AMI 7300 MICROASSEMBLER

ASM73 REFERENCE MANUAL

American Micro-systems, Inc.
Microprocessor Group
3800 Homestead Road
Santa Clara, CA 95051

February 1973

TABLE OF CONTENTS

		Page
I	SYNTAX	1
	A. Operating Characteristics	1
	B. Source Language Format	2
	C. Listing Format	3
	D. Character Set	3
	E. Symbols	4
	F. Expression Constants	4
	G. IMA Instruction Decode Constants	4
	H. Expressions	5
	I. Comments	5
II	STATEMENTS	6
	A. Equate Symbol	6
	B. Set Symbol	6
	C. Polynomial Symbol	6
	D. End of Program	6
	E. Register Control Format	6
	F. Literal Format	7
	G. Branch Unconditional	8
	H. Branch Conditional Format	8
	I. Branch and Load Step Counter Format	9
	J. Branch and Load Stack Pointers Format	11
	K. Polynomial Address Generation	11
	L. Instruction Mapping Array	12
III	PREDEFINED SYMBOLS	15
	A. A, B-D Source (Destination) Fields	15
	B. ALU - I/O Control Field	16
	C. Read-Only Memory Origins	18
	D. Error Flags and Their Meanings	18

		Page
IV	BINARY RECORD FORMAT	19
	A. Microinstruction ROM	19
	B. Starting Address ROM	19
	C. IMA 1 Address ROM	20
	D. IMA 1 Data ROM	20
	E. IMA 2 Address ROM	20
	F. IMA 2 Data ROM	20
V	PUNCH RECORD FORMAT	21
VI	ERROR FLAGS AND THEIR MEANINGS	22
	APPENDIX	23

I SYNTAX

A. OPERATING CHARACTERISTICS

ASM73 is used by loading and executing the FORTRAN load module as required by the host operating environment. Prior to assembly, a control card is read (from the job stream); this card contains several fields as follows:

COL 01-02	The logical unit onto which each deck will be copied for pass 2 processing.
COL 04-05	The logical unit onto which the listing is to be written; zero (blank) suppresses the listing.
COL 07-08	The logical unit onto which the binary output (The SIMIC Simulator Formatted Data) is to be written; zero (blank) suppresses the binary output.
COL 10-11	The logical unit onto which the punch output (The ROM-Mask Formatted Data) is to be written; zero (blank) suppresses the output.
COL 13-14	The logical unit from which the symbolic decks are to be read; zero (blank) defaults to the job stream.
COL 16-17	The maximum number of lines per page; less than 20 defaults to 54.
COL 19-26	The project ID.
COL 28-30	Branch location for machine error. All unused words in the microinstruction ROM will be filled with a branch and mark to this sequential location.

COL 41-72 Up to 16 character pairs which specify special character translation to be performed upon the input data. A card character which matches an odd-column punch is translated into the character in the subsequent even column. The default is no mapping.

B. SOURCE LANGUAGE FORMAT

The ASM73 source language is a series of 80 column card images. Each card contains a comment, a data-generating command, or an assembler directive. A comment card is denoted by an asterisk in column one; the entire card appears in, and may affect, the listing but is otherwise not processed. All other cards (statements) contain one to four fields in columns 1-72:

LABEL This field begins in column one and ends with the first blank column.

OPERATION This field begins in the first non-blank column following the label and ends with the next blank column (or column 72). If no label is present, this field begins with the first non-blank column.

OPERAND This field is considered empty if 16 blank columns follow the operation field; otherwise, it begins with the first non-blank character following the operation field and ends with the next blank column (or column 72).

COMMENTS This field begins with the first non-blank column following the operand field and ends with column 72.

C. LISTING FORMAT

The optional listing consists of a series of lines, each containing the following fields (where applicable) in the sequence specified:

COL 01-06	Error Flags
COL 09-11	Sequential location in hex (see Appendix)
COL 14-16	ROM identification A. Polynomial address for Microinstruction ROM B. SAR for Starting Address ROM C. MA1 for Decode Mapping Array 1 D. MA2 for Decode Mapping Array 2
COL 19-24	Generated data or value (6 hex digits)
COL 29-32	Card number (generated by the assembler)
COL 35-114	Source card image

D. CHARACTER SET

The following characters are permitted in statements; others may appear only in comments:

A-Z, $	Alphabetic
0-9	Numeric
#	Hex Flag
%	Binary Flag
@	IMA Instruction Decode
(BLANK)	Field terminator
,	Parameter separator
(	Begin subexpression
)	End subexpression
*	Multiply, Comment
/	Divide
+	Add
-	Subtract

E. SYMBOLS

A symbol is a string of 1 to 10 alphanumeric characters, the first of which must be alphabetic. Usually, a symbol is defined when it appears in the label field of a statement and is referenced by its appearing in the operand field. Normally a symbol has a value ranging between -2**23 and 2**24-1. A special symbol $ is pre-defined in ASM73 to have the current instruction or directive sequential location value.

F. EXPRESSION CONSTANTS

An expression constant is a self-defining, absolute value which may be used in an operand field expression:

#N	Hexadecimal	(e.g., #10 = 16)
N	Decimal	(e.g., 10 = 10)
%N	Binary	(e.g., %10 = 2)
@L	IMA Instruction Decode	(L is a list of 1s, 0s and Xs)

Expression constants are unsigned with a value ranging from zero to a maximum of 2**24-1.

G. IMA INSTRUCTION DECODE CONSTANTS

The IMA address ROM consists of 16 (18 for IMA 2) input lines which define both the true and the false input to NOR gates for each of the 8 (9 for IMA2) bits to be decoded. In particular each X is converted to 00, each 0 is converted to 01, and each 1 is converted to 10. (e.g., @10XXX0X1 = %1001000000010010)

H. EXPRESSIONS

An expression consists of one or more terms (symbols or constants), each separated from its neighboring terms by an operator. Parentheses permit alteration of the normal evaluation order, which is left-to-right within Hierarchy:

Unary + And -	Hierarchy = 3
Multiply and Divide	Hierarchy = 2
Add and Subtract	Hierarchy = 1

Expressions are terminated by a comma, a blank, or column 72. Restrictions on expressions include:

A division must have a non-zero denominator.

No intermediate value is produced when evaluating the expression may exceed 10**9.

The final expression value must lie between -2**23 and 2**24-1, inclusively.

I. COMMENTS

The assembler examines the second column of each comment card and interprets its contents as follows:

/	Save the card image in the title buffer.
*	Skip to the top of the next page before listing this card
OTHER	No special action

The listing includes the title, the ASM73 revision code, and the page number on the top of each page. Two blanks separate this line from the remainder of the listing on the page.

II STATEMENTS

A. EQUATE SYMBOL

```
LABEL       EQU       E
```

The label value is set to that of the expression. The symbol must not be referenced before this directive unless the expression is evaluatable in pass one. The symbol in the label field cannot be redefined with an equate.

B. SET SYMBOL

```
LABEL       SET       E
```

This directive is identical to the equate except that the symbol in the label field may have been previously defined. Note that the load origin, $, may be altered with a set.

C. POLYNOMIAL SYMBOL

```
LABEL       POLY      E
```

This directive is identical to the equate except that the expression value is converted to a polynomial address before it is assigned to the label.

D. END OF PROGRAM

```
END
```

This directive defines the end of a single program deck.

E. REGISTER CONTROL FORMAT

```
LABEL       OPCODE    E1, E2, E3
```

Where the expressions represent:

E1 - A Source Field

E2 - ALU - I/O Field

E3 - B-D Source/Destination Field

And the OpCode may be any of the following:

RN	Null
RS1	Status Operation 1
RS2	Status Operation 2
RS3	Status Operation 3
RS4	Status Operation 4
RIF	Return to Instruction Fetch (RIF)
RFS1	RIF and Status Operation 1
RFS2	RIF and Status Operation 2
RFS3	RIF and Status Operation 3
RFS4	RIF and Status Operation 4
RCC	CoCall
RMR	Microreturn
RD1	Decode I
RD2	Decode II
RD3	Decode III
RLR	Return from Microsubroutine Loop

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory.

F. LITERAL FORMAT

LABEL OPCODE E1, E2, E3

Where the expressions represent:

E1	Literal value (8 bits)
E2	ALU - I/O Field
E3	B-D Source/Destination Field

And the opcode can be any of the following:

LN Null

LMR Microreturn

LCC CoCall

LLR Return From Microsubroutine Loop

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory.

G. BRANCH UNCONDITIONAL

LABEL OPCODE E1

Where the expression represents:

E1 Branch Address (to be converted to Polynomial Address) and the OpCode can be any of the following:

BU Branch Unconditional

BMU Branch and Mark Unconditional

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory.

H. BRANCH CONDITIONAL FORMAT

LABEL OPCODE E1, E2, E3

Where the expressions represent:

E1 Bit selection (0-7)

E2 A Source Field

E3 Branch Address (converted to Polynomial Address)

And the OpCode can be any of the following:

BR	Branch-On Reset
BS	Branch-On Set
BMR	Branch and Mark on Reset
BMS	Branch and Mark on Set

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory.

I. BRANCH AND LOAD STEP COUNTER FORMAT

LABEL OPCODE E1, E2

Where the expressions represent:

E1	Step count (1-16)
E2	Branch Address (to be converted to Polynomial Address)

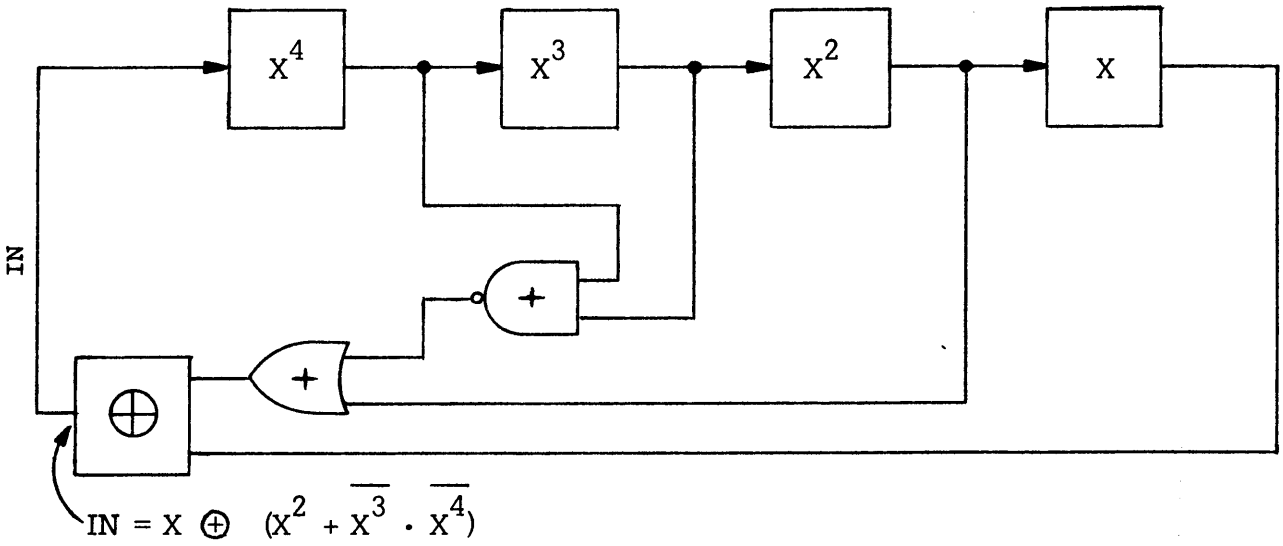
And the OpCode can be any of the following:

BC	Branch and Load Step Count
BMC	Branch and Mark and Load Step Count

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory. The polynomial counts and corresponding decimal equivalent counts are shown on the following page.

STEP COUNTER - MODIFIED $X^4 + X + 1$



BIT 19 X^4	BIT 18 X^3	BIT 17 X^2	BIT 16 X	DECIMAL COUNT	STEP COUNT (Times Around Loop)
0	0	0	0	0	1
1	0	0	0	1	16
0	1	0	0	2	15
0	0	1	0	3	14
1	0	0	1	4	13
1	1	0	0	5	12
0	1	1	0	6	11
1	0	1	1	7	10
0	1	0	1	8	9
1	0	1	0	9	8
1	1	0	1	10	7
1	1	1	0	11	6
1	1	1	1	12	5
0	1	1	1	13	4
0	0	1	1	14	3
0	0	0	1	15	2

J. BRANCH AND LOAD STACK POINTERS FORMAT

LABEL OPCODE E1, E2

Where the expressions represent:

E1 Stack Pointer ROM Address. Decimal values 1 through 16 will address RALU chip Stack Pointer ROM locations 0 through 15 respectively.

E2 Branch Address (to be converted to Polynomial Address)

and the OpCode can be any of the following:

BP Branch and Load Stack Pointer

BMP Branch and Mark and Load Stack Pointer

The label is assigned the current sequential location value. The expression values are right justified and truncated where required.

These directives are restricted to use within the Microinstruction Read-Only Memory.

K. POLYNOMIAL ADDRESS GENERATION

PADR E1, E2,...,EN

This directive is identical to the data directive except that the expression values are converted to Polynomial Addresses before being stored.

This directive is restricted to use within the Starting Address Read-Only Memory.

L. INSTRUCTION MAPPING ARRAY

IMA E1, E2, E3

Where the expressions represent:

E1 IMA Instruction Decode

E2 Mapping Address (converted to Polynomial Address)

E3 Instruction Inhibit

The operand expressions are evaluated and stored in the next two memory locations. The first location contains E1, and the second location contains E3*2**10+E2. Values are right justified and truncated where required.

This directive is restricted to use within the Instruction Mapping Arrays. In the case of Mapping Array 2 a twelfth bit is set in the second location (2**11); this bit corresponds to the Select line in Mapping Array 2. Although only 25 locations are represented for each array, the Microassembler provides memory space as if each array were 80 locations. Array overflow are detected on the basis of 25 locations.

Figures 1 and 2 are diagrams of the Instruction Mapping Array and Microinstruction ROM sections of a typical breadboard which uses the Intel 1701 ROMs in conjunction with the Microassembler. These ROMs must be wired as shown in order to be compatible with Assembler paper tapes.

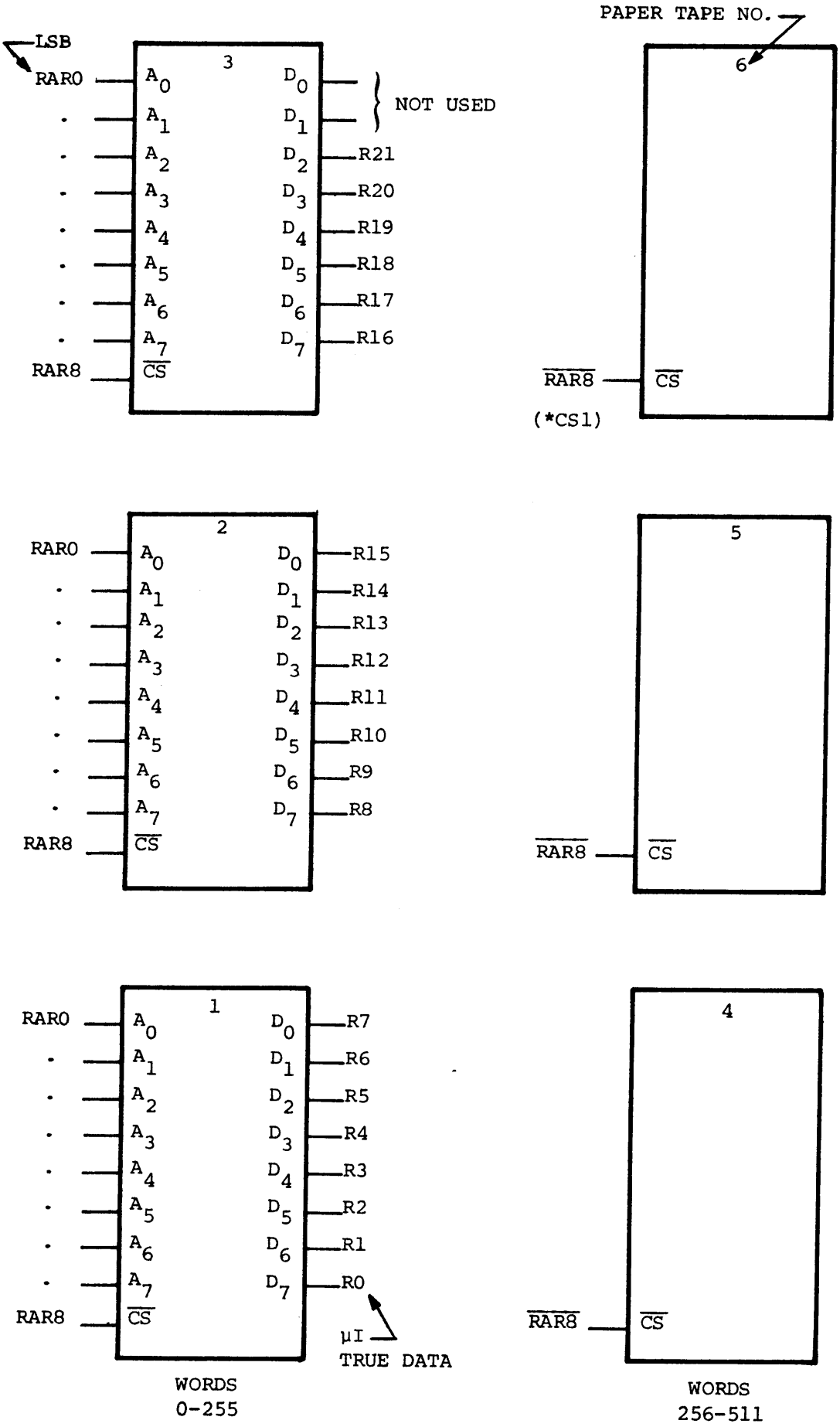


FIGURE 1 - MICROINSTRUCTION ROM

D1-DECODE 1; D2-DECODE 2; D3-DECODE 3

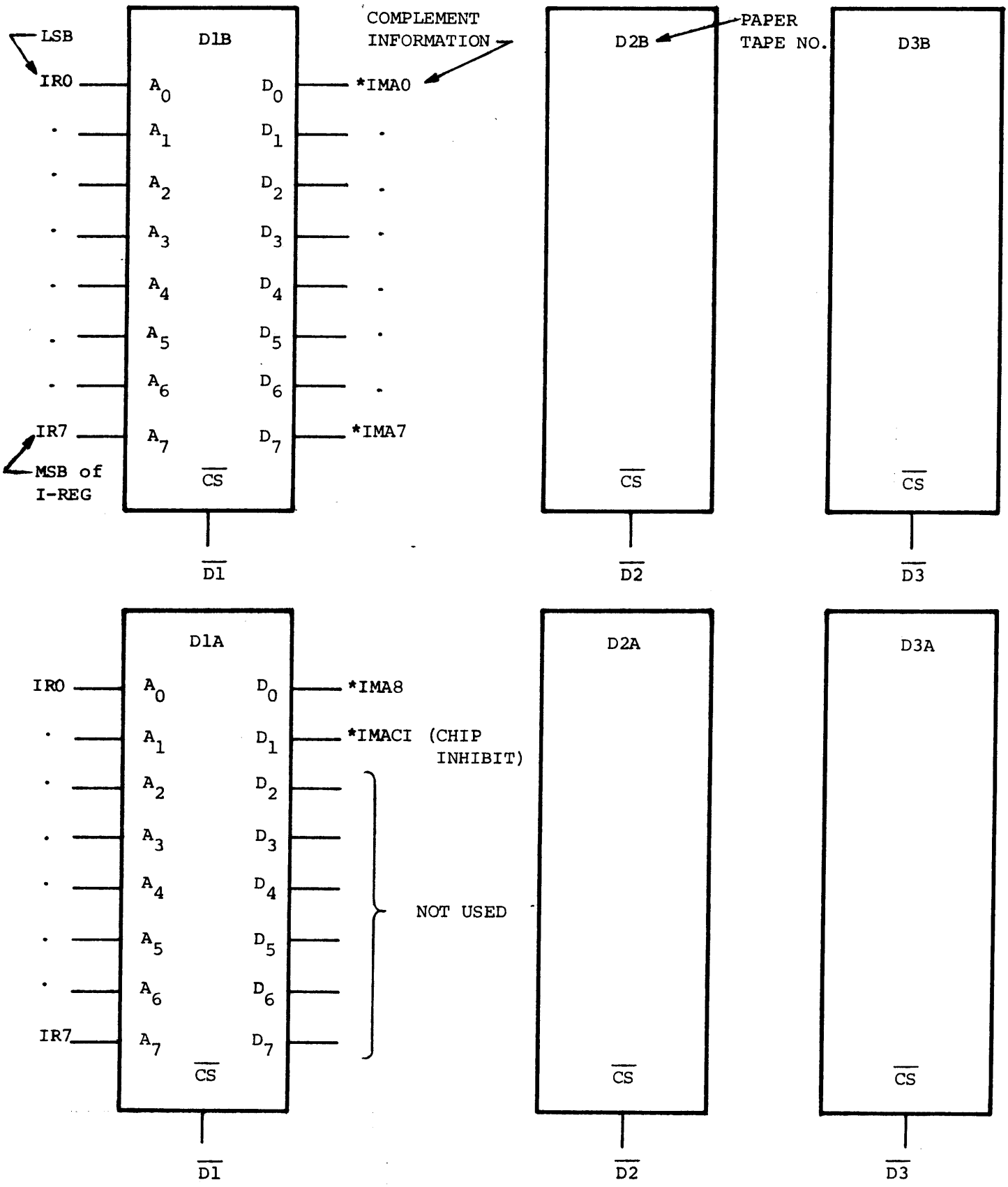


FIGURE 2 - 7300 INSTRUCTION MAPPING ARRAY

III PREDEFINED SYMBOLS

A. A, B-D SOURCE (DESTINATION) FIELDS

SYMBOL	HEX CODE	FUNCTION
R0	00	General Register 0
R1	01	General Register 1
R2	02	General Register 2
R3	03	General Register 3
R4	04	General Register 4
R5	05	General Register 5
R6	06	General Register 6
R7	07	General Register 7
R8	08	General Register 8
R9	09	General Register 9
R10	0A	General Register 10
R11	0B	General Register 11
R12	0C	General Register 12
R13	0D	General Register 13
R14	0E	General Register 14
R15	0F	General Register 15
SK1	10	Stack 1
SK1I	11	Stack 1, Increment Pointer
SK1D	12	Stack 1, Decrement Pointer
SP1	13	Stack Pointer 1
SK2	14	Stack 2
SK2I	15	Stack 2, Increment Pointer
SK2D	16	Stack 2, Decrement Pointer
SP2	17	Stack Pointer 2
FS	19	Final Status

SYMBOL	HEX CODE	FUNCTION
TS	1A	Temporary Status
PALU	1B	Previous ALU Result
DEL	1C	Data Exchange Low
M1	1D	2s Complement -1
LSP	1E	Load Stack Pointers (Format 1 Only)
Z	1F	Zero

Note that LSP (Code 1E) will cause the Stack Pointers to be loaded for Format 1 Microinstructions. If a Format 3 Branch and Load Stack Pointers are performed, it should use the BP and BMP Codes. The two cases are summarized below:

FORMAT 1

```
LABEL    OPCODE    LSP, E2, E3
```

FORMAT 3

```
LABEL    BP        E1, E2
LABEL    BMP       E1, E2
```

B. ALU - I/O CONTROL FIELD

SYMBOL	HEX CODE	FUNCTION
TAD	00	A + B
ADI	01	A + B + 1
ADTC	02	A + B + TC
ADC	03	A + B + C
A	08	A + 0
AI	09	A + 0 + 1
ATC	0A	A + 0 + TC
AC	0B	A + 0 + C

C. READ-ONLY MEMORY ORIGINS

SYMBOL	HEX CODE	FUNCTION
$MIROM	0	Microinstruction ROM origin
$SADR	400	Starting Address ROM origin
$MA1	4A0	Mapping Array 1 origin
$MA2	540	Mapping Array 2 origin

D. ERROR FLAGS AND THEIR MEANINGS

C	Command (No OpCode, illegal OpCode)
D	Duplicate label definition
E	Expression syntax
L	Label (illegal character, missing, ignored)
N	A single numeric field exceeds 2**24-1
O	Operand (too many, missing)
P	Parenthesis (extra right or left, or too many levels)
R	Restriction (directive cannot be used in this ROM)
S	Symbol exceeds ten characters
T	Truncation (excessive value formed in an expression)
U	Undefined symbol in operand field
V	Assembler table overflow
Z	Attempted division by zero
$	Location counter (overflow, illegal set)

IV BINARY RECORD FORMAT

The optional binary output consists of a series of records, each containing data to be loaded into one of the six ROMs as described below.

A. MICROINSTRUCTION ROM

CHAR	CONTENTS
01	Blank
02-04	Load location of the next word (ROM binary order)
06-11	6-digit hex value of next word
13-18	6-digit hex value of next word + 1
20-25	6-digit hex value of next word + 2
27-32	6-digit hex value of next word + 3
34-39	6-digit hex value of next word + 4
41-46	6-digit hex value of next word + 5
48-53	6-digit hex value of next word + 6
55-60	6-digit hex value of next word + 7
67-71	ROM ID, MIROM
73-80	Project ID

B. STARTING ADDRESS ROM

CHAR	CONTENTS
01	Blank
02-04	#200 (Control Store Address of Starting Addresses)
06-11	6-digit hex value for RESET
13-18	6-digit hex value for Interrupt 1
20-25	6-digit hex value for Interrupt 2
27-32	6-digit hex value for Interrupt 3
34-39	6-digit hex value for Return to Instruction Fetch
67-71	ROM ID, MIROM
73-80	Project ID

C. IMA 1 ADDRESS ROM

CHAR	CONTENTS
01	Blank
02-04	Card number
06-64	Ten 5-digit hex values separated by one blank
67-71	ROM ID, IMA1A
73-80	Project ID

D. IMA 1 DATA ROM

CHAR	CONTENTS
01	Blank
02-04	Card number
06-64	Ten 5-digit hex values separated by one blank
67-71	ROM ID, IMA1B
73-80	Project ID

E. IMA 2 ADDRESS ROM

CHAR	CONTENTS
01	Blank
02-04	Card number
06-64	Ten 5-digit hex values separated by one blank
67-71	ROM ID, IMA2A
73-80	Project ID

F. IMA 2 DATA ROM

CHAR	CONTENTS
01	Blank
02-04	Card number
06-64	Ten 5-digit hex values separated by one blank
67-71	ROM ID, IMA2B
73-80	Project ID

The last record in binary output file contains @@ in columns one and two.

V PUNCH RECORD FORMAT

The optional punch output consists of twelve groups of records, each containing data for one of twelve breadboard ROMs. The format of the records within the groups are as follows:

REC	CHAR	CONTENTS
1	1-80	Blank
2	1-5	Blank
	6-8	Chip ID
	11-16	ROM ID
3	1-80	Rubout
4-67	1-3	Blank
	4-5	2-digit Hex Address or next byte
	11-65	Four bytes consisting of a leading B, eight P or N characters and a trailing F separated by five blanks
68	1-80	Rubout
69	1-80	Blank

Each of the twelve ROMs contains the following:

CHIP ID	ROM	ID	CONTENTS
A02	D-1	A	Decode 1 Array, left half (Bits 8, CI)
A02	D-1	B	Decode 1 Array, right half (Bits 7-0)
A02	D-2	A	Decode 2 Array, left half (Bits 8, CI)
A02	D-2	B	Decode 2 Array, right half (Bits 7-0)
A02	D-3	A	Decode 3 Array, left half (Bits 8, CI)
A02	D-3	B	Decode 3 Array, right half (Bits 7-0)

See Figure 2 for a complete definition of Intel ROM designations which equate to the ROM IDs shown above.

CHIP ID	ROM	ID	CONTENTS
A03	ROM	3	MIR words 0-255, Bits 16-21
A03	ROM	2	MIR words 0-255, Bits 8-15
A03	ROM	1	MIR words 0-255, Bits 0-7
A03	ROM	6	MIR words 256-511, Bits 16-21
A03	ROM	5	MIR words 256-511, Bits 8-15
A03	ROM	4	MIR words 256-511, Bits 0-7

In the Decode Arrays a bit value of zero is translated to a P and a value of one to an N while in the Microinstruction Memory the opposite is true, a zero translates to an N and a one translates to a P. Also, in the Decode Arrays the most significant bit of the 8-bit byte is outputted first while, in the Microinstruction Memory, the least significant bit is outputted first. See Figure 1 for a complete definition of the Intel ROM designations which relate to the table shown above.

VI ERROR FLAGS AND THEIR MEANINGS

C	Command (no OpCode, illegal OpCode)
D	Duplicate label definition
E	Expression syntax
L	Label (illegal character, missing, ignored)
N	Single numeric field exceeds 2**24-1
O	Operand (too many, missing)
P	Parenthesis (extra right or left, or too many levels)
S	Symbol exceeds ten characters
T	Truncation (excessive value formed in an expression)
U	Undefined symbol in operand field
V	Assembler table overflow
Z	Attempted division by zero
$	Location counter (overflow, illegal set)

APPENDIX

The assembler assumes the following arbitrary assignment of sequential addresses for the four different Read-Only Memories.

ADDRESS	BITS	CONTENTS
0-1FF	0-21	Microinstruction ROM
400	0-9	Initialize Starting Address (reset)
401	0-9	Interrupt 1
402	0-9	Interrupt 2
403	0-9	Interrupt 3
404	0-9	Instruction Fetch Starting Address
405	0-9	Not Used
406		Unassigned
420-471		Reserved for simulator use
472-49F		Unassigned
4A0,4A2,...	0-15	Input lines for Mapping Array 1
4A1,4A3,...	0-9	Mapping Address for Mapping Array 1
	10	Instruction Inhibit
540,542,...	0-17	Input lines for Mapping Array 2
541,543,...	0-9	Mapping Address for Mapping Array 2
	10	Instruction Inhibit
5E0		Unassigned