

# 928 Graphical Display System

MARCONI  
ELLIOTT  
COMPUTER  
SYSTEMS

# **928 Graphical Display System**

## **Key to contents**

---

Introduction 

---

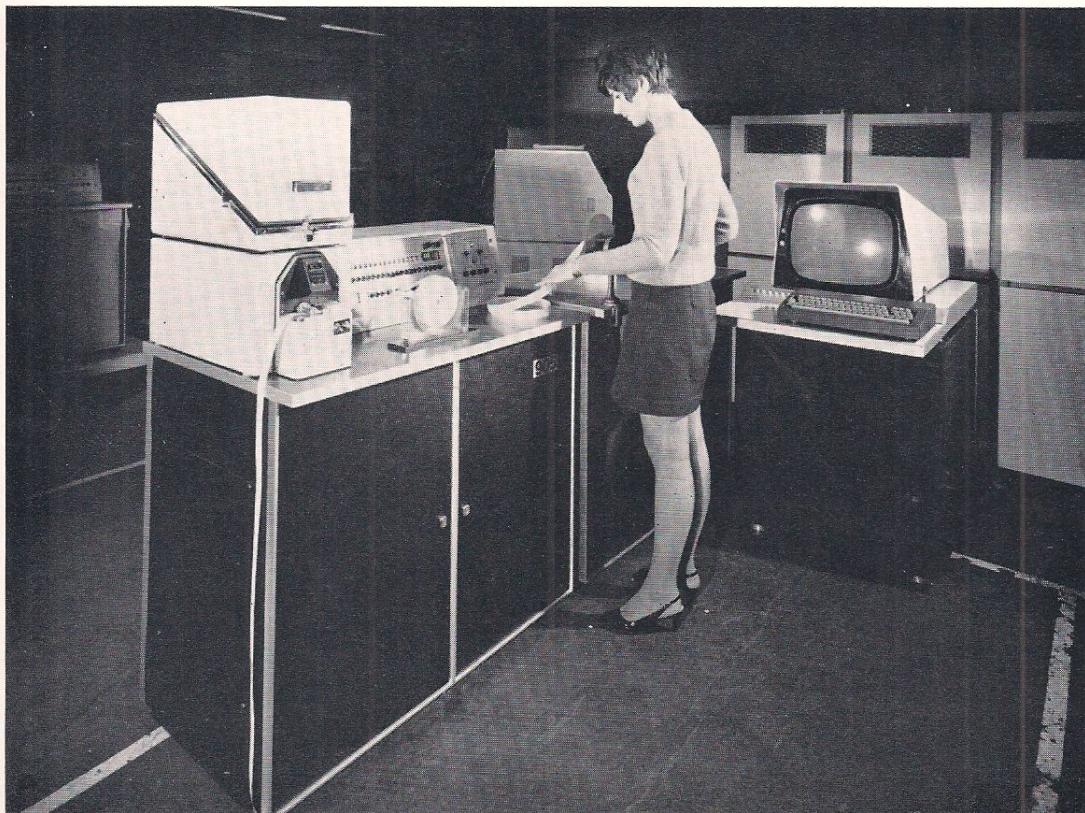
Description 

---

Software 

---

# 928 Graphical Display System Introduction

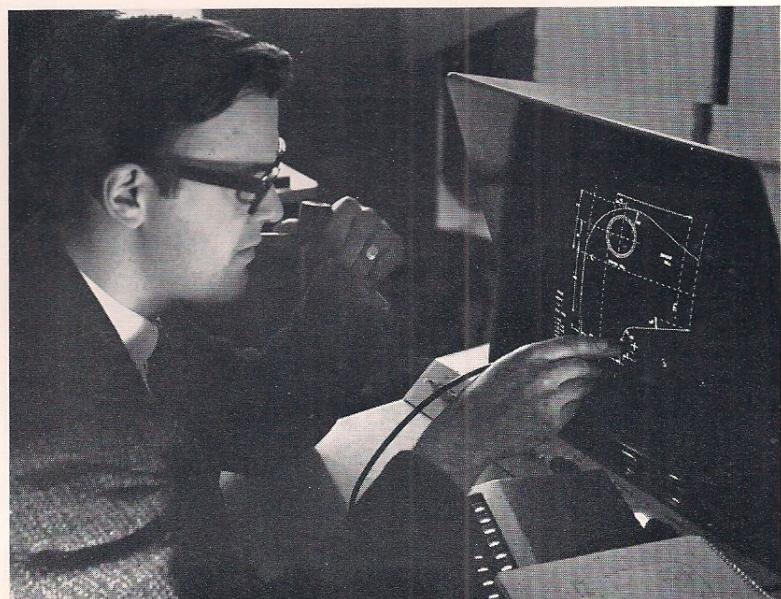


The 928 Graphical Display system is a powerful graphical processing system. It operates in association with a Marconi-Elliott 905 Computer which provides the computation power necessary for graphic manipulation and for the related calculations. The system includes a graphic console with a light pen, character, vector and arc generators, a graphic control unit, and a keyboard. The system can be extended by the addition of further keyboards and graphical consoles.

An extensive graphical programming repertoire complements the hardware. A full range of display capabilities can be performed including circles, enlargement, contraction and rotation about the X, Y or Z axis.

The 928 Display has been developed as a satellite to central processors of all makes. Having its own compact computer to perform all the routine tasks concerned with the display manipulation eliminates the problem of surrendering store capacity and time in the parent computer to these duties. All information necessary for solving the problem is stored in the parent computer and only the specific information needed to display a

picture to the user need be transmitted to the 928. The user may modify the displayed image by use of the light pen and when he is



satisfied with the revised pattern the 928 can communicate the revision to the parent computer which will up-date its stored information.

### Operation

The 928 display functions by autonomously extracting data from any area of the core store (referred to as the display file) on a cycle stealing basis. The data are used to generate an image on the cathode ray tube and to control the display system, particularly with respect to the input devices such as the light pen and the function keys.

The display file is created and changed by programs obeyed by the 905 computer either in response to the graphical input devices or as the result of calculations performed. The operation of the 905 is concurrent with the operation of the display so that there is no loss of picture while the 905 is computing. The computation could, if required, be by a completely independent program.

### Packaging

The 928 display consists of four items:

- (1) The viewing unit and character generator with their associated power supplies.
- (2) The light pen and its power supplies.
- (3) The keyboard.
- (4) The controller.

Items (1) and (2) are housed in the graphical console which is connected by a single cable, normally 7·6 m (25 feet) long. The 17-inch cathode ray tube has an orange phosphor and is retained by an amber-tinted PVC implosion shield to give good contrast and to reduce operator fatigue when using the display for periods of many hours. The light pen is a small hand-held device similar to a writing pen. It is connected by a flexible light guide to a light sensitive circuit. The light pen can be used as a pointer for selecting purposes or as a means of effectively 'drawing' upon the c.r.t screen.

The keyboard may be placed on any suitable working surface or upon the shelf in front of the viewing unit. The keyboard is connected by a single cable normally 10 m (33 feet) long. The associated logic boards and power supply occupy a space 222 mm (8·75 inches) high by 343 mm (14 inches) deep of a 19-inch rack.

The controller occupies 19-inch racks and may be housed in a 900 Series cabinet.

Each 928 graphical display system is, of course, tailored to suit the exact requirements of the user. Taking a typical arrangement as shown in the block diagram on page 3 the units would be as indicated in the dimension drawing.

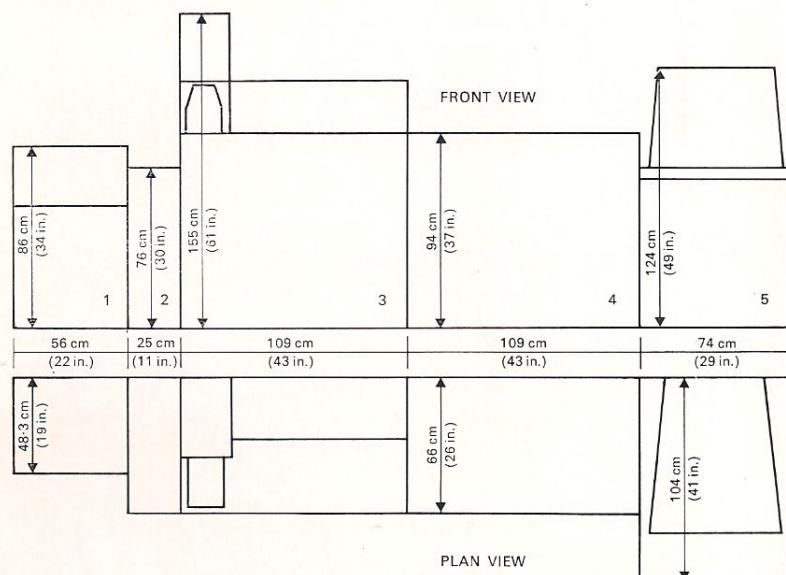
Each 928 Graphical Display System is, of course, tailored to suit the exact requirements of the user. Taking the typical arrangement shown in the block diagram on page 5, without options, however, the units would be as indicated in the dimension drawing below. The units are 1) teleprinter; 2) paper tape bin; 3) 905 computer (16384 word store basic)

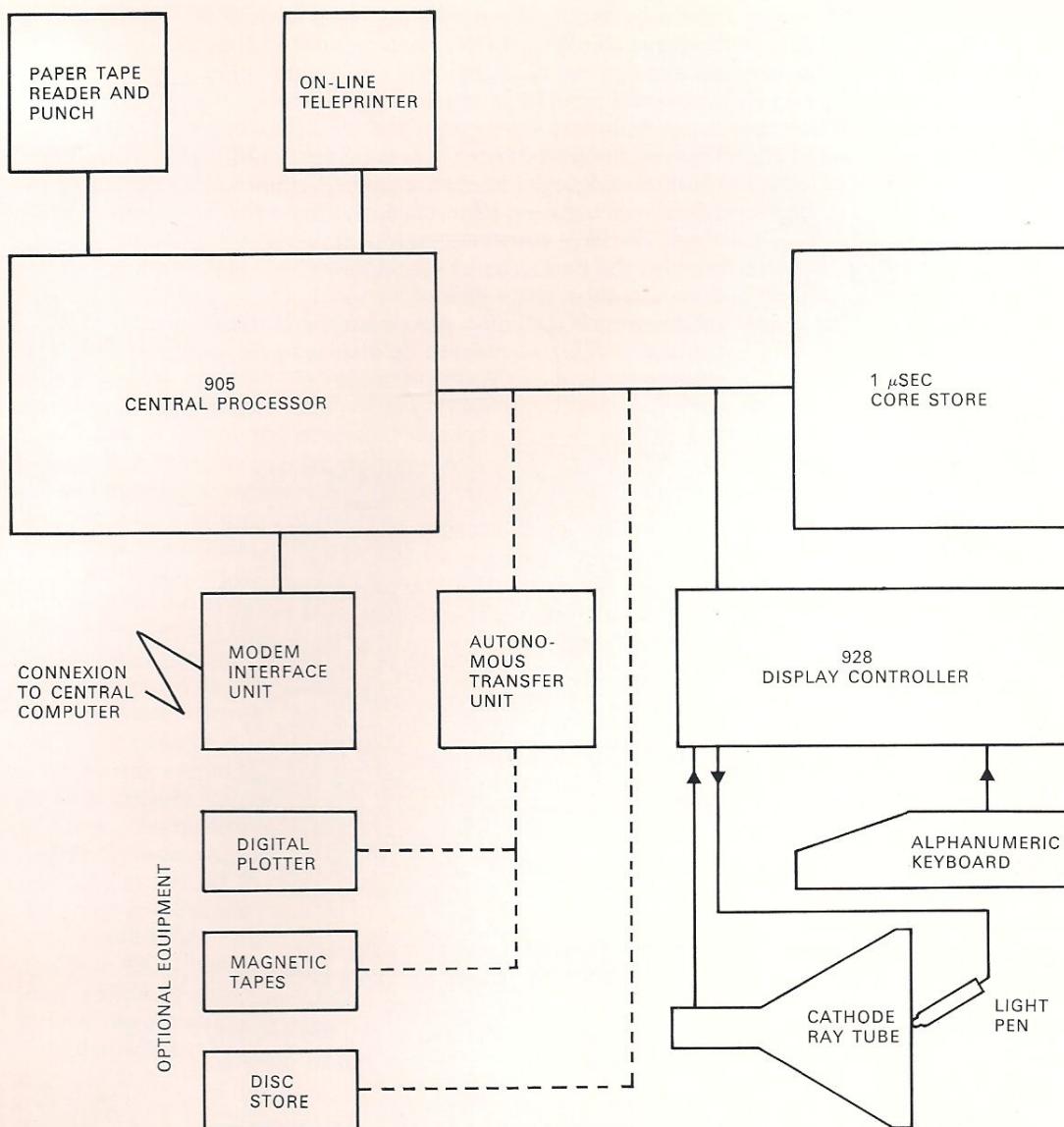
with control box and paper tape equipment on top; 4) cabinet containing the graphical display controller and the modem interface and its power supplies; 5) viewing unit. The alphanumeric keyboard is not shown and can stand on any convenient surface – the bench top of cabinet 4 or 5 for example. The viewing unit may be up to 9·1m (30 feet) away from the graphical display controller and is connected to it by a single cable.

This typical system, without options, draws a power consumption of about 3kVA. Ordinary commercial single-phase a.c mains is usually 'clean' enough. Special environmental conditions such as a sprung floor or air-conditioning are not necessary – reasonable laboratory or office conditions are quite satisfactory.

The typical system may be expanded by adding further viewing units together with additional core store or else backing store in the form of a magnetic disc. Further expansion is possible by adding one or more of the following peripheral devices, magnetic tape, digital plotter, line printer or card reader. These devices can be operated autonomously when the system incorporates an autonomous transfer unit.

### Dimensions



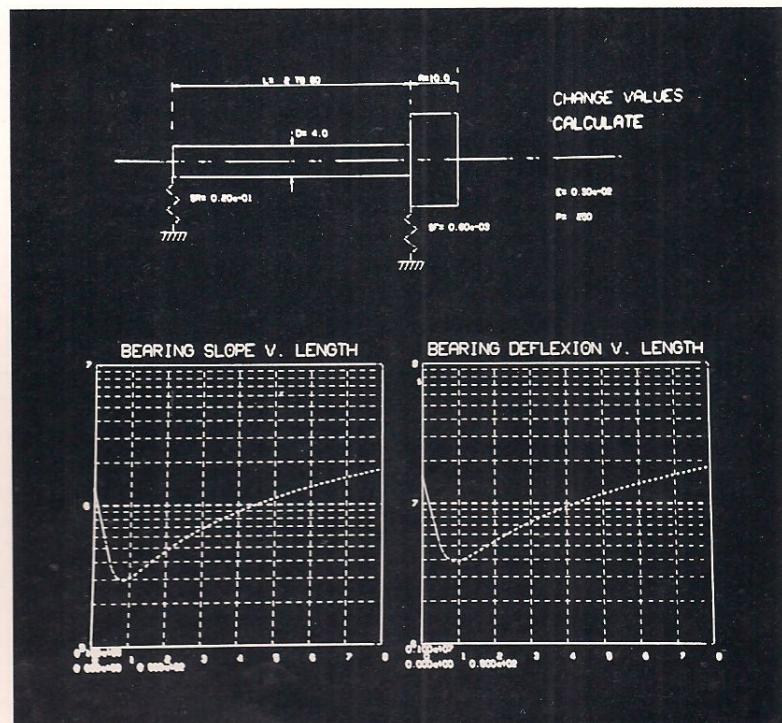


*A Typical Satellite System*

### The Graphical Console

This is the operator's console which contains the cathode ray tube viewing unit and light pen, and has been carefully designed for ease of operation. The cathode ray tube is a 17-inch tube coated with a special long persistence orange phosphor, which enables the display to be refreshed at the low rate of twelve frames per second. This low rate enables a large flicker-free picture to be generated. This special phosphor has a sharp fall-off in light output after a period of 160 milliseconds, so that there is no smear on rapidly changing displays.

The picture on the screen is built up by means of points, lines and characters positioned on a basic raster of 1024 by 1024 points occupying an area of 245 mm (10 inches) square. Two 10-bit binary numbers are required to specify a position on the screen. The picture is generated mainly by changing the contents of the horizontal and vertical position registers so that the electron beam traces out the required picture. This is known as 'directed scan' as opposed to the 'line scan' method used with television. Different parts of



the picture can be displayed at different brightness levels under program control. This can be used to accentuate details by using the brightest level, or to fade background information using the lower level of brightness.

### The Light Pen

This is contained in the graphic console and is a small hand-held device similar to a pen. It is connected by a flexible light guide to a light-sensitive circuit. This circuit is sensitive to a particular wavelength emitted by a small percentage of a very short persistence phosphor (less than 1 microsecond persistence) which is mixed with the long persistence phosphor. When the light pen is activated by a foot switch and pointed at a visible picture part, this part is detected by the pen the instant it is drawn so that the graphic processor can determine what feature is currently being drawn. It can then interrupt the processor to take the appropriate action. The pen is used in this mode as a pointer, but it can also be used to input new positions by means of a simple program which continuously tracks the position of the pen by using a small cross drawn on the screen.

### The Keyboard

There are sixty alphanumeric keys, two shift keys and a separate block of twelve function keys. A mechanical interlock prevents the simultaneous depression of more than one key. There is no practical limit to the typing speed as the keyboard is capable of fifty operations per second.

Each key on the keyboard outputs two 6-bit words to the controller. The first 6-bit word is the control word which identifies the keyboard and defines whether it is a character code or a function code. The other 6-bit word is the data word which contains the 6-bit character code or the function code. Whenever a key is depressed, an interrupt is sent to the computer which then reads both the control word and the data word.

### The Character and Vector Generators

Any picture could be made up entirely of individual spots positioned close together. This is, however, inefficient in both speed of generation and in the amount of core store necessary to define the picture. The 928 display is, therefore, equipped with special items of hardware to draw the most frequently used graphic features, characters and straight lines.

### Character Generator

The character generator is located in the console. It generates an alphanumeric symbol at the current beam position and increments the horizontal deflection register an appropriate amount ready for the next character. The characters are formed by short strokes on a basic 7 by 4 grid. The description of each character is held in a fixed diode store. This particular method produces characters of good legibility with the greatest

reliability and freedom from the need for frequent adjustment. The character repertoire consists of eighty-eight visible characters plus space and newline, and all of these are available in two sizes controlled by program. The visible character set is extensible to 120 characters. Special hardware ensures uniform brightness between characters of different sizes. The high speed deflection circuitry enables the basic size of character to be drawn in an average time of 11 to 21 microseconds including positioning, and this enables 8192 characters to be displayed simultaneously.

### Vector Generator

The vector generator is incorporated in the control unit. It draws straight lines automatically on the screen at any angle and of any length. The co-ordinates of the end point are simply specified by program relative to the starting point. The display code for a vector is therefore independent of its position on the screen, and this is very useful in the subpicture feature which is available. The lines are drawn by digitally incrementing or decrementing the contents of the horizontal and vertical deflection registers at a constant rate of 1.3 increments per microsecond. This gives accurate lines of constant brightness with complete stability. As with the character generator, no periodic adjustments are necessary.

An extension of the vector generator hardware enables arcs or complete circles to be drawn using only four store words. The arcs are relatively specified, as are the vectors, and are produced digitally in a similar manner.

### The Graphic Control Unit

This control unit functions by extracting the data from the display file held in the core store autonomously every 100 milliseconds. Once initiated by the 905 this refreshing is completely automatic. The data are in the form of single-address instructions with a format almost identical to that used by the 905, with a 4-bit instruction field, a 13-bit address or operand and a modify bit which determines whether the operand is the 13-bit address or the contents of the location specified by it.

This direct addressing technique is relatively unknown with computer graphic displays and adds considerable power to the instruction code. It enables direct reference to a variable in a complex data structure to be made, without the need to change the display file when this variable is altered.

The range of instructions enables all the graphic features, such as co-ordinate positioning, vectors and characters, to be generated with relatively few words. For instance a single character instruction displays two alphanumeric symbols. As well as the graphic instructions, there are instructions specially designed to aid the program. These include 'jumps', which have provision for preserving the current address, so enabling

subroutines to be used in the display code, and control instructions to vary parameters like character size and brightness levels, as well as the provisions for interrupting the 905 computer.

A further innovation is the introduction of an arithmetic instruction which can be used, for instance, to create conditional jumps by operating on the sequence control register, or to store the registers of the 928 in the core store.

There is an occasional need for the temporary interruption of picture generation to perform some function like informing the 905 computer of a 'light pen see', or the pressing of a function key by the operator. This is accomplished in the 928 by storing the current instruction address and transferring control to an interrupt routine (still obeyed by the 928). The cause of the interrupt is determined by using the conditional jumps and appropriate action is then taken.

Depending on the reason, the 928 may handle the interruption without using the 905 (for instance in regular tracking of the light pen) or it will interrupt the 905 to take action (as in the case of a 'light pen see').

Other features of the 928 control units include 'picture scissoring' and the 'naming' of items. The horizontal and vertical deflection registers are actually of 13-bit length, enabling points off the screen to be specified. By using this feature the display can be used as a window on a much larger area (up to 2.125 m (80 inches) square. If programmed using relative instructions, this window can be moved rapidly over the whole area by altering only the initial starting co-ordinates.

Instructions are provided to enable sections of the display code, called items, and also collections of these items such as subpictures, to be assigned an identifier or 'name'. This naming structure functions regardless of the nested depth of the subpictures. By using the 'naming' feature, rapid identification of picture parts can be made either by the program or from the operator via the light pen.

Construction of the 928 control unit is similar to that of the 905 central processor. High reliability integrated circuits are used and full engineers' facilities are provided.

By careful design many improvements in processing speed have been attained. Whenever possible the next instruction is extracted from the store while the present instruction is being obeyed, thus saving access time. In addition all non-graphic instructions can be completely overlapped with the settling of the electron beam during co-ordinate positioning.

#### Satellite Operation

The 928 Graphical Display System is eminently suitable for use as a satellite display to a larger central computer. When used in the satellite mode the 905 computer deals with the graphical operations such as pen tracking, drawing or deleting lines, etc., while the central machine performs any lengthy calculations and administers the 'data base'.

The 905 may be attached to any large computer, either via a GPO modem attached to a modem controller, via a British Standard Interface, or by a specially designed Marconi-Elliott Interface Matching Unit.

**Marconi-Elliott Computer Systems Limited**

A GEC-English Electric Company

Elstree Way, Borehamwood, Hertfordshire  
Telegrams and Cables: Elliotauto Borehamwood  
Telex: 22777 Answerback: Ellauto Borhwd  
Telephone: 01-953 2030

**928 M TL/1**

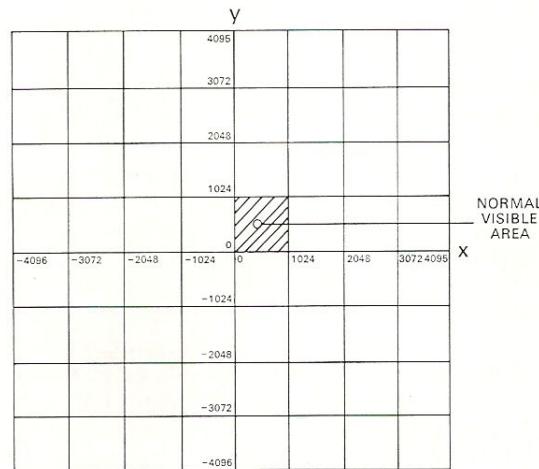


# 928

# Graphical Display

## CONTENTS

	Pages
Display Organization	1
Instruction code	2
Holds	5
System control	5
Performance and operating characteristics	6
Controls and lamps	6
Physical details	6



*Display Co-ordinate Addressing.*

0 to +1023 but may be larger depending on a scaling parameter.

The following registers are of interest to the programmer.

### X-Register

This is of 13 bits length and controls the horizontal deflection of the electron beam. This it does by using ten consecutive bits to feed a digital to analogue converter in the console. The 10 bits are specified by the scaling factor.

### Y-Register

This is again 13 bits and is identical in function to the X-Register except that it controls vertical deflection.

### H-Register

This is of 18 bits length and is usually used to hold a 13-bit signed horizontal vector component or line co-ordinate although it is used by the Jump instruction and also as a store address register in the Reg. move instruction.

### V-Register

This is of 13 bits length and contains the vertical vector component or line co-ordinate.

### P-Register

This is an 18-bit parameter register which holds details of character size and brightness level.

### S-Register

This is a 16-bit sequence control register which holds the address of the next instruction to be obeyed.

## DISPLAY ORGANIZATION

A picture is built up as a series of spot positions, lines and characters. There is a square array of possible spot positions arranged in 1024 rows and 1024 columns giving a resolution of approximately 0.01 inch on a 10-inch square picture. Any spot, once illuminated will remain bright for some 100 milliseconds and then fade. The entire picture must be redrawn at intervals no greater than this if a flicker-free picture is to be produced. This is done automatically. A single output of the complete picture is referred to as a 'frame'.

Information is extracted from the computer store one word at a time. Words usually come from consecutive locations, but provision is made to break sequence where necessary by the use of 'Jump' instructions and subroutines instructions. The instructions include vector and line generation, arc generation, co-ordinate positioning and character drawing as well as 'non-graphical' instructions such as naming, parameters, and simple arithmetic operations.

Although the visible picture area is only  $1024 \times 1024$  spot positions in size, provision is made to allow parts of the picture to be off the screen. All possible spot positions can be specified by a pair of 13-bit signed integers held in 2's complement form. The co-ordinates therefore lie in the range +4095 to -4096.

The visible section is normally in the range

## Instruction Format

The instruction format is shown below and consists of the address field N which is the lower 13 bits of the word, the function part which is bits 14–17 and the address mode bit which is bit 18. The 'Function' part determines which instruction is to be performed and defines sixteen instructions. The address mode bit 18 determines how the address bits 1–13 are to be interpreted, if bit 18 is a 0 then the address bits define a 13-bit operand (Q) to be used in the instruction. If however bit 18=1 then the 13 address bits specify a location within an 8192 word module of store, the contents of which are then used to define the operand as stated below. The actual 8192 word module used is specified by bits 12–9 of the parameter register (see Function 10, page 3).

If the word specified above has bits 18–1 and bit 17=0 then bits 16–1 define a further location whose contents again define the operand by bits 18 and 17 in the same manner.

If, however, bits 18 and 17 are other than 1,0 in the first word specified or subsequent word then bits 18–1 specify the operand which depending on the instruction may not use all 18 bits.

This addressing system therefore gives three modes of operand specification. These are Literal, Direct Addressing and Multi-level Indirect Addressing.

The instruction format and addressing modes are summarized in the following table.

*Instruction Format*

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
M O D E																	ADDRESS    N

If bit 18=0 Q=N.

If bit 18=1 Q=N' or Q=((N')...') to the specified depth.

## INSTRUCTION CODE

Details of all the instructions are given below with their effects on the registers and the display.

### Function 0 – Jump

This instruction stores the current contents of the sequence control register into the location specified by the contents of the system control location P+2 and then loads S with the operand. It enables a break sequence to occur and also performs a naming function.

$(P+2)' := S$   
 $S := Q$

*Note:* Bits 14–16 of S can only be affected by a Jump instruction having bit 18=1 (Direct or Indirect) Literal jumps can only set a value in the current 8192 word module of store.

### Function 1 – Subroutine Enter

This instruction increments the system control location P+2 by one and writes the current value of the sequence control register S into the location whose address is now in P+2, i.e. into the next location to the one currently being used for naming. Location P+2 is then again incremented, it then becomes a pointer for naming at the new subroutine level. Finally, the sequence control register S is loaded with the operand, which may be literal, direct or indirect.

$(P+2)' + 1 := S \quad (1)$

$(P+2)' := (P+2)' + 2 \quad (2)$

$S := Q \quad (3)$

This instruction may be repeated within a subroutine to a depth of seventeen nested subroutines, but one level is always reserved for 'hold' routines and hence only sixteen levels are available for general display file use.

Protection against corrupting the software by entering a subroutine at a level greater than sixteen is provided by hardware which does not increment P+2 in stage (2) above when the level is sixteen. Thus links are overwritten and routines cannot be correctly unnested but no damage occurs to any part of the display software.

### Function 2 – LOAD H

This loads the horizontal vector component register H and is issued normally just prior to a draw vector instruction (Function 5) but can be used for other functions.

$H := Q$

### Function 3 – LOAD V and DRAW LINE

This loads the vertical co-ordinate register V with the operand and draws a line to the absolute position as specified by H and V.

The line drawing logic interprets the H and V registers as containing 13-bit 2's complement co-ordinates and a line is drawn by suitable digital increments in the X and Y registers. At the end of the line drawing H and V are reset to zero.

Hence  $V := Q$   
 $X := H$   
 $Y := V$  } with a visible line  
 $H := V := 0$

### Function 4 – LOAD H and DRAW VECTOR

This loads the horizontal vector-component register H and then draws a vector as specified by H and V. The vector logic interprets H and V registers as holding 13-bit signed (2's complement) vector components which cause the vector to be drawn by suitable digital increments in the X and Y registers. At the end of the vector drawing H and V are both reset to zero, this enables



horizontal and vertical vectors to be specified with only one word per vector

```
H:=Q
then X:=X+H
      Y:=Y+V (with a visible line)
then H:=V:=O
```

#### **Function 5 – LOAD V & DRAW VECTOR**

This loads the vertical vector component register V and then draws a vector as defined in Function 4 above.

```
V:=Q
then X:=X+H
      Y:=Y+V (with a visible line)
then V:=H:=O
```

#### **Function 6 – INVEC X (Invisible Vector)**

This causes a relative co-ordinate change in the horizontal axis with no visible output on the screen

X:=X+Q

#### **Function 7 – INVEC Y (Invisible Vector)**

This causes a relative co-ordinate change in the vertical axis with no visible output on the screen.

Y:=Y+Q

#### **Function 8 – SPOT X**

This loads the horizontal deflection register

X:=Q

#### **Function 9 – SPOT Y**

This loads the vertical deflection register Y

Y:=Q

#### **Function 10 – LOAD PARAMETERS**

This loads the parameter register P with the operand Q, the action of these parameter bits is given below :

##### *Brightness Control*

- Bits 2,1 are 0,0 no change
- are 0,1 set brightness dim
- are 1,0 set brightness normal
- are 1,1 set brightness bright

##### *Character Size*

- Bits 4,3 are 0,0 no change
- are 0,1 set small characters
- are 1,0 set medium characters
- are 1,1 undefined

##### *Pen Control*

- Bits 6,5 are 0,0 no change
- are 0,1 pen not enabled
- are 1,0 pen enabled
- are 1,1 undefined

##### *Line Type*

- Bits 8,7 are 0,0 no change
- are 0,1 line type continuous
- are 1,0 line type dotted thirty-two increments frequency
- are 1,1 line type dotted sixteen increments frequency

Note this applies to arcs as well as Vectors.

##### *Address Modified*

Bits 12–9 specify a modifier which is used to allow direct instructions to specify any part of the store.

Bits 11–9 are shifted left to occupy bits 16–14 of the address field and are added to the 13-bit address in any direct instruction.

Bit 12=0	no change
Bit 12=1	bits 11, 10, 9=
	000 1st 8192 word module
	001 2nd 8192 word module
	010 3rd 8192 word module
	011 4th 8192 word module
	100 5th 8192 word module
	101 6th 8192 word module
	110 7th 8192 word module
	111 8th 8192 word module

##### *Slave Status*

bits 16, 15, 14, 13 are 0000 no change otherwise

bit 13 controls Master Screen

- 1 is ON
- 0 is OFF

bit 14 controls Slave Screen 1

- 1 is ON
- 0 is OFF

bit 15 controls Slave Screen 2

- 1 is ON
- 0 is OFF

bit 16 controls Slave Screen 3

- 1 is ON
- 0 is OFF

##### *Blink Status*

Bits 18, 17 specify a 'blink status' which when in the 'ON' state causes the particular part of the picture to be displayed once every fourth frame and thus makes it flash on and off.

Bits 18, 17 are 0,0 no change

- 0,1 blink OFF
- 1,0 \*
- 1,1 blink ON

\*Note bits 18, 17=1,0 specifies an indirect instruction.

#### **Function 11 – CONTROL**

This loads another parameter register with the operand Q, but since this instruction is meant mainly for systems control there is no formal register which is available to programmers and no necessity for preserving in sub-routines or 'Holds'. The interpretation of the bits is given below:

- Bit 3 is a 1 interrupt on Level 3
- Bit 4 is a 1 stop until a Processor 'Continue' (See SYSTEM CONTROL section)

- Bit 5 is a 1 stop until frame timer expires then start frame timer.

Note the frame timer is of 80 milliseconds duration and if bit 5 is a 1 then the stop action is performed before the timer is restarted. Such an instruction will normally occur at the end of the display file just before a Jump back to the start of the display file. HOLDS will be allowed when controller is stopped waiting for Frame Timer to expire.

- Bit 6 is a 1 EXIT from subroutine and permit holds.

Important: bit 5 and bit 6 must not occur simultaneously in one instruction.

The action of the exit instruction is as

follows. The system control location P+2 is decremented by 1 and used as an address to extract the subroutine link from the previous level which is loaded into the S register. P+2 is then decremented again by 1 to be a pointer to the naming location at the previous subroutine level.

$$S := (P+2)' - 1$$

$$P2 := (P+2) - 2$$

Bits 9-11 of the control word specify a scaling factor which is applied to the deflection system. The outputs of the X and Y registers are shifted down by up to four places thus causing a contraction of the picture as follows:

BITS 11, 10, 9	000	no change
	001	normal
	010	$\div 2$
	011	$\div 4$
	100	$\div 8$

In the  $\div 8$  mode, the whole of the picture area is visible on the screen.

### Function 12 – DISPLAY CHARACTERS

This displays two alphanumeric characters at the current position and in each case increments the X register an amount appropriate to the character size. The first character is specified by  $Q_{12-7}$  and the second character by  $Q_{6-1}$  the character code being given below.

#### Codes

The display system uses the standard 900 internal code. The sixty-four symbols of the in-shift are standard.

The complete list of symbols and their values generated by the character generator is given below. Characters left blank are undefined:

FUNCTION	BITS	
Jump	18	17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Subroutine Enter	0 0 0 0	Address
Load H	0 0 0 1	Address
Load V and Draw Line	0 0 1 0	X or $\Delta X$
Load H and Draw Vector	0 0 1 1	Y
Load V and Draw Vector	0 1 0 0	$\Delta X$
Spot X Relative	0 1 0 1	$\Delta Y$
Spot Y Relative	0 1 1 0	$\Delta X$
Spot X Absolute	0 1 1 1	$\Delta Y$
Spot Y Absolute	1 0 0 0	X
Load Parameters	1 0 0 1	Y
Control	1 0 1 0	Parameters
Display Characters	1 0 1 1	Function
Register Move	1 1 0 0	Character 1
Arc	1 1 0 1	Character 2
Store H	1 1 1 0	Function
	1 1 1 1	Parameters
	1 1 1 1	Address

928 Graphical Display Instruction Code

IN-SHIFT				OUT-SHIFT			
Value	Symbol	Value	Symbol	Value	Symbol	Value	Symbol
0	Space	32	@	0	(Escape)	32	
1		33	A	1		33	a
2	Newline	34	B	2		34	b
3	$\frac{1}{2}$	35	C	3		35	c
4	\$	36	D	4		36	d
5	%	37	E	5		37	e
6	&	38	F	6		38	f
7	(acute)	39	G	7		39	g
8	(	40	H	8		40	h
9	)	41	I	9		41	i
10	*	42	J	10		42	j
11	$\div$	43	K	11		43	k
12	,	44	L	12		44	l
13	-	45	M	13		45	m
14	.	46	N	14		46	n
15	/	47	O	15		47	o
16	0	48	P	16		48	p
17	1	49	Q	17		49	q
18	2	50	R	18		50	r
19	3	51	S	19		51	s
20	4	52	T	20		52	t
21	5	53	U	21		53	u
22	6	54	V	22		54	v
23	7	55	W	23		55	w
24	8	56	X	24		56	x
25	9	57	Y	25		57	y
26	:	58	Z	26		58	z
27	;	59	[	27		59	"
28	<	60	£	28		60	†
29	=	61	]	29		61	
30	>	62	NULL	30		62	NULL
31	10	63	NULL	31		63	NULL



Q13 specifies the shift status of the two characters in the operand.

Q13=0 INSHIFT

Q13=1 OUTSHIFT

There are also four NULL CHARACTERS, i.e the X register will not be incremented and the character will not be drawn.

These are :

111110

111111 in both

#### **INSHIFT AND OUTSHIFT**

Character grid :  $7 \times 4$  short strokes.

Character size : Two sizes controlled by programme namely :

Small : 128 char/line 64 lines.

Medium : 64 char/line 32 lines.

Character repertoire : 88 visible characters plus 'space' and new line.

#### **Function 13 – REGISTER MOVE**

This is a simple register manipulation instruction which enables simple arithmetic to be performed by the display. The various bits in the operand control gating of the registers to the 13-bit Adder/Subtractor via the two inputs A and B to the adder, and also controls the setting of the result. Reference can be made to store locations whose addresses are in H or S and this is treated as a register of the system. If this location is indicated then H or S is afterwards incremented by 1. A particular use of this is to store all the registers of the display into a block of consecutive locations pointed to by H, this occupies one instruction per register stored.

*Bits 16, 15, 14, 3, 2, 1 A Input to Adder.*

The binary value of bits 3, 2, 1 determine the register to be gated to the A input

Value 0 zero

1 +1

2 X

3 Y

4 P

5 S

6 H'

7 S'

*Bit 14=1 Sense Keys* (there are eight of these on the console and they occupy bits 8–1 of the input).

*Bit 15=1 Keyboard* (the character from the keyboard occupies bits 7–1 of the input).

If a character has not been entered, then bit 13=1.

*Bit 16=1 Scaling Factor* (the Scaling Factor occupies bits 9–11 of the input).

If a character has not been entered, then bit 13=1.

*Note:* Sense key, keyboard and scaling factor operations can only be done by a direct register move instruction.

*Bits 4, 5 B Input to Adder*

Value 0 zero

1 +1

2 H

*Bit 6 SUBTRACT B*

If bit 6 is 1 then the B input is subtracted from the A input.

*Bits 9, 8, 7 SET REGISTER*

These bits define the register into which the result is to be set.

Value 0 zero

1 X

2 Y

3 P

4 S

5 H

6 H'

*Bits 12, 11, 10 CONDITIONS*

These bits control whether the B input to the Adder (as specified by bits 4 and 3) is actually gated to the Adder. This only occurs if the condition is true.

Value 0 unconditional

1 light pen seen (this clears the marker)

2 light pen footswitch depressed

3 light pen timer hold (this clears the marker)

4 Processor demanded Hold (this clears the marker)

5 negative marker already set

6 visible condition (i.e X and Y  $\leq 1023$ )

Note the negative marker is set on the result of every reg. move.

*Bit 13 Shift*

If this bit is a 1 then the result of the register move is shifted one place right arithmetically, this is used for programming pen tracking.

#### **Function 14 – ARC**

This instruction uses a digital interpolation method to draw any arc of circle or complete circle. It requires the horizontal position of the starting point relative to the centre to be loaded into H first. The operand in the arc instruction specifies the vertical starting position relative to the centre and the following two words in store specify an increment count and an accumulator starting value.

*H := V := 0*

The extraction of these following two words is considered to be part of the arc instruction therefore the display cannot have a hold during the extraction of these words.

#### **Function 15 – STORE H**

This stores the value of the H register into a location specified by the operand Q, this is useful for preserving the H register particularly in subroutines and for Item Identification 'Naming'. If instruction is literal then the current store module will be addressed.

*Q' := H*

#### **HOLDS**

It is occasionally necessary to break the normal sequence of obeying display instructions in order to do some other task such as light pen tracking or interrupting the central processor when a visible output has

been 'seen' by the light pen. This break in sequence is called a 'Hold' and is accomplished by a method similar to that used in most central processors for interrupts.

When a reason for a 'hold' occurs then the current contents of the S register are stored in the location whose address is held in the System control location P+2 in the same manner as a subroutine entry. The contents of location P+3 are then read and used as a pointer to access the address of the 'hold' routine which is then loaded into S.

The hold routine can determine the reason for the hold by means of register moves that provide additional skips or jumps and then take specific action by storing certain registers as necessary and possibly interrupting the central processor and waiting for a processor 'continue' instruction.

When a 'hold' occurs, further holds are inhibited until an 'exit' instruction is given.

When a register move instruction refers to a condition that caused a hold then that hold marker is cleared. The reasons for holds are mentioned below.

#### Light Pen See

If the light pen has been enabled and the footswitch depressed, then a hold will occur whenever the light pen detects a visible effect from the current instruction, and the light pen see marker is set true. If the instruction being obeyed is a vector then the X and Y values at the first instant of the see are stored in fixed locations, this occurs even if holds are inhibited, the marker is also set if holds are inhibited although it is cleared by a step on timer (end of frame) parameter instruction.

#### Light Pen Timer

If the output control instruction has enabled the light pen timer then every 20 milliseconds a hold will occur and the light pen timer marker is set. This will be used to instigate a light pen tracking programme without using the central processor.

#### Processor Hold

If the processor issues a peripheral instruction demanding a hold (see SYSTEM CONTROL section) then the marker is set and a hold occurs.

## SYSTEM CONTROL

### Fixed Locations

Four fixed locations are used and these are normally locations 60-63 inclusive within the first store module. However, the exact addresses used may be changed by linking certain pins on edge connectors to OV. The first of these locations is referred to as P and must have an address which is a multiple of 4.

Location P      This is used to store the X register value when a vector is seen by the light pen.

Location P+1      This is used to store the Y register value when a vector is seen by the light pen.

Location P+2      This is used as an address pointer to store the link on holds and subroutine and to store the starting address of the display file to start the display operating initially.

*Note:* The contents of P+2 must be a multiple of 64 at subroutine level 0.

Location P+3      This is used to hold the starting address of the hold routine.

### Control Instruction

Control of the display is exercised by a direct output instruction described below.

#### Control Characters

This uses the 900 series 15 4992 output instruction (variable by back wiring links). The bits of the accumulator are interpreted as below.

Bits 2, 1 are 0, 0 no change  
Bits 2, 1 are 0, 1 interrupts inhibited  
Bits 2, 1 are 1, 0 interrupts permitted  
Bits 2, 1 are 1, 1 undefined

#### Bit 3 Stop

If bit 3 is a 1 then the display stops at the end of the current instruction, until a continue or start instruction is given. The display is set to this state at switch on.

#### Bit 4 Continue

If the display has been stopped either by bit 3 above or by a control instruction (bit 4 see page 3), then it can be restarted if this bit is a 1. This is only defined if the display is 'stopped'.

#### Bit 5 Start

If the display is 'stopped' and this bit is a 1 then the S-register will be loaded from location P+2.

#### Bit 6 Processor Hold

If a peripheral control instruction is issued with bit 6 present, then a 'processor hold' will be instigated.

#### Bits 8, 7, Pen Timer Control

The pen timer may be set on or off by outputting a peripheral control word from the computer to the display.

Bits 8, 7=0, 0 no change  
                  =0, 1 turn pen timer off  
                  =1, 0 turn pen timer on

When the pen timer is set 'on' the pen timer marker is set every 20 milliseconds and subsequently a pen timer hold will occur.

#### Input Status

This uses the 900 series 15 896 input instruction.

Bit 1 of the status word is set to 1 when a display interrupt occurs. The input status instruction resets this bit and the interrupt.



## PERFORMANCE AND OPERATING CHARACTERISTICS

The times in microseconds taken by functions are given below.

Spot moves	30 min, 150 max
Vectors time is	0.75 max (h.v) if max (h.v) > 15 or 2 max (h.v) if max (h.v) ≤ 15

Characters take between 8 and 35 microseconds, depending on the character and size of the character. This includes positioning the beam for the next character. Arc time is 2 microseconds for each incremented step.

The tolerance on the above times is ±10%.

## CONTROLS AND LAMPS

### Operator's Controls

These are mounted on the viewing unit.

*Focus* – rotary control to adjust focus of the displayed picture.

*Brightness* – rotary control to adjust the brightness of the displayed picture.

*Emergency Off* – red push button fitted with a guard ring. When depressed switches off the viewing unit.

*Mains On* – lamp coloured red indicates a.c mains is applied to viewing unit.

*d.c On* – lamp coloured green indicates that d.c power supplies have been correctly established in viewing unit.

*Sense keys* – Eight 3-position lever keys locking up, non-locking down. The position of these keys can be determined by a display function 13 (Register Move) see Function 13 page 5. These keys are normally associated with the light pen software routines.

*Pen Enable* – this is a foot-operated switch which when depressed enables the light pen.

### Engineer's controls in Viewing Unit

*OFF/LOCAL ON/REMOTE* – three-position lever key controlling the d.c power supplies within the viewing unit. In LOCAL ON all supplies are up sequenced regardless of any external signals. In REMOTE the power supplies will up and down sequence in response to an external signal from the display controller.

*RESET* – two-position non-locking lever key – depressing this key causes d.c power to be re-established after Emergency Off has been depressed.

### Engineer's controls in Display Controller

*Mains On* – red lamp indicates that a.c mains is applied to the display controller.

*MAINS ON/OFF* – two-position lever key.

*LOCAL/REMOTE* – two-position lever key.

*PARTY RESET* – two-position non-locking lever key which when depressed clears the parity error bistable.

*SLOW STORE* – two-position lever key

when depressed inserts a delay between successive store demands. This gives a steadier oscilloscope triggering source when 928 is running time shared with 905 C.P.U.

*NORMAL/INST. STEP/ORD. STEP* – three-position lever key. When in INST. STEP the display microprogram can be stepped on one instruction at a time whilst ORD. STEP allows the microprogram to be advanced in small discrete steps.

*STEP* – two-position non-locking lever key which when depressed causes the display microprogram to advance one step. The size of the step is controlled by the switch described in the previous paragraph.

*Engineer's Display*. This is an assembly of 200 neon lamps which enables the logic state of all bits in each of the ten registers and of the main timing and control signals to be monitored.

The mechanical construction allows for the easy removal of power supply units. Provision has also been made to hinge down connector panels at the rear of the logic racks, to increase accessibility.

## PHYSICAL DETAILS

### Basic System

The basic 928 graphical display is comprised of two units, the display controller and the graphical console. The display controller houses all interface logic for communicating with the 905 data processing system as well as the display control logic and vector/arc generator. The graphical console houses the cathode ray tube, the character generator, light pen, keyboard and operator's controls.

### Keyboard

There are sixty alphanumeric keys, two shift keys plus a separate block of twelve function keys. A mechanical interlock prevents the simultaneous depression of more than one key. The maximum possible typing speed is fifty characters per second.

### Light Pen

This is a small hand-held device connected to the graphical console by a flexible light guide. The light pen detects the short persistence blue phosphor and when the foot switch is activated the portion of the display seen by the light pen is detected at the instant it is drawn. The sensitivity of the pen is adjustable to exclude activation by external light.

### Optional Units

The number of graphical consoles on the display controller may be increased to four total. Each of these three additional graphical consoles is complete with character generator, light pen, keyboard and operator's controls.

The 928 display controller may be equipped to regenerate the 900 series Peripheral Interface. In this case one standard peripheral connector is available for a 900 series peripheral (i.e. a digital plotter). However, no

facilities are provided within the 928 for specifically identifying any interrupts generated by such a peripheral (the interrupts are OR-ed together on each level). Software must identify the relevant device by inputting status words.

### Interfaces

The graphical display is connected to the 905 store highway as an autonomous device. It is thus able to share access to the main core store of the computer with other devices on a cycle stealing basis. The display controller must be so arranged that the maximum length of the store bus cables is not exceeded.

The display controller is also connected to the 905 peripheral interface for control purposes (maximum length 20 feet (6 m)).

There is a further interface between the display controller and the viewing units. This is arranged on the bus principal. The cable being looped from one viewing unit to the next.

Normally each viewing unit is supplied with a 25-foot (7.5 m) cable, though lengths of 16 (5 m) or 30 feet (9 m) are available. The total length of such cables is restricted to 100 feet (30 m) per controller. The keyboard is normally housed on the graphical console desk and is connected to the display controller by a 33 foot (10 m) cable.

### Power supplies

#### Mains supply

The 928 graphical display system requires an a.c mains supply as follows:—

Voltage: 100 to 125 in 5's,  
200 to 250 in 10V steps  $\pm 10\%$ .  
Frequency: 50Hz  $\pm 4\%$ .

#### Power consumption

Viewing Unit: 500W.

Display controller: 200W.

#### Protection

The incoming mains is fused at 5A (slow blow). The d.c power supplies are individually protected and current limited. An SCR over voltage protection system is incorporated.

#### Interference

The incoming mains supply is fitted with a filter for interference suppression (filter to PS.444).

#### Failure

The display system will tolerate interruptions in the mains supply not exceeding 10 ms in duration. For longer interruptions the system will down sequence in such a way as to retain store information.

#### On sequence

The display controller and graphical console

may be remotely switched on and off with the CPU. The Power On signal in the peripheral interface is used for this purpose.

The display console has facilities for distributing an a.c mains supply to one graphical console. Additional graphical consoles must be connected to a separate mains supply but all are controlled by the remote switch-on signal.

### Installation

#### Dimensions and weights

Viewing unit 49·0in. (124 cm) H  $\times$  28·5in. (72·5 cm) W  $\times$  41·0in. (104 cm) D.

Weight: 400lb. (180 kg)

Controller 56in. (142 cm) H  $\times$  2·10in. (5·2 cm) W  $\times$  22·75in. (58 cm) D.

to match the standard extension cabinet.

Alternatively the controller can be supplied in a double pedestal desk along with a 916 single modem controller or else a Data Interchange Unit. The dimensions are then 36in. (91 cm) H  $\times$  42in. (107 cm) W  $\times$  26in. (66 cm) D.

Light pen length=5in (12.7 cm), diameter  $\frac{9}{16}$ in.(1·4 cm).  
length of light guide=37in.  
(94 cm).  
min. bending radius of guide=  
1in. (2·5 cm).  
weight of pen and guide=6·5oz.  
(184 g).

Keyboard 17in. (43 cm). L  $\times$  9 $\frac{1}{2}$  + 6in.  
(24 + 16 cm) W  $\times$  1 $\frac{1}{2}$ in. (4 cm).  
H (front) and 2 $\frac{1}{2}$ in. (7 cm) H  
(back).

Weight 12lb. (5·4 kg).

Key operating force 250g.

Key stroke  $1\frac{3}{16}$ in.  
(3 cm).

Key centre spacing  $\frac{3}{4}$ in.  
(1·9 cm).

#### Heat dissipation

Cooling clearance. A gap of at least two feet must be left between the rear of each unit and any vertical surface to allow egress of air from the rear mounted cooling fan.

#### Environment

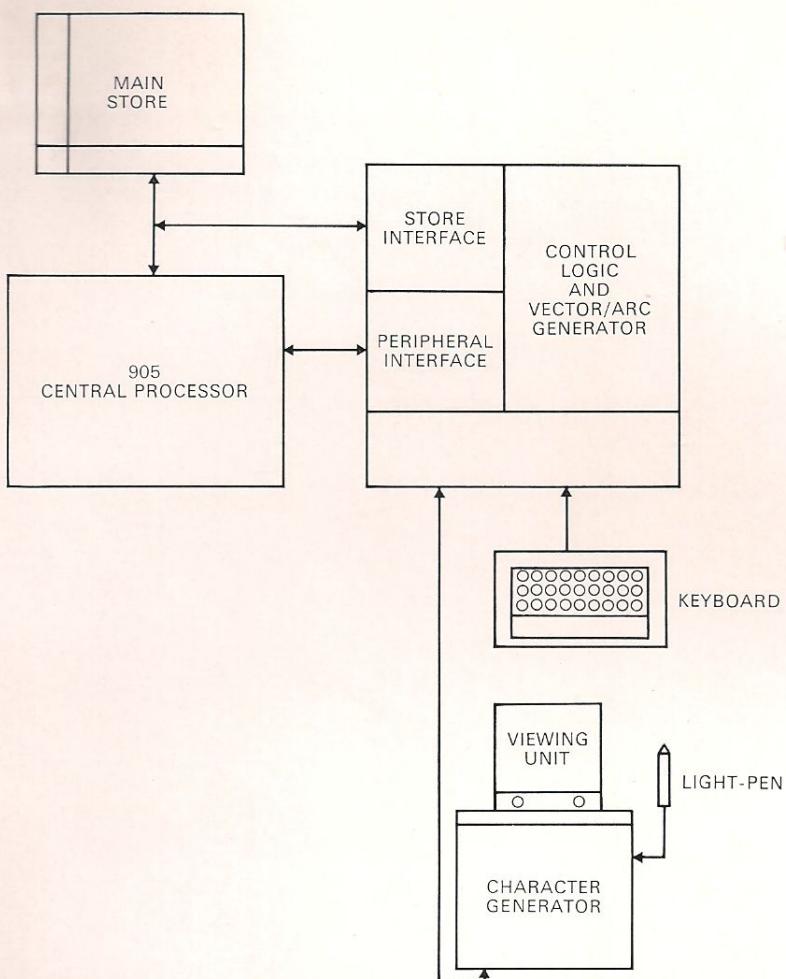
The graphical display will operate in the temperature range 5°C–35°C. The humidity limits are 20–90% without condensation.

#### Storage

The display may be stored without detriment at temperatures between –10°C and 60°C and humidity up to 95% R.H.

#### Air Filtration

A nylon washable air filter is fitted horizontally beneath the mains entry panel in the units. It is exchangeable from the rear of the unit.



928 Display System block diagram

**Marconi-Elliott Computer Systems Limited**

A GEC-English Electric Company

Elstree Way, Borehamwood, Hertfordshire  
Telegrams and Cables : Elliotauto Borehamwood  
Telex : 22777 Answerback : Ellauto Borhwd  
Telephone : 01-953 2030

**928 C TL/1**

# 928 Software

In the design of the flexible order code of the Marconi-Elliott 928 Graphical Display, programming requirements have received as much consideration as the graphics. Although it is anticipated that most users will use one of the programming systems available, the display is nevertheless easily programmed directly by the user. One of the main features of the available programs is that the user does not need to have a detailed knowledge of the 928 hardware; a broad outline of its operation and features is sufficient to enable full use to be made of the system. The communication with, and the control of, the 928 is handled entirely by these graphic programs. The nucleus of all these program systems is the Marconi-Elliott 905 program DISMAN 900, the Display Manipulation System. This is supplemented by ALGOL and FORTRAN routines.

## **Disman 900**

DISMAN 900 is a comprehensive device routine package which acts as an interface between the users' SIR (Symbolic Input Routine) program and the 928 graphical display. It enables the programmer to utilize fully the capabilities of the display without having a detailed knowledge of the hardware or the complex 'housekeeping' software. DISMAN is written in Symbolic Input Code and as such can be used directly by another SIR program. The advantages of programming in this language are mainly in respect of efficiency in speed and use of core store. It is ideal for many real-time applications where speed is important. DISMAN consists of a modular group of routines: its main functions are the control of the 928 hardware and the maintenance of the display file.

The picture is built up as a series of display commands held in the main store. These commands, known collectively as the 'display file', are accessed autonomously by the display controller, which in turn sends signals to the viewing unit to draw a picture on the face of the cathode ray tube.

The picture can be altered by modifying the display file as necessary, and the user is freed from any of the routine problems of storage allocation associated with a dynamic display file.

The facilities provided by DISMAN form three main groups:

- (1) Basic housekeeping routines to start the display and allocate areas in which to construct the display file.

- (2) Code generation routines to generate display commands for items such as points, lines, arcs and text.
- (3) An interrupt routine to react to the interrupts produced by the display and activate the appropriate users' routines.

DISMAN contains a number of display code generation routines which can be used to generate code for an 'item' in a buffer. The item can consist of any mixture of points, lines and alphanumeric characters as well as control functions. This item is then inserted into the display file where required and at the same time it is given an identifying name by the user. This name can be used later to reference the item if, for instance, it is required to delete it, or the name may be handed to the users' program as a reference if this item is seen by the light pen. Using DISMAN the user is able to insert items into, or delete items from, the display file, and items, once in the display file, can be replaced, renamed or modified at will by the user. Facilities exist for defining a collection of items as a subpicture, which can then be named and called up using that name within a subsequent item or items. This subpicture facility can be nested to any depth. Beside the above-mentioned routines there are control routines mainly associated with the input hardware like the light pen and the function keys. These routines are set up by the user to cause entry to his own routines when an input occurs, making available all the information associated with it. Pen tracking can be initiated by the user at any time. The pen will be tracked automatically and continuously by a small cross on the screen, without further action on the part of the users' program. By this method the co-ordinates of the pen position are always available. Communication between the users' program and DISMAN is basically via a number of global identifiers and labels.

Objects on the screen may be at one of two brightness levels, thus certain features of the picture may be visually emphasized relative to the rest of the picture. Characters may be displayed in two sizes - small or normal. Lines may be plain or dotted, either at 16 or 32 increments frequency. Objects on the screen may be made to blink on and off to attract attention to any given part of the picture, and may be made visible or invisible to the light pen.

## **Edgar 900 and Fred 900**

All the facilities of DISMAN are available to

the ALGOL user via a procedure package known as EDGAR 900 and to the FORTRAN user via a subroutine package known as FRED 900. These packages of procedures enable the ALGOL or FORTRAN programmer to use the 928 Graphical System for input and output of graphical information, including conversational operations using the light pen and console keys, and autonomous pen tracking. Procedures are provided for creating and modifying a file of display commands in the store, detecting when a picture part is indicated with the pen, interrogating the console keys in various modes, and tracking the pen.

Examples of the instructions for EDGAR 900 are

NEWBUF (A) : Start a new buffer: array A.  
 POINT (X, Y) : Generate display code for point (X, Y).  
 VECTOR (DX, DY, B) : Generate a vector with DX and DY as relative components.  
 ARC (X1, Y1, X2, Y2, WAY) : Generate an arc from X1, Y1, to X2, Y2.  
 DELETE (N) : Delete item N from display file.  
 SUBPIC (N) : Enter Subroutine N.  
 PENTRAK (X, Y) : Sets X and Y to the current position of the tracking cross.  
 Similar instructions are available to the FORTRAN user via FRED 900.

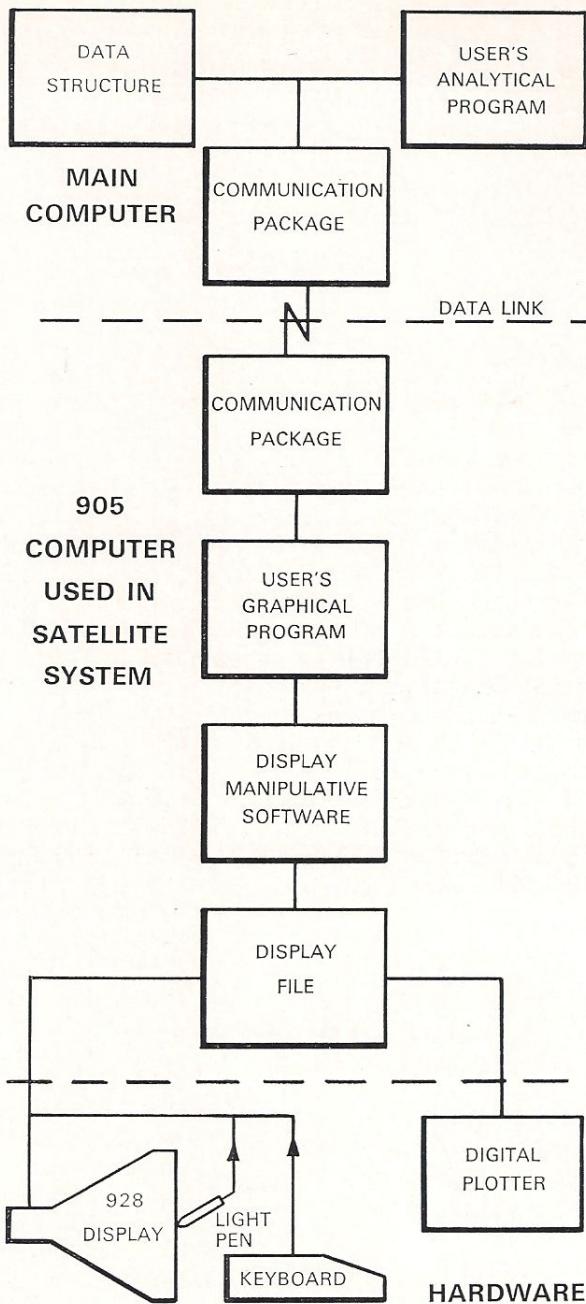
The provision of graphical features in ALGOL and FORTRAN gives the user a very powerful high level programming system which has the advantages of ease of writing and ease of understanding by other users and establishments. Using EDGAR 900 and FRED 900 the Graphical Display can be programmed with almost the same ease as any other input or output peripheral, while maintaining the unique interactive capabilities that make the display such a flexible and powerful device.

### The Software System

The diagram shows the normal structure of the software system for the 928 display when it is used in a 'satellite' configuration.

When the 928 display is used as a satellite to a central computer the data structure and the analytical routines for operating on the data structure are usually held in the central computer. The display file together with the Display Manipulation System, DISMAN 900 and the users' graphical routines reside in the core store of the Marconi-Elliott 905 computer. It may also be necessary to hold certain parts of the data structure (for example those parts which are being viewed) in the satellite computer. For certain applications a 928 configuration with at least 16 384 words of store can be used on its own when the main computer is temporarily not available. Special data structure program aids are available which provide for the automatic linking of items and correct storage allocation.

DISMAN 900, the Display Manipulation package for use with programs written in SIR (Symbolic Input Routines) code, can be used on a 928 system with 8192 words of core store. A system with 16 384 or more words of



store allows use to be made of the powerful ALGOL and FORTRAN programming systems.

Other graphics programs include diagnostic routines, and programs to produce hard copy of the displayed picture on an on-line digital plotter.

**Marconi-Elliott Computer Systems Limited**

A GEC-English Electric Company

Elstree Way, Borehamwood, Hertfordshire  
Telegrams and Cables: Elliauto Borehamwood  
Telex: 22777 Answerback: Ellauto Borhwd  
Telephone: 01-953 2030

**928 S TL/1**

**Marconi-Elliott Computer Systems Limited**

A GEC-English Electric Company

Elstree Way, Borehamwood, Hertfordshire  
Telegrams and Cables: Elliotauto Borehamwood  
Telex: 22777 Answerback: Ellauto Borhwd  
Telephone: 01-953 2030