SERIAL LINE CONTROL SWITCHES

Switch	Normal	Description
1	ON	Transmitter clock frequency select
2	ON	Receiver clock frequency select
3	ON	Stop bit select
4	OFF	Transmitter mode select
5	OFF	Receiver mode select
6	ON	Serial line circuitry enabled
7	ON	Initialize circuitry enabled*
8	ON	System clock enabled*

*The OFF position for test only.

MACHINE LANGUAGE TO ASSEMBLY LANGUAGE MNEMONIC

LSD / MSD	0	1	2	3	4	5	6	7
00	HLT	HLT	RLC	RFC	ADI	RST	LAI	RET
01	INB	DCB	RRC	RFZ	ACI	RST	LBI	RET
02	INC	DCC	RAL	RFS	SUI	RST	LCI	RET
03	IND	DCD	RAR	RFP	SBI	RST	LDI	RET
04	INE	DCE		RTC	NDI	RST	LEI	RET
05	INH	DCH		RTZ	XRI	RST	LHI	RET
06	INL	DCL		RTS	DRI	RST	LLI	RET
07				RTP	CPI	RST	LMI	RET
10	JFC	INP0	CFC	INP1	JMP	INP2	CAL	INP3
11	JFZ	INP4	CFZ	INP5	JMP	INP6	CAL	INP7
12	JFS	OUT0	CFS	ION	JMP	IOF	CAL	OUT3
13	JFP	OUT4	CFP	OUT5	JMP	OUT6	CAL	OUT7
14	JTC	OUT10	CTC	OUT11	JMP	OUT12	CAL	OUT13
15	JTZ	OUT14	CTZ	OUT15	JMP	OUT16	CAL	OUT17
16	JTS	OUT20	CTS	OUT21	JMP	OUT22	CAL	OUT23
17	JTP	OUT24	CTP	OUT25	JMP	OUT26	CAL	OUT27
20	ADA	ADB	ADC	ADD	ADE	ADH	ADL	ADM
21	ACA	ACB	ACC	ACD	ACE	ACH	ACL	ACM
22	SUA	SUB	SUC	SUD	SUE	SUH	SUL	SUM
23	SBA	SBB	SBC	SBD	SBE	SBH	SBL	SBM
24	NDA	NDB	NDC	NDD	NDE	NDH	NDL	NDM
25	XRA	XRB	XRC	XRD	XRE	XRH	XRL	XRM
26	ORA	ORB	ORC	ORD	ORE	ORH	ORL	ORM
27	CPA	CPB	CPC	CPD	CPE	CPH	CPL	CPM
30	LAA	LAB	LAC	LAD	LAE	LAH	LAL	LAM
31	LBA	LBB	LBC	LBD	LBE	LBH	LBL	LBM
32	LCA	LCB	LCC	LCD	LCE	LCH	LCL	LCM
33	LDA	LDB	LDC	LDD	LDE	LDH	LDL	LDM
34	LEA	LEB	LEC	LED	LEE	LEH	LEL	LEM
35	LHA	LHB	LHC	LHD	LHE	LHH	LHL	LHM
36	LLA	LLB	LLC	LLD	LLE	LLH	LLL	LLM
37	LMA	LMB	LMC	LMD	LME	LMH	LML	HLT

MDP—MICROPROCESSOR DEBUG PROGRAM

Instruction	Form	Description
B	B ADDR	Set breakpoint at address
B R	B R	Remove breakpoint
D	D ADDR1; ADDR2	Dump memory from ADDR1 to ADDR2
E	E	Punch end block and trailer on paper tape
G	G ADDR	Go and execute program at address
L	L ADDR1; ADDR2;N	Load memory from ADDR1 to ADDR2 with constant N
P	P ADDR1; ADDR2	Punch memory from ADDR1 to ADDR2
R	R	Read paper tape
S	S	Open status storage register
T	T	Punch trailer
X	X	Open index register A
<CTRL/C>	↑C	Aborts current command
<RUBOUT>	\X	Ignore previous character, X
/	ADDR/ XXX	Open location address
.	ADDR/ .	Open current location again
<LF>	ADDR/ <LF>	Open next location
↑	ADDR/ ↑	Open previous location

MRP—MICROPROCESSOR ROM PROGRAMMER

Instruction	Form	Description
C	C	Check for clear PROM
D	D ADDR1; ADDR2	Dump memory from ADDR1 to ADDR2
E	E	Punch end block and trailer
F	F ADDR1; ADDR2	Fetch data from PROM
L	L ADDR1; ADDR2;N	Load data buffer from ADDR1 to ADDR2 with N
P	P ADDR1; ADDR2	Punch data buffer from ADDR1 to ADDR2
Q	Q	Read the queued paper tape
R	R	Read paper tape
T	T	Punch trailer
V	V ADDR1; ADDR2	Verify the contents of PROM
W	W ADDR1; ADDR2	Write data into PROM
/	ADDR/ XXX	Open location ADDR
.	ADDR/ .	Open current location again
<LF>	ADDR/ <LF>	Open next location
↑	ADDR/ ↑	Open previous location
<RUBOUT>	\X	Ignore previous character
<CTRL/C>	↑C	Abort current command

MRP—Switch Register Assignment

Bit	Function
9	Output select 0-TTY 1-Line Printer
11	Punch select 0-Low Speed 1-High Speed

MPL—MICROPROCESSOR PROGRAM LOADER

Starting Address Block 77 Offset 000

MHL—MICROPROCESSOR HOST LOADER

Starting Address 7777 Base 8
Bit 0 set, select low speed reader

MTD—MASTER TAPE DUPLICATOR

Starting Address 0200 Base 8

Switch Register Assignments

Bit	Function
0	Create master
1	Duplicate master
2	Verify tape

7-BIT ASCII CODE

Octal Code	Char	Octal Code	Char	Octal Code	Char	Octal Code	Char
000	NUL	040	SP	100	@	140	`
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ECS	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	↑	176	~
037	US	077	?	137	←	177	DEL

PRINTED IN USA 5813 00674 3894/Y 03 05

digital

MPS MICROPROCESSOR SERIES POCKET REFERENCE CARD

INSTRUCTION SET

Mnemonic	Time States	# Bytes	Instruction Codes			Condition Bits CZSP	Description
Lr1r2	5	1	11	DDD	SSS	****	r1←r2
LrM	8	1	11	DDD	111	****	r←M
LMr	7	1	11	111	SSS	****	M←r
LrI	8	2	00	DDD	110	****	r←<B2>
LMI	9	2	00	111	110	****	M←<B2>
INr	5	1	00	DDD	000	*YYY	r←r+1 r≠A,M
DCr	5	1	00	DDD	001	*YYY	r←r−1 r≠A,M
ADr	5	1	10	000	SSS	YYYY	A←A+r
ADM	8	1	10	000	111	YYYY	A←A+M
ADI	8	2	00	000	100	YYYY	A←A+<B2>
ACr	5	1	10	001	SSS	YYYY	A←A+r+C
ACM	8	1	10	001	111	YYYY	A←A+M+C
ACI	8	2	00	001	100	YYYY	A←A+<B2>+C
SUr	5	1	10	010	SSS	YYYY	A←A−r
SUM	8	1	10	010	111	YYYY	A←A−M
SUI	8	2	00	010	100	YYYY	A←A−<B2>
SBr	5	1	10	011	SSS	YYYY	A←A−r−C
SBM	8	1	10	011	111	YYYY	A←A−M−C
SBI	8	2	00	011	100	YYYY	A←A−<B2>−C
NDr	5	1	10	100	SSS	0YYY	A←A AND r
NDM	8	1	10	100	111	0YYY	A←A AND M
NDI	8	2	00	100	100	0YYY	A←A AND <B2>
XRr	5	1	10	101	SSS	0YYY	A←A EXCLUSIVE OR r
XRM	8	1	10	101	111	0YYY	A←A EXCLUSIVE OR M
XRI	8	2	00	101	100	0YYY	A←A EXCLUSIVE OR <B2>
ORr	5	1	10	110	SSS	0YYY	A←A OR r
ORM	8	1	10	110	111	0YYY	A←A OR M
ORI	8	2	00	110	100	0YYY	A←A OR <B2>
CPr	5	1	10	111	SSS	YYYY	A-r } Set condition bits
CPM	8	1	10	111	111	YYYY	A-M } Z=1 if A=r, M, <B2>
CPI	8	2	00	111	100	YYYY	A-<B2> } C=1 if A<r, M, <B2>
RLC	5	1	00	000	010	Y***	A(m+1)←A(m), A(0)← A(7), C←A(7)
RRC	5	1	00	001	010	Y***	A(m)←A(m+1), A(7)← (0), C←A(0),
RAL	5	1	00	010	010	Y***	A(m+1)←A(m), C←A(7), A(0)←C
RAR	5	1	00	011	010	Y***	A(m)←A(m+1), C←A(0), A(7)←C
JMP	11	3	01	XXX	100	****	P←<B3> <B2>
JFc	9/11	3	01	$0C_4C_3$	000	****	P←<B3> <B2>, If Flag c clear else P←P+3
JTc	9/11	3	01	$1C_4C_3$	000	****	P←<B3> <B2>, If Flag c set, else P←P+3
CAL	11	3	01	XXX	110	****	STACK←P+3, P←<B3> <B2>
CFc	9/11	3	01	$0C_4C_3$	000	****	STACK←P,P←<B3><B2>, If Flag c clear, else P←P+3
CTc	9/11	3	01	$1C_4C_3$	010	****	STACK←P,P←<B3><B2>, If Flag c set, else P←P+3
RET	5	1	00	XXX	111	****	P←STACK
RFc	3/5	1	00	$0C_4C_3$	011	****	P←STACK, If Flag c clear, else P←P+1
RTc	3/5	1	00	$1C_4C_3$	011	****	P←STACK, If Flag c set, else P←P+1
RST	5	1	00	AAA	101	****	STACK←P, P←AAA000
INP	8	1	01	00M	MM1	****	A←Contents of Port MMM
OUT	6	1	01	RRM	MM1	****	Port RRMMM←ARR≠00
HLT	4	1	00	000	00X	****	Enter Stopped State
HLT	4	1	11	111	111	****	Enter Stopped State

DEFINITIONS

r	Index Register A,B,C,D,E,H,L
M	Memory location pointed to by H, L
c	Condition flags C,Z,S,P
C_4C_3 =	00 — C(carry) 10 — S(sign) 01 — Z(zero) 11 — P(parity)
<B2>	Byte 2 of a multibyte instruction
<B3>	Byte 3 of a multibyte Instruction
A(m)	Bit m of the A Register
P	Program Counter
XXX	Don't care
STACK	Pushdown register for P storage
SSS	Source Register
DDD	Destination Register
←	Is replaced by
*	Remains the same
Y	Can change

MLA MICROPROCESSOR LANGUAGE ASSEMBLER

RESERVED I/O INSTRUCTIONS

Reserved I/O	Instruction Code	Description
INP0	01 000 001	Read data from serial line
OUT0	01 010 001	Output data to serial line
INP1	01 000 011	Read serial line status register
ION	01 010 011	Enable external event interrupt
IOF	01 010 101	Disable external event Interrupt

PSEUDO INSTRUCTIONS

Pseudo Instruction	Form	Description
PAUSE		End of tape
OCT,HEX,DEC		Radix Control base 8,16,10 Respectively
EXPUNGE		Delete operation code table
OPDEF	Mnemonic; Value; Type	Programmer defined operation code
DATA	N0;N1; . . . Nn	Data statement
BLOCK	Size; Initial; Increment	Block of data set to constants
TEXT	/Literal/	Literal specification
ADDR	A0;A1; . . . An	Address constants

OPERATORS

Operator	Definition
$	End of program
*	Origin statement
=	Direct assignment statement
!	Inclusive-or
+	Addition
−	Subtraction
.	Current pointer
#	Block offset separator
&	Logical and
'	Label designator
;	Pseudo instruction argument separator
/	Comment
↑	High byte selector
<	Begin imbedded literal in text
>	End imbedded literal in text

ERROR MESSAGES

Error	Description
AD	Address out of range
AW	Address warning
DT	Duplicate tag
IC	Illegal character
ID	Illegal definition
IN	Illegal numeric constant
OA	Operand error
OV	Input buffer overflow
PE	Pseudo op expression error
PO	Pushdown stack overflow
PU	Pushdown stack underflow
RD	Redefinition warning
ST	Symbolic table overflow
UO	Undefined operation code
US	Undefined symbol

MLA SWITCH REGISTER ASSIGNMENTS

Bit	Definition
0,1	Pass Designation 01 PASS 1, 10 PASS 2, 11 PASS 3
2	Suppress symbol table listing
9	Output listing to line printer
10	Output spaces instead of tab rubout combination
11	Output on high speed punch

MICROPROCESSOR LANGUAGE EDITOR—MLE

Command	Form	Description
READ	R	Read & append text until form feed
APPEND	A	Add text from keyboard until form feed
LIST	L	List entire buffer
	nL	List line n
	m,nL	List lines m to n, m<n
PUNCH	P	Punch entire buffer
	nP	Punch line n
	m,nP	Punch lines m to n, m<n
FORM FEED	F	Punch trailer & ASCII 214_8
TRAILER	T	Punch trailer
NEXT	N	Perform the functions of P,F,K and R
	nN	Perform the functions of P,F,K and R,n times
KILL	K	Kill text buffer
DELETE	nD	Delete line n
	m,nD	Delete lines m to n m<n
INSERT	I	Insert before line 1 until form feed
	nI	Insert before line n until form feed
CHANGE	nC	Change line n until form feed
	m,nc	Delete lines m to n, replace with text
MOVE	m,n$kM	Move lines m to n before line k
GET	G	Get & list the next line with a tag
	nG	Get & list the next line with a tag start at line n
SEARCH	S	Search buffer for character specified
	nS	Search line n for character specified
	m,nS	Search lines m to n for character specified

MLE—SWITCH OPTIONS

Switch	Description
0	Convert spaces to tabs on input
1	Output tab & rubout for each tab
2	Suppress output
10	High-speed punch output
11	High-speed reader input

SERIAL LINE STATUS WORD

Bit	State	Description
7	1	Error
6	1	Overrun
5	1	Data available
4	1	Transmit buffer empty