Summary of Minutes of TMDS Meeting
(Television Monitor Display System)

DATE: October 20, 1969

TIME 10:30 - 11:30 a.m.

PLACE: Rm. 1206, Bldg. 113

SPEAKERS: John Wade x8583, Bldg. 113, Rm 3076
Gale Marshall x8582, Bldg. 113, Rm. 2111
Pete Keller x8581, Bldg. 113, Rm. 1171

PURPOSE

(1) To present a status report of TMDS for general information.

(2) To inform the users and programmers of the capabilities and limitations of the TMDS system so they will know what to expect and can plan properly.

(3) To ask for suggestions and proposals for usage so that adequate software can be provided. (Without your comments you cannot expect timely or adequate software.)

WHAT IS TMDS?

Physically, the TMDS console is a television set with a 12-inch square screen. The console is an output device capable of displaying computer-generated information. A frame of information will continue to be displayed until the computer-generated command ERASE frame is sent to the TMDS. There are 16 consoles whose screens are refreshed 30 times/sec by the TMDS disk. This disk stores only one frame of information for each console, a maximum of 16 frames. TMDS can display up to 85 characters per line and up to 51 lines per frame.

TMDS also has a graphic capability; the frame consists of 512 points by 512 points—any one of these points may be illuminated by computer instruction. Thus, any graph which can be described by an array of 512 × 512 points can be displayed.

It is expected that TMDS will be used in conjunction with a teletype. There will be 16 consoles placed throughout the Lab (the consoles are being allotted by John Burk, ext. 8581). Only one user at a time can reserve a single console for his use; however, this user may link other consoles to his program, thus displaying the same information on several consoles at the same time. At most, 16 users can be using the TMDS system at one time.

The TMDS is an experimental system. This paper describes the current state of TMDS which is subject to change.

The TMDS was designed to handle alphameric displays. Although TMDS will display graphics and can be made interactive through the use of the teletype, it is expected to be slow in response and frustrating to the graphic or interactive user.

WHERE DOES TMDS FIT INTO THE OCTOPUS SYSTEM?

The TMDS is attached directly to the PDP-6 (Fig. 1), and will be available first through the CDC 7600. Since the PDP-6 has neither the time nor the core available to generate TMDS data, the CDC 7600 must generate the TMDS data. This data is then

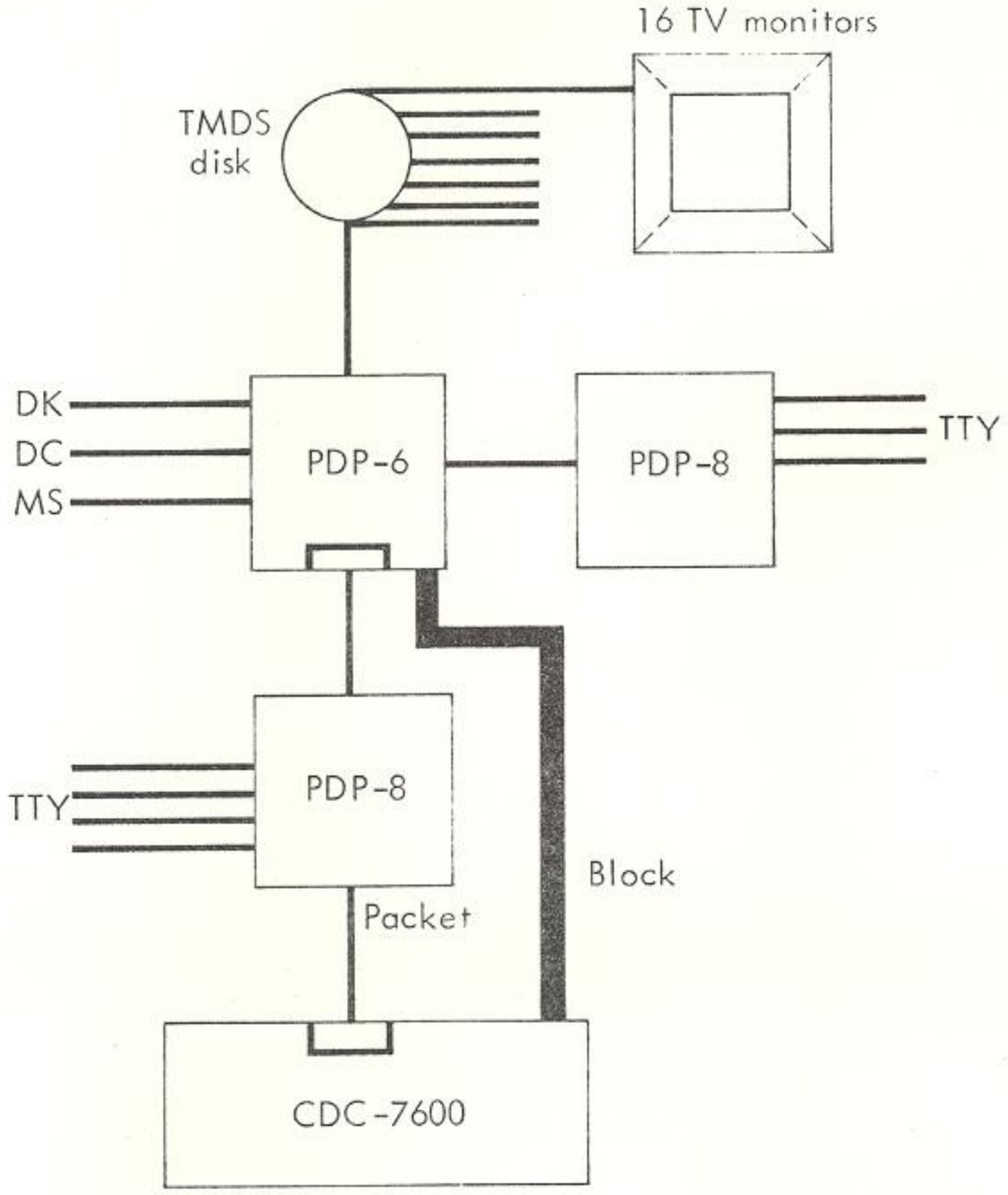


Fig. 1. Block diagram of TMDS and Octopus.

sent to the TMDS via the packet (uses the teletype message line) or via the TMDS block (uses the file transport line). A packet consists of no more than 15 60-bit words.* A block consists of no more than 612 60-bit words (block size limited by PDP-6 core available).

The file transport line is a physical cable used by the file transport system to send large blocks of data (no more than 275,251 60-bit words and not to be confused with a TMDS block). The file transport system is used to send the retrieve computer information from the data cell and mass store.

The TMDS will become available on the CDC 6600 when the new file transport system (shortly to become available on the 7600) is thoroughly checked out and made available on the 6600. (There may be a delay in bringing the new file transport system to the 6600 since this involves an extensive hardware modification.)

WHEN WILL TMDS BECOME AVAILABLE?

Four requirements must be met before the TMDS can accept user-generated TMDS data sent by the 7600.

(1) TMDS must pass an acceptance test. This test is a 1-month trial period during which a test code (program) will exercise the TMDS. This test will insure that all contractural specifications are met or arbitrated. This test is expected to begin in mid-December.

(2) The TMDS/PDP-6 memory interface must be completed. This is a hardware device which makes the TMDS compatible with the PDP-6. The estimated completion data for this hardware is February 1970.

(3) The new file transport system must be operational. This should be completed shortly.

(4) The PDP-6/7600 software link must be operational. The completion date for this is February 1970.

WHAT CAN TMDS DO?

If programs are written, TMDS will be able to display the following ouputs:

- Octal, decimal, alphameric dumps.
- Source listings.
- HSP output files (85 characters/line).

WHAT MIGHT THE TMDS DO?

If programs are written, TMDS might be able to handle graphic interaction (drawing a graph, manipulating graphic information) or alphameric interaction (report editing, program editing). By the word "might" we wish to suggest that TMDS will not be responsive to the extent expected or need for interactive work; and while these types of programs will execute and be interactive, they are expected to be slow and frustrating to use.

What are some of the things which slows the interactive usage? TMDS has no hardware vector generator; all vectors must consist of a string of points. One can

*Possibly 63 60-bit words.

think of a line on the TMDS console as consisting of 64 columns of 8 points each. Now, to illuminate one point, the user must specify the group of eight points. (See Appendix A.) A full frame of graphic information requires about 4600 60-bit words of storage and requires that 8 TMDS blocks be transmitted to the PDP-6. Interaction can be expected to be slower than the usual teletype/problem-program interaction or file transport because TMDS will be time-sharing the PDP-6, as well as using the teletype and file transport lines.

WHAT WON'T THE TMDS DO?

TMDS will not be able to display directly any information stored on the data cell or mass store. This data must first be brought to the 7600 via the file transport, converted to the TMDS format by the 7600 (if not already in TMDS format), and then sent to the TMDS. Why this indirect route—the PDP-6 has neither the core nor the time available to send directly to the TMDS.

TMDS will not be able to display character by character as typed because the PDP-8 is line oriented. The only way to have a single character displayed as typed is to type the character and then line feed (which creates an end-of-line sentinal). The problem program would then get this message, convert to TMDS format, and send to TMDS. This is not practical; it is slow and costly.

TMDS will not be able to graph a vector given the vector end points; the PDP-6 is unable to generate this sequence of points given the end points. (PDP-6 has neither time nor core available.)

STATE OF TMDS HARDWARE AND SOFTWARE

Appendix A describes the TMDS instruction format and hardware in detail. This was written for the programmer needing detailed specifications.

WHAT SOFTWARE IS CURRENTLY PLANNED?

A routine is being designed which will make the complete TMDS capability available to the 7600. This is simply a routine which communicates with the PDP-6 to establish the availability of core space, to send the TMDS-formatted data to the PDP-6, and lastly to determine whether the data was acceptable (or in error) to TMDS. This routine places the burden on the user or programmer to properly format the TMDS data.

USER INSTRUCTION FORMAT

What must the user specify to cause a display on a TMDS console? This is up to the user. Currently, we have one proposal on how TMDS should look to the user (see Appendix B).

REQUEST FOR PROPOSALS

What user-oriented software is written depends on your demands. Since TMDS appears to become available in February 1970, now is the time to begin designing the software. Submit your written software proposals for consideration to Ed Schoonover, L-61, Bldg. 113, Room 1152. To insure proper PDP-6/7600 communication, also submit a copy of your proposal to Gale Marshall, L-61, Bldg. 113, Room 2111 and to Pete Keller, L-61, Bldg. 113, Room 1171. Programming must begin now.

to have the software available in time. Your proposal must be received no later than December 15, 1969, to be considered.

Appendix A

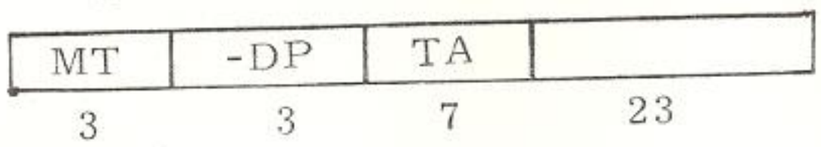
State of TMDS Hardware and Software

TMDS DATA FORMAT

In order to modify a display, the PDP-6 must send the TMDS a list of 36-bit word messages. The maximum data rate is 30 lists per second.

Each message in a list addresses one or more of the 512 horizontal scan lines of the 512 × 512 display raster. The two rules concerning a list are (1) no two messages may effect the same scan line, and (2) messages must be ordered in the same way the raster lines are scanned (i.e., top to bottom, line 0 to line 511).

The first word of a message is called the message header. Every header contains the following four pieces of information: MT, D, P, and TA.

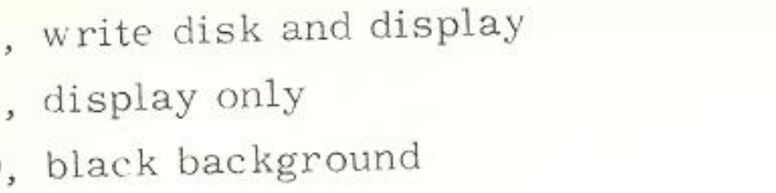


DEFINITIONS OF ABBREVIATIONS

Term	Meaning	Values
MT	Message Type	2, erase frame
		4, multi-graphic
		6, single-graphic
		5, multi-alphameric
		7, single-alphameric
D	Display Mode	0, write disk and display
		1, display only
P	Polarity	0, black background
		1, white background
TA	Terminal Address	0-15
LA	Line Address	0-511
CA	Column Address	0-63, low capacity (LC)—large characters and graphics
		0-84, high capacity (HC)—small characters
G	Graphic Mode	0, add
		1, replace
A	Alphameric Mode	0, LC
		1, HC
TWC	Text Word Count	1-255

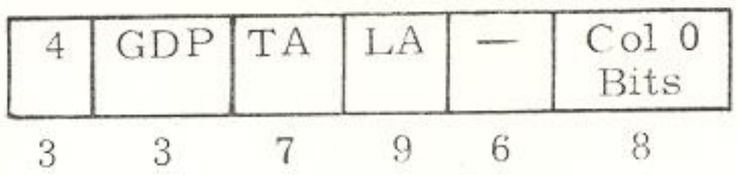
There are 5 kinds of messages, 3 consist of a header only, 1 consists of a header plus 14 words, and 1 consists of a header plus 1-255 words.

(1) Erase Frame:



Erases lines 0-511

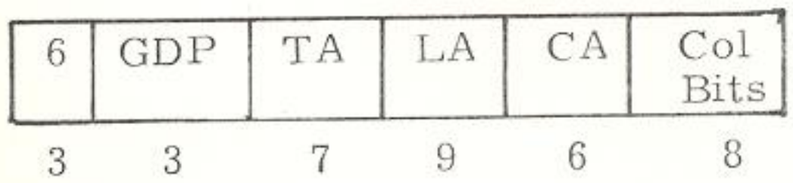
(2) Multi-graphic: word 0



words 1-14

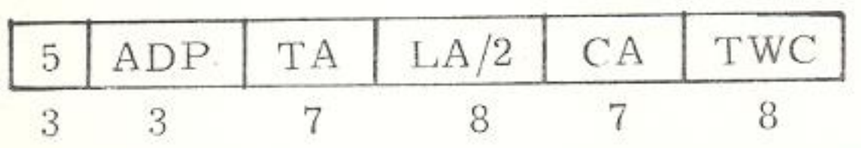
Writes all 512 points of one line.

(3) Single graphic:



Writes eight consecutive points of one line.

(4) Multi-alphameric: word 0

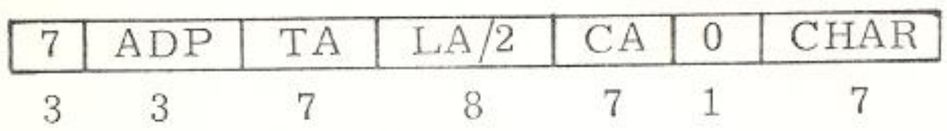


Words 1—TWC

Writes a string of characters.

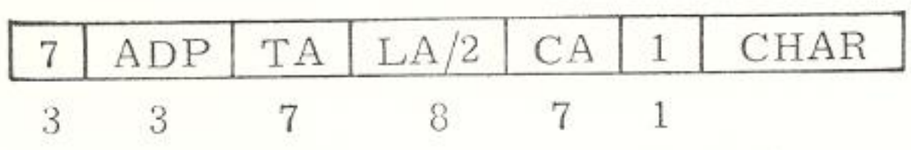
(5) Single alphameric:

(a) Character:



Writes one character.

(b) Cursor:



Writes the top line of one character and erases the rest of the line. (A cursor may be obtained by plotting the top line of an E, F, T, Z, 5, 7, or %.)

Graphic Messages

Graphic messages contain bits which correspond to points of the display raster. These bits may replace or be "added" to existing data.

In the replacement mode, the bits of the message replace the old data. In the additive mode, the bits of the message are inclusively or'ed with the old data.

Alphameric Messages

Alphameric messages contain 7-bit ASCII characters from either a 64-character set or a 96-character set, respectively referred to as high capacity (HC) and low capacity (LC). These characters are converted into point arrays by the TMDS hardware. Table 1 gives the specification of these characters. The scan lines required to display a character comprise the character row, and the message line address refers to the first, or top, such line. This line must be an even numbered line. If a message modifies more than one row, the hardware assumes a standard number of lines between rows.

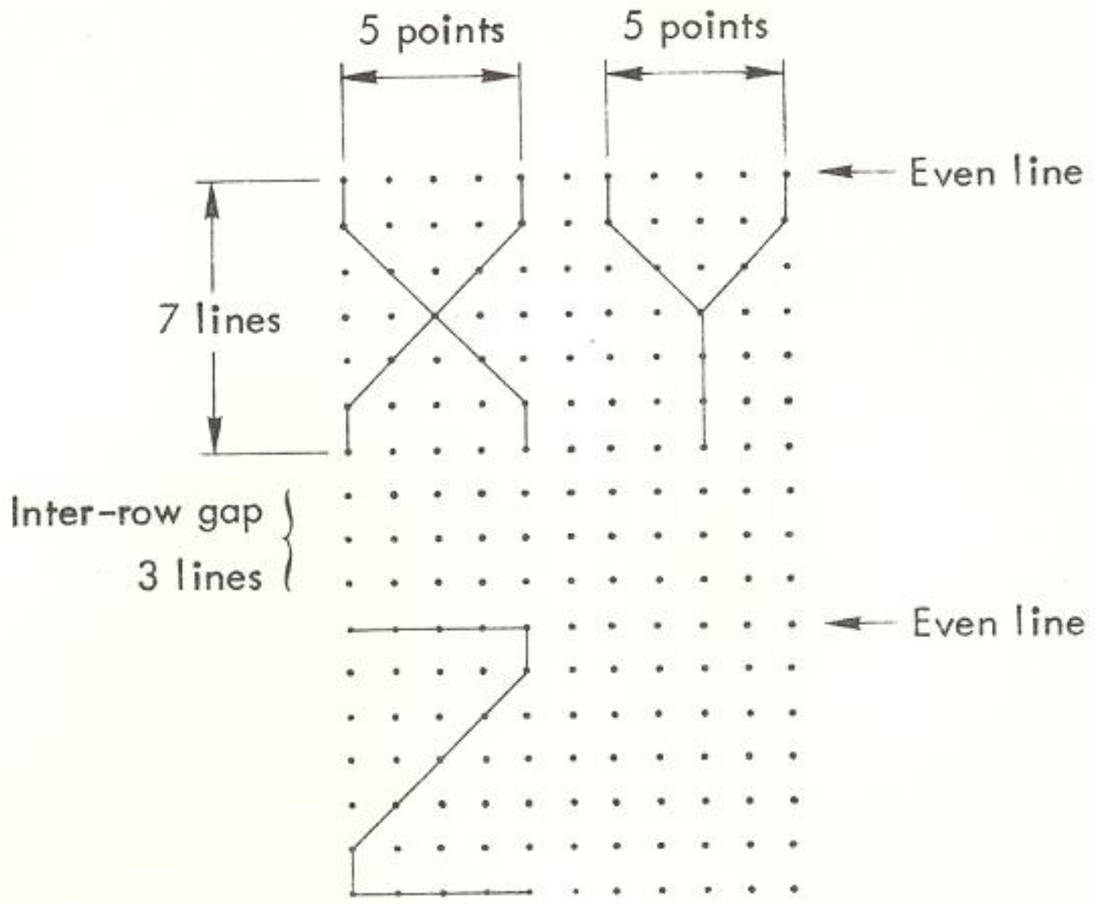
Example of Character Sizes

Figure 2 diagrammatically shows the letters both in high-capacity and low-capacity modes. Figures 3 and 4 show photographs of the TMDS screen. These

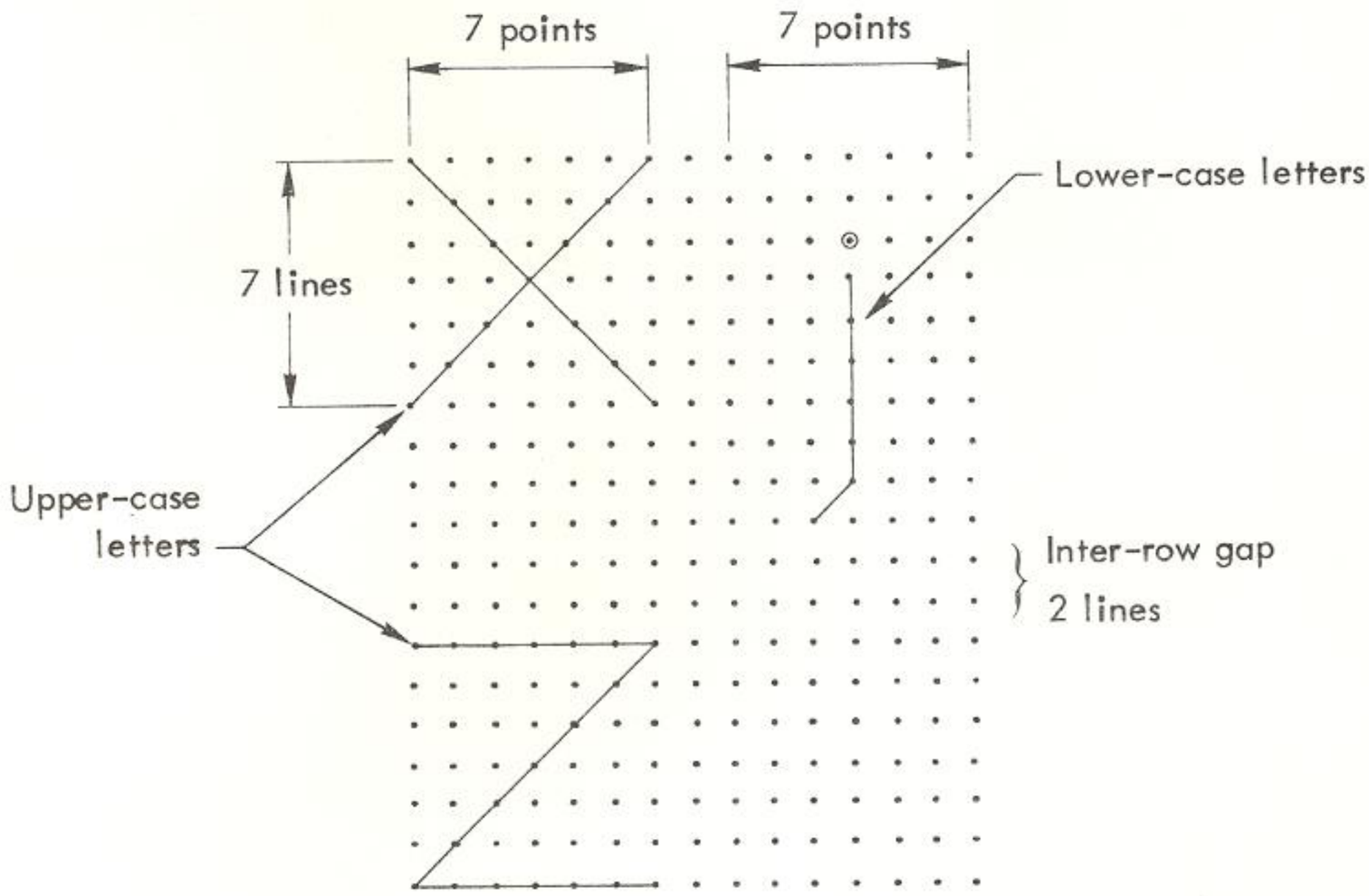
Table 1. Description of high- and low-capacity characters.

Item	HC	LC
No. of columns	85	64
Width of character	5 points	7 points
Inter-column gap	1 point	1 point
No. of rows	51	42
Height of character	7 lines	10 lines
Inter-row gap	3 lines	2 lines

photographs were taken before the character set hardware was debugged. Stray marks were caused by these hardware bugs.



(a) High capacity (small characters). All upper-case letters.



(b) Low capacity (large characters). Upper- and lower-case letters.

Fig. 2. Specifications of high-capacity and low-capacity letters.

000	040	100 @	140 @
001 !	041 !	101 A	141 A
002 "	042 "	102 B	142 B
003 #	043 #	103 C	143 C
004 $	044 $	104 D	144 D
005 %	045 %	105 E	145 E
006 &	046 &	106 F	146 F
007 '	047 '	107 G	147 G
010 (	050 (	110 H	150 H
011)	051)	111 I	151 I
012 LF	052 *	112 J	152 J
013 +	053 +	113 K	153 K
014 ,	054 ,	114 L	154 L
015 CR	055 -	115 M	155 M
016 .	056 .	116 N	156 N
017 /	057 /	117 O	157 O
020 0	060 0	120 P	160 P
021 1	061 1	121 Q	161 Q
022 2	062 2	122 R	162 R
023 3	063 3	123 S	163 S
024 4	064 4	124 T	164 T
025 5	065 5	125 U	165 U
026 6	066 6	126 V	166 V
027 7	067 7	127 W	167 W
030 8	070 8	130 X	170 X
031 9	071 9	131 Y	171 Y
032	072 :	132 Z	172 Z
033 ,	073 ;	133 [	173 [
034 <	074 <	134 \	174 \
035 =	075 =	135]	175]
036 >	076 >	136 ↑	176 ↑
037 ?	077 ?	137 ←	177 ←

Fig. 3. High capacity (small characters) with only upper case letters and numerals.

000	@	040		100	`	140	@
001	A!	041	!	101	A	141	a
002	B"	042	"	102	B	142	b
003	C#	043	#	103	C	143	c
004	D$	044	$	104	D	144	d
005	E%	045	%	105	E	145	e
006	F&	046	&	106	F	146	f
007	G'	047	'	107	G	147	g
010	H(	050	(	110	H	150	h
011	I)	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	K+	053	+	113	K	153	k
014	L,	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	N.	056	.	116	N	156	n
017	O/	057	/	117	O	157	o
020	P0	060	0	120	P	160	p
021	Q1	061	1	121	Q	161	q
022	R2	062	2	122	R	162	r
023	S3	063	3	123	S	163	s
024	T4	064	4	124	T	164	t
025	U5	065	5	125	U	165	u
026	V6	066	6	126	V	166	v
027	W7	067	7	127	W	167	w
030	X8	070	8	130	X	170	x
031	Y9	071	9	131	Y	171	y
032	Z:	072	:	132	Z	172	z
033	[;	073	;	133	[	173	{
034	\<	074	<	134	\	174	|
035	]=	075	=	135	]	175	}
036	↑>	076	>	136	↑	176	~
037	←?	077	?	137	←	177	→

Fig. 4. Low capacity (large characters) with upper and lower case letters and numerals.

Appendix B

Proposal for the Implementation of TMDS

The following proposals are made regarding the forthcoming implementation of TMDS.

(1) The initiation and use of TMDS should be as simple as that of any other output device, i.e., for alphamerics, the user should be required to specify only:

- a BCD list,
- a unit number, and
- a format.*

(2) Provisions should be made for an additional new hardware-type specification for TMDS which would allow the user to select TMDS in addition to the existing output hardware devices, during problem execution, by simply changing the unit number. This can now be done for the TTY, HSP, on-line-printer, and dd80.

(3) Provisions should also be made such that the user may either specify those instructions commonly employed for all output devices, or he could use a smaller subset of instructions required for the inclusion of a select few output devices. The latter would satisfy those users who wish to run their program at minimum size, whereas the former would satisfy users who wish to employ any combination of output devices.

These proposals are meant to continue the now existing trend of I/O usage at LRL as far as alphamerics are concerned. A similar approach could also be worked out in the future for graphics. For example, the user's dd80 instructions could be rerouted to any other display device, i.e., the TMDS.

---

*A buffer-out statement requires only a BCD list and a unit-number.