A new graphic display/plotter for small digital computers

by GRANINO A. KORN, STEVENS SIMONS, RUSSELL STEINBACH, and CLAUDE WIATROWSKI

The University of Arizona
Tucson, Arizona

INTRODUCTION

This report describes a new inexpensive cathode-ray-tube-display and recorder interface designed to provide graphical output for many of the increasingly popular small digital computers in the 12-to 24-bit class. Our original design was developed to produce differential-equation solutions, phase-plane plots, correlation functions, spectra, and amplitude distributions for on-line digital simulation with the 18-bit PDP-9; but the interface logic is flexible enough to serve many other computers and applications.

Our display design does not involve the use of a storage oscilloscope, so that "dynamic" or changing displays are possible. A single "packed ' 18- or 16-bit display word sets both X and Y coordinates of a display point, which halves our refresher-memory requirements. As a novel feature, the cathode-ray beam intensity, line-segment (vector) generation, and X-coordinate incrementing can be controlled not only by programmed instructions, but also by special data words corresponding to unused coordinate combinations. Since the display then requires only data words, completely automatic data-channel operation is possible, i.e., the display can be refreshed or changed with the cycle-stealing automatic data channels built into many of the newer small computers with little or no programming. The display will also operate xy (servo) recorders and a four-channel stripchart recorder.

The entire cost of the display interface, when built with Digital Equipment Corporation logic cards, is less than $1,900—exclusive of power supplies and display oscilloscope. The use of DEC cards is convenient for interfacing with the PDP-9, but logic costs could be halved if the interface were assembled with integrated-circuit logic. In this case, the entire display interface would fit on two or three logic cards.

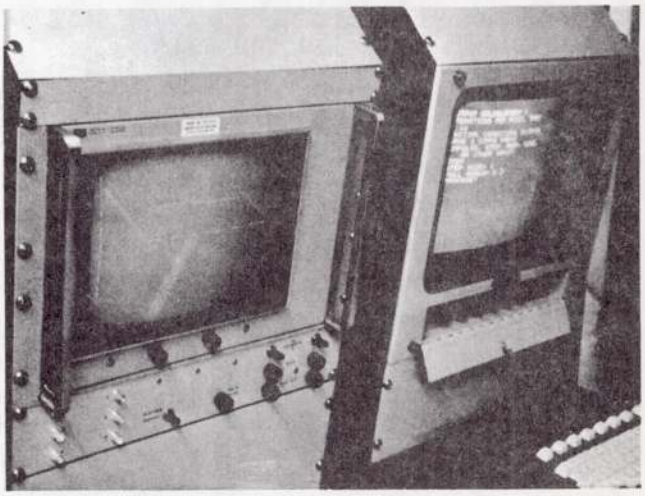
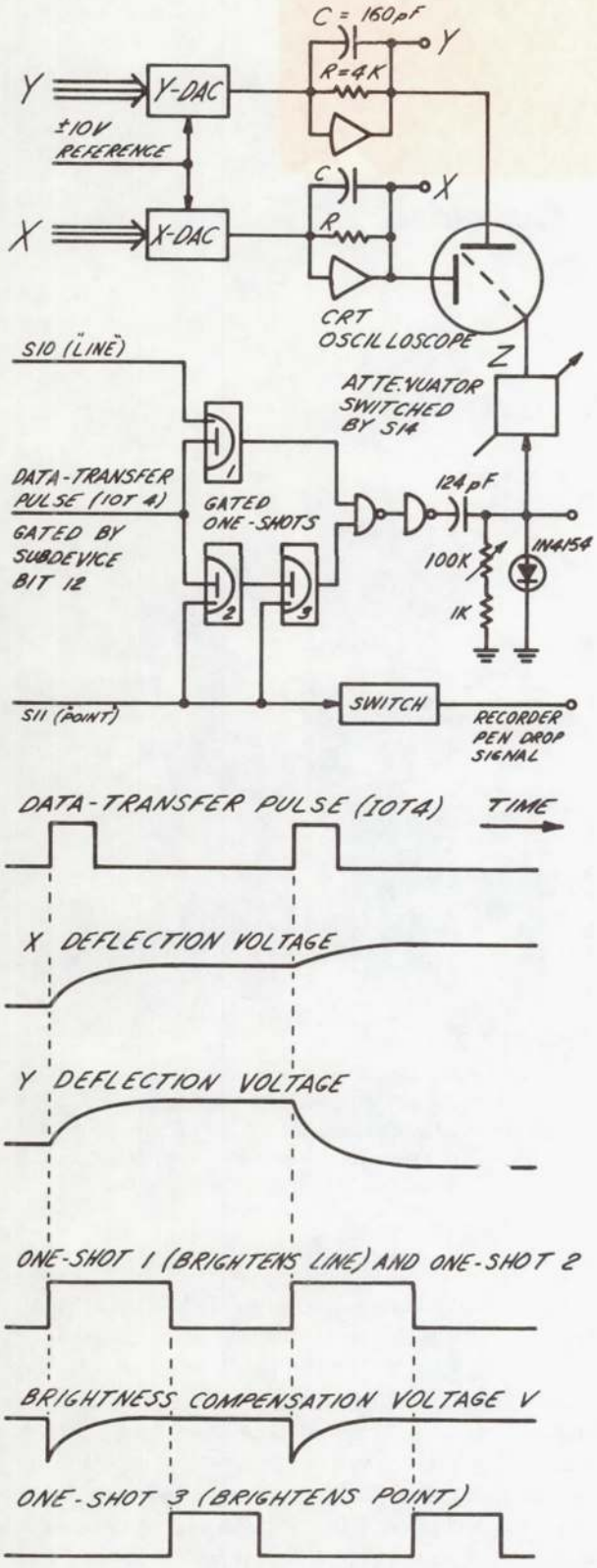


Figure 1—Display unit (left) built into the Univers.ty of Arizona's DARE (Differential Analyzer REplacement) console. When differential equations or analog-block interconnections are typed onto the CRT typewriter (television monitor at right), the display shows solution curves on-line

Basic display operation

Figure 2 illustrates the design of the digital-computer driven cathode-ray-tube display. Referring to Figure 2a, a point (X, Y) on the oscilloscope screen is positioned through deflection amplifiers driven by 9-bit digital-to-analog converters (X- and Y-DACs). Matched RC delay feed-back networks on the X and Y amplifiers cause both deflection voltages to approach new values exponentially with the same time constant of about 0.7 μsec. (Figure 2b). *When the X- and Y-DAC registers are updated simultaneously, the point (X, Y) then generates a straight-line segment useful for curve interpolation and vector generation,* as suggested by Dertouzos.[1]



Figures 2a, b—Basic display operation. The data-transfer pulse IOT4 from device selector A or B is gated into one-shots 1 and/or 2 and 3 by the enabling subdevice bit 12 and by S-register bits S10 and S11. DAC's are Pastoriza Electronics Minidac's

Referring again to Figure 2a, the cathode-ray beam intensity is determined by a switched attenuator at the input to the oscilloscope Z-axis amplifier and three gated one-shot multivibrators, which permit *selective brightening of points and/or lines*. These operations are controlled by a 9-bit digital control register, the S register, as follows:

1. S-register bit S14 sets two different beam intensities by shorting a portion of a Z-axis attenuator resistor with a simple FET switch.
2. S-register bit S10 gates the DAC updating pulse into one-shot No. 1 to brighten the screen while a line segment is drawn. As a new feature, *a clamped differentiating network decreases the beam intensity as the beam slows down exponentially* (Figure 2b). This circuit results in more uniform lines than the original Dertouzos circuit.
3. S-register bit S11 enables monostable multivibrators 2 and 3 to brighten the new display point or end point of a line segment (Figure 2b).

Suitable instructions loading the S-register will thus permit the cathode-ray beam to draw points, lines, dash and/or dotted lines, and also to brighten calibration marks on coordinate axes. It is also possible to cause portions of the display to blink.

A primer on program-controlled data transfers

We must next discuss how computer-supplied data words and S-register words are entered into the display registers. As it turns out, a little study of computer-data-transfer techniques will permit us to develop a very versatile set of display instructions.

Practically all computer-interface systems employ a "party-line" I/O bus of the general type illustrated in Figure 3. Here, all DAC and control registers intended to receive data words are permanently wired to the parallel computer I/O data bus. Additional party-line wires carry control-logic signals, which select a specific device and its mode of operation and also synchronize data transmission with the digital-computer operating cycle.

For a minimum of linkage hardware, interfaces work with programmed processor instructions. This method permits great flexibility at the cost of some ingenuity in assembly-language programming. Referring to Figure 3, each input/output (IOT) instruction in the processor instruction register places a specific device-selection code on a set of device selection lines. The *device selector* associated with each individual device is essentially an AND gate which recognizes the device selection-code and gates one, two, or three pulses (IOT pulses) from

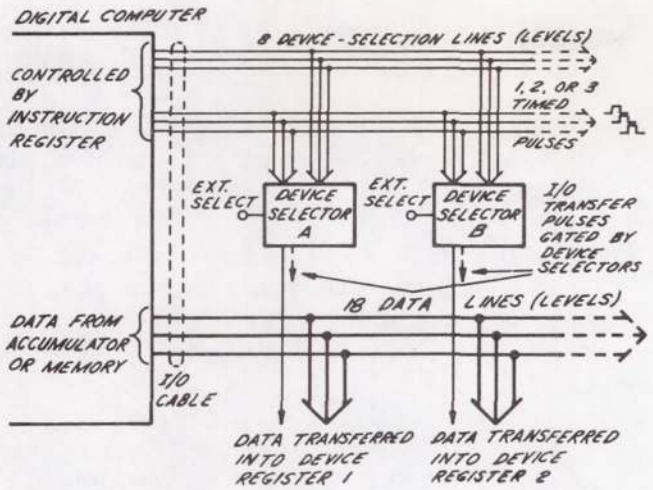


Figure 3—Program-controlled operation of a digital computer with party-line I/O bus. An I/O instruction addressed to a specific device is recognized by a device selector, which gates data-transfer pulses to the device in question, as shown in Figure 4

the processor into the selected device to effect data transfers and/or other operations.

Figure 4 shows in more detail how the parallel-connected device-selection and control lines of a typical laboratory computer system (Digital Equipment Corporation PDP-9) correspond to the format of an input/output-transfer (IOT) instruction word in the processor instruction register.[2–5] Bits 0 to 3 of the instruction word inform the processor that an input-output operation is wanted. Bits 6 to 11 place levels (0 or 1) on five *device-selection lines* parallel-connected to all devices on the I/O bus. When these lines carry the *device-selection code* associated with a specific device, its *device selector* (essentially an AND gate, Figure 4) gates (and regenerates) a set of one, two, or three successive processor-timed *command pulses* (IOT pulses) used to effect data transfers and other operations in the selected device in accordance with instruction bits 15 to 17. Not all instructions and devices utilize all three pulses. Bits 12 and 13 of the instruction word ("subdevice bits") can similarly serve to select subdevices or can further gate the command pulses to select device operating modes.

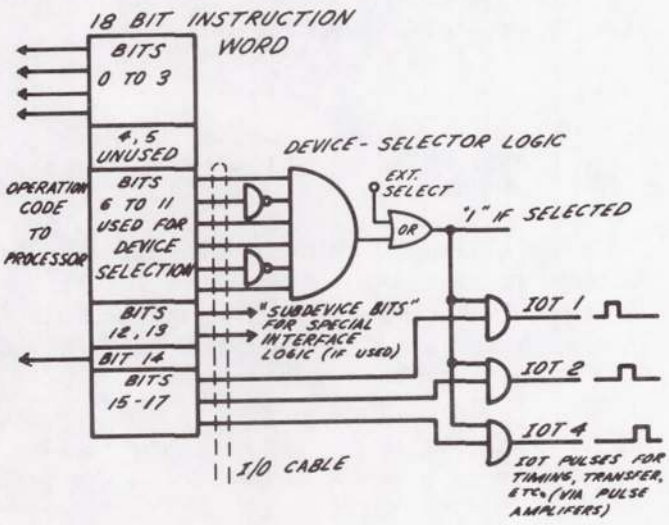
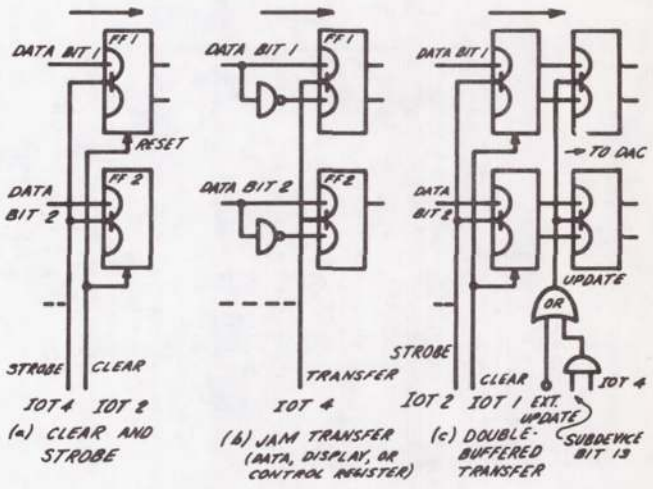


Figure 4—Program-controlled selection of device address and function: device-selector operation

The most common application of the device-selector-gated command pulses is *data transfer* from and to the processor; note that the pulses are synchronized with the processor operation cycle and, thus, with the processor's ability to transmit or accept data. Figure 5 illustrates the principal data-transfer techniques:

1. *Clear-and strobe Transfer* from the I/O bus parallel data lines into the flip-flops of a device register (Figure 5a). Each flip-flop is first cleared by IOT2; then IOT4 strobes the 1's on the data bus into the flip-flop register.
2. *Jam Transfer* (Figure 5b). A single command pulse (IOT4) sets or resets the device-register bits in accordance with the data-bus levels. Jam transfers require slightly more complex electronics than clear-and-strobe, but need only one pulse period for transfer. Jam transfer *must* be used whenever the register resetting opera-



Figures 5a, b, c—Parallel data transfer: clear-and-strobe (a), jam transfer (b), and transfer into a double-buffered DAC (c). Flip-flops with diode-capacitor gates, like those used in the Digital Equipment Corporation PDP-9, are shown. Level inputs are "0" at 0 volts and "1" at −3 volts, but flip-flop gates set or reset the flip-flop when the level input is "0," and the pulse input goes UP to "0"

tion would disturb device functions. This is true, for instance, with DACs required to have a continuous voltage output, and also with control registers which continuously establish a device status.

3. *Double-buffered-register Transfer* (Figure 5c). Data are transferred intothe buffer register by either a clear-and-strobe or jam-transfer operation and are then jam-transferred into the device register. Double-buffered DACs permit simultaneous transfer ("updating") of the analog output of two or more DACs.

As we shall see, suitable device-selecting instructions can also gate IOT pulses into a counter to increment the count.[2]

Display interface and display instructions

Figure 6 shows all the registers of the display interface. The interface employs two separate device selectors, A and B. Each device selector can be addressed by several different display instructions from the processor. Each instruction will generate a different combination of the two subdevice bits and the three IOT pulse obtainable from the device selector addressed, so that many different display instructions can be implemented (Figure 6 and Table I).

Referring to Figure 6, the most important mode of operation involves *simultaneous updating of the 9-bit*

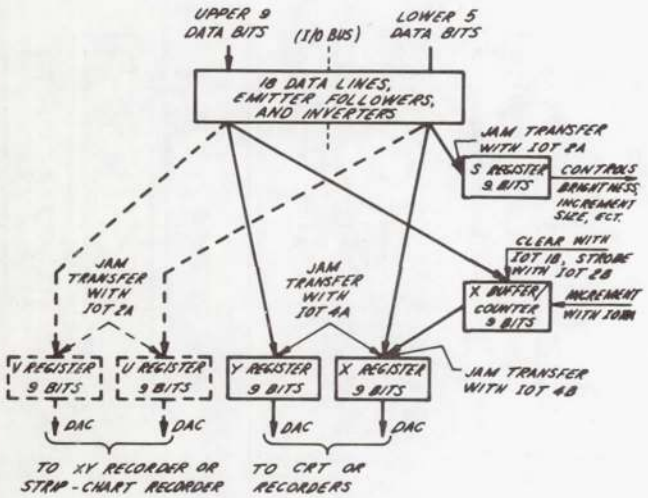


Figure 6—The display registers, with the data-transfer (IOT) pulse effecting data transfers from the I/O bus to the various registers. Subdevice bits can produce many different combinations of the six IOT pulses available from two separate device selectors A and B, so that many different display instructions can be implemented

X-and Y-DACs with "packed" 18-bit PDP-9 words from the I/O data bus:

> "Load Y-DAC with data bits 0 to 8, X-DAC with 9-17, and display (i.e., permit either point or line brightening)."

Using octal notation for the instruction-register bits in Figure 3 (each octal digit stands for three bits), the required instruction is

DXYS:	7 0	0 6	4	4
	Code for I/O Instruction	Device Selector A	Subdevice bit 12	IOT 4 (code is sum of IOT pulse numbers)

Such use of a single "packed" 18-bit word for each display point is very advantageous: compared to separate X and Y transfers, we are *halving* data storage requirements, and we are *doubling* the maximum possible point-display rate. If one still desires, however, to load the display with separate X and Y words (say, to avoid the extra time required for word-packing operations when points are displayed in the course of a fast computation), the instruction DXB (Table I) employs device selector B to *clear and strobe the X-buffer*, which is thus loaded with data-bus bits 0-8:

DXB:	7 0	0 5	0	3
	Code for I/O Instruction	Device Selector B	No Subdevice bits are "1"	IOT Pulses 1, 2

(see also Figure 7a). Next, one can *simultaneously load the Y-DAC with data-bus bits 0-8 and the X-DAC from the X-buffer, and display the resulting point or line*:

DYST:	7 0	0 6	6	4
	Code for I/O Instruction	Device Selector A	Subdevice bits 12, 13	IOT Pulse 4

To plot a curve Y(X) with equal X-increments, the X-buffer functions as a *counter* (Figure 7b). The instruction DYSI *increments the X-buffer, then loads Y from the data bus and X from the X-buffer, and displays*:

DYSI:	7 0	0 6	6	5
	Code for I/O Instruct on	Device Selector A	Subdevice bits 12, 13	IOT Pulses 1, 4

Table I lists the most commonly useful display instructions. Figure 7a shows in detail how the IOT

Table I—Display Instructions
(see also Figures 3 and 6)

(*a*) *Basic Instructions*

Mnemonic	Instruction	Octal Code	Device Selector	IOT Pulses	Subdevice Bits
DXYS	Load Y-DAC with data bits 0–8, X-DAC with 9–17, and display	700644	A	4	12
DSL	Load S register with data bits 9–17	700622	A	2	13
DXB	Load X-buffer with data bits 0–8	700503	B	1, 2	—
DYST	Load Y-DAC with data bits 0–8, transfer X-buffer to X-DAC, and display	700664	A	4	12, 13
DXC	Clear X-buffer and X-register	700505	B	1, 4	—
DYSI	Load Y-DAC with data bits 0–8, increment X, and display	700665	A	1, 4	12, 13

(*b*) *Instructions Which Move the Display Point without Brightening*

Mnemonic	Instruction	Octal Code	Device Selector	IOT Pulses	Subdevice Bits
DXYL	Load Y with data bits 0–8, X with 9–17	700604	A	4	—
DYLT	Load Y and transfer X-buffer	700624	A	4	13
DYLI	Load Y and increment X	700625	A	1, 4	13
DXL	Load X (Y is unchanged)	700507	B	1, 2, 4	—

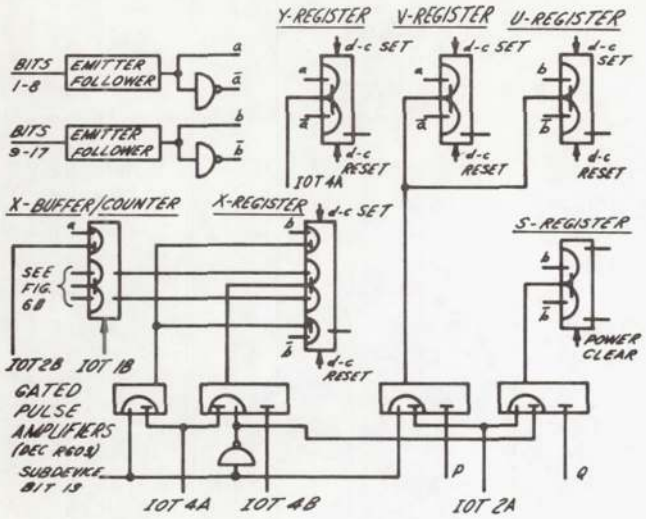
(*c*) *Instructions for Fast Drawing of Horizontal Lines*

(Note: Instruction bit 14 clears all data bits in the PDP-9. To draw *vertical* line segments, use DYST.)

Mnemonic	Instruction	Octal Code	Device Selector	IOT Pulses	Subdevice Bits
DXYC	Clear X and Y, and display	700654	A	4	12
DYCT	Clear Y, transfer X-buffer, and display	700674	A	4	12, 13
DYCI	Clear Y, increment X, and display	700675	A	1, 4	12, 13
DXS	Load X with data bits 0–8, and display (Y is unchanged)	700547	B	1, 2, 4	12

(*d*) *Instructions for U and V Channels*

Mnemonic	Instruction	Octal Code	Device Selector	IOT Pulses	Subdevice Bits
DUVL	Load V DAC with data bits 0–8, U with 9–17	700602	A	2	—
DAL	Load Y and V with data bits 0–8, X and U with 9–17	700606	A	2, 4	—
DAS	Load Y and V with data bits 0–8, X and U with 9–17, and display	700646	A	2, 4	12



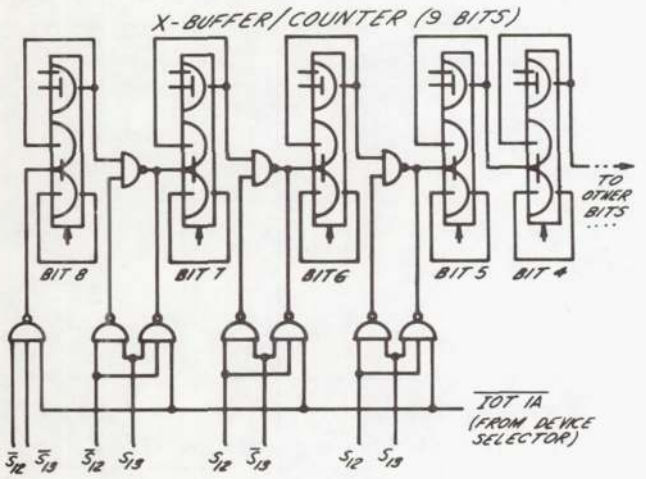


Figure 7a—This figure shows in detail how the different IOT pulses are gated to produce data transfers into the various display registers. Only one flip-flop of each 9-bit register is shown. Calibration switches can set the X, Y, U, and V registers to 0 and −10 V for calibration. The gated pulse amplifiers transmit pulses (−3 V to 0 V to −3 V) when an ungated pulse input is pulsed, or when a gated pulse input is pulsed with the level input at 0 V. Pulse inputs P and Q are for automatic data-channel operation (Figure 8)

Figure 7b—The X-buffer functions as a counter whose increment size is controlled by S-register bits 12 and 13

pulses and subdevice bits corresponding to each instruction gate data into the display registers. Note the following special features:

1. Instructions with subdevice bit 12 = "0" will *move the display point without brightening*. This is useful for creating gaps in lines or curves without any need to reload the intensity-controlling S-register.
2. Instructions clearing either X or Y are useful for *drawing coordinate axes*; selective point brightening produces *scale markers*.
3. Instructions updating Y or X alone can quickly create *vertical and horizontal lines* (coordinate lines, bar charts).

The instruction DSL loads the S-register with data-bus bits 9-17; Table II lists the functions of the individual S-register bits.

Table II—S-register bit functions

Bit	Function
9	enable data-channel clock
10	display POINT
11	display LINE, or recorder PEN DOWN
12} 13}	X-increment size: 00 = 1, 01 = 2, 10 = 4, 11 = 8
14	Beam intensity
15, 16, 17	Spares

The large number of different instructions possible with only two device selectors has also permitted us to transfer data into two additional DACs (U and V registers in Figure 6), which are intended for use with a four-channel strip-chart recorder but could, in principle, be employed to generate a different display on an additional oscilloscope.

Automatic data-channel operation

While program-controlled display instructions can conveniently alternate with other processor operations (computations and data taking) through program branching and interrupts, much more elegant and efficient display operation may be possible through direct memory access with the *automatic-data-channel hardware* available with many modern small digital computers. Our display is, therefore, designed to work with one of the cycle-stealing data channels built as a standard feature into the PDP-9. Only two processor instructions are needed to specify the *starting address* and *word count* of a block of memory associated with a display picture. The processor then steals memory cycles to output an 18-bit word directly from memory whenever an external *data-request pulse* is sensed. When a complete

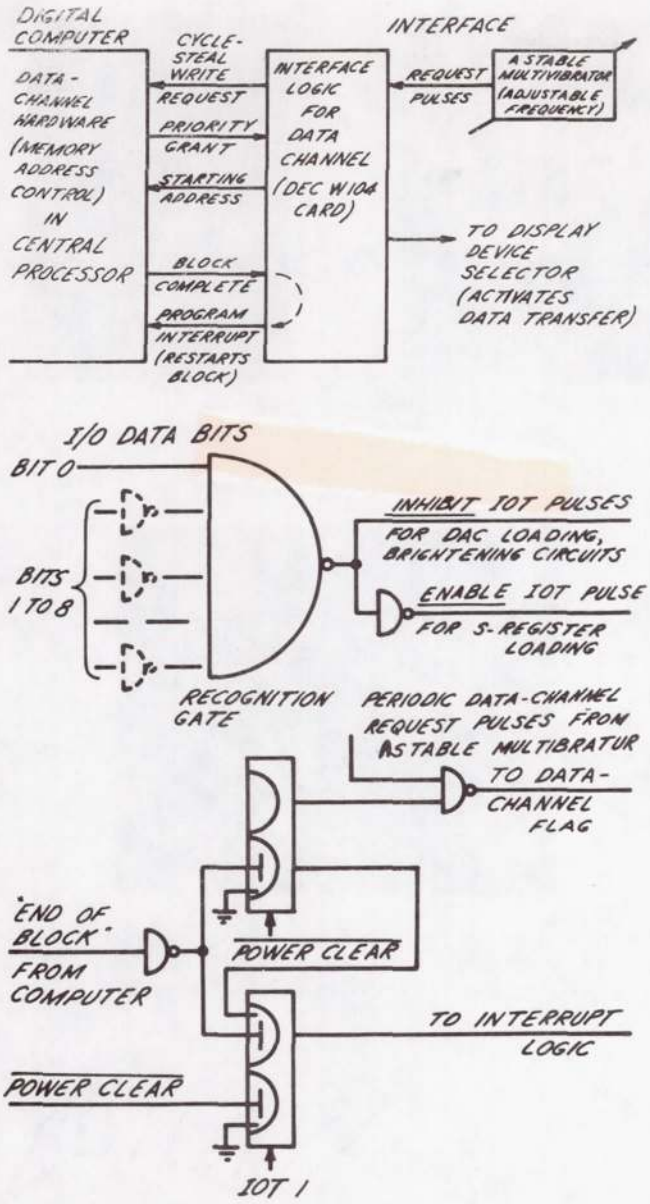


Figure 8a—Automatic-data-channel operation. Processor instructions set the number of words and the starting-word address in a display block and enable the display clock. From then on, data words are transferred to the display on each display-clock request pulse without any need for processor instructions. A data-channel end-of-block signal is used to interrupt the processor, which then restarts the sequence

Figure 8b—Recognition gate for data-channel S-register transfers (see text), and end-of-block interrupt flag

block of, say, 2,000 data words is finished, an *end-of-block pulse* can be used to interrupt the computer program and to reinitiate the block transfer (Figure 8a). No other display instructions are required, and memory locations corresponding to display points can be updated at any time between data-channel outputs. While data-channel word transfer rates as high as 250,000 words per second are possible, we usually set the transfer rate (which is determined by a simple astable multivibrator clock, Figure 8a) at about 100,000 words per second, both to give the computer a chance to compute and to permit cleaner line-segment generation.

On the face of it, data-channel outputs involve only *data words* (in our case packed 18-bit X, Y words, one corresponding to each display point). Since the data channel cannot transmit different display *instructions*, it would seem that we have lost the ability to change between the LINE and POINT display modes and to vary display intensities. If one had more than 18 bits, say with a 24-bit computer, one could use extra data bits as control bits, but this is not possible with the 18-bit PDP-9 and similar small processors. *A special trick, however, permits the use of data-channel produced data words for control purposes.* We recall that, with the 2's-complement code generally employed with DACs, the largest positive 9-bit excursion is 011111111, corresponding to + 9.98 volts, while the largest negative excursion is 100000000 or − 10 volts. This maximum negative excursion, thus, has no positive counterpart and is, in a sense, redundant. *We will, then, employ the last 9 bits of an 18-bit data word of the form 100000000-*sssssssss (Y = − 10 volts) *to set our 9S-bit register bits, rather than for positioning the CRT beam.* Referring to Figure 8b, a simple recognition gate detects the fact that the first 9 bits are 100000000 and gates the remaining bits into the S-register to control the beam intensity and the POINT or LINE mode of subsequent display points.

A second mode of data-channel operation employs the X-buffer/counter *to plot any one-dimensional array automatically against its index, without word packing.*

Operation with external oscilloscopes

To permit the use of our display interface with external cathode-ray oscilloscopes and recorders, the X and Y deflection voltages, an intensity-control voltage, and one of the S-register bits are brought out to front-panel terminals (Figure 2a). In particular, the display can thus operate with a standard five-inch oscilloscope equipped with a Polaroid Land camera to produce quick hard copy. For class demonstrations and visiting admirals, the display also operates with a large 24-inch display oscilloscope at reduced plotting rates. Depending on the external oscilloscope used, the beam-intensifying signal can be amplified and/or inverted.

Operation with xy recorders

To obtain hard copy directly at low cost, one can simply apply the X and Y deflection voltages to an xy recorder (servo table) at a suitably low word rate. For more flexible recorder operation, however, we added two additional digital-to-analog-converter channels (U-and V-DACs and registers, Figure 6).

For servo-table recording, one of the S-register bits is brought out to control the recorder pen-lift solenoid, with the pen DOWN whenever the display instructions call for the LINE mode (Figure 2a). The instruction DUVL produces IOT 2A (Figure 7a) to feed the 9-bit U and V registers with "packed 18-bit words for xy recorder operation. Since the relatively slow servo recorder cannot plot more than 5 points per second, the instructions must be tied to a suitably slow real-time-clock interrupt routine; slow data-channel operation in the manner of Figure 8 is also possible. For the same reason, the 160 pF delay capacitors shown with the X and Y DACs in Figure 2a are replaced with 5 μF plug-in capacitors in the U and V channels, so that line segments are drawn with a time constant of 0.1 sec.

Figure 10 shows a permanent record prepared with the new display unit.

Operation with stripchart recorders

For simultaneous plotting of up to four time-history records on a multichannel strip-chart recorder, the X, Y channels* and the U, V channels are alternately fed packed 18-bit words at a clock-controlled combined maximum rate of 200 words/second. Such multiple time records are especially valuable for recording variables in digital simulation of dynamical systems.

Discussion and follow-on program

Figures 9 and 10 show examples of display operation. As noted earlier, our display was originally intended mainly to produce solution curves and phase-plane diagrams for on-line simulation of dynamical systems. As it turned out, the display is very useful for producing much more general pictures (Figure 9), so that we are planning the addition of a light pen and protractor dial, plus SKETCHPAD-type software[6] for computer-aided drawing operations with the PDP-9.

The most useful feature of our simple display is the possibility of refreshing the display with packed 18-bit words from the standard automatic data channel of the 18-bit PDP-9. Since these 18-bit words fully utilize

* The small capacitors in the X and Y channels (Figure 2a) have essentially no effect at the 50 to 200 Hz word rates used for strip-chart recording.

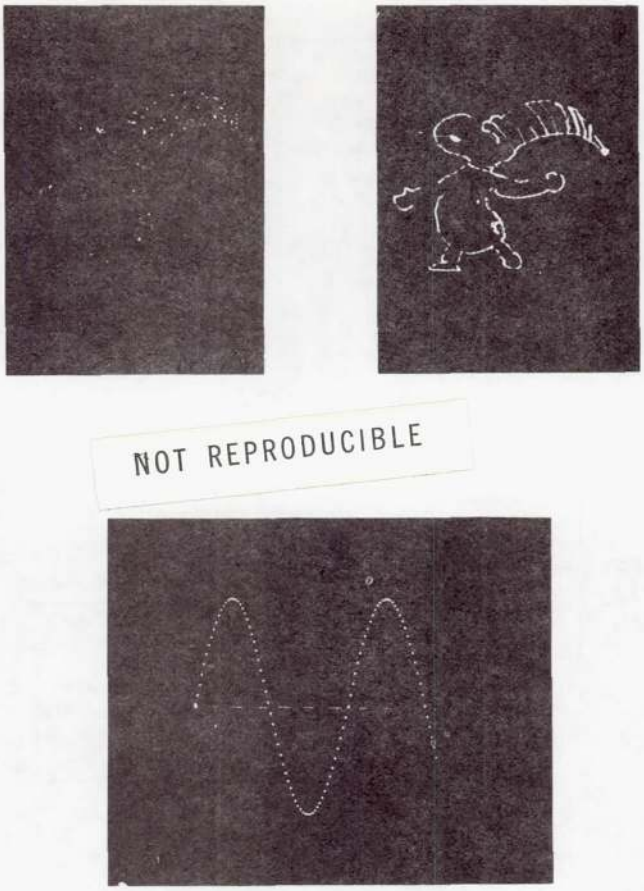


Figure 9—Cathode-ray-tube displays using the PDP-9 data channel at 125,000 points/second: POINT mode (a), LINE mode (b), and mixed POINT and LINE mode (c)

Figure 10—Hard copy produced on a Moseley xy recorder (servo table) in the LINE mode

their PDP-9 registers, we can afford to eliminate the hardware and programming complication of a display-refreshing core or delay-line buffer. With processor and display sharing a common memory, our automatic data channel can do much more than refresh a single display. The channel can also transmit and mix several displays or *parts* of displays (curves, figures, characters), each corresponding to a memory block with a program-selected starting address and block length.[3] Indeed, a block or blocks can be displayed with the automatic data channel while the processor modifies another portion of the display.

The Dertouzos line-segment-generating technique, as modified by our intensity-compensating circuit, was thought to be especially suitable for curve interpolation and stroke-implemented characters. In our actual operating experience, though, most users have employed the LINE mode mainly for servorecorder plots; they seem to prefer the simpler programming possible when using only the POINT mode for CRT display. If we need not alternate LINE and POINT modes it is, for instance, simpler to obtain "incremental" display operation, with successive packed words

$$({}^{k+1}\mathrm{X},\ {}^{k+1}\mathrm{Y}) = ({}^{k}\mathrm{X} + {}^{k}\Delta\mathrm{X},\ {}^{k}\mathrm{Y} + {}^{k}\Delta\mathrm{Y})$$

2's-complement-accumulated in the PDP-9 accumulator before transmission to the display or to a data-channel block.

In view of the popularity of small 16-bit digital computers, we have also tried the display with our 9-bit DACs restricted to 8-bit operation. A display thus obtained with packed 16-bit words has, of course, less resolution than the 18-bit display, but still appears to be acceptable for many purposes. This opens the interesting possibility of combining our display system permanently with a small 16-bit processor as a moderately sophisticated stand-alone display unit

The provision of the X-buffer/counter in our display permits *display operation with separate X and Y data transfers*. This is mandatory with 12-bit computers (like the popular PDP-8 series), and even speeds up some PDP-9 programs, since no word-packing is needed (see Appendix A for explicit programmed-data-transfer routines). It is only fair to state, though, that all our users to date have preferred packed-word operation with the PDP-9, so that we could have omitted the entire X-buffer logic (including our extra device selector).

ACKNOWLEDGMENTS

The display system described in this paper is part of a continuing study of digital-, analog-, and hybrid-computer simulation at the University of Arizona. For their support of this study, we are very grateful to the National Aeronautics and Space Administration (Grant NsG-646 and Institutional Grant to the University of Arizona), the National Science Foundation (Grant GK-1860 and Institutional Grant), and to Drs. G. W. Howard, R. H. Mattson, D. L. Patrick, A. B. Weaver, and E. N. Wise of the University of Arizona for allocating institutional-grant funds and University resources.

REFERENCES

1 M L DERTOUZOS
PHASEPLOT: An on-line graphical display technique
IEEETEC April 1967

2 G A KORN
Digital-computer interface systems
Simulation December 1968

3 *PDP–9 user handbook*
Digital Equipment Corporation Maynard Mass 1967

4 *Logic handbook*
Digital Equipment Corp Maynard Mass 1968

5 *Small Computer Handbook*
Digital Equipment Corp Maynard Mass 1968

6 I E SUTHERLAND
SKETCHPAD, a man-machine graphical communication system
Proc S J C C 1963

APPENDIX A: BASIC PDP–9 DISPLAY ROUTINES

LAC X	/load accumulator with X	– 2 μsec
DXB	/load X-buffer with X	– 4 μsec
LAC Y	/load accumulator with Y	– 2 μsec
DYST	/transfer X-buffer and Y, and display	– 4 μsec
		12 μsec

LAC Y	/load accumulator with Y	– 2 μsec
DYSI	/increment X-buffer, transfer	– 4 μsec
	/X-buffer and Y, and display	
		6 μsec

LAW 17000_8	/load accumulator with mask	– 1 μsec
AND Y	/mask last 9 bits of Y out	– 2 μsec
DAC TEMP	/ and save result	– 2 μsec
LAC X	/load accumulator with X	– 2 μsec
CLL	/clear the link*	– 1 μsec
LRS 9	/shift 9 places*	– 6 μsec
ADD TEMP	/combine with Y	– 2 μsec
DXYS	/transfer X, Y, and display	– 4 μsec
		20 μsec

LAC XY	/deposit X, Y in accumulator	– 2 μsec
DXYS	/transfer X, Y, and display	– 4 μsec
		6 μsec

* not needed if X was scaled previously