40"

"A DEWITT A"

"ZIP HELLER"

"ZIP GOODSTUFF"

"ASS AGT"

"GLOAD I.DRAW5".... Warm start

"SEG #PANT012"

Now that running the system has become a familiar and ordinary event, suppose you would like to change the mode of system operation.

Multiplexor Programming Details:

accessing the control port (see Fig. 4)

Data is either strobed into the data output latch or the control word latch depending on the PUS (parallel unit select) line.

PUS = high----datamode

PUS = low----control mode, this mode routes data on the mux board.

The PUS is bit four of output port (FA)hex.

*code to switch to data mode

PUSH PSW; save A and flags

MVI A,8; bit 4=1

OUT OFAH; write it out

POP PSW; restore Acc and flags

*code to switch to control mode

PUSH PSW;

XRA A; clear A

OUT OFAH;

POP PSW;

Control and data are written out the parallel port with an OUT OFDH instruction.

The control words are as follows:

0-clocks SOL out the latch to CAM H

1-clocks SOL out latch the to CAM V

2-clocks SOL out the latch to CAM COL

3-enables

4-enables bits 0 to 3 from PDP-8 to SOL input

5-enables bits 4 to 11 from PDP-8 to SOL input

6-clocks SOL out latch to D4 to 11 of PDP-8 input and sets interrupt flag

7-clocks SOL out to D1-3 of PDP-8

To run the 8080 assembler type

"R ASM8080 source"

A binary file suitable for downloading the SOL-10 should be created. Typing

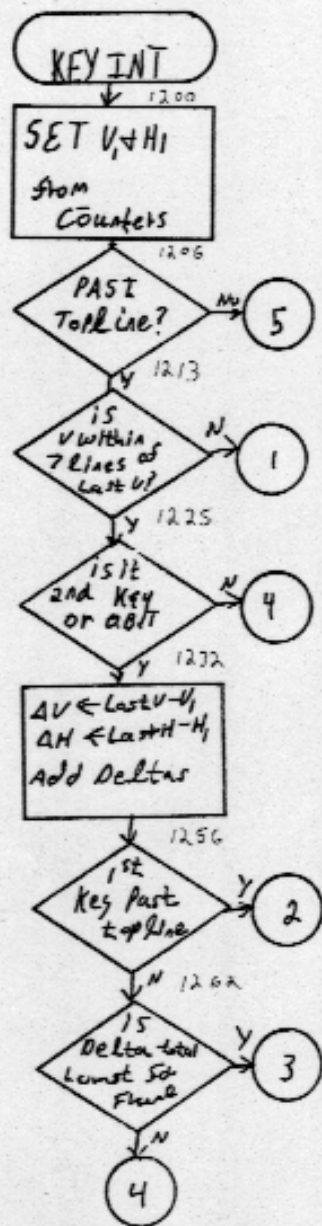
"R *LOADSOL B source

should download the SOL.

Appendix A

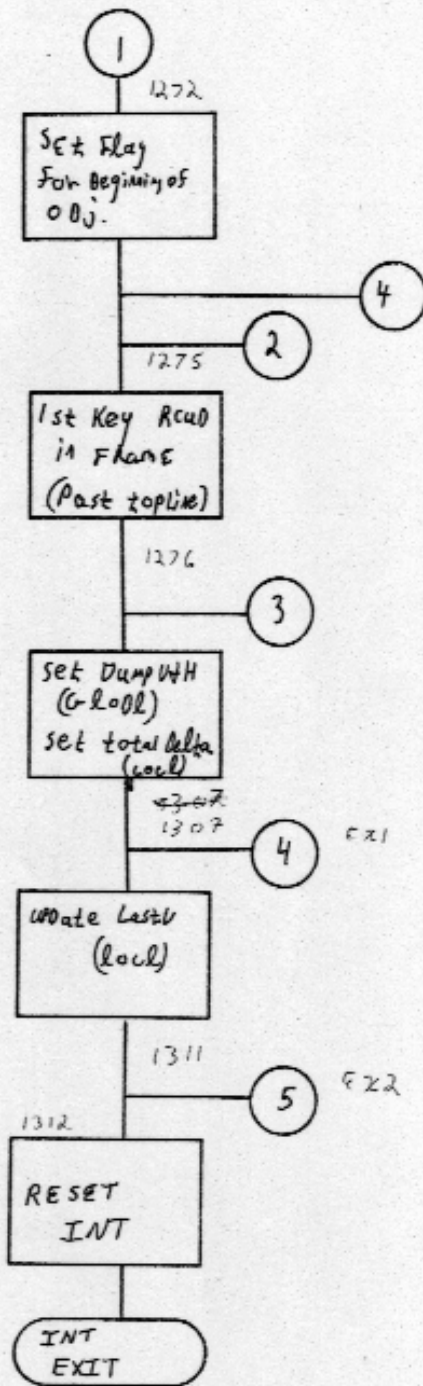
The PDP8-L Pantomation Software

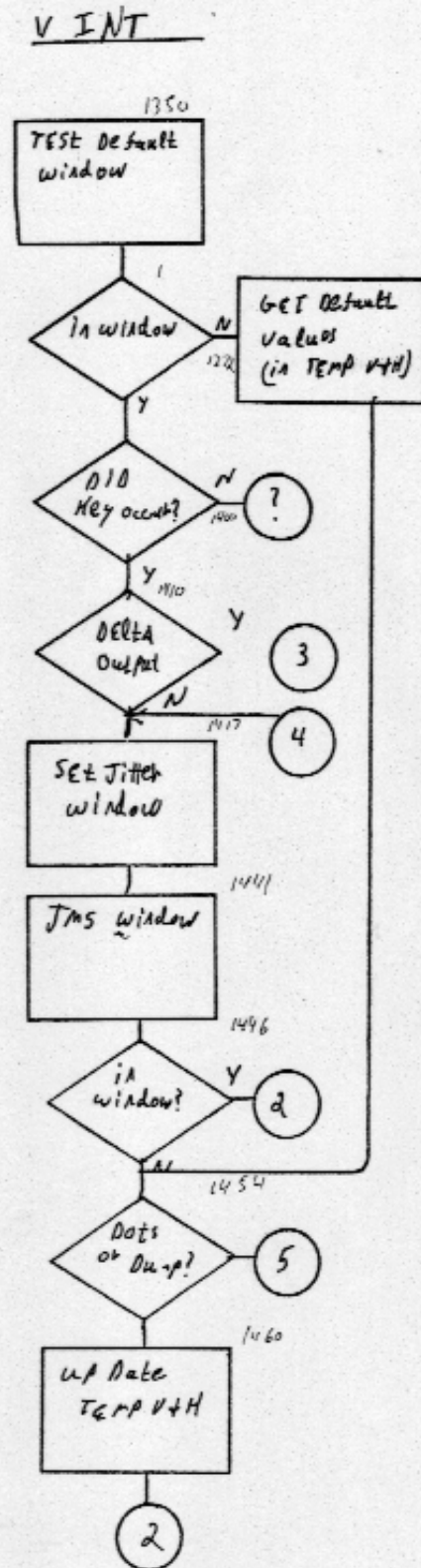
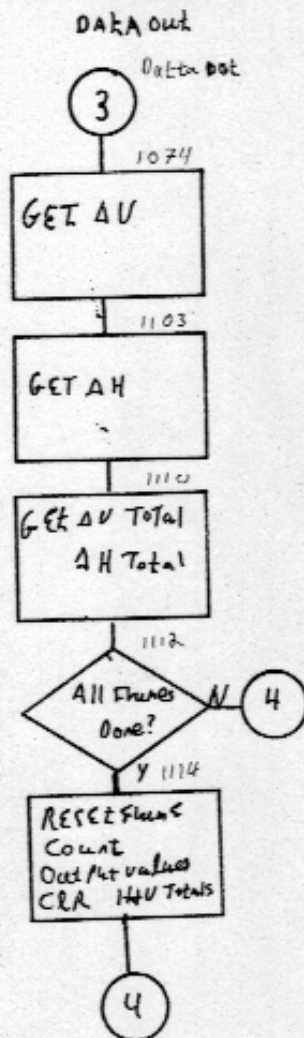
the software within is the product of Roger Meyers



Please Note that all future flowcharts
will comply with the precedent set here or
top down programming with no arrow required.

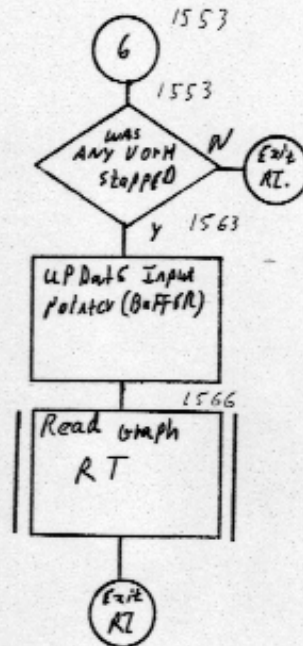
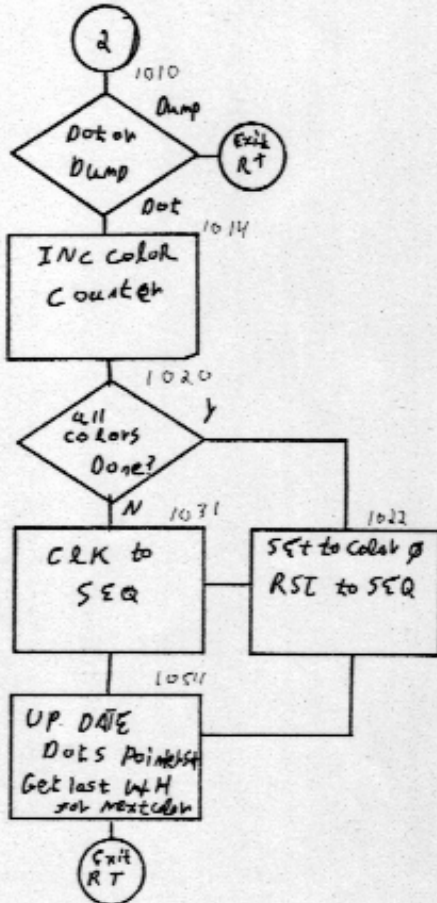
A1.

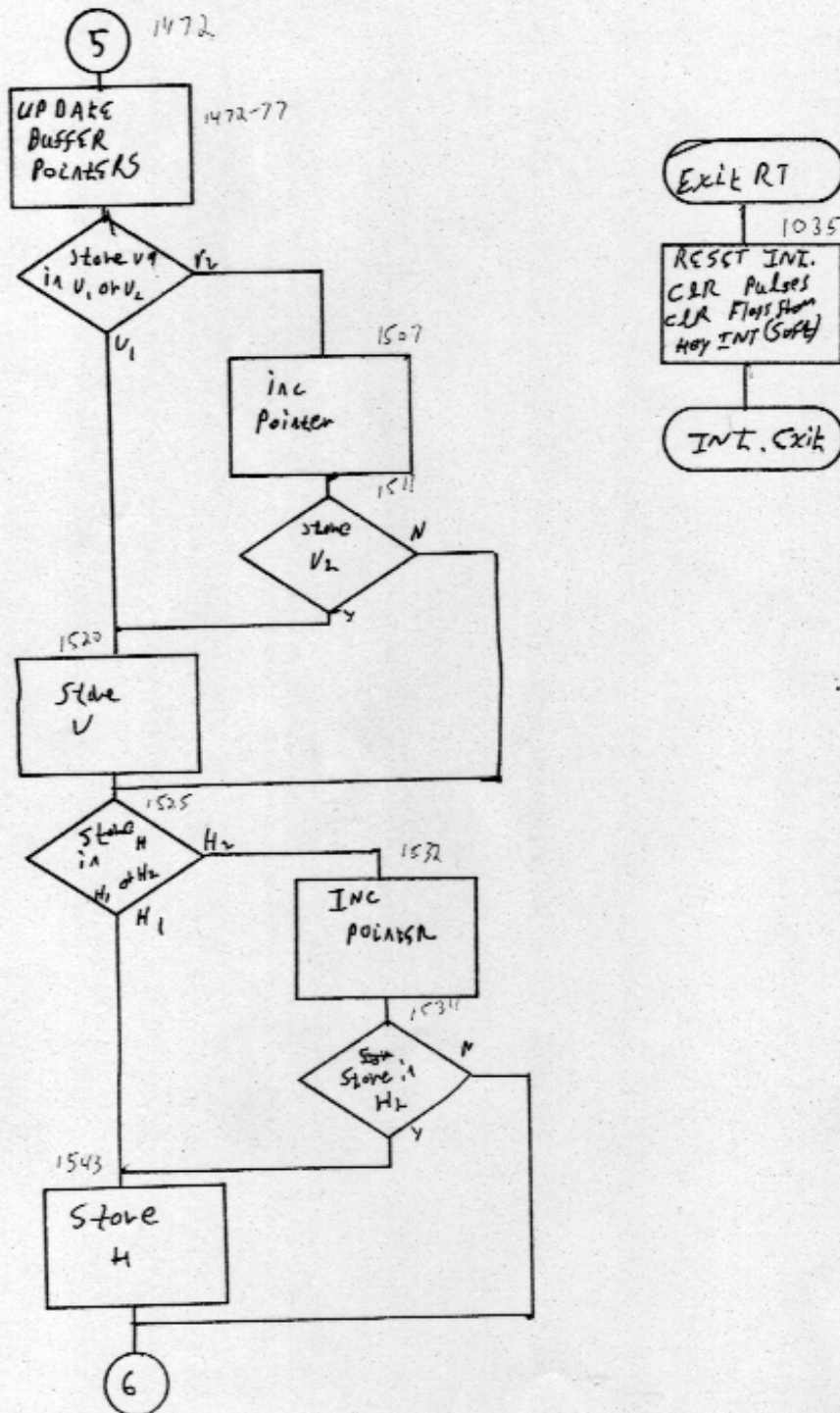




$$V_2 = V_1 + V_2 \quad H_2 = H_1 + H_2$$

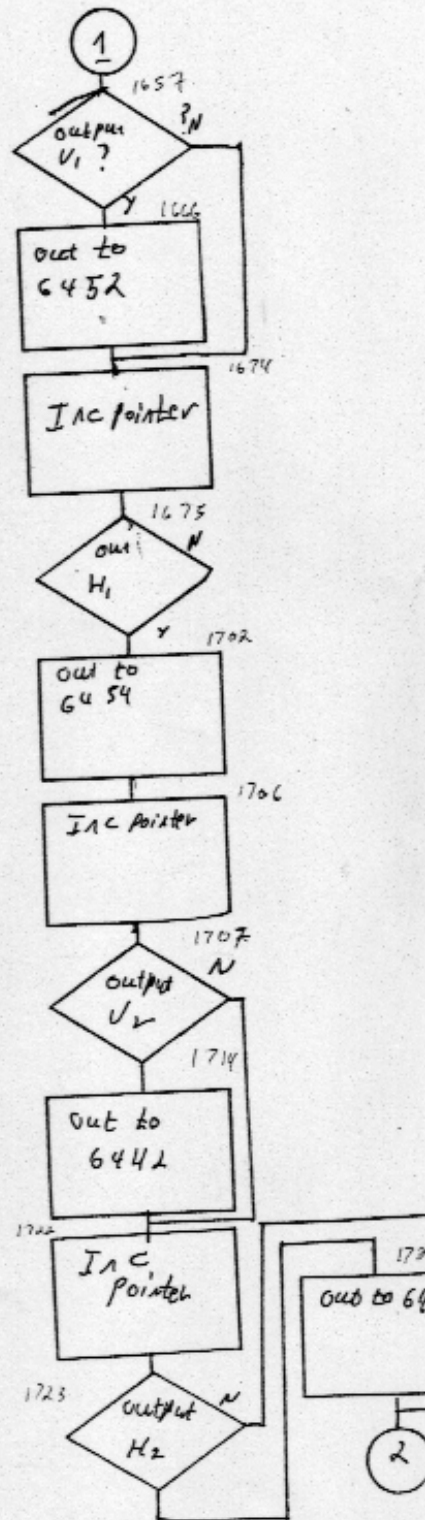
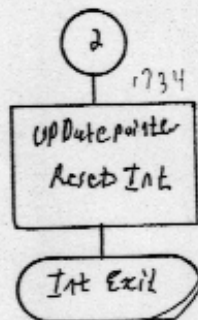
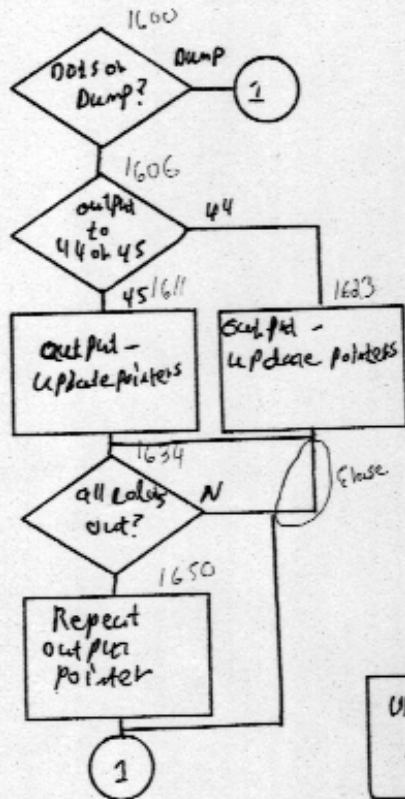
Dots continued



Buffer Input

CLK INT

PAGE 1



A.6

