

## List of Appendices

Appendix 1	Basic Input/Output Coding Chart
2	Coding Chart for IBM
3	B.C.L. Serial Printer Code Set
4	Line Printer Code Set
5	Tally Roll Printer Code Set
6	Visual Display Terminal Code Set
7	Alpha/Numeric Keyboard Code Set
8	Code List for all IBM Characters
9	Bit No. and Decimals Values Chart
10:1,10:2	2 - Character storage
11:1,11:2,11:3	Peripheral Character Sets
12:1,12:2,12:3	Decimal/Octal Conversion Tables
13:1,13:2	Complete List of Program Instructions
14	Standard Tape Dimensions
15	Octal Addresses of all Pages from 0 to 77

THE BASIC INPUT OUTPUT CODING FOR THE MOLECULAR 18

	...	..5	.6.	.65	7..	7.5	76.	765
....	NUL	DLE	SPACE	∅	@	P	`	p
...1	SOH	DC1	!	1	A	Q	a	q
..2.	STX	DC2	"	2	B	R	b	r
..21	ETX	DC3	£	3	C	S	c	s
.3..	EOT	DC4	\$	4	D	T	d	t
.3.1	ENQ	NAK	%	5	E	U	e	u
.32.	ACK	SYN	&	6	F	V	f	v
.321	BEL	ETB	'	7	G	W	g	w
4...	BS	CAN	(	8	H	X	h	x
4..1	HT	EM	)	9	I	Y	i	y
4.2.	LF	SUB	*	:	J	Z	j	z
4.21	VT	ESC	+	;	K	[	k	{
43..	FF	FS	,	<	L	½	l	
43.1	CR	GS	-	=	M	]	m	}
432.	SO	RS	.	>	N	^	n	~
4321	SI	US	/	?	O	_	o	DEL

TAPE FEED:- FEED HOLE ONLY  
 PARITY IS EVEN:- ADD BIT 8 WHERE NECESSARY

CODING CHART FOR IBM (INPUT AND OUTPUT) BCL HEAD

	...	..5	.6.	.65	7..	7.5	76.	765
....	L/C	W ½	B ½		- _		SPACE	* :
...1	' "			11 £		S ½	H ½	
..2.								
..21								
.3..	, +			9 (		I ½	K ½	
.3.1		z ;	E ½		10 \$			Z ½
.32.		7 ½	N ½		J ½			O )
.321	Y ½			8 &		4 @	X ½	
4...	BS							
4..1	TAB	O ½	L ½		G ½			6 ½
4.2.	LF							
4.21								
43..		A ½	C ½		P ½			5 ½
43.1	Q ½			3 ½	CR	R ½	D ½	
432.	F ½			2 ½		V ½	U ½	
4321	U/C	M ½	T ½		. /			1 ?

Note :- ▲ only to have parity bit (Bit 8) true added.

Characters on left and right hand side (' ') are Lower Case and Upper Case, respectively.

BCL SERIAL PRINTER CODE SET

	...	..5	.6.	.65	7..	7.5	76.	765
....			Space	∅	@	P		
...1			!	1	A	Q		
..2.			"	2	B	R		
..21			£	3	C	S		
.3..			\$	4	D	T		
.3.1			%	5	E	U		
.32.			&	6	F	V		
.321			'	7	G	W		
4...			(	8	H	X		
4..1			)	9	I	Y		
4.2.	LF1		*	:	J	Z		
4.21	LF2		+	;	K	[		
43..			,	<	L	\		
43.1	C/R		-	=	M	]		
432.			.	>	N	^		
4321			/	?	O	_		

LINE PRINTER CODE SET

	...	..5	.6.	.65	7..	7.5	76.	765
....			SPACE	Ø	@	P		
...1			!	1	A	Q		
..2.			"	2	B	R		
..21			#	3	C	S		
.3..			\$	4	D	T		
.3.1			%	5	E	U		
.32.			&	6	F	V		
.321			'	7	G	W		
4...			(	8	H	X		
4..1			)	9	I	Y		
4.2.	LF		*	:	J	Z		
4.21			+	;	K	[		
43..	FF		,	<	L	\		
43.1	Print		-	=	M	]		
432.			.	>	N	]		
4321			/	?	O	-		

Print cycle occurs automatically if 132  
into the buffer store.

characters are transferred

LF    Line Feed  
FF    Form Feed

TALLY ROLL PRINTER CODE SET

	...	..5	.6.	.65	7..	7.5	76.	765
....			SPACE	∅	@	P		
...1			!	1	A	Q		
..2.			"	2	B	R		
..21			£	3	C	S		
.3..			\$	4	D	T		
.3.1			%	5	E	U		
.32.			&	6	F	V		
.321	Bleep		'	7	G	W		
4...			(	8	H	X		
4..1			)	9	I	Y		
4.2.			*	:	J	Z		
4.21			+	;	K	[		
43..			,	<	L	\		
43.1			-	=	M	]		
432.			.	>	N	^		
4321			/	?	O	_		

## CODE SET FOR VISUAL DISPLAY TERMINAL with Keyboard (92423)

Cols.	0	1	2	3	4	5	6	7
	...	..5	.6.	.65	7..	7.5	76.	765
....	NUL	DLE	SPACE	Ø	Ⓐ	P	`	p
...1	SOH	DC1	!	1	A	Q	a	q
..2.	<u>STX</u>	DC2	"	2	B	R	b	r
..21	<u>ETX</u> *	DC3	#	3	C	S	c	s
.3..	EOT	DC4	\$	4	D	T	d	t
.3.1	ENQ	<u>Skip</u> *	%	5	E	U	e	u
.32.	ACK	<u>Line Clear</u> *	&	6	F	V	f	v
.321	BEL	ETB	'	7	G	W	g	w
4...	<u>BS</u> *	<u>Clear</u> *	(	8	H	X	h	x
4..1	HT	<u>Reset</u> *	)	9	I	Y	i	y
4.2.	<u>LF</u> *	<u>Cursor Up</u> *	*	:	J	Z	j	z
4.21	VT	<u>ESC</u>	+	;	K	[	k	{
43..	FF	<u>FS</u>	,	<	L	\	l	!
43.1	<u>CR</u> *	<u>GS</u>	-	=	M	]	m	}
432.	<u>SO</u> *	<u>RS</u>	.	>	N	^	n	~
4321	<u>SI</u> *	<u>US</u>	/	?	O	_	o	■

**Notes**

- 1) All Control Codes in Columns 0 and 1 can be generated by pressing corresponding Keys in Columns 4 and 5 together with the CNTRL Key. In addition certain codes (those under-lined in above chart) may also be produced by specially marked Keys.
- 2) SO = Inverse Video ON (shows the symbol < on the Display)  
SI = Inverse Video OFF (shows the symbol > on the Display)  
ETX shows the symbol ▲  
Rubout (also Parity) shows the symbol ■  
Break Key gives the symbol ■■
- 3) Codes marked with an \* show cursor movement.
- 4) Codes marked with \* produce special characters (see Note 2 above)

CODE SET FOR ALPHA/NUMERIC KEYBOARD

	...	..5	.6.	..65	7..	7.5	76.	765
....	NUL	DLE	SPACE	∅	@	P		
...1	SOH	DC1	!	1	A	Q		
..2.	STX	DC2	"	2	B	R		
..21	ETX	DC3	£ or #	3	C	S		
.3..	EOT	DC4	\$	4	D	T		
.3.1	ENQ	NAK	%	5	E	U		
.32.	ACK	SYN	&	6	F	V		
.321	BEL	ETB (Accept)	'	7	G	W		
4...	BS	CAN (ER)	(	8	H	X		
4..1	TAB	EM	)	9	I	Y		
4.2.	LF	SUB	*	:	J	Z		
4.21	VT	* <sub>2</sub>	+	;	K	[		{
43..	FF		,	<	L	½		
43.1	CR	◊	-	=	M	]		}
432.			.	>	N	^		~
4321	‰		/	?	O	_		

Parity is even - Bit 8 is added where necessary



CODE LIST FOR ALL IBM CHARACTERS

<u>LOWER CASE</u>	<u>UPPER CASE</u>	<u>BITS 87654321</u>		<u>BITS 87654321</u>
0	)	1110110	L/C	10000000
1	?	1111111	B. SPACE	10001000
2	!	0111110	TAB	10001001
3	@	0111101	LINE FEED	10001010
4	#	1010111	U/C	10001111
5	\$	1111100	CR	11001101
6	%	1111001	SPACE	11100000
7	&	0010110		
8	'	0110111		
9	(	0110100		
10	)	1000101		
11	*	0110001		
A	A	0011100		
B	B	0100000		
C	C	0101100		
D	D	1101101		
E	E	0100101		
F	F	0001110		
G	G	1001001		
H	H	1100001		
I	I	1010100		
J	J	1000110		
K	K	1100100		
L	L	0101001		
M	M	0011111		
N	N	0100110		
O	O	0011001		
P	P	1001100		
Q	Q	0001101		
R	R	1011101		
S	S	1010001		
T	T	0101111		
U	U	1101110		
V	V	1011110		
W	W	0010000		
X	X	1100111		
Y	Y	0000111		
Z	Z	1110101		
-	-	1000000		
*	*	1110000		
,	,	0000001		
.	.	0000100		
/	/	0010101		
.	.	1001111		

<u>Bit No.</u>	<u>Decimal Value of Bit</u>	<u>Decimal Capacity of all Bits Up to and including Bit Shown</u>
1	1	1
2	2	3
3	4	7
4	8	15
5	16	31
6	32	63
7	64	127
8	128	255
9	256	511
10	512	1,023
11	1,024	2,047
12	2,048	4,095
13	4,096	8,191
14	8,192	16,383
15	16,384	32,767
16	32,768	65,535
17	65,536	131,071
18	131,072	262,143
19	262,144	524,287
20	524,288	1,048,575
21	1,048,576	2,097,151
22	2,097,152	4,194,303
23	4,194,304	8,388,607
24	8,388,608	16,777,215
25	16,777,216	33,554,431
26	33,554,432	67,108,863
27	67,108,864	134,217,727
28	134,217,728	268,435,455
29	268,435,456	536,870,911
30	536,870,912	1,073,741,823
31	1,073,741,824	2,147,483,647
32	2,147,483,648	4,294,967,295
33	4,294,967,296	8,589,934,591
34	8,589,934,592	17,179,869,183
35	17,179,869,184	34,359,738,367
36	34,359,738,368	68,719,476,735
37	68,719,476,736	137,438,953,471
38	137,438,953,472	274,877,906,943
39	274,877,906,944	549,755,813,887
40	549,755,813,888	1,099,511,627,775
41	1,099,511,627,776	2,199,023,255,551
42	2,199,023,255,552	4,398,046,511,103
43	4,398,046,511,104	8,796,093,022,207
44	8,796,093,022,208	17,592,186,044,415
45	17,592,186,044,416	35,184,372,088,831
46	35,184,372,088,832	70,368,744,177,663
47	70,368,744,177,664	140,737,488,355,327
48	140,737,488,355,328	281,474,976,710,655

2-CHARACTER STORAGE

A	0404 101
B	0410 102
C	0414 103
D	0420 104
E	0424 105
F	0430 106
G	0434 107
H	0440 110
I	0444 111
J	0450 112
K	0454 113
L	0460 114
M	0464 115
N	0470 116
O	0474 117
P	0500 120
Q	0504 121
R	0510 122

S	0514 123
T	0520 124
U	0524 125
V	0530 126
W	0534 127
X	0540 130
Y	0544 131
Z	0550 132
Ø	0300 060
1	0304 061
2	0310 062
3	0314 063
4	0320 064
5	0324 065
6	0330 066
7	0334 067
8	0340 070
9	0344 071

½	0560 134
,	0260 054
.	0270 056
SPACE	0200 040
TAB	0044 011
C/R	0064 015
VT	0054 013
L/F	0050 012
BS	0040 010
'	0234 047
(	0240 050
)	0244 051
:	0350 072
;	0354 073
?	0374 077
"	0210 042
!	0204 041
/	0274 057

£	0214 043
\$	0220 044
@	0400 100
[	0554 133
]	0564 135
^	0570 136
┌	0574 137
%	0224 045
&	0230 046
*	0250 052
+	0254 053
-	0264 055
<	0360 074
>	0370 076
=	0364 075
BEL	0034 007
CAN	0140 030
ETB	0134 027

2-CHARACTER STORAGE (continued)

NUL	0000 000	g	0634 147	u	0724 165	ENQ	0024 005
DEL	0774 177	h	0640 150	v	0730 166	ACK	0030 006
EOT	0020 004	i	0644 151	w	0734 167	DLE	0100 020
FF	0060 014	j	0650 152	x	0740 170	DC1	0104 021
ESC	0154 033	k	0654 153	y	0744 171	DC2	0110 022
GS	0164 035	l	0660 154	z	0750 172	DC3	0114 023
SO	0070 016	m	0664 155	`	0600 140	DC4	0120 024
SI	0074 017	n	0670 156	{	0754 173	NAK	0124 025
a	0604 141	o	0674 157	}	0764 175	SYN	0130 026
b	0610 142	p	0700 160		0760 174	EM	0144 031
c	0614 143	q	0704 161	⌋	0770 176	SUB	0150 032
d	0620 144	r	0710 162	SOH	0004 001	FS	0160 034
e	0624 145	s	0714 163	STX	0010 002	RS	0170 036
f	0630 146	t	0720 164	ETX	0014 003	US	0174 037

These tables will be found very useful when storing messages, literals, etc. in core, at two characters to a word. It should be used as follows :-

For a character in the top (most significant) half of the store, use the upper four-figure number, and for a character in the bottom (least significant) half of the store add the lower three-figure number in the appropriate position, so that a six-digit octal number is produced.

e.g. Should you wish to store the word 'STOP' in 2 words of core, it should be stored as follows :-

S - 0514      0474      - O  
T -    124      120      P

051524	047520
--------	--------

PERIPHERAL CHARACTER SETS

<u>I.B.M. Code Bit Structure</u>	<u>I.B.M.</u>	<u>Line Printer</u>	<u>Dot Printer</u>	<u>C.R.T. V.D.U.</u>	<u>Alpha Numeric Keyboard</u>	<u>Numeric Keyboard</u>	<u>Tally Roll Printer</u>	<u>ASCII Code Bit Structure</u>
876.....	Space	Space	Space	Space	Space		Space	8.6.....
		!	!	!	!		!	..6....1
.....1 ■	"	"	"	"	"		"	..6...2.
..65...1 ■	£	#	#	#	#		£	8.6...21
.7...3.1 ■	\$	\$	\$	\$	\$		\$	..6...3..
...5.3.1	%	%	%	%	%		%	8.6...3.1
..65.321 ■	&	&	&	&	&		&	8.6...3.
.....1	'	'	'	'	'		'	..6..321
..65.3.. ■	(	(	(	(	(		(	..6.4...
.765.32. ■	)	)	)	)	)		)	8.6.4..1
.765... ■	*	*	*	*	*		*	8.6.4.2.
.....3.. ■	+	+	+	+	+		+	..6.4.21
.....3..	,	,	,	,	,		,	8.6.43..
.7.....	-	-	-	-	-		-	..6.43.1
.7..4321	.	.	.	.	.	DP	.	..6.432.
.7..4321 ■	/	/	/	/	/		/	8.6.4321
.765.32.	∅	∅	∅	∅	∅	∅	∅	..65....
.7654321	1	1	1	1	1	1	1	8.65...1
..65432.	2	2	2	2	2	2	2	8.65...2
..6543.1	3	3	3	3	3	3	3	..65...21
.7.5.321	4	4	4	4	4	4	4	8.65.3..
.76543..	5	5	5	5	5	5	5	..65.3.1
.7654..1	6	6	6	6	6	6	6	..65.32.
...5.32.	7	7	7	7	7	7	7	8.65.321
..65.321	8	8	8	8	8	8	8	8.654...
..65.3..	9	9	9	9	9	9	9	..654..1

Note :- ■ = Upper Case

Cont.

<u>I.B.M. Code Bit Structure</u>	<u>I.B.M.</u>	<u>Line Printer</u>	<u>Dot Printer</u>	<u>C.R.T. V.D.U.</u>	<u>Alpha Numeric Keyboard</u>	<u>Numeric Keyboard</u>	<u>Tally Roll Printer</u>	<u>ASCII Code Bit Structure</u>
.765.... ■	:	:	:	:	:		:	..654.2.
...5.3.1 ■	;	;	;	;	;		;	8.654.21
		<	<	<	<		<	..6543..
		=	=	=	=		=	8.6543.1
		>	>	>	>		>	8.65432.
.7654321 ■	?	?	?	?	?		?	..654321
.7.5.321 ■	@	@	@	@	@		@	87.....
...543..	A	A	A	A	A		A	.7.....1
..6.....	B	B	B	B	B		B	.7.....2.
..6.43..	C	C	C	C	C		C	87....21
.76.43.1	D	D	D	D	D		D	.7...3..
..6..3.1	E	E	E	E	E		E	87...3.1
....432.	F	F	F	F	F		F	87...32.
.7..4..1	G	G	G	G	G		G	.7...321
.76....1	H	H	H	H	H		H	.7..4...
.7.5.3..	I	I	I	I	I		I	87..4..1
.7...32.	J	J	J	J	J		J	87..4.2.
.76..3..	K	K	K	K	K		K	.7..4.21
..6.4..1	L	L	L	L	L		L	87..43..
...54321	M	M	M	M	M		M	.7..43.1
..6..32.	N	N	N	N	N		N	.7..432.
...54..1	O	O	O	O	O		O	87..4321
.7..43..	P	P	P	P	P		P	.7.5....
....43.1	Q	Q	Q	Q	Q		Q	87.5...1
.7.543.1	R	R	R	R	R		R	87.5..2.
.7.5...1	S	S	S	S	S		S	.7.5..21
..6.4321	T	T	T	T	T		T	87.5.3..
.76.432.	U	U	U	U	U		U	.7.5.3.1
.7.5432.	V	V	V	V	V		V	.7.5.32.
...5.....	W	W	W	W	W		W	87.5.321

<u>I.B.M. Code Bit Structure</u>	<u>I.B.M.</u>	<u>Line Printer</u>	<u>Dot Printer</u>	<u>C.R.T. V.D.U.</u>	<u>Alpha Numeric Keyboard</u>	<u>Numeric Keyboard</u>	<u>Tally Roll Printer</u>	<u>ASCII Code Bit Structure</u>
.76..321	X	X	X	X	X		X	87.54...
.....321	Y	Y	Y	Y	Y		Y	.7.54..1
.765.3.1	Z	Z	Z	Z	Z		Z	.7.54.2.
		[	[	[	[		[	87.54.21
.76543...■	½	\	\	\	\		\	.7.543..
		]	]	]	]		]	87.543.1
		⎵	↑	^	^		^	87.5432.
.7.....■	-	-	-	-	-		-	.7.54321
8...4...	Back Space			Back Space	Back Space			8...4...
8...4..1	TAB			TAB	TAB			....4..1
8...4.2.	LF	LF	LF A	LF	LF			....4.2.
			LF B	VT	VT			8...4.21
87..43.1	CR		CR	CR	CR			8...43.1
				BEL	BEL	BEL	Bleep	8....321
					Error	Error		...54...
					◊	◊		...543.1
					*	*		...54.21
					Accept	Accept		...5.321

Note :- ■ = Upper Case

DECIMAL/OCTAL CONVERSION TABLES

The Tables on the next two pages allow easy Decimal/Octal and Octal/Decimal conversion to be made.

The Decimal to Octal Table (on next page) converts up to a maximum of 139,999 decimal (421,337 octal), and the Octal to Decimal Table (Appendix 12:3), converts up to a maximum of 377,777 octal (131,071 decimal).

The tables are used as follows :-

Decimal to Octal (Appendix 12:2)

Split the decimal number into three pairs of digits, and use each pair as the matrix reference to the three sections of the chart, adding together the three figures so referenced.

For instance :-        Decimal 129,748

Split into 3 pairs :-    12    97    48

Use most significant pair on top section of the chart	- gives 352300
Use middle pair on middle section of chart	- gives 22744
Use least significant pair on bottom section of chart	- gives 60

---

Total 375324

So decimal 129,748 = octal 375324

Note that the addition of the three referenced numbers must be done in octal (6+4=12).

Octal to Decimal (Appendix 12:3)

Split the octal number into three pairs of digits, and use each pair as the matrix reference to the three sections of the chart, adding together the three figures so referenced.

For instance :-        Octal 375324

Split into 3 pairs :-    37    53    24

Use most significant pair on top section of chart	- gives 126,976
Use middle pair on middle section of chart	- gives 2,752
Use least significant pair on bottom section of chart	- gives 20

---

Total 129,748

So octal 375342 = decimal 129,748

Note that the addition of the three referenced numbers must be done in decimal.



DECIMAL to OCTAL CONVERSION TABLE

0	0	23420	47040	72460	116100	141520	165140	210560	234200	257620
1	303240	326660	352300	375720						
	0	1	2	3	4	5	6	7	8	9

0	0	144	310	454	620	764	1130	1274	1440	1604
1	1750	2114	2260	2424	2570	2734	3100	3244	3410	3554
2	3720	4064	4230	4374	4540	4704	5050	5214	5360	5524
3	5670	6034	6200	6344	6510	6654	7020	7164	7330	7474
4	7640	10004	10150	10314	10460	10624	10770	11134	11300	11444
5	11610	11754	12120	12264	12430	12574	12740	13104	13250	13414
6	13560	13724	14070	14234	14400	14544	14710	15054	15220	15364
7	15530	15674	16040	16204	16350	16514	16660	17024	17170	17334
8	17500	17644	20010	20154	20320	20464	20630	20774	21140	21304
9	21450	21614	21760	22124	22270	22434	22600	22744	23110	23254
	0	1	2	3	4	5	6	7	8	9

0	0	1	2	3	4	5	6	7	10	11
1	12	13	14	15	16	17	20	21	22	23
2	24	25	26	27	30	31	32	33	34	35
3	36	37	40	41	42	43	44	45	46	47
4	50	51	52	53	54	55	56	57	60	61
5	62	63	64	65	66	67	70	71	72	73
6	74	75	76	77	100	101	102	103	104	105
7	106	107	110	111	112	113	114	115	116	117
8	120	121	122	123	124	125	126	127	130	131
9	132	133	134	135	136	137	140	141	142	143
	0	1	2	3	4	5	6	7	8	9

OCTAL to DECIMAL CONVERSION TABLE

0	0	4096	8192	12288	16384	20480	24576	28672
1	32768	36864	40960	45056	49152	53248	57344	61440
2	65536	69632	73728	77824	81920	86016	90112	94208
3	98304	102400	106496	110592	114688	118784	122880	126976
	0	1	2	3	4	5	6	7

0	0	64	128	192	256	320	384	448
1	512	576	640	704	768	832	896	960
2	1024	1088	1152	1216	1280	1344	1408	1472
3	1536	1600	1664	1728	1792	1856	1920	1984
4	2048	2112	2176	2240	2304	2368	2432	2496
5	2560	2624	2688	2752	2816	2880	2944	3008
6	3072	3136	3200	3264	3328	3392	3456	3520
7	3584	3648	3712	3776	3840	3904	3968	4032
	0	1	2	3	4	5	6	7

0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63
	0	1	2	3	4	5	6	7

COMPLETE LIST OF PROGRAM INSTRUCTIONS

Memory Ref. Instructions

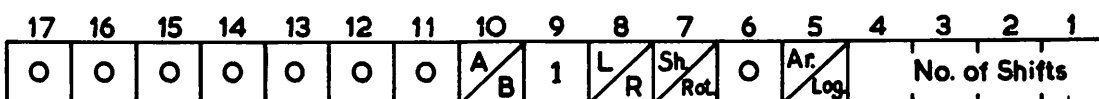
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
OPERATION CODE					I D	Z C	MEMORY ADDRESS											
0	0	0	1	0		02											JUMP	Jump
0	0	0	1	1		(03)											JSBR	Jump to Subroutine
0	0	1	0	0		(04)											INSZ	Increment, Skip if Zero
0	0	1	0	1		(05)											DESZ	Decrement, Skip if Zero
0	0	1	1	0		(06)											ANDA	'And' to A
0	0	1	1	1		(07)											IORA	'Inclusive OR' to A
0	1	0	0	0		(10)											XORA	'Exclusive OR' to A
0	1	0	0	1		(11)											ADA	Add to A
0	1	0	1	0		(12)											ADB	Add to B
0	1	0	1	1		(13)											SFA	Subtract from A
0	1	1	0	0		(14)											SFB	Subtract from B
0	1	1	0	1		(15)											ADAC	Add to A, with Carry
0	1	1	1	0		(16)											ADBC	Add to B, with Carry
0	1	1	1	1		(17)											SFAC	Subtract from A with Carry
1	0	0	0	0		(20)											SFBC	Subtract from B with Carry
1	0	0	0	1		(21)											LDA	Load A
1	0	0	1	0		(22)											LDB	Load B
1	0	0	1	1		(23)											CMPA	Compare A, Skip if not equal
1	0	1	0	0		(24)											CMPB	Compare B, Skip if not equal
1	0	1	0	1		(25)											STA	Store A
1	0	1	1	0		(26)											STB	Store B
* 1	0	1	1	1		(27)											UNSTK	Unstack

\* Input/Output Instructions

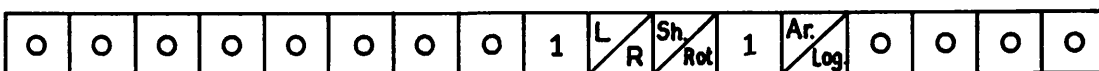
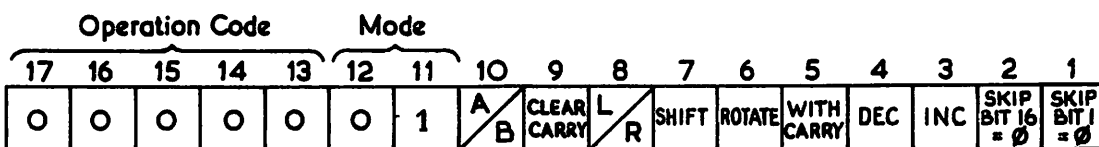
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	1	A B	FUNCT.	MODE	DEVICE CODE								
								0	0	0	No I/O Transfer, just Function					
								0	0	1	DATI 1					
								0	1	0	DATI 2					
								0	1	1	DATI 3					
								1	0	0	DATO 1					
								1	0	1	DATO 2					
								1	1	0	DATO 3					
								0	0	0	Skip Mode					
								0	1	1	Skip Mode					
								1	0	0	Skip Mode					
								1	1	1	Skip Mode					

All instructions marked with an \* are 'Privileged' instructions

Literal Instructions



Register Instructions



Control Instructions (Octal Code)

000000	No Op	* 000014	SK. Limit
* 000001	Halt	* 000015	SK. MA=SW
* 000002	Mask Out	* 000016	SK. Cont. Int.
* 000003	Ack. Int.	* 000017	I O Reset
* 000004	Int. On	* 000020	SK. Overstack
* 000005	Int. Off	* 000021	SK. >15 Indirects
* 000006	SK. Int. On	* 000022	SK. Timer Int.
* 000007	SK. Int. Off	* 000023	SK. Illegal Op
* 000010	SK. Mains Fail	* 000024	SK. Extra Code
* 000011	SK. Mains Ret.		
* 000012	SK. Mem. Parity	001000	Extracode
* 000013	Reserved	001001	Set Greater Than

Interrupt Addresses

000205	Cont. Int.
000206	MA=SW Int.
000207	Mains Return
000210	Extra Code
000211	I/O Int.
000212	Illegal Op
000213	> 15 Indirects
000214	Limit
000215	Mem. Parity
000216	Mains Fail
000217	7th Level Int.

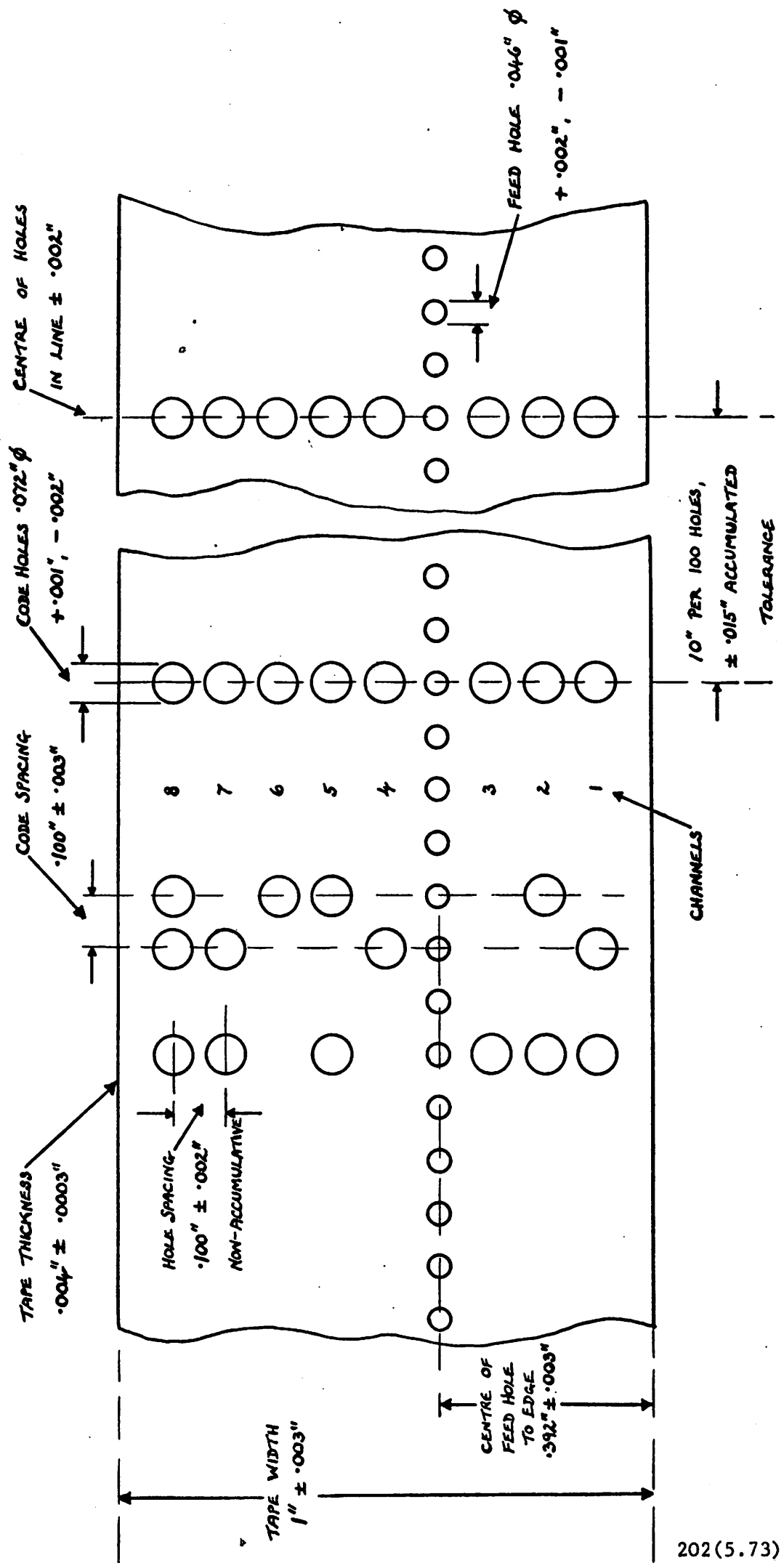
Interrupt Stack

Stack Pointer Address = 000030  
 1st. Word Bit 1=Carry, Bit 2=G.Than  
 2nd Word PC & Mode  
 Bit 17=1 Absolute Add. Mode  
 3rd. Word Base  
 4th Word B Register  
 5th Word A Register

Mains Return Auto Start Address 000002  
 Auto Inc. Mem. Addresses 000010 to 000017  
 Auto Dec. Mem. Addresses 000020 to 000027

Limit Register loaded with 1st. word of program +Base

# STANDARD TAPE DIMENSIONS



Page 0	-	000000	to	001777
Page 1	-	002000	to	003777
Page 2	-	004000	to	005777
Page 3	-	006000	to	007777
Page 4	-	010000	to	011777
Page 5	-	012000	to	013777
Page 6	-	014000	to	015777
Page 7	-	016000	to	017777
Page 10	-	020000	to	021777
Page 11	-	022000	to	023777
Page 12	-	024000	to	025777
Page 13	-	026000	to	027777
Page 14	-	030000	to	031777
Page 15	-	032000	to	033777
Page 16	-	034000	to	035777
Page 17	-	036000	to	037777
Page 20	-	040000	to	041777
Page 21	-	042000	to	043777
Page 22	-	044000	to	045777
Page 23	-	046000	to	047777
Page 24	-	050000	to	051777
Page 25	-	052000	to	053777
Page 26	-	054000	to	055777
Page 27	-	056000	to	057777
Page 30	-	060000	to	061777
Page 31	-	062000	to	063777
Page 32	-	064000	to	065777
Page 33	-	066000	to	067777
Page 34	-	070000	to	071777
Page 35	-	072000	to	073777
Page 36	-	074000	to	075777
Page 37	-	076000	to	077777
Page 40	-	100000	to	101777
Page 41	-	102000	to	103777
Page 42	-	104000	to	105777
Page 43	-	106000	to	107777
Page 44	-	110000	to	111777
Page 45	-	112000	to	113777
Page 46	-	114000	to	115777
Page 47	-	116000	to	117777
Page 50	-	120000	to	121777
Page 51	-	122000	to	123777
Page 52	-	124000	to	125777
Page 53	-	126000	to	127777
Page 54	-	130000	to	131777
Page 55	-	132000	to	133777
Page 56	-	134000	to	135777
Page 57	-	136000	to	137777
Page 60	-	140000	to	141777
Page 61	-	142000	to	143777
Page 62	-	144000	to	145777
Page 63	-	146000	to	147777
Page 64	-	150000	to	151777
Page 65	-	152000	to	153777
Page 66	-	154000	to	155777
Page 67	-	156000	to	157777
Page 70	-	160000	to	161777
Page 71	-	162000	to	163777
Page 72	-	164000	to	165777
Page 73	-	166000	to	167777
Page 74	-	170000	to	171777
Page 75	-	172000	to	173777
Page 76	-	174000	to	175777
Page 77	-	176000	to	177777