

8092 TELEPROGRAMMER

PROGRAMMING

TRAINING MANUAL

- - -

CORPORATE MARKETING

PROGRAMMING AND APPLICATIONS TRAINING DEPARTMENT

MINNEAPOLIS, MINNESOTA

JUNE 1, 1966

Copyright 1966, Control Data Corporation  
Printed in the United States of America  
Publication No. 60183900  
Revision - July 1967

## CONTENTS

	Page
I. General Characteristics	1
II. Basic Concepts in Programming the Teleprogrammer	
A. Features	2
B. Word Format	3
C. Instruction Repertoire	6
D. Block Diagram	8
III. Central Processor Commands	
A. Load Instructions	9
B. Store Instructions	18
C. Jump Instructions	26
D. Shift Instruction	32
E. Arithmetic Instructions	33
F. Logical Instructions	42
G. Input/Output Instructions	49
H. Control Instructions	58
IV. Operation Console	63
V. External Equipment	70
A. 8096 Teletype Model 33 Page Printer	71
B. 350 Page Tape Reader	72
C. 8098 Tally Reader	74
D. BRPE-11 Paper Tape Punch	75
E. 161 I/O Typewriter	76

	Page
F. 8093 Magnetic Tape Synchronizer	79
G. 162 Magnetic Tape Synchronizer	87
H. 166-2 Line Printer	93
I. 167-2 Card Reader	98
J. 170 Card Punch Control Unit	103
K. 177 Card Reader Controller	105
L. 8094 Peripheral Adaptor	110
M. 8060 Series Digital Communications Terminals	111
N. 8095 Record Transmission Control Panel	112
O. 8022/915 Page Reader	117
Appendix A - TAS Instruction Repertoire	A-1
Appendix B - MOD 33 Codes	B-1
Appendix C - External BCD Codes	C-1

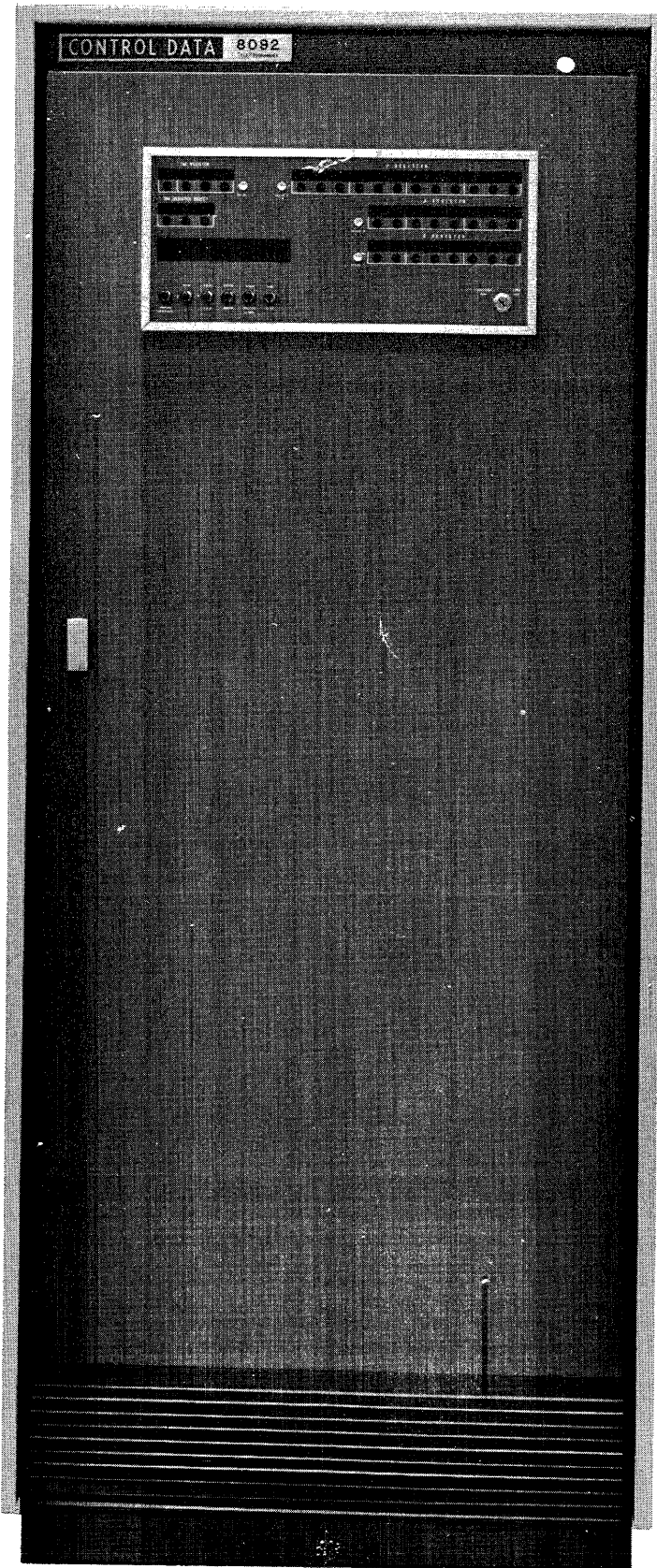
TABLE OF CONTENTS  
(INSTRUCTIONS)

<u>Instruction</u>	<u>Mnemonic</u>	<u>Octal</u>	<u>Page</u>
Load	LD-		
No Address	LDN	20	9
Memory	LDM	21	10
Indirect	LDI	22	11
Load Complement	LC-		
Memory	LCM	25	13
Indirect	LCI	26	14
Store	ST-		
Memory	STM	41	18
Indirect	STI	42	19
Add	AD-		
No Address	ADN	30	32
Memory	ADM	31	33
Indirect	ADI	32	34
Subtract	SB-		
No Address	SBN	34	35
Memory	SBM	35	36
Indirect	SBI	36	37
Replace			
Add one	RAO	55	40
Add	RAM	51	39

Shift	SHA	01	31
Logical	L--		
Logical Sum	LS-		
No Address	LSN	14	45
Memory	LSM	15	46
Indirect	LSI	16	47
Logical Product	LP-		
No Address	LPN	10	41
Memory	LPM	11	42
Indirect	LPI	12	43
Tag Register			
A to tag	ATT	02	21
Tag to A	TTA	03	16
Jumps	--P		
Zero	ZJP	60	26
Non-Zero	NZP	61	27
Positive	PJP	62	28
Negative	NJP	63	29
Unconditional	UJP	64	30
Control			
Clear Interrupt	CIL	013 or 113	60
Error Halt	ERR	00	62
Halt	HLT	77	62

## Input/Output

External Function	EXF	75	58
Normal			
Input to A	INA	76	56
Input to Memory	INN	72	49
Output No Address	OTN	74	57
Output from Memory	OUT	73	52
Buffer			
A to BER	ABR	04	22
A to BXR	ABX	05	24
BER to A	BER	06	17
Clear Buffer Controls	CBC	07	61
Input to Memory	IBI	70	54
Output from Memory	IBO	71	55



8092 TELEPROGRAMMER



I. 8092 Teleprogrammer General Characteristics

A. The CONTROL DATA\* 8092 Teleprogrammer is a highly flexible and versatile stored program processor specially designed as a high speed buffer memory system for use in a variety of data communication applications.

Among the more important features are the following:

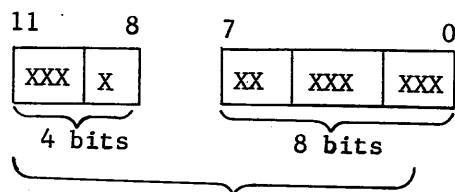
1. Stored program
2. Parallel mode of operation
3. 8-bit word length
4. 2048 words of core storage - with 4096 (optional)
5. 1 normal I/O channel (8 bits)
6. 1 buffer I/O channel (8 bits)
7. Versatile instruction repertoire of 42 instructions
8. 3 auxiliary tag registers of 4 bits each
9. No address, memory, and indirect ADDRESSING MODES
10. 4 Interrupts
11. 12 bit external function address codes
12. 7 internal program registers
13. Physical size: height, 68 inches; width, 34 inches; depth, 30 inches
14. Storage reference cycle time of 4 microseconds
15. The ability to use the OSAS or OSAS-A assembler for those who have a 160 or 160-A computer

\*Registered Trademark of Control Data Corporation

## II. Basic Concepts in Programming the Teleprogrammer

A. The Teleprogrammer has some unique features for programming. Most of these center around the word length of 8 bits. In order to carry addresses for 4096 locations, 12 bits are required ( $2^{12} = 4096$ , where highest address is  $2^{12} - 1$ ). To provide for 12 bits, the Teleprogrammer makes use of three 4-bit tag registers (tag registers 1, 2 and 3). Combining the 8 bits in the instruction word to the 4 bits in the tag register causes a split in the second octal digit from the left. This is indicated below:

Tag Reg. Designation



Complete 12 bit address  
XXX XXX XXX XXX

The 8-bit word length will be represented as two full octal digits and one quartic digit (the leftmost 2 bits). The tag registers will be generally represented as one full octal digit (on the left) and a single bit (0 or 1) on the right.

B. Instruction Word Format

The Teleprogrammer operates on two word instructions. The instructions are contained in two sequential storage locations. The first word contains the Function Code (F), in the lower two octal digits and the Tag Register designator ( $F^1$ ), in the upper quartic digit. The second word of the instruction holds: an 8 bit operand, or a partial address which is 8 bits. Three modes of operation are possible in the 8092; NO ADDRESS, MEMORY, and INDIRECT.

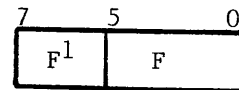
Examples:

NO ADDRESS MODE

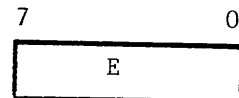
Memory Location

Contents

P



P+1



$F^1$  = TAG REGISTER DESIGNATOR (2 bits)

F = Function Code (6 bits)

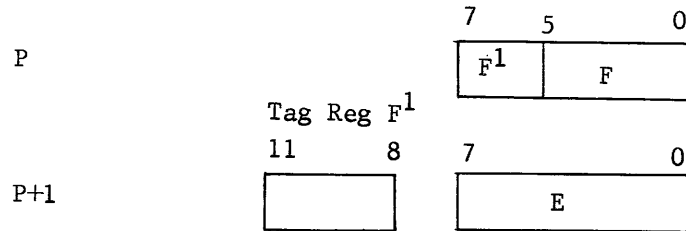
E = Operand (8 bits)

$F^1$  would equal 0, since there is no Auxiliary Tag Register used in this mode. The operand must contain 3 digits in the octal range of 000 through 377.

MEMORY ADDRESS MODE

Memory Location

Contents



F<sup>1</sup> = TAG REGISTER DESIGNATOR (2 bits)

F = Function Code (6 bits)  
Partial Execution

E = Address (8 bits)

F<sup>1</sup> could equal 0, 1, 2, or 3. If F<sup>1</sup> = 0, the operand address is fully contained in the second word of the instruction. If F<sup>1</sup> = 1, 2, or 3 the operand address is fully contained in the designated Tag Register and the second word. Therefore, we obtain a 12 bit address, of which is 8 bits of the second word and 4 bits of the Tag Register.

INDIRECT ADDRESS MODE

Memory Location

Contents

	7	5	0
P	F <sup>1</sup>		F
P+1	E		

F<sup>1</sup> = TAG REGISTER DESIGNATOR (2 bits)

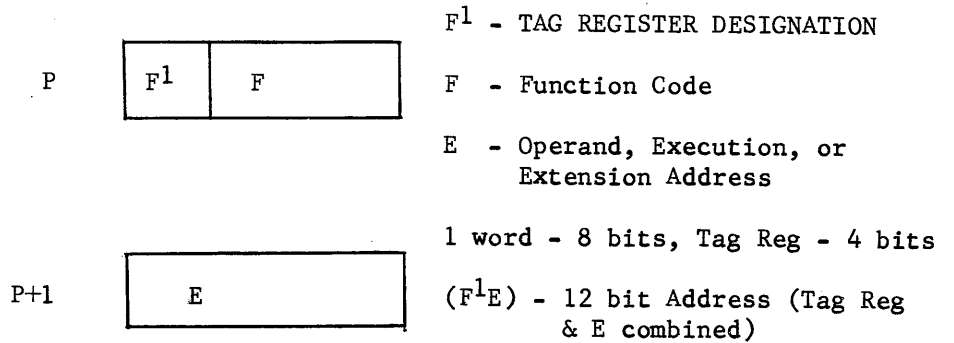
F = Function Code (6 bits)

E = Indirect Address (8 bits)

F<sup>1</sup> could equal 0, 1, 2, or 3. One of the first 256<sub>10</sub> core locations designated by E is referenced for the lower 8 bits of the operand address. The upper 4 bits of the operand address comes from the designated Tag Register, thus giving us a 12 bit operand address. If the Tag Register designation were 0, the upper 4 bits of the 12 bit operand address would be 0's.

C. Instruction Repertoir

Normal Instruction 2 Words



SYMBOLS

E = 2nd word 8 bits	N NO ADDRESS X = E NI P+2
X = Operand	
Y = Address to be used	M MEMORY Y = (Tag F <sup>1</sup> E) NI P+2
( ) = Contents	
NI = Next Instruction	I INDIRECT Y = (Tag F <sup>1</sup> (E)) NI P+2

cycle time = 4 usec.

ARITHMETIC/LOGICAL

	N	M	I	
LP	10	11	12	Logical Product
LS(SC)	14	15	16	Logical Sum
LD	20	21	22	Load A
LC		25	26	Load Comp.
AD	30	31	32	ADD
SB	34	35	36	Subtract
ST		41	42	Store
RA*		51		Replace Add
AO*		55		Replace Add one
	2	3	4	Cycles

\* add 1 cycle

SENSE JUMPS

	P	
ZJ	60	Zero Jump
NZ	61	Non-Zero Jump
PJ	62	Positive Jump
NJ	63	Negative
UJ	64	Unconditional
	2	Cycles (if Jump made)
	1	Cycle otherwise

SHIFT 1 cycle

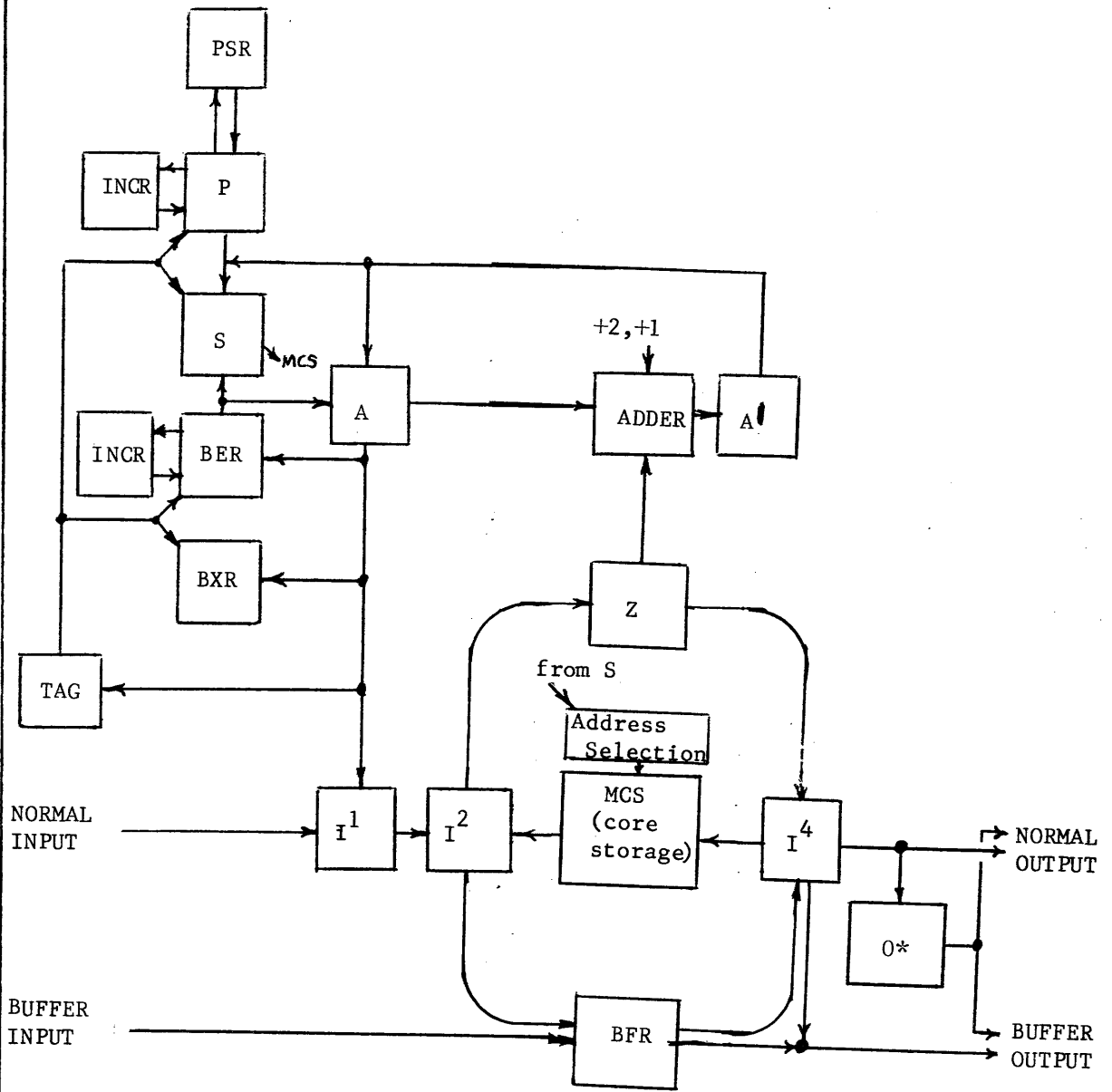
SHA 01 A left shift one TAG REG 1 cycle

ATT 02 A to Tag Reg

TTA 03 Tag Reg to A

INPUT/OUTPUT		CONTROLS	
Normal		Buffer	
INN	72	IBI	70
	Initiate Input from (Tag		Initiates
	F <sup>1</sup> ) (P+1) to (Tag F <sup>1</sup> )		Buffer
	(P+2)-1, NI P+3		Input
	**		NI P+2.***
OUT	73		2 Cycles
	Initiate Output from (Tag	IBO	71
	F <sup>1</sup> ) (P+1) to (Tag F <sup>1</sup> )		Initiate
	(P+2)-1, NI P+3		Buffer
	**		Output
INA	76		NI P+2.***
	Input to A		2 Cycles
	2 Cycles	ABR	04
OTN	74		(A) to
	Output X		BER
	X = E		NI P+2.***
	2 Cycles		1 Cycle
		ABX	05
			(A) to BXR
			NI P+2.***
			1 Cycle
		BER	06
			BER to A
			1 Cycle
**Cycle Time 3+2(X+1)+Terminate time X = No. of words		*** If Buffer Active Jump to Y, Y = (Tag F <sup>1</sup> ) (P+1)	
		EXF	75
			External Fun.
			Code 2nd & 3rd
			words
			UU at 2nd
			TT at 3rd
			3 Cycles
		CIL	13
			Clear Interrupt
			013
			113
			1 Cycle
		CBC	07
			Clear Buffer
			Controls
			1 Cycle
		ERR	00
			Error Stop
			1 Cycle
		HLT	77
			Halt
			1 Cycle
		UUTT	
			UU - Unit designator
			TT - Function designator

D. 8092 TELEPROGRAMMER BLOCK DIAGRAM



\* upper 6 bits of Function Code

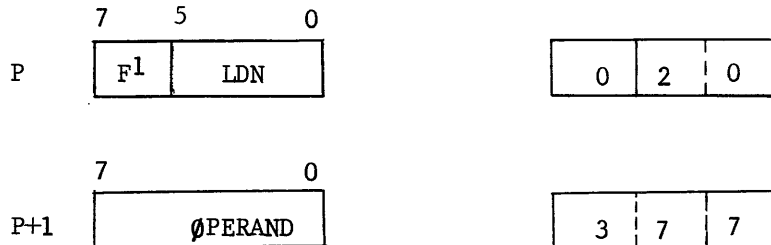


### III. 8092 Teleprogrammer Central Processor Commands

#### A. Load Instructions

##### 1. Load A - No Address Mode

8 usec

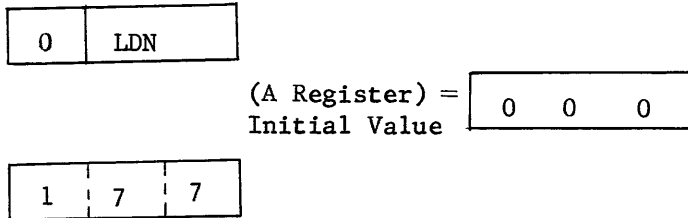


F<sup>1</sup> not used

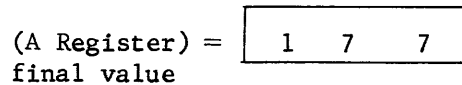
#### Description:

Load the A Register with the second word of the instruction. Octal numbers 000 through 377 can be entered into A by this instruction, RNI @ P+2

#### Example:

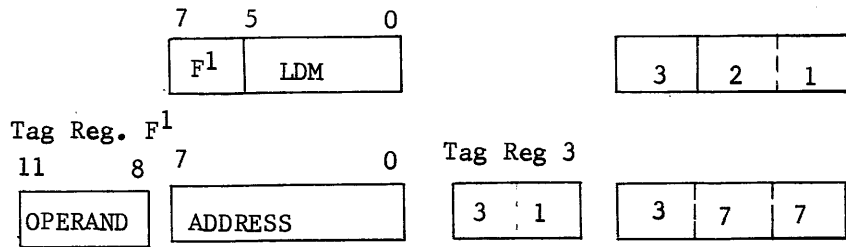


After the execution of the above command the A Register would be:



2. Load A - Memory Mode

12 usec.

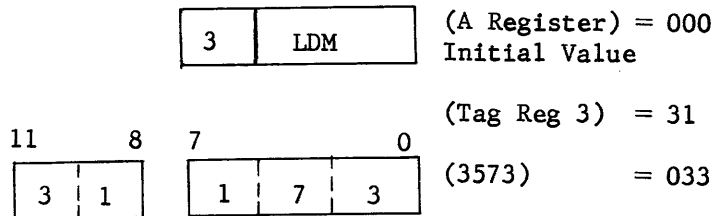


Description:

Load the A register with the contents of memory address whose lower eight bits are given in the second instruction word and whose upper four bits are contained in the designated Auxiliary Tag Register.

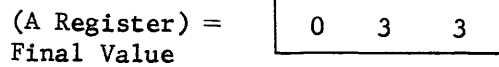
RNI @ P+2.

Example:



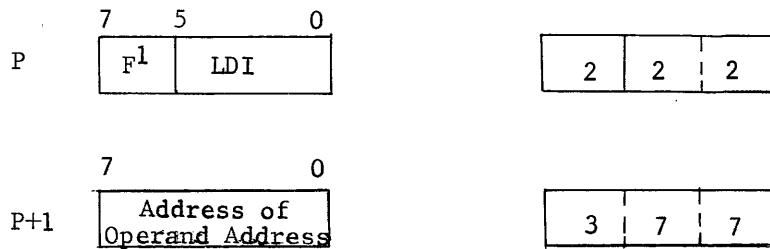
3573 = OPERAND ADDRESS

After the execution of the above command the contents of 3573 would be loaded into the A Register.



3. Load A - Indirect Mode

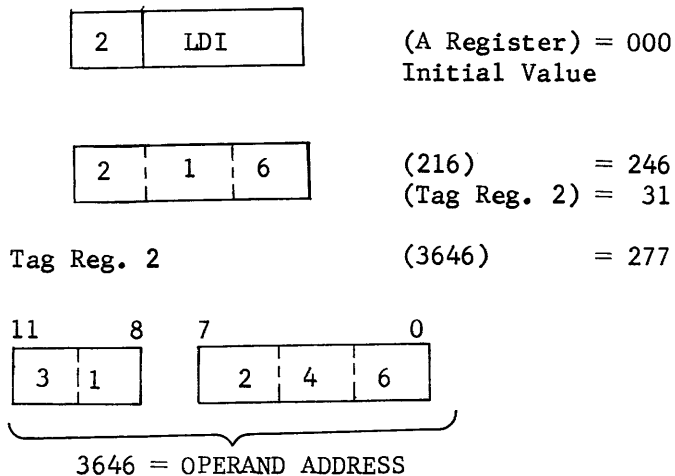
16 usec.



Description:

Load the A Register with the contents of the Address whose lower 8 bits are contained in one of the first  $256_{10}$  ( $400_8$ ) addresses, and whose upper 4 bits are contained in a designated Auxiliary Tag Register. The location in the core (one of the first  $256_{10}$  addresses) is given in the second instruction word. The Auxiliary Tag Register is indicated in the first word bits 6 and 7. RNI @ P+2.

Example:



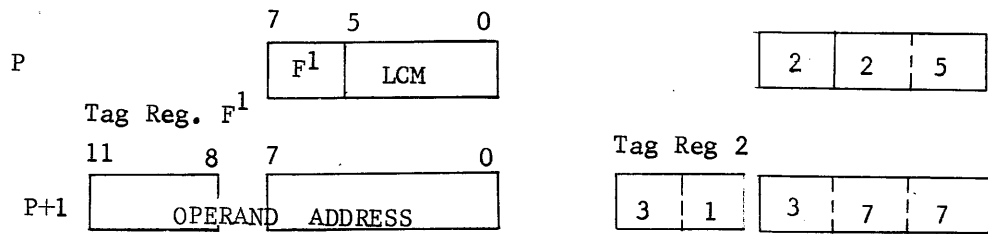
After the execution of the above command, the contents of 3646 is loaded into the A Register. (NOTE: The 1 bit of Tag Reg. 2 and the quartic digit of address 216, form the bits, 110, which gives the octal digit 6.)

(A Register) = 

2	7	7
---	---	---

  
Final Value

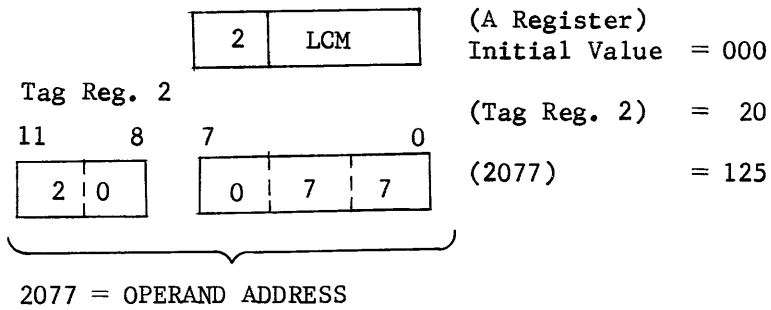
4. Load Complement to A - Memory Mode 12 usec.



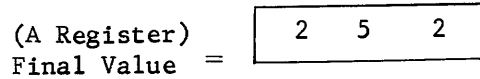
Description:

Load the A Register with the complement of the contents of the memory address whose lower 8 bits are given in the second instruction word and whose upper 4 bits are contained in the designated Auxiliary Tag Register. RNI @ P+2.

Example:

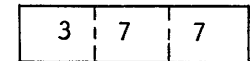
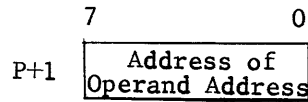
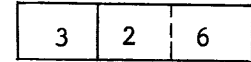
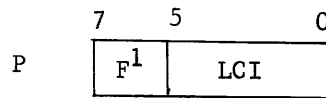


After the execution of the above command the complement of the contents of 2077 would be loaded into the A Register.



5. Load Complement to A - Indirect Mode

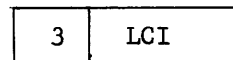
16 usec



Description:

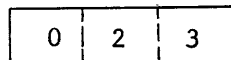
Load the A Register with the complement of the contents of the address whose lower 8 bits are contained in one of the first  $256_{10}$  ( $400_8$ ) addresses and whose upper 4 bits are contained in a designated Auxiliary Tag Register. The location in the core (one of the first  $256_{10}$  addresses) is given in the second instruction word. The Auxiliary Tag Register is indicated in the first word bits 6 and 7. RNI @ P+2.

Example:



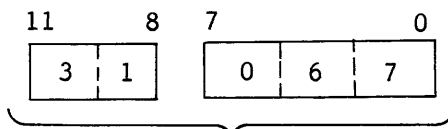
(A Register)  
Initial Value = 000

(023) = 067  
(Tag Reg. 3) = 31



(3467) = 053

Tag Reg 3



3467 = OPERAND ADDRESS

After the execution of the above command, the complement of the contents of 3467 would be loaded into the A Register.

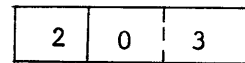
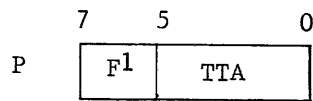
NOTE: The 1 bit of Tag Reg 3 and the quartic digit at address 023, form the bits, 100, which gives the octal digit 4. Also the complement of the quartic digit 0, at address 3467 is 3; whereas, the complements of the octal digits 5 and 3 are respectively 2 and 4.

(A Register)  
Final Value = 

3	2	4
---	---	---

6. Tag Register to A

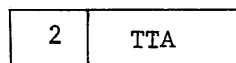
4 usec.



Description:

Load the contents of the designated Auxiliary Tag Register into the A - Register. With the A - Register being 8 bits, and the Tag Register being 4 bits, zero's are packed into the upper 4 bits of A. RNI @ P+1.

Example:



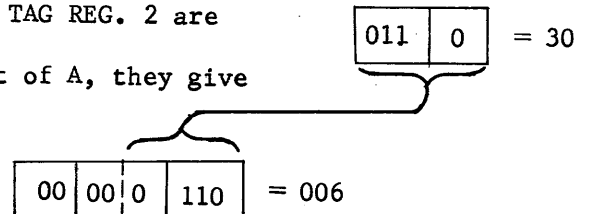
(A Register)

Initial Value = 000

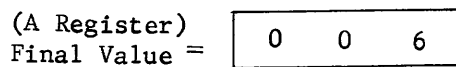
(Tag Reg. 2) = 30

After the execution of the above command the contents of the designated Tag Register would be loaded into the A Register.

NOTE: The four bits of TAG REG. 2 are when packed to the right of A, they give the following results:



A would be:





7. Buffer Entrance Register to A

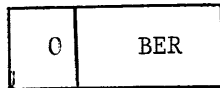
4 usec.



Description:

Load the A Register with the lower 8 bits of the Buffer entrance Register. RNI @ P+1.

Example:



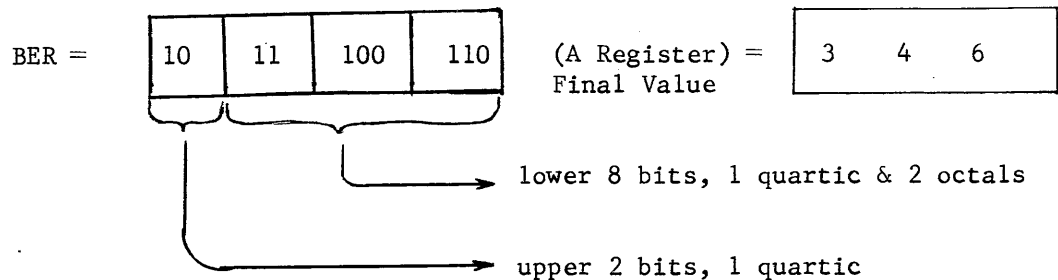
(A Register)  
Initial Value = 000

(BER ) = 1346  
10 bits

After the execution of the above command the lower 8 bits of the BER would be loaded into the A-Register.

NOTE: On this instruction, the lower 8 bits, (1 quartic and 2 octals) of the BER, are transferred into the A Register.

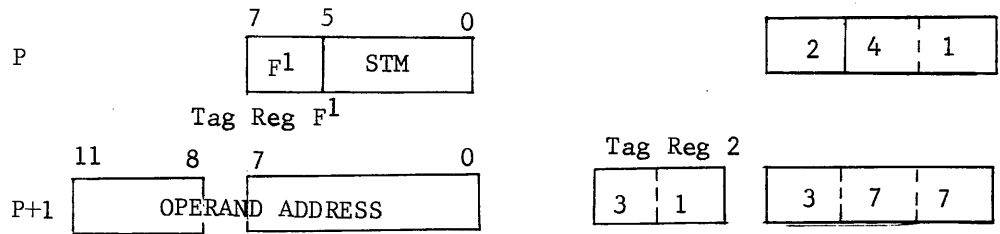
The upper 2 bits (1 quartic) are not transferred.



B. Store Instructions

1. Store A - Memory Mode

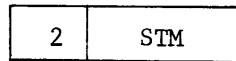
12 usec.



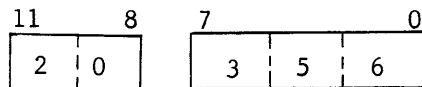
Description:

Store the contents of the A-Register into the the location whose address is equivalent to the combined contents of the designated Tag Register and the second word of the instruction, RNI @ P+2.

Example:



(A Register)  
Initial Value = 155



(Tag Reg. 2) = 20  
(2356 ) = 377  
Initial Value

2356 = OPERAND ADDRESS

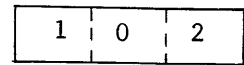
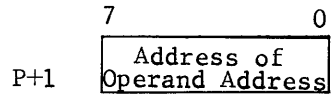
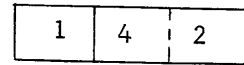
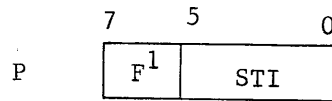
After the execution of the above command, the contents of A will replace the value 377 in Memory Location 2356. The A Register remains unchanged. 2356 would be:

(2356)  
Final Value = 

1	5	5
---	---	---

2. Store A - Indirect Mode

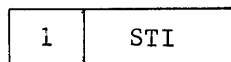
16 usec.



Description:

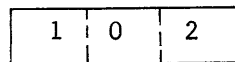
Store the contents of the A Register into the location whose address is equivalent to the combined contents of the designated Tag Register and the contents of one of the first 256<sub>10</sub> core registers. The exact location of one of these 256 registers is given, through its address, in the second instruction word. RNI @ P+2.

Example:



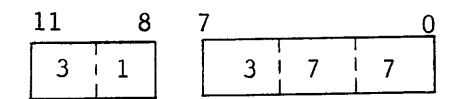
(A Register) = 037  
Initial Value

(102 ) = 377  
(Tag Register) = 31



(3777 ) = 377  
Initial Value

Tag Reg 1



3777 = OPERAND ADDRESS

After the execution of the above command, the contents of A will replace the value 377 in Memory Location 3777. The A Register remains unchanged. 3777 would be:

(3777 ) = 

0	3	7
---	---	---

  
Final Value

3. A to Tag Register

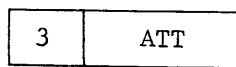
4 usec.



Description:

Transfer the lower 4 bits (2 quartic digits) of the A Register into the designated Auxiliary Tag Register. The upper 4 bits of the A Register are ignored. RNI @ P+1.

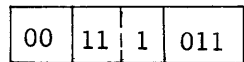
Example:



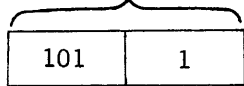
(A Register ) = 073  
Initial Value

(Tag Reg. 3 ) = 000  
Initial Value

After the execution of the above command the lower 4 bits of the contents of A are sent to the designated Tag Register. NOTE: The contents of A are:



= 073. When packed in the Tag Register, the lower 4 bits give the result:



= 51. Tag Register 3 would be:

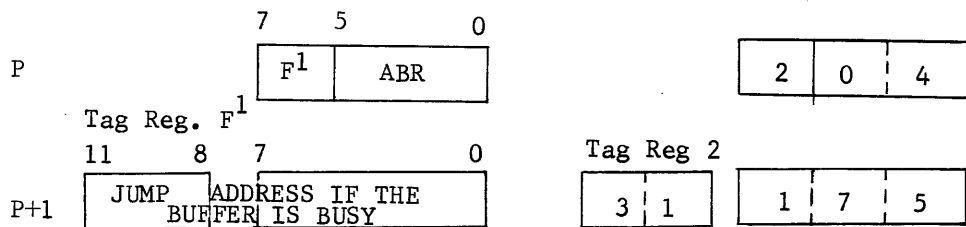
(Tag Reg. 3) = 

5	1
---	---

  
Final Value

4. A to Buffer Entrance Register

4 usec. (no jump)  
8 usec. (jump )



Description:

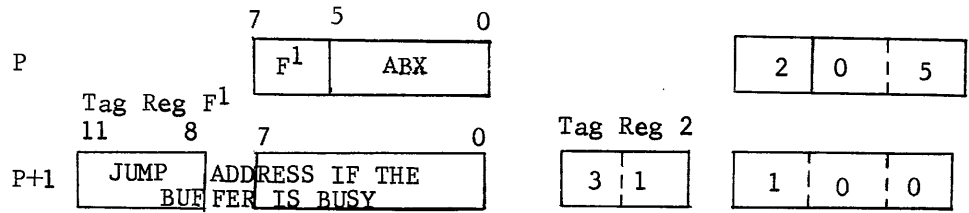
Transfer the contents of A to the lower 8 bit positions of the Buffer Entrance Register (BER). The Buffer Entrance Register is a 10 bit register; therefore, the rightmost 2 bits of Tag Register 3 becomes the 9th and 10th bits of the BER, the upper 2 bits of Tag Register 3 are referenced for bits 11 and 12 to complete the ability to reference 4K. If the Buffer is busy, a jump occurs to the combined address contained in the second word of the instruction set and the designated Tag Register. If the buffer is not busy, control goes to the next instructional set @ P+2.

Previous to executing this instruction Tag Register 3 should be loaded with the effective upper 4 bits of the starting address, and then the BER with the lower 8 bits. The Tag Register designated in the instruction word should also be prepared in case the buffer is busy.



5. A to Buffer Exit Register

4 usec. (No jump)  
8 usec. (jump )



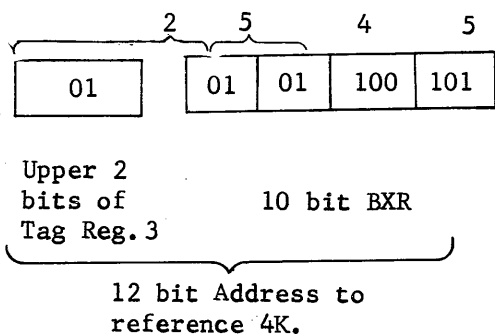
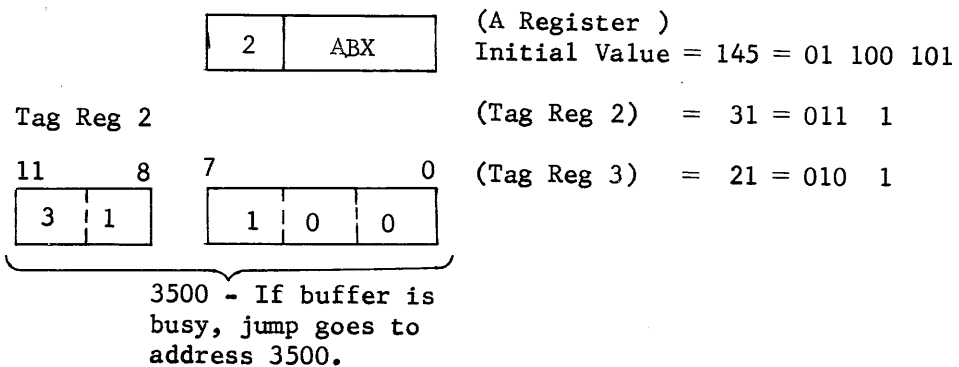
Description:

Transfer the contents of A to the lower 8 bit positions of the Buffer Entrance Register (BXR). The Buffer Entrance Register is a 10 bit register; therefore, the rightmost 2 bits of Tag Register 3 becomes the 9th and 10th bits of the BXR, the upper 2 bits of Tag Register 3 are referenced for bits 11 and 12 to complete the ability to reference 4K. If the Buffer is busy, a jump occurs to the combined address contained in the second word of the instruction and the designated Tag Register. If the buffer is not busy, control goes to the next instruction @ P+2.

Previous to executing this instruction Tag Register 3 should be loaded with the effective upper 4 bits of the starting address, and then the BXR with the lower 8 bits. The Tag Register designated in the instruction word should also be prepared in case the buffer is busy.



Example:

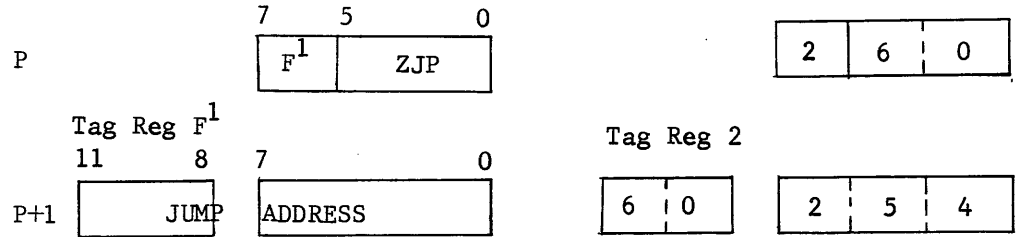


After the execution of the above command the 10 bit BXR and the upper 2 bits of Tag Register 3 would give a 12 bit address 2545. If the Buffer is busy, a jump address is provided by the designated Tag Register and the second word of the instruction. The jump address would be 3500.

C. Jump Instructions

1. Zero Jump

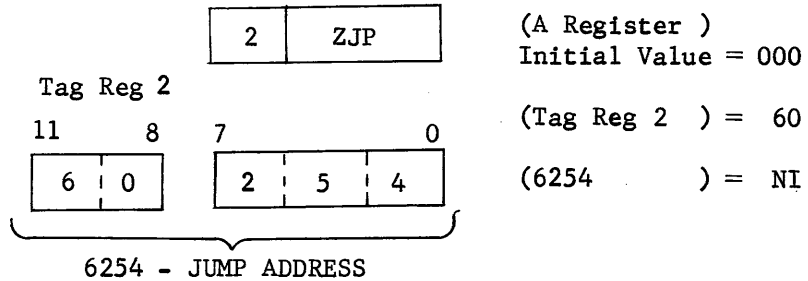
4 usec. (No jump)  
8 usec. (jump )



Description:

If the contents of A equals zero, jump to the combined address contained in the designated Tag Register and the second word of the instruction. If the contents of A are not zero, continue in sequence with the next instruction @ P+2.

Example:

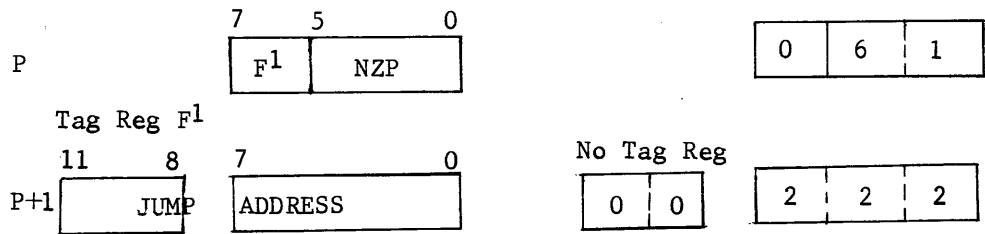


After the execution of the above command control would transfer to Memory Location 6254 to read the next instruction.

NOTE: A zero: the contents of A are equal to 000, or a plus zero. Minus zero is not considered equivalent to plus zero to meet the jump condition.

2. Not Zero Jump

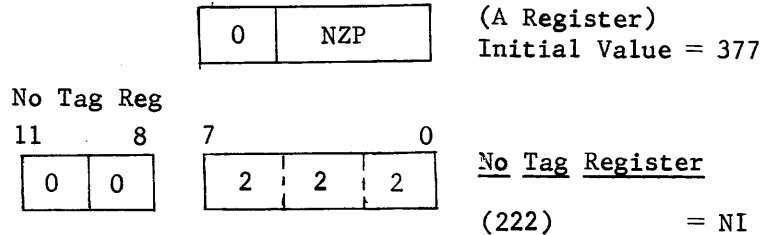
4 usec. (No jump)  
8 usec. (jump)



Description:

If the contents of A are not zero, jump to the combined address contained in the designated Tag Register and the second word of the instruction. If the contents of A are zero, continue in sequence with the next instruction @ P+2.

Example:



0222 = JUMP ADDRESS

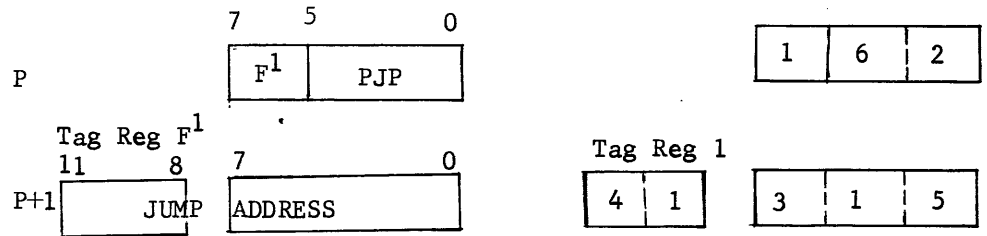
After the execution of the above command control would transfer to Memory Location 0222 to read the next instruction.

NOTE: A not zero: A contains any quantity other than 000.

Since the complete jump address can be expressed in 8 bits, no Tag Register is required. Thus, the Tag designation = 0, in the first instruction word.

3. Positive Jump

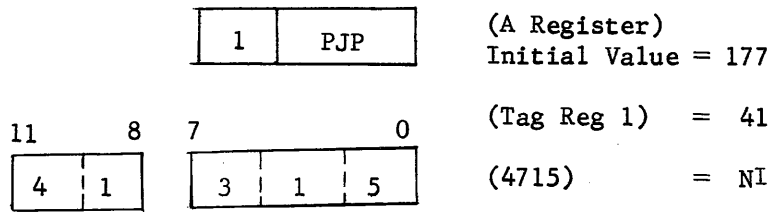
4 usec. (No jump)  
8 usec. (jump)



Description:

If the contents of A are positive (equal or greater than zero), jump to the combined address contained in the designated Tag Register and the second word of the instruction. If the contents of A are not positive, continue in sequence with the next instruction @ P+2.

Example:



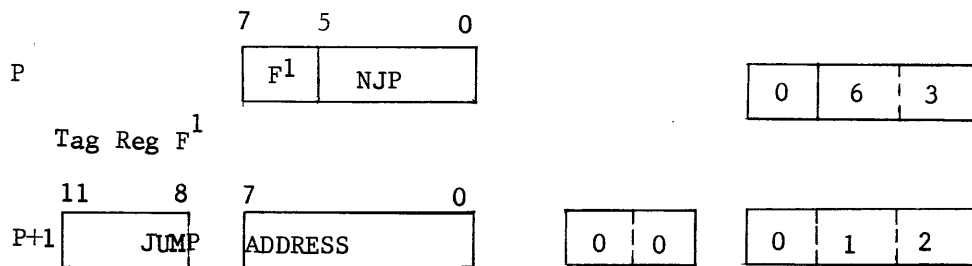
4715 = JUMP ADDRESS

After the execution of the above command control would transfer to Memory location 4715 to read the next instruction.

NOTE: A positive: bit 7 of A is 0.

4. Negative Jump

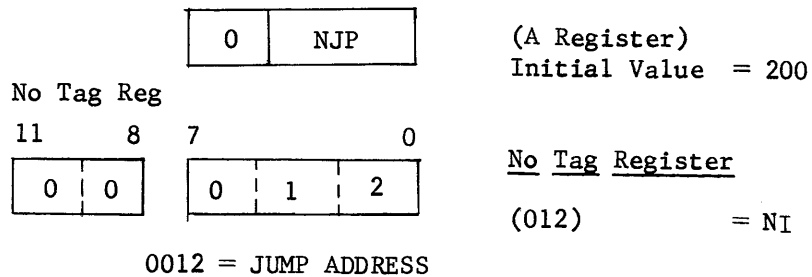
4 usec. (No jump)  
8 usec. (jump)



Description:

If the contents of A are negative, jump to the combined address contained in the designated Tag Register and the second word of the instruction. If the contents of A are not negative, continue in sequence with the next instruction @ P+2.

Example:



After the execution of the above command control would transfer to Memory Location 0012 to read the next instruction.

NOTE: A negative: bit 7 of A is 1. Since the complete jump address can be expressed in 8 bits, no Tag Register is required. Thus, the Tag designation = 0, in the first instruction word.

5. Unconditional Jump

8 usec.



Description:

Jump to the combined address contained in the designated Tag Register and the second word of the instruction.

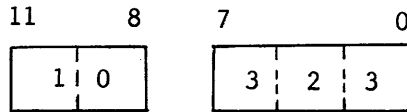
Example:



(Tag Reg 2) = 10

Tag Reg 2

(1323) = NI



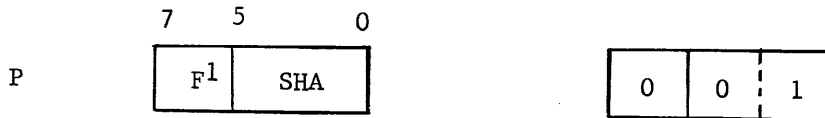
1323 = JUMP ADDRESS

After the execution of the above command control would transfer to Memory Location 1323 to read the next instruction.

D. SHIFT Instruction

1. Shift A left 1

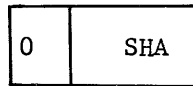
4 usec.



Description:

Shift the contents of A left, end around one bit position. Bits coming off the left end of the A register enter the lowest bit position on the right end of the register, and each bit moves on up the line respectively. RNI @ P+1.

Example:



(A Register  
Initial Value) = 023 = 00010011

After the execution of the above command the A Register would be:

(A Register  
Final Value) = 

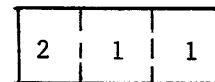
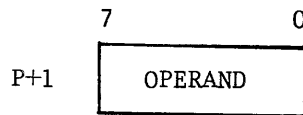
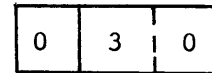
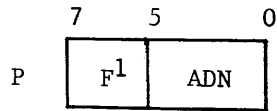
0	4	6
---	---	---

NOTE: One shift instruction is required to shift A one place (1 bit) to the left. Each shift is equivalent to one multiplication by 2. To shift 5 bits left, it is necessary to give 5 shift instructions, or loop through the single shift instruction.

E. Arithmetic Instructions

1. Add - No Address Mode

8 usec.



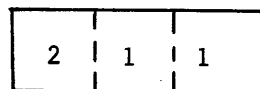
Description:

Add to the previous value of the A Register the 8 bit number given in the second word of the instruction. The sum is left in the A Register. RNI @ P+2.

Example:

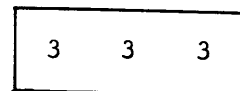


(A Register Initial Value) = 122



After the execution of the above command the A Register would be:

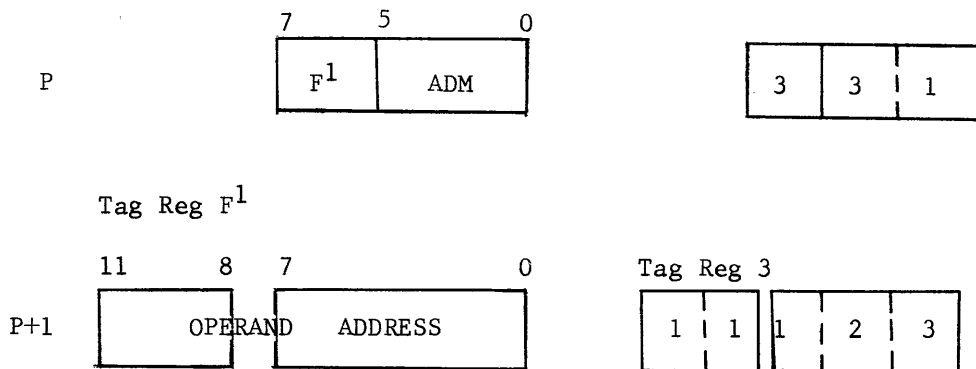
(A Register Final Value) =





2. Add - Memory Mode

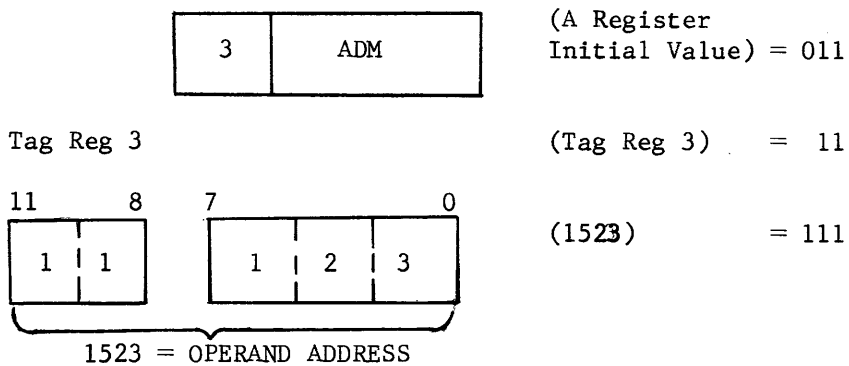
12 usec.



Description:

Add to the previous value of the A Register the contents of the combined address given in the designated Tag Register and the second word of the instruction. The sum is left in the A Register. RNI @ P+2.

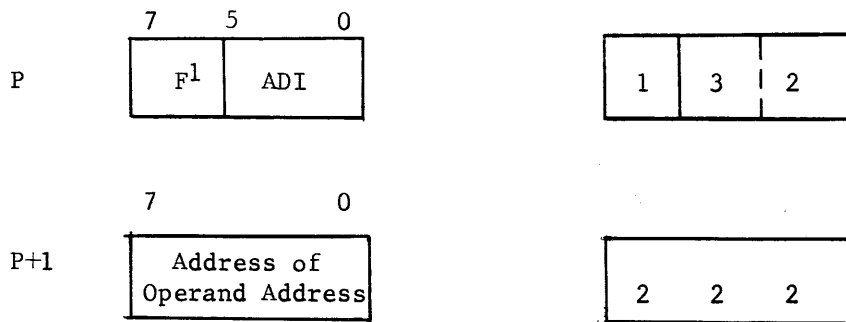
Example:



After the execution of the above command the contents of 1523 would be added to the contents of A.

3. Add - Indirect Mode

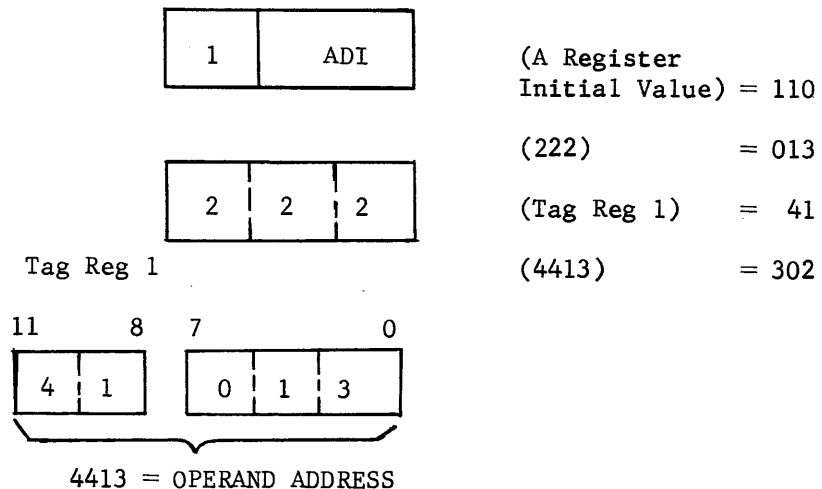
16 usec.



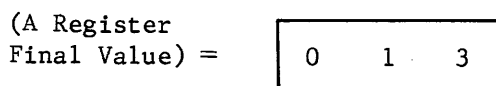
Description:

Add to the previous value of the A Register, the contents of the combined address contained in the designated Tag Register and one of the first 256<sub>10</sub> (400<sub>8</sub>) locations indicated in the second word of the instruction. The sum is left in the A Register. RNI @ P+2.

Example:

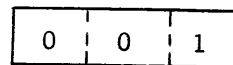
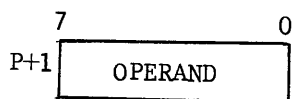
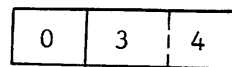
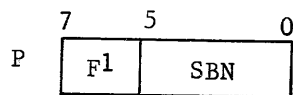


After the execution of the above command, the contents of 4413 would be added to the contents of A.



4. Subtract - No Address Mode

8 usec.

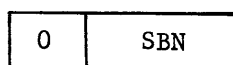


Description:

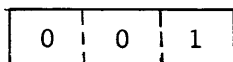
Subtract from the previous value of the A Register the 8 bit number contained in the second word of the instruction.

The difference is left in A. RNI @ P+2.

Example:



(A Register  
Initial Value) = 003



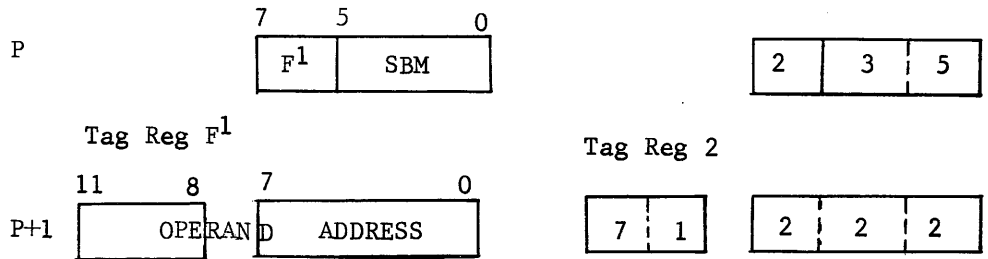
After the execution of the above command the A Register would be:

(A Register  
Final Value) = 

0	0	2
---	---	---

5. Subtract - Memory Mode

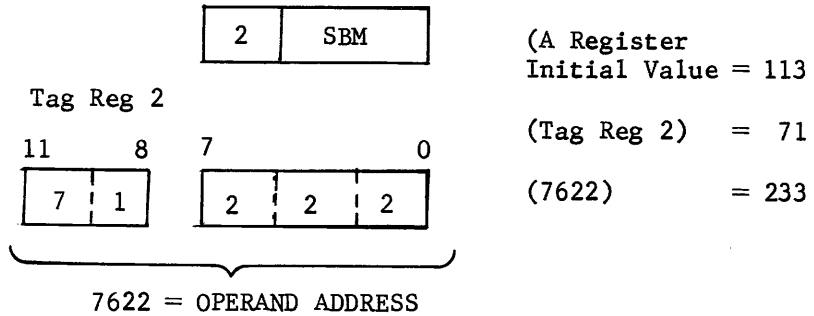
12 usec.



Description:

Subtract from the previous value of the A Register the contents of the combined address contained in the designated Tag Register and the second word of the instruction. The difference is left in A. RNI @ P+2.

Example:



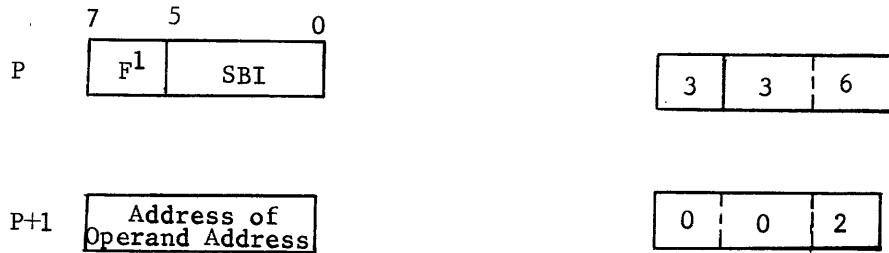
After the execution of the above command the contents of 7622 would be subtracted from the A Register.

(A Register Final Value) = 

2	5	7
---	---	---

6. Subtract - Indirect Mode

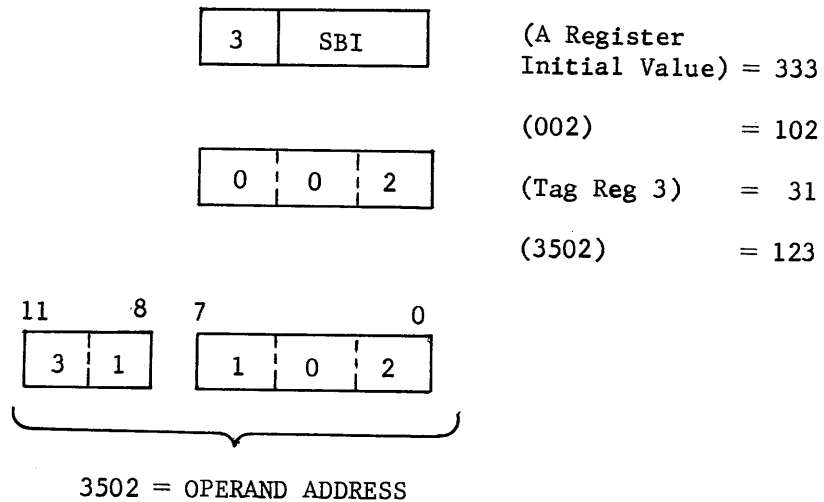
16 usec.



Description:

Subtract from the previous value of the A Register, the contents of the combined address contained in the designated Tag Register and the location of one of the first  $256_{10}$  ( $400_8$ ) locations indicated by the second word of the instruction. The difference is left in A. RNI @ P+2.

Example:



After the execution of the above command the contents of 3502  
would be subtracted from A.

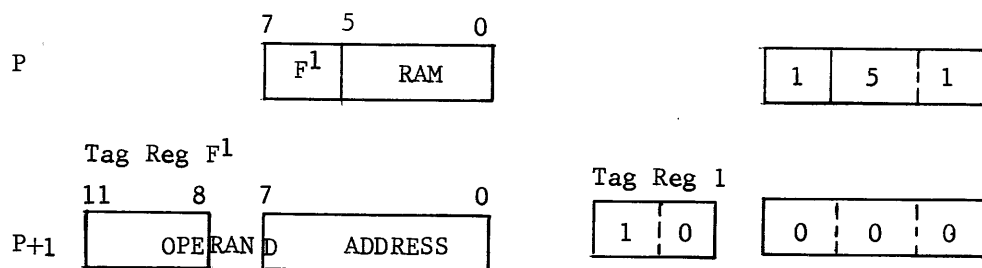
(A Register = 

2	1	0
---	---	---

  
Final Value)

7. Replace ADD - Memory Mode

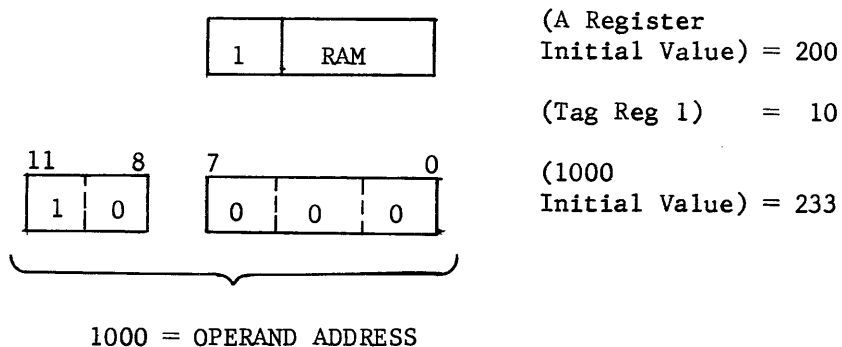
16 usec.



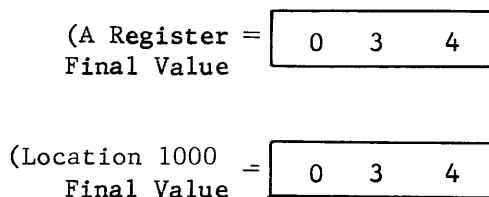
Description:

Add to the previous contents of A, the contents of the memory address formed by the contents of the designated Tag Register and the second word of the instruction. The sum thus formed, remains in A, and replaces the initial contents of the memory address. RNI @ P+2.

Example:

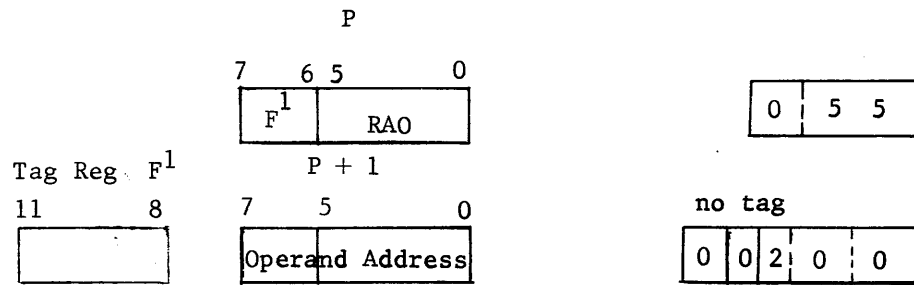


After the execution of the above command the contents of 1000 would be added to the A Register and replaced.



8. Replace Add One - MEMORY MODE

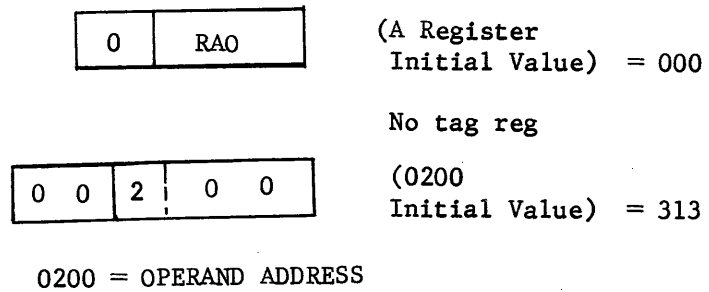
16 usec.



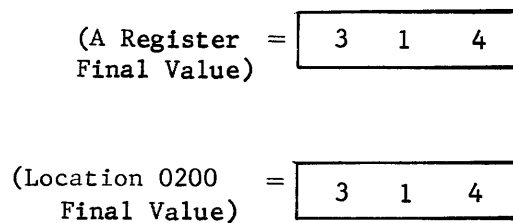
Description:

Add 1 to the contents of the memory address indicated by the combined contents of the designated Tag register and the second word of the instruction. This sum is formed in A and remains in A at the end of the instruction and replaces the initial contents of the memory address. RNI @ P+2

Example:



After the execution of the above command the contents of 0200 would be sent to A, 1 added to it and the new result replaced. The registers would be:

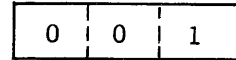
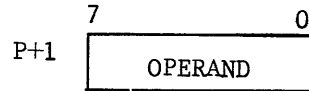
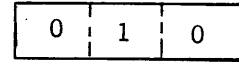
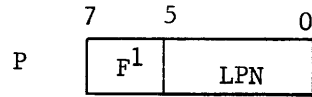




F. Logical Instructions

1. Logical Product - No Address Mode

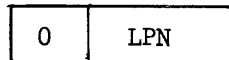
8 usec.



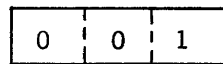
Description:

Form in A the Logical Product of the contents of A and the contents of the second word of the instruction. RNI @ P+2

Example:



(A Register  
Initial Value) = 377



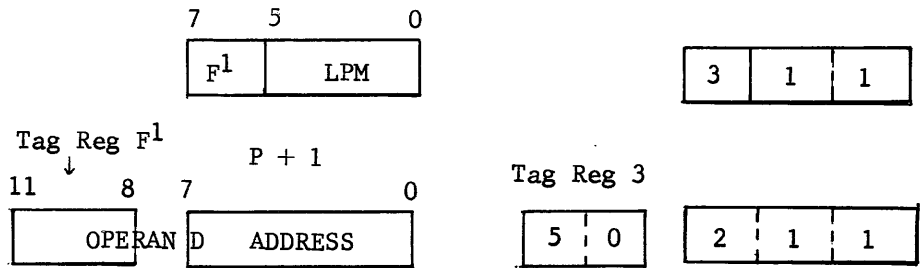
After the execution of the above command, the bit by bit logical product is performed and the A register contains:

(A Register =  
Final Value) 

0	0	1
---	---	---

2. Logical Product - Memory Mode

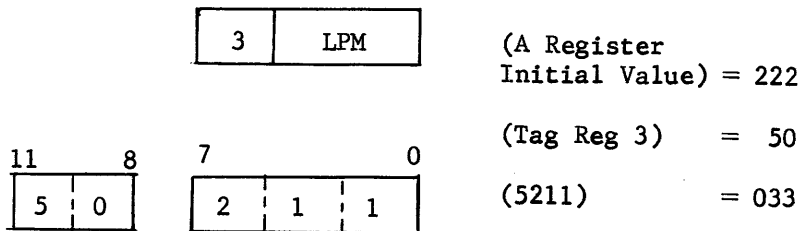
12 usec.



Description:

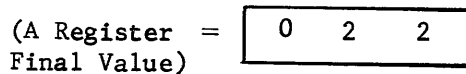
Form in A, the logical product of the contents of A and the contents of the memory address indicated by the combined contents of the designated Tag Register, and the second word of the instruction. The initial contents of the memory address remains unchanged. RNI @ P+2.

Example:



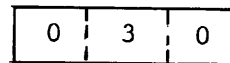
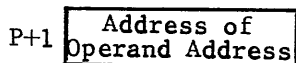
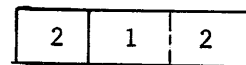
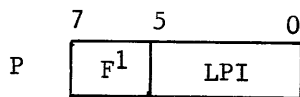
5211=OPERAND ADDRESS

After the execution of the above command the bit by bit multiplication is performed and the A Register contains:



3. Logical Product - Indirect Mode

16 usec.



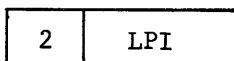
Description:

Form in A the logical product of the contents of A and the contents of the memory location whose address is the combined contents of the designated Tag Register and the contents of one of the first  $256_{10}$  ( $400_8$ ) locations. The address of this location is given in the second word of the instruction.

The initial contents of the memory location remains unchanged.

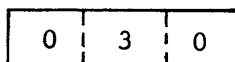
RNI @ P+2.

Example:



(A Register Initial Value) = 133

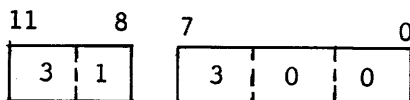
(030) = 300



(Tag Reg 2) = 31

(3700) = 012

Tag Reg 2



3700 = OPERAND ADDRESS

After the execution of the above command the bit by bit multiplication is performed with the A Register and the contents of memory location 3700.

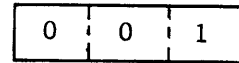
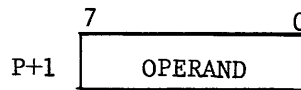
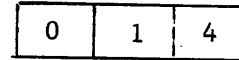
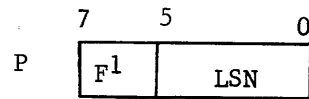
(A Register = 

0	1	2
---	---	---

  
Final Value)

4. Logical Sum - No Address Mode

8 usec.



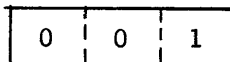
Description:

Form in A the logical sum (partial sum) of the contents of A and the second word of the instruction. RNI @ P+2.

Example:



(A Register  
Initial Value = 002



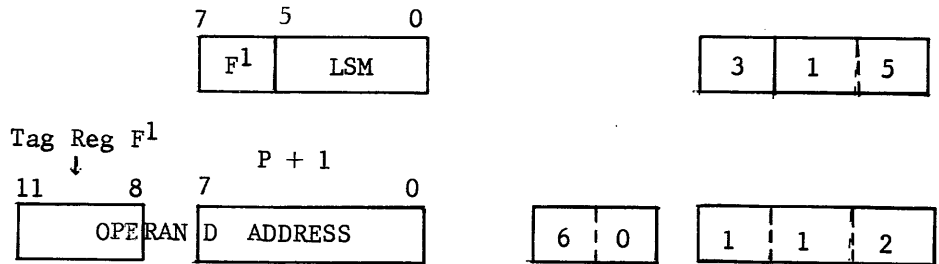
After the execution of the above command, the partial add is performed and the A Register contains:

(A Register Final Value) = 

0	0	3
---	---	---

5. Logical Sum - Memory Mode  
P

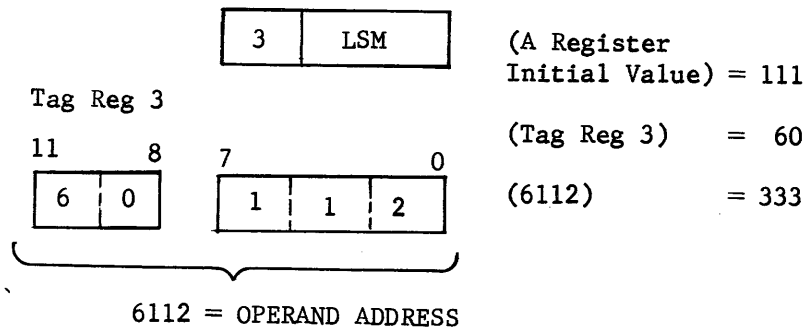
12 usec.



Description:

Form in A the logical sum (partial sum) of the contents of A and the contents of the memory location whose combined address is given in the designated Tag Register and the second word of the instruction. RNI @ P+2.

Example:



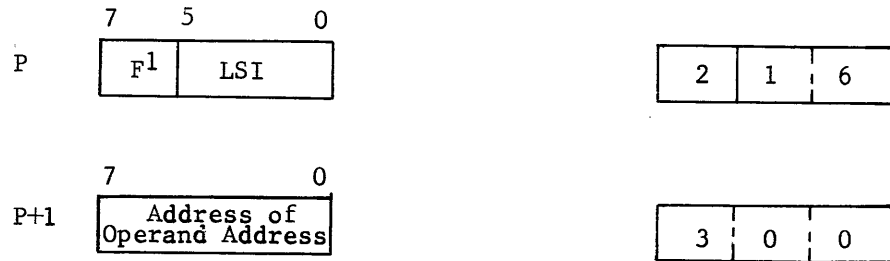
After the execution of the above command, the partial add is performed with the contents of A and the contents of the operand address, 6112.

(A Register = Final Value) 

2	2	2
---	---	---

6. Logical Sum - INDIRECT MODE

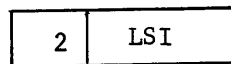
16 usec.



Description:

Form in A the Logical Sum (Partial Sum) of the contents of A and the contents of the memory location whose address is the combined contents of the designated Tag Register and one of the first  $256_{10}$  ( $400_8$ ) locations. The location of one of these  $256_{10}$  locations is given in the second word of the instruction. RNI @ P+2.

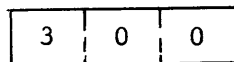
Example:



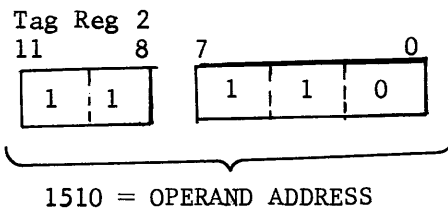
(A Register )  
Initial Value = 010

(300 ) = 110

(Tag Reg 2 ) = 11



(1510 ) = 301



After the execution of the above command, the Partial Add is performed with the contents of A and the contents of 1510.

Thus, the A Register contains:

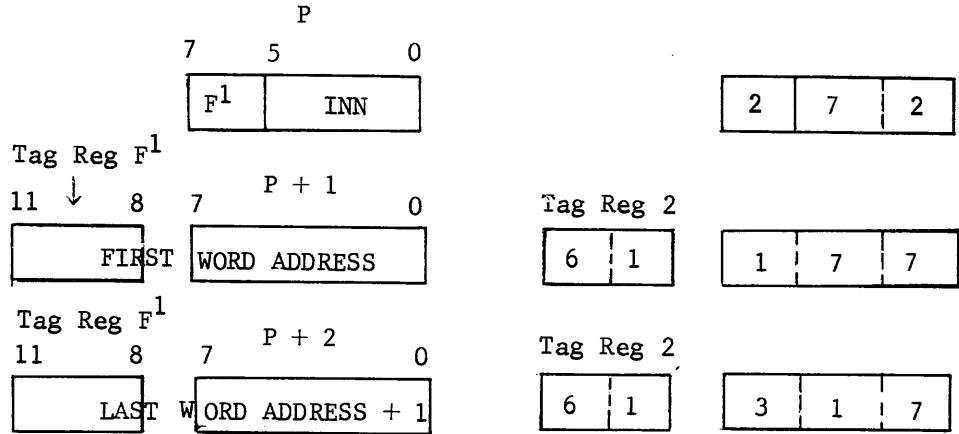
(A Register)  
Final Value = 

3	1	1
---	---	---



G. Input - Output Instructions

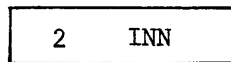
1. Input Normal 3 + 2 (X + 1) + Terminate Time  
Where X = No. of words



Description:

Input a number of words to memory starting at the memory address contained in the designated Tag Register and the second word of the instruction. The last word address plus 1, is contained in a third word immediately following the second word. Thus, this instruction is composed of three words. (The Tag Register designation indicated in the first word is automatically assigned as the Tag Register designation for the last word address plus 1, in the third word.) RNI @ P+3 at completion of instruction.

Example:



(Tag Reg 2) = 61

$$\begin{array}{r}
 \text{Tag Reg 2} \\
 11 \quad 8 \quad 7 \quad \quad \quad 0 \\
 \boxed{6} \quad \boxed{1} \quad \boxed{1} \quad \boxed{7} \quad \boxed{7} \quad 0 \\
 = \text{FWA} + 6577 \quad \begin{array}{r} 6717 \\ -6577 \\ \hline 120 \end{array} \text{ Words Input}
 \end{array}$$

$$\begin{array}{r}
 \text{Tag Reg 2} \\
 11 \quad 8 \quad 7 \quad \quad \quad 0 \\
 \boxed{6} \quad \boxed{1} \quad \boxed{3} \quad \boxed{1} \quad \boxed{7} \quad 0 \\
 = \text{LWA} + 1 = 6717
 \end{array}$$

After the execution of the above command  $120_8$  word will have been input into memory.

NOTE: The length of the input or output block, that is from the FWA to the LWA + 1, may result in a "gap" -- that is, no input or output can come from a certain memory location. The reason is that addresses such as, 1377, falls at a "boundary address" as far as the addressing logic of the Tele-programmer is concerned. "Boundary addresses" are those, which when incremented by 1, would cause a change to occur in any one of the 4 leftmost, Tag Register, address bits. This in turn, requires a change in the Tag Register. There are 16 such "boundary addresses" in the whole 4096 registers. This condition is not serious due to the following alternatives:

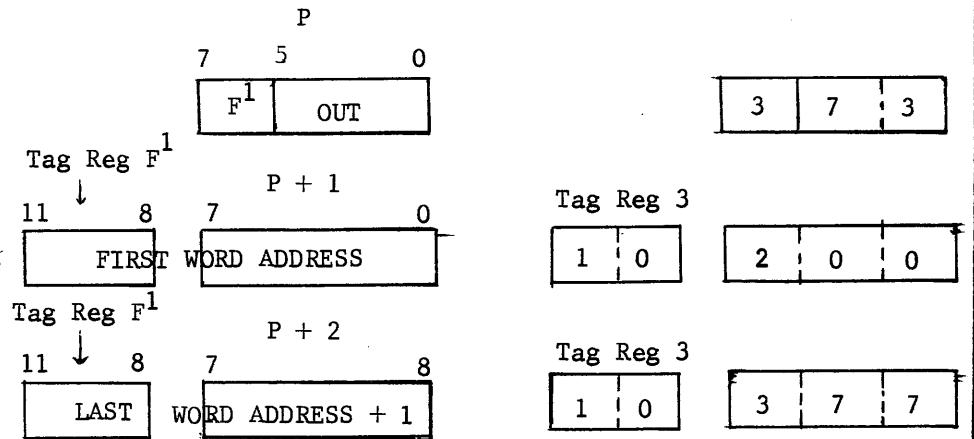
- a. If output follows input or vice versa, such "gaps" would have existed in the identical places anyway, and thus are of no consequence.

- b. If one wishes, he can fill the gap location by loading one word into A and storing at the gap address.
- c. By effective memory allocation, boundary addresses can often be entirely avoided.
- d. Buffered operations do not have this situation.

It should be apparent that the maximum transfer without changing the Tag Register is  $255_{10}$  ( $377_8$ ) words. On an input and a completion by an input Disconnect the A Register would contain the last word address plus 2. Last word address + 1 = 0?

2. Output Normal

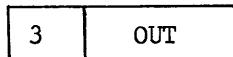
$3 + 2(X + 1) + \text{Terminate Time}$   
 Where X = No. of words



Description:

Output a number of words from memory starting at the memory address contained in the designated Tag Register and the second word of the instruction. The last word address plus 1, is contained in a third word immediately following. Thus, this instruction set is composed of three words. (The Tag Register designation, indicated in the first word is automatically assigned as the Tag Register designation for the last word address plus 1, in the third word.) RNI @ P + 3 at completion of instruction.

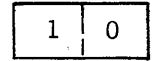
Example:



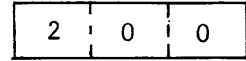
(Tag Reg 3) = 10

Tag Reg 3

11 8



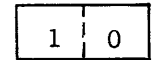
7 0



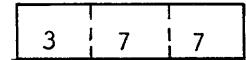
$$= FWA = 1200 - \frac{1377 - 1200}{177_8} \text{ words output}$$

Tag Reg 3

11 8



7 0



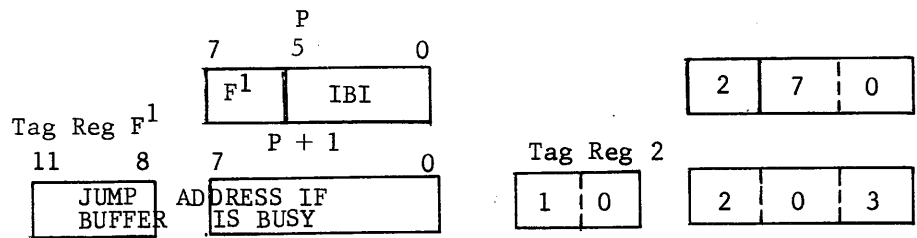
$$= LWA + 1 = 1377$$

After the execution of the above command  $177_8$  words will have been output from memory.

NOTE: SEE NOTE ON NORMAL INPUT.

3. Initiate Buffer Input

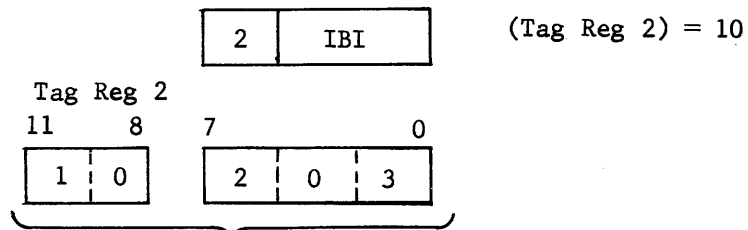
8 usec.



Description:

Before using this instruction, the FWA of the transfer area is sent to the BER, and the LWA + 2 is sent to the BXR, and Tag this instruction initiates the input buffer cycle. If the buffer channel is not busy, control goes to the next instruction following the second word of the instruction, @ P+2. If the buffer channel is busy, a jump occurs to the memory location whose combined address is contained in the designated Tag Register and the second word of the instruction.

Example:



1203 = Jump Address if the buffer is busy

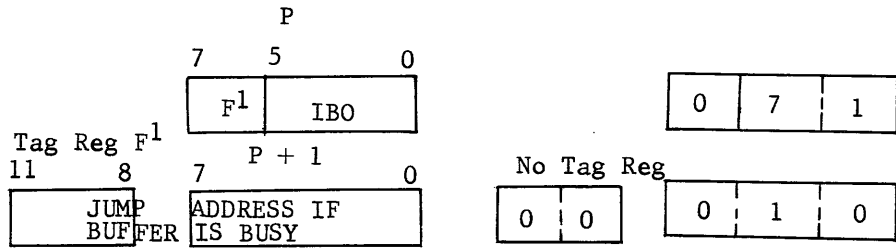
Not busy initiate buffer and RNI @ P+2

NOTE: The maximum word block would be 1024 before a Tag Register change would be needed.

On a buffer input completion by an input disconnect the A Register would contain the last word address plus 2.

4. Initiate Buffer Output

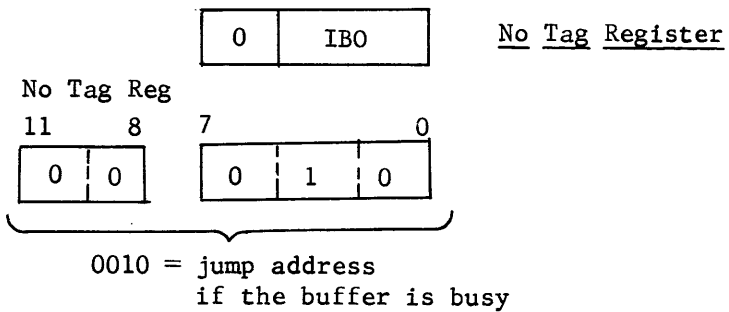
8 usec.



Description:

Before using this instruction, the FWA of the transfer area is sent to the BER, and the LWA + 2 is sent to the BXR. This instruction initiates the output buffer cycle. If the buffer channel is busy, a jump occurs to the combined memory address given in the designated Tag Register and the second word of the instruction. If the buffer channel is not busy, control goes to the next sequential instruction following the instruction @ P+2.

Example:

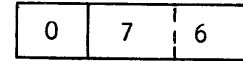
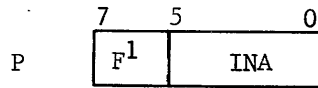


Not busy initiate buffer and RNI @ P+2.

NOTE: A Tag Register was not required here. Also see note on IBI.

5. Input to A

8 usec. + equipment time

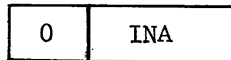


Description:

This instruction inputs one word from a previously selected input device to the A register. This is a single word instruction, and the Tag Register designation is always zero.

RNI @ P + 1.

Example:



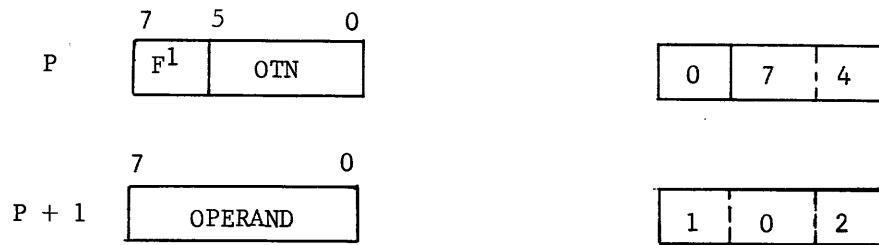
(A Register)  
Initial Value = 000

(A Register)  
Final Value = Input word



6. Output No Address

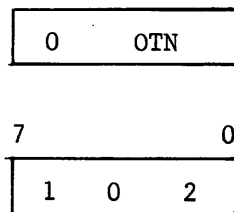
8 usec + equipment time



Description:

This instruction outputs one word. This word is the second word of the instruction. The Tag Register designation is always zero in this instruction. RNI @ P+2.

Example:

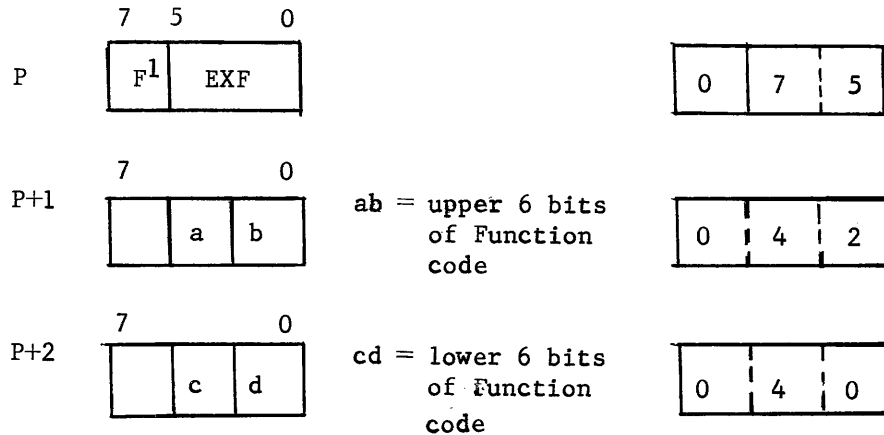


After the completion of this command the previously selected device would have received the word 102.

## H. Control Instructions

### 1. External Function

12 usec. + equipment time



#### Description:

This instruction is used to select an external unput or output device to communicate with the Teleprogrammer. The select function is accomplished by sending out on the output lines a 12-bit "function code." Each external device is capable of recognizing and interpreting only its own unique code. Thus, the programmer by selecting different external function codes can use this same instruction to select all external devices. The 12-bit function code is contained in the second and third words of the three words which make up this instruction.

RNI @ P+3.

Example:

0	EXF
---	-----

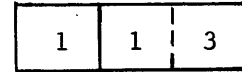
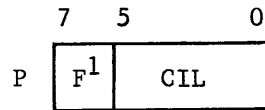
0	4	2
---	---	---

0	4	0
---	---	---

After the execution of the above command, the 12-bit Status Select, 4240, of the typewriter is sent out. RNI @ P+3.

2. Clear Interrupt Lockout

4 usec.



Description:

This instruction clears the interrupt lockout flip-flop (FF). This instruction must be programmed at the end of every routine which is initiated by the interrupt. This instruction may return control to the main program. In this instruction, the Tag Register designation becomes a part of the instruction itself. It can only be 0 or 1. Thus, to return to the main program after clearing interrupt lockout, the Tag designation must be 1. If 0, control continues in sequence.

Example:



After the execution of the above command control continues to the main program.

### 3. Clear Buffer Controls

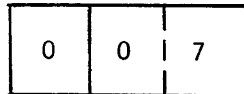
4 usec.



#### Description:

This instruction has the effect of sending a zero to the buffer control and thus putting that device in a "ready state." If this instruction is used during a buffer operation, it will stop the buffer. The Tag Register designation is ignored. RNI @ P+1.

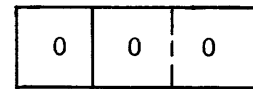
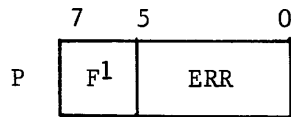
#### Example:



After the execution of the above command control continues @ P+1.

4. Error Stop

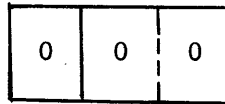
4 usec.



Description:

This is an illegal instruction, as such, it can be used as an Error Stop.

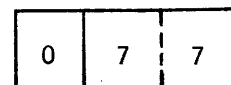
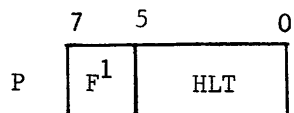
Example:



After the recognition the computer stops.

5. Halt Stop

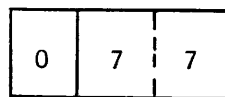
4 usec.



Description:

This instruction is used to bring the program to a halt.

Example:



After the recognition the computer stops.

#### IV. Operation

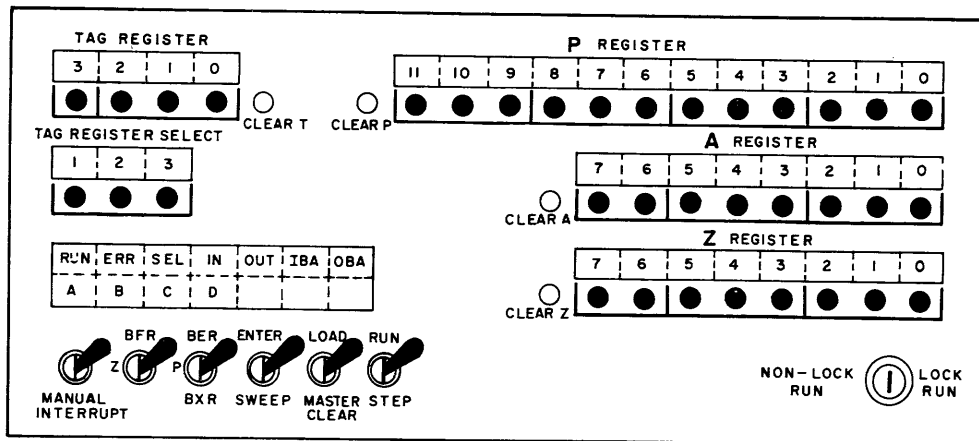


Figure 2-1 8092 Operator's Panel

#### OPERATOR'S CONSOLE

The 8092 Teleprogrammer Operator's Panel consists of several displays and switches necessary for the operation of the Teleprogrammer. The panel (see Figure 2-1) contains six display windows, six switches, and a lock switch. Four of the display windows can display in binary the contents of nine 8092 registers. Buttons beneath these displays clear and enter data into the P, A, Z, and Tag registers (the only registers into which data may be entered or cleared). A fifth window contains information as to which Tag register has been selected. The sixth window contains the operating lights which indicate the status of operation of the Teleprogrammer. At the bottom of the panel is located all the operating and mode switches. The operation of these switches is explained on the following page.

SWITCHES

Manual Interrupt

- Momentary depression causes the Teleprogrammer to enter an interrupt routine to determine the nature of the interrupt.

BFR,Z

- This 3-position switch chooses the register that is to be displayed in the 8-bit Z register display.

UP - Displays the last word processed during the last buffer operation (BFR register).

Center - Shows the current contents of the Z register (Z register).

Down - Not assigned.

BER, P, BXR

- This 3-position switch chooses the register to be displayed in the 12-bit P register display.

Up - Displays the address of the last word transferred out, the next word to be transferred in on the buffer channel (BER register).

Center - Displays the address of the current instruction (P register).

Down - Displays the LWA + 1 of the last buffer operation (BXR register).

ENTER/SWEEP

- Sweep is used to display the contents of core storage locations. Enter is used for entering information into core storage from the console.

LOAD/MASTER CLEAR

- LOAD position allows specially prepared paper tapes to be read into storage by the paper tape reader.

Master CLEAR performs a Teleprogrammer master clear which -



- a. Clears the registers
- b. Clears the control flip flops
- c. Clears all waiting interrupts and removes interrupt lockout.

Note: The master clear does not alter core storage.

RUN/STEP

Up - In RUN position a program is executed at high speed starting at the location specified by the P register.

Center - Center position stops the computer program. If the switch is in RUN and an ERR or HLT instruction is executed, the switch must be returned to neutral and then placed in RUN to continue computation.

Down - In STEP position, one storage cycle of an instruction is executed each time the switch is set; a program may be executed one instruction at a time for debugging.

NON-LOCK RUN  
RUN LOCK

- In the Lock position all other switches are disabled and the Teleprogrammer is locked in the RUN position.

In the non-lock position, the console switches are enabled and the Teleprogrammer programs can be operated and modified from the console.

DISPLAYS

Z REGISTER

- This display known as the Z register group displays the Z and BFR registers in accordance with the setting of the BFR, Z switch.

A Register

- Displays the current contents of the A register.

P Register

- This display known as the P register group displays the BER, P, and BXR registers in accordance with the setting of the BER, P, BXR switch.

TAG REGISTER

- This display indicates the Tag register currently being referenced by an instruction. The contents of any Tag register may be displayed by depressing one of the buttons directly below the select indicators. Depressing one of the select buttons also enables the Tag registers to be manually set or cleared.

STATUS INDICATORS

RUN

- Indicates that the Teleprogrammer is in RUN status. This does not necessarily indicate that instructions are being executed.

ERR

- Indicates that a timing fault has occurred.

SEL

- Displayed each time an EXF instruction is executed; remains until selection is completed. A constant display of SEL with no apparent input/output action usually indicates the Teleprogrammer has attempted an illegal selection.

IN

- Displayed during all normal input operations. A constant display of IN with no apparent input action usually indicates that input was attempted without proper unit selection. IN is also displayed when the Teleprogrammer is waiting for an external device to supply data.

OUT

Displayed during all normal output operations. A constant display of OUT with no apparent output action usually indicates that output was attempted without proper unit selection.

IBA

- Displayed during all buffer input operations.

OBA

- Displayed during all buffer output operations

A, B, C, or D

- Indicates which storage reference cycle will be executed at the next operation of the Run/Step switch. When a master clear is performed, D is displayed indicating that the next operation to be executed, when the Run/Step switch is operated, will be to fetch the instruction from memory at the address indicated by the P register.

#### STARTING THE 8092 TELEPROGRAMMER

1. Be sure the Teleprogrammer is plugged into proper power source and room temperature is within the prescribed limits.
2. Turn Power switch on power supply to ON.
3. Master clear by momentarily pressing Load/Clear switch to Clear.
4. When the ERR Light goes out, the Teleprogrammer is ready to operate. If repeated master clears do not turn the Red ERR light off, turn off the 8092 and call maintenance.

#### LOADING A PROGRAM OR DATA

##### Paper Tape Load Format

1. Master Clear
2. Turn on reader
3. Insert paper tape in reader
4. Set P to starting location
5. Set Load/Clear switch to LOAD
6. Set Run/Step switch to RUN. Paper tape will load and Teleprogrammer will stop.

#### ENTERING DATA FROM THE TELEPROGRAMMER CONSOLE

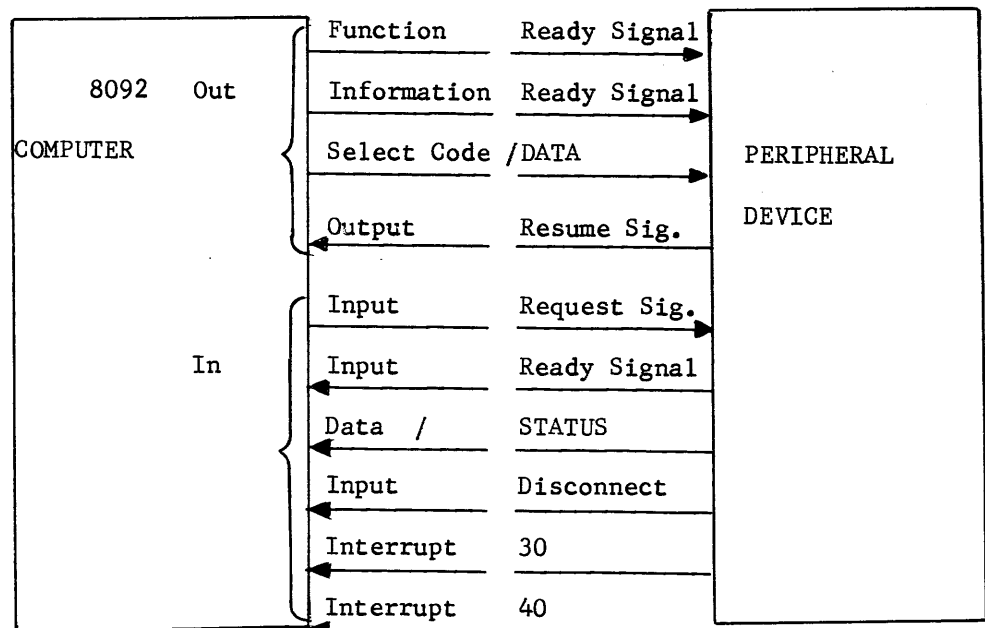
1. Master clear. Set Enter/Sweep switch to ENTER
2. Set P to location into which data is to be entered.
3. Enter one word of data into the A register.
4. Set Run/Step switch to STEP, once. At this point A is clear and the data word is in storage and in Z.
5. If data is to be entered into consecutive locations, go to step 3 and P will be advanced by one on step 4. If data is to be entered into non-consecutive locations, clear P. Go to step 2.

EXAMINING THE STORAGE CONTENTS

1. Master clear. Set Enter/Sweep switch on SWEEP.
2. Set P to location to be examined.
3. Press Run/Step switch to STEP, once. The contents of the location specified by P will appear in Z.
4. To examine consecutive locations, go to step 3 and P will be advanced by one on step 3. To examine non-consecutive location, clear P, go to step 2.

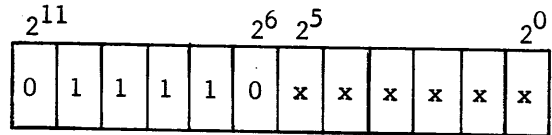
V. External Function Codes and Status Responses for PERIPHERAL EQUIPMENT

Two I/O channels are available on the 8092, NORMAL and BUFFER. The Buffer channel transmits either Buffer or Normal I/O operations. The NORMAL channel transmits only normal I/O instructions. Two cables connect the computer with the peripheral equipment. They are designed as "IN" and "OUT". To transmit data from the computer to a piece of peripheral equipment, or vice versa, more is necessary than just putting the data on the lines. First one piece of equipment of many must be chosen. Next, it must know whether its an Input or Output operation. So we have a code called a Function Code. Also at times it is necessary for the computer to know the condition or STATUS of the piece of equipment with which it is going to or has been communicating. NOW with a Function Code, STATUS CODE, and Data on the lines, it is necessary to have signals for communication.



A. 8096 - TELETYPE MODEL 33 PAGE PRINTER

External Function	Code
Select	3605
De-Select	36XX
Status Request	3603



Status Responses	Code
Break	001
Ready for Input	002
Ready for Input	004
Ready for Output	010

Function Code:

Select (3605)

This code selects the equipment. When the equipment is selected information can be outputted from the computer. For an input operation the computer waits for an interrupt 30 from the equipment. The interrupt 30 is produced by depressing the first key and is repeated at a pre-set rate (normally 90 milliseconds) until the unit is De-Selected.

De-Select (36XX)

This code de-selects the equipment. The XX portion of the code can be anything except 03 or 05.

Status Request (3603)

This code causes the equipment to place the status on the lines which can be inputted by the 8092.

Status Code

001 - A break in progress. The character from 8092 contains all 0's and the teletype loop is open; or no input or output operations have occurred for the past 8 seconds.

002 - A previous word ready for transmission to 8092 is not sent because the input request was not received.

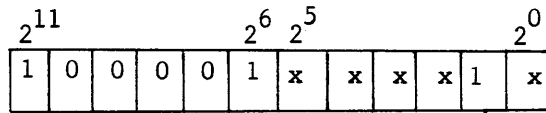
004 - A data word ready for transmission to 8092.

010 - Prepared to accept a new data word from 8092.



B. 350 PAPER TAPE READER

External Function	Code
Select Reader	4102



Select Reader

Status Responses	Code
NONE	

Function Code:

The Master Bits for the Reader is a 41. If a bit is present in the  $2^1$  bit position the reader will be selected for Input. The Input Operation will take place upon the initiation of an Input Instruction.

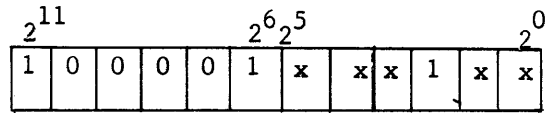
Status Code

NONE



D. BRPE-11 PAPER TAPE PUNCH

External Function	Code
Select Paper Tape Punch	4104



Select Paper Tape Punch

Status Responses	Code
NONE	

Function Code:

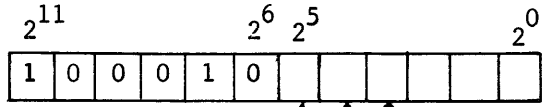
The Master Bits for the punch is a 41. If a bit is present in the  $2^2$  bit position the punch will be selected for output. The output operation will take place upon the initiation of an Output Instruction.

Status Code

NONE

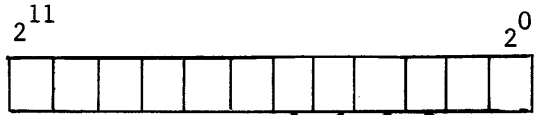
E. 161 INPUT/OUTPUT TYPEWRITER

External Function	Code
Select Typewriter Output	4210
Select Typewriter Input	4220
Request Typewriter Status	4240
Alternate Unit Selection	43XX



Req. Type Status  
 Select Type Input  
 Select Type Output

Status Response	Code
Ready	000
Power Off	XX4
Not In Computer Status	X1X
Input Char. Ready	X2X
Output in Use	X4X



Output in Use  
 Input Char. Ready  
 Type not in Comp. Status  
 Power Off

Function Codes:

Select Typewriter Output (4210)

The Master Bits for the typewriter is a 42. When a bit is present in the  $2^3$  bit position of the select code, the typewriter will be selected to do an output operation upon the receipt of an output instruction.

Select Typewriter Input (4220)

Whenever a bit is present in the  $2^4$  bit, the typewriter will be selected to do an Input Operation upon the receipt of an output instruction.

Request Typewriter Status (4240)

At times it is necessary to know the conditions or the status of the typewriter with which we are going to or have been communicating. When a bit is present in the  $2^5$  bit position, the typewriter will put the status response on the lines, and the computer must follow this instruction with an input instruction to get and interrogate the response.

Alternate Unit Selection (43XX)

The Alternate Master Bits for another typewriter is a 43. The lower 6-bit codes would be the same as above on the other selection.

Status Codes

Ready (0000)

If no bits are present in the status code, the typewriter is connected and I/O operations may proceed. The mode switch is set to the computer status, and the typewriter is not being used by the computer.

POWER OFF (XXX4)

If a bit is present in the  $2^2$  bit position of the response, the typewriter power switch has not been turned on and the typewriter will be stalled until the condition is corrected.

Typewriter not in computer status (XX1X)

If a bit is present in the  $2^3$  bit position, the mode switch has not been put in the computer status, and until the condition is corrected the typewriter will be stalled.

Input Character Ready (XX2X)

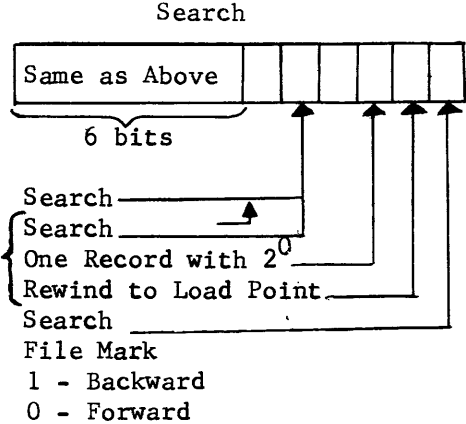
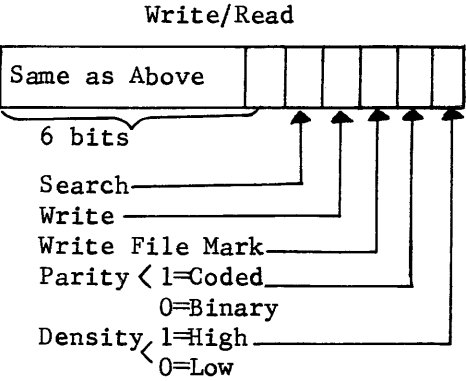
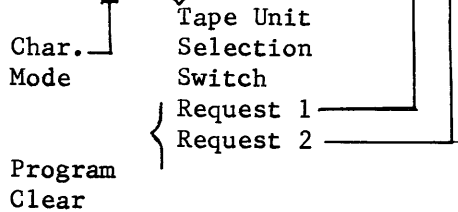
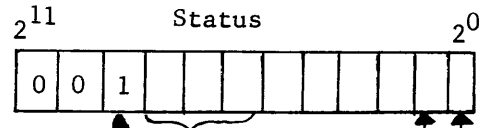
Upon receipt of a status response, and a bit is present in the  $2^4$  bit position an input character is available to the computer.

Output in Use (XX4X)

If a bit is present in the  $2^5$  bit position, the typewriter is performing a character output operation.

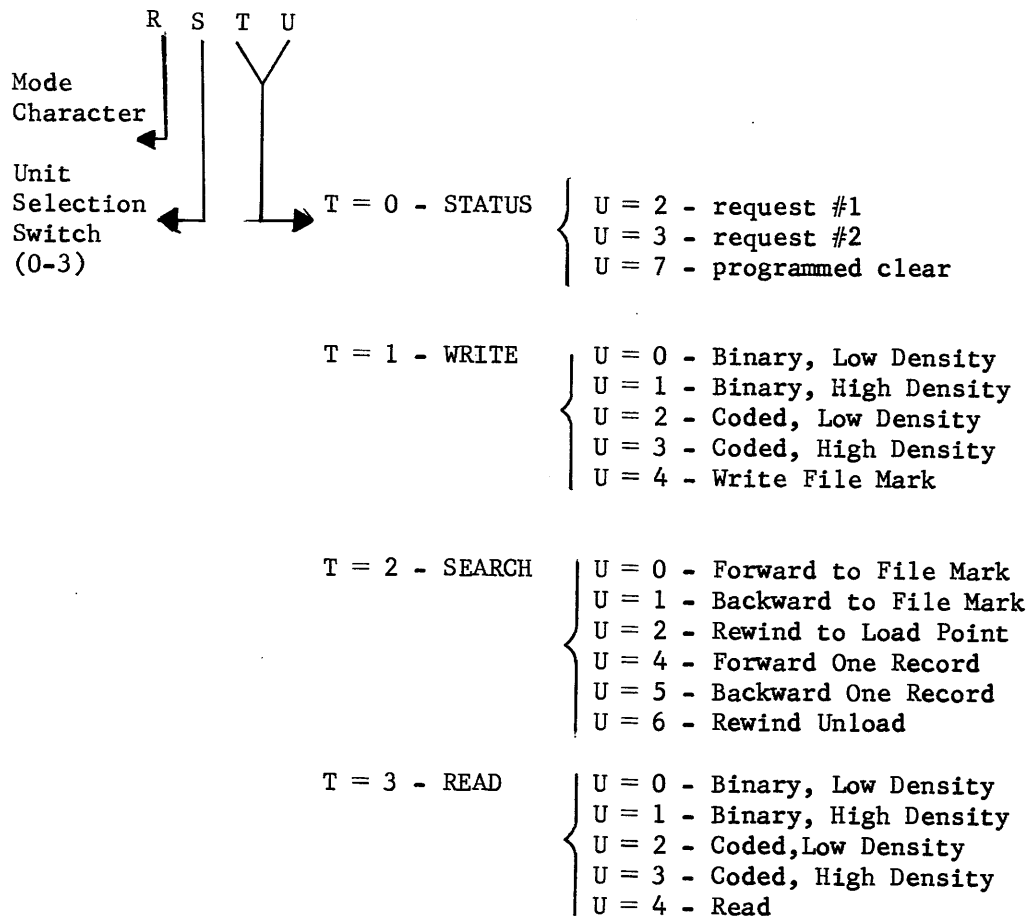
F. 8093 Magnetic Tape Synchronizer  
(For one 603)

External Function	Code
/////Status/////	
Status Request 1	1X02
Status Request 2	1X03
Programmed Clear	1X07
/////Write/////	
Write, Binary, Low Density	1X10
Write, Binary, High Density	1X11
Write, Coded, Low Density	1X12
Write, Coded, High Density	1X13
Write File Mark	1X14
/////Search/////	
Forward to File Mark	1X20
Backward to File Mark	1X21
Rewind to Load Point	1X22
Forward One Record	1X24
Backward One Record	1X25
Rewind Unload	1X26
/////Read/////	
Read, Binary, Low Density	1X30
Read, Binary, High Density	1X31
Read, Coded, Low Density	1X32
Read, Coded, High Density	1X33
Read	1X34



both bits  
rewind  
unload

The above External Function Code may be represented by:







Function Code:

Status Request 1 (1X02)

Status Request 2 (1X03)

A 1X02/1X03 code will place a 8-bit status response on the lines to allow the programmer to check certain conditions that may appear. The response will stay on the line until the program brings it into the computer with an input request signal, or drops it with a EF clear. X designates the setting of the tape unit, which will be 0 - 3.

Programmed Clear (1X07)

This code clears out the operational logic and establishes initial operating conditions within the control unit.

Write, Binary, Low Density (1X10)

Write, Binary, High Density (1X11)

Both of these codes allow all information to be written in binary notation. A parity generator makes the total number of "1" bits odd in the vertical direction of the seven tracks on the tape. Depending upon the code the information is either written at 200 bpi or 556 bpi, when the file protection ring is on the tape reel.

Write, Coded, Low Density (1X12)

Write, Coded, High Density (1X13)

These codes allow all information to be written in binary coded decimal notation. A parity generator makes the total number of "1" bits even in the vertical direction of the seven tracks on the tape. Depending upon the code the information is either written at 200 bpi or 556 bpi, when the file protection ring is on the tape reel.

Write File Mark (1X14)

This code writes  $17_8$  as an end of file mark to signify the end of a record or a group of records.

Search Forward to File Mark (1X20)

Search Backward to File Mark (1X21)

These codes search forward/reverse until an end of file mark is detected. If no file marks are detected in the Backward Motion, tape motion continues until load point is reached and then stops. In the forward direction it would be possible to run the tape off the supply reel.

Rewind to Load Point (1X22)

A 1X22 code rewinds tape at high speed to load point. Any further rewind instructions will not affect the tape when it is already at Load Point.

Search Forward One Record (1X24)

Search Backward One Record (1X25)

These codes move tape forward/reverse one record length.

Rewind Unload (1X26)

This code rewinds tape at high speed until all the tape is on the supply reel. All further operations on this unit are locked out until the tape has been reloaded manually.

Read, Binary, Low Density (1X30)

Read, Binary, High Density (1X31)

Both of these codes allow all information to be read in the binary notation. The information is either read at 200 bpi or 556 bpi. The file protection ring has no effect on this operation.

Read, Coded, Low Density (1X32)

Read, Coded, High Density (1X33)

These codes allow all information to be read in the binary coded decimal notation. The information is either read at 200 bpi or 556 bpi. The file protection ring has no effect on the operation.

Read (1X34)

This code will read the information according to the previous settings.

Status Codes

CODED (XX1)

If a bit is present in the  $2^0$  bit position of the response, the previous parity generation was even. When no bit is present, the parity to be or that was generated was odd.

Transport Not Ready (XX2)

If a bit is present in the  $2^1$  bit position of the response, the tape unit is not in the ready mode. A ready indicator on the tape unit lights when it is in a ready condition; i.e., power has been applied and the tape unit is in Automatic Mode. When in the automatic mode the tape unit is under control of synchronizer. The ready signal is not present when an operator manually controls the 603 from the console of the unit.

Parity Error (XX4)

If a bit is present in the  $2^2$  bit position, the number of bits encountered in the word read/written does not agree with the parity that was to be generated. This signal drops when a new record is read or written, or by a clear instruction.

Illegal Coded on Write (X1X)

If a bit is present in the  $2^3$  bit position, the program sent a (000) code to the tape unit, and there must be at least one bit present to generate a parity bit.

End of File (X2X)

If a bit is present in the  $2^4$  bit position, the End-of-File code  $17_8$  has been detected.

Tape Mark (X4X)

If a bit is present in the  $2^5$  bit position the load point marker or the End of Tape Marker has been sensed.

High Density (1XX)

If a bit is present in the  $2^6$  position, the operation to be or was performed was in high density. The absence of this bit says the operation is in low density.

Busy (2XX)

If a bit is present in the  $2^7$  bit position, the unit is busy moving tape.

Program Error (277)

If all the bits are present in this response, a program error has occurred.

For example we send an Information Ready.

Load Point (XX1)

On Request Number 2 if a bit is present in the  $2^0$  bit position, the tape is at load point. This signal drops when tape motion begins again.

End of Tape (XX2)

On Request Number 2 if a bit is present in the  $2^1$  bit position the end of tape marker has been detected. The signal drops when tape has been rewound past the end of tape marker; i.e., the end of tape marker is sensed during rewinding.

Write Not Ready (XX4)

Searching for File Mark (X1X)

8193-601 MAGNETIC TAPE FUNCTION CODES

FUNCTION	SYMBOLIC	OCTAL	FUNCTION	SYMBOLIC	OCTAL
Request Status	RST	1X00	Write	WRT	1X20
Set Interrupt Lockout	SLO	1X02	Write File Mark	WFM	1X21
Clear Interrupt Lockout	CLO	1X03	Read Forward	RF	1X24
Set Odd Parity	SOX	1X10	Read Backward	RB	1X25
Set Odd Parity, Low Density	SOL	1X11	Search Forward	SFR	1X30
Set Odd Parity, Medium Density	SOM	1X12	to Record Gap		
Set Odd Parity, High Density	SOH	1X13	Search Backward	SBR	1X31
Set Even Parity	SEX	1X14	to Record Gap		
Set Even Parity, Low Density	SEL	1X15	Search Forward	SFF	1X32
Set Even Parity Medium Density	SEM	1X16	to File Mark		
Set Even Parity High Density	SEH	1X17	Search Backward	SBF	1X33
			to File Mark		
			Rewind to Load	RWL	1X34
			Point		
			Rewind Unload	RWU	1X35
			Note: (X) is designated by the Unit Select Switch at the top of the Magnetic Tape Transport cabi- net and is limited to Units 0-3.		

8193-601 MAGNETIC TAPE STATUS RESPONSE CODES FOR 8092

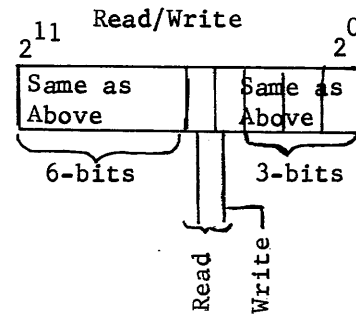
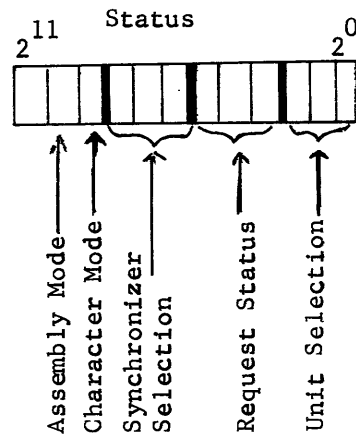
OCTAL

XX1	Even Parity
XX2	Transport Not Ready
XX4	Parity Error
X1X	Program Error
X2X	File Mark
X4X	Tape Mark
1XX	Write Not Ready
2XX	Unit Busy

For description of responses, see 8093, page 84.

G. 162 Magnetic Tape Synchronizer (used with 8094 will handle 1 to 4 603)

External Function	Code
///Status////////////////////////////////////	///
Request Status (6-Bit)	114X
Request Status (12-Bit)	214X
///Write////////////////////////////////////	///
Write, 6-bits, when followed by an Output Instr.	111X
Write End-of-File Mark (no output Instr.)	111X
Write, 12-bits, when followed by an Output Instr.	211X
///Read////////////////////////////////////	///
Read, 6-bits, when followed by an Input Instr.	113X
Read, 12-bits, when followed by an Input Instr.	213X
///Search////////////////////////////////////	///
Backspace one record when followed by an INA Instr.	112X
Backspace one file (No INA Instr.)	112X
Search forward one file	113X
Rewind Unload	115X
Rewind Load	116X

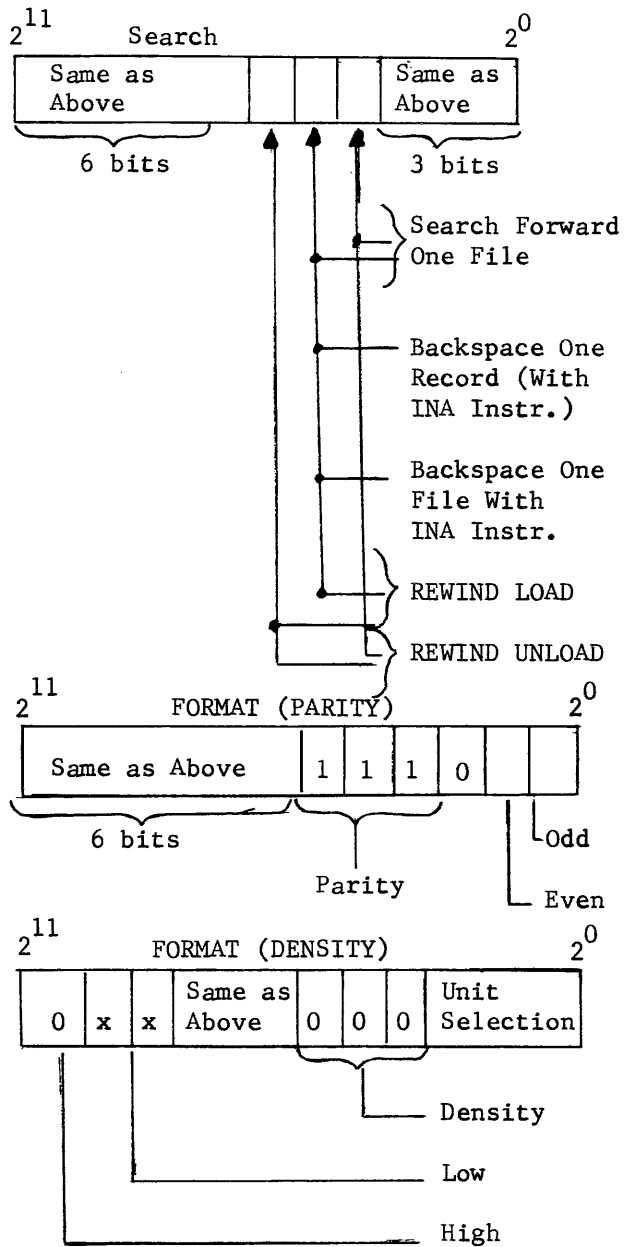


Note: Read and Write must be followed by Input or Output instruction respectively.

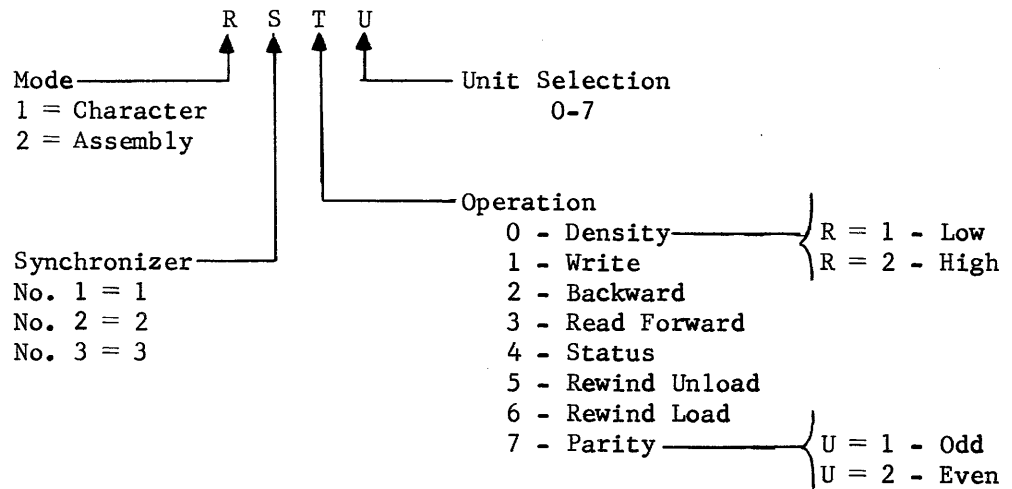
Continued



External Function ////////Format////////	Code ////////
Set to Odd Parity	1171
Set to Even Parity	1172
High Density	210X
Low Density	110X

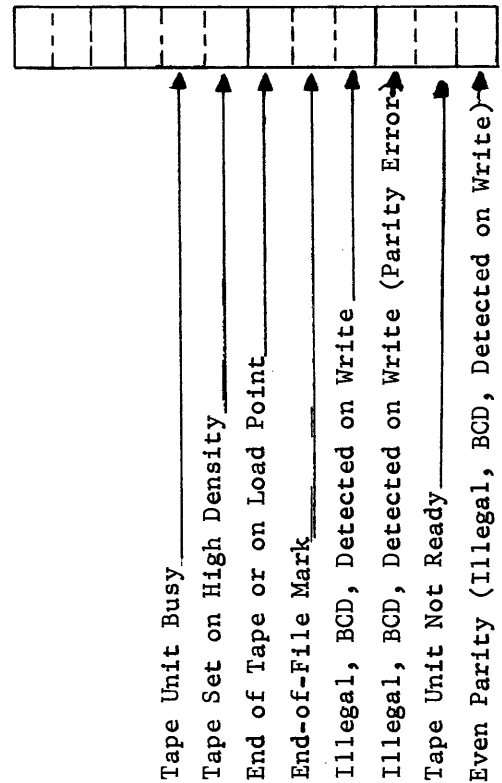


The above external function codes may be represented by:



Status Response	Code
Odd Parity Selected - No Errors	000
Even Parity Selected - No Errors	001
Tape Unit Not Ready	002
Parity Error	004
Illegal BCD Detected on Write	015
End-of-File Mark	020
End of Tape or On Load Point	040
Tape Set On High Density	100
Tape Unit Busy	200

STATUS RESPONSE



Function Code

Request Status (6-bits) (114X)

Request Status (12-bits) (214X)

The 114X/214X code will place a 6-bit or 12-bit status response code, respectively, on the lines. In order for the programmer to check the certain conditions that may appear in the tape units, he must follow the request with an input instruction. However, the response will stay on the lines until a master clear is performed, or taken off the lines by the input instruction. The highest octal group of the select code determines if the request is to be 6, or 12 bits.

WRITE (6-bits) (111X)

WRITE (12-bits) (211X)

These codes select the tape unit and start tape in motion. The information will be written, when the file protection ring is on the tape reel, in either assembly mode (12-bits) or character mode (6-bits) depending upon the highest octal group on the select code. START time is approximately 3 ms and stop time is approximately 2 ms, on the units. The information may be written in high or low density. Both of these codes must be followed with an output instruction or another instruction is interpreted.

WRITE END-OF-FILE MARK (111X)

This code writes  $17_8$  as an end of file mark to signify the end of a record or a group of records.

READ (6-bits) (113X)

READ (12-bits) (213X)

These codes allow the information to be read in either assembly or character mode, depending upon the highest octal group. The density depends upon the previous settings. The file protection ring has no effect on this operation.

BACKSPACE ONE RECORD (112X)

This code will backspace the mag. tape one record, when followed by an input instruction. This instruction differs from the backspace one file by the input instruction that must follow. The input instruction does bring a word into the computer, it may be the check character for part of the record in reverse.

BACKSPACE ONE FILE (112X)

This code will backspace the tape one file, if no file marks are present, the tape will be backspaced to load point.

SEARCH FORWARD ONE FILE (113X)

This code will move tape forward one file, if no file marks are present, the tape will run off the supply reel.

REWIND UNLOAD (115X)

This code will remove all the tape from the takeup reel.

REWIND LOAD (116X)

This code will move the tape to load point.

SET TO ODD/EVEN PARITY (1171/1172)

These codes select the parity to be used under program control.

SELECT HIGH/LOW DENSITY (210X/110X)

These codes select the density to be used under program control.

STATUS CODES

ODD PARITY SELECTED WITH NO ERRORS (0000)

With a status response of all zeros back, the previous parity selection was odd. This would be a pre-status check.

EVEN PARITY SELECTED WITH NO ERRORS (XXX1)

As opposed to the above response, this response says the unit is setting with even parity chosen previously.

TAPE X NOT READY (XXX2)

If a bit is present in this bit position of the response, the tape unit is not in the ready mode. The ready indicator on the tape unit must be pushed at the beginning of the operations to take the unit out of the manual state.

Illegal BCD detected on Write (XX15)

If a bit is present in this bit position, the programmer tried to send the tape unit a code of (0000) while the unit is in the coded or even parity selection. The unit must have at least one bit present to record a frame.

END-OF-FILE MARK READ (XX2X)

On the last read operation the tape unit read an end-of-file mark, which is a  $17_8$  code, used to indicate the end of a record, or a group of records.

END-OF-TAPE OR LOAD POINT SENSED (XX4X)

On the last operation the load point marker or the end-of-tape marker was sensed. Thus, the program can act accordingly.

HIGH DENSITY (X1XX)

This bit indicated that the tape unit is in high density.

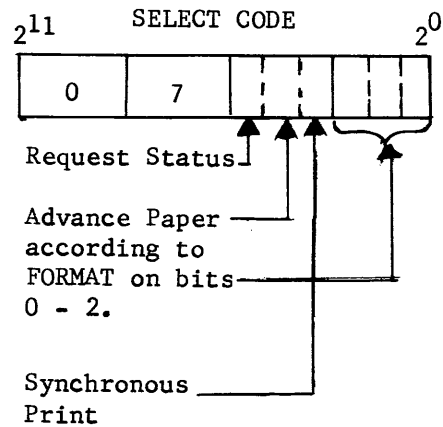
If this bit is not present, the unit is in low density.

TAPE X BUSY (X2XX)

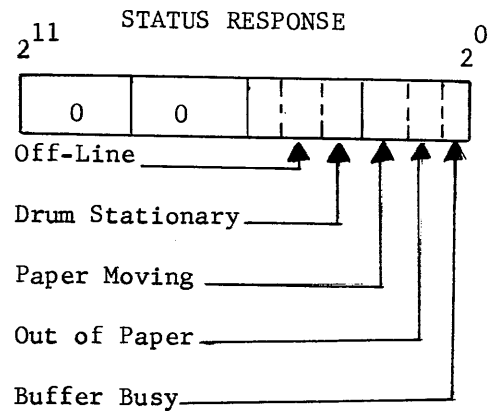
This bit indicates the unit is busy moving tape.

H. 166-2 Line Printer

External Function	Code
Asynchronous print	0700
Synchronous print	0710
Advance Forms	072X
Status Request	0740



Status Response	Code
Printer Ready	000
Buffer Busy	XX1
Out of Paper	XX2
Paper Moving	XX4
Drum Stationary	X1X
Off-Line	X2X



**Function Code:**

**ASYNCHRONOUS PRINT (0700)**

This code will print one line for every revolution of the drum. The print cycle may start at any character on the drum and printing will continue until the starting character is reached. The printer need not receive a full 120 characters. The printer accepts 12 bit words, where bits 6 - 11 designate an odd number hammer, and bits 0 - 5 designate the even numbered hammers. It would take 60 computer words to print one complete line.

**SYNCHRONOUS PRINT (0710)**

On this code the print cycle begins at the character specified by the 62nd word from the computer and terminate prior to printing the character specified by the 61st word from the computer. The 61st and 62nd words use only the lower 6 bits of the computer word. The layout of the drum and code wheel must be known to use this select code.

#### ADVANCE FORMS (072X)

The programmer has seven possible choices for forms line spacing patterns - channels 0-7. The X in the select code above represents the channel which contains the paper advance pattern desired by the programmer.

Level 7 is the format homing channel, to advance paper to the top of the page. The other levels 0-6 may contain a variety of paper advance patterns, one level for each type of form.

#### STATUS REQUEST (0740)

Before selecting or reselecting a print or paper advance operation, the computer should sense conditions within the printer by executing a status request followed by an input to the A-Register or Memory. Then coded information is transferred to the computer's input lines.

#### STATUS CODES

##### PRINTER READY (0000)

A code of all zero's back indicates that the printer is on-line and ready to print.

##### BUFFER BUSY (XXX1)

A bit response in the 2<sup>0</sup> position indicates the buffer is doing an input or output cycle.

##### OUT OF PAPER (XXX2)

This response is used to indicate the forms compartment must be re-filled.



PAPER MOVING (XXX4)

A response of 4 would indicate that a paper advance is in progress.

DRUM STATIONARY (XX1X)

This response says the drum motor is not running and the printer is inoperative.

OFF-LINE (XX2X)

This response indicates the printer is in the off-line mode. The printer generates pseudo select codes to control the tape system and the card reader. These operations may be performed:

TAPE-TO-PRINTER (Asynchronous)

CARD READER-TO-PRINTER (Asynchronous)

CARD READER-TO-TAPE

OFF-LINE OPERATIONS

TAPE-TO-PRINTER

Both READ/WRITE operations with tapes are done in the assembly (12 bit) mode. Information sent to the printer must contain a minimum of 16 to maximum of 120 characters for continuous printing.

The off-line operation may be started by pushing the TAPE/PRINT switch to print, the TAPE/CARD switch to tape and the continuous switch. The stop switch initiates one off-line print cycle (one line). To search for the tape file marks:

1. Place TAPE/CARD switch in TAPE position
2. Place TAPE/PRINT switch in PRINT position
3. Press and hold MC switch

4. Press BACKSPACE switch for a search back or step  
for a search forward and FINALLY
5. Release MC switch when tape motion stops.

#### CARD-TO-PRINTER

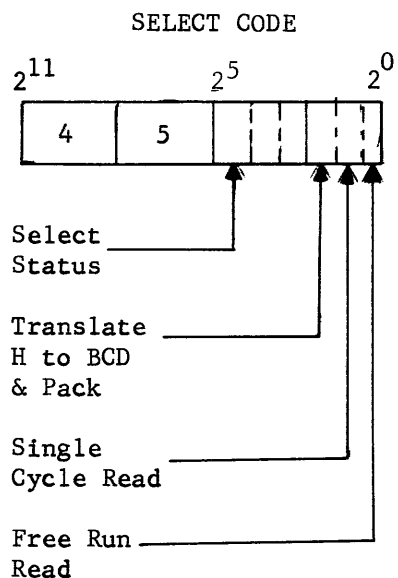
The card reader will read and send all 80 columns of information to the printer, however, only the first columns are used in printing a line. Hollerith coded cards may be used with the 167-2 which contains a H to BCD and pack translator. To initiate this operation place the TAPE/PRINT switch in the print position, the TAPE/CARD switch in the card position, and hit either step (for one line) or continuous.

#### CARD-TO-TAPE

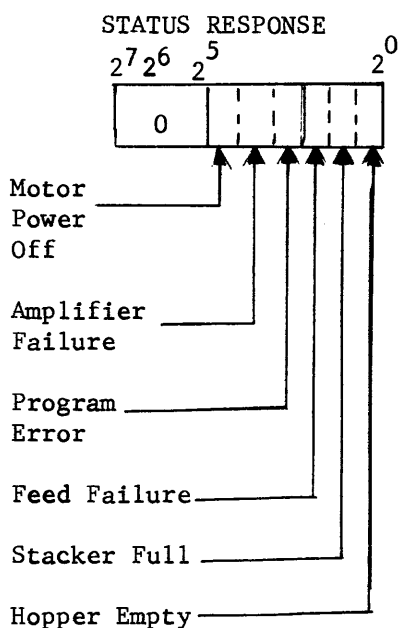
When writing on tape from BCD coded cards, use the binary mode if any unpunched character locations are on the card. To initiate the operation, place the TAPE/PRINT switch to TAPE, THE TAPE/CARD switch to card and press the continuous switch. To terminate the operation press the STOP switch on the card reader.

I. 167-2 Card Reader (8094 required)

External Function	Code
EF Clear	4500
Free Run Read	4501
Single Cycle Read	4502
FRR, H to BCD & Pack	4505
SCR, H to BCD & Park	4506
Check Status	4540



Status Response	Code
Card Reader Ready	000
Input Hopper Empty	XX1
Output Stacker Full	XX2
Feed Failure	XX4
Program Error	X1X
Amplifier Failure	X2X
Motor Power Off	X4X



FUNCTION CODE

EF CLEAR (4500)

This code sets up the initial conditions in the hardware, and is also used to terminate a Free Run Read Operation. If the card reader is to stop immediately after reading a card, the code must be received within 1.5 ms after reading the 80th column of the card. It may be also used on the single cycle read operation at any time during the cycle.

FREE RUN READ (4501)

This code actuates clutch and starts a card in movement. Cards move continually without need for reselection between cards. The operation terminates when the card reader receives an EF CLEAR or generates an input disconnect. An input disconnect results from:

1. a feed failure
2. a late input request (program error), or
3. an amplifier failure

SINGLE CYCLE READ (4502)

This code also actuates the clutch and starts a card in movement. However, only one card is read. To read the next card, another select must be issued.

Full speed operation is possible if reselection is within 4 ms after the 80th column is read.

FRR, H to BCD and PACK (4505)

In this mode, data from cards punched in Hollerith code format is translated into a 6-bit BCD format. After the translation the reader

packs two 6-bit words into one word. The complete card is held in  $40_{10}$  words, as opposed to the binary read where each column is assembled into one word and the complete card is held in  $80_{10}$  words. The operation is terminated as stated in the FREE RUN READ CODE.

SCR, H to BCD and PACK (4506)

In this mode, the data is translated and packed, and one card is read for each selection.

CHECK STATUS (4540)

A Status Request directed to the card reader and followed by an input to A provides the computer with information related to the card reader status.

STATUS RESPONSES

CARD READER READY (0000)

A code of all zero's back to the computer indicates that the card reader is ready to read cards.

INPUT HOPPER EMPTY (XXX1)

This response indicates to the programmer that he has completed his operation, that is read the last card in the hopper, and he may refill the hopper.

OUTPUT STACKER FULL (XXX2)

This response indicates to the programmer that the card stacker is full and before any more cards may be read the stacker should be emptied to read the next card for no card will be read.

FEED FAILURE (XXX4)

This response indicates that an input instruction was given, and for some mechanical reason the card failed to move. An INPUT DISCONNECT signal will be generated.

PROGRAM ERROR (XX1X)

The program error response is issued if an input request (INA or INP) is not present as each column pulse rises. An INPUT DISCONNECT Signal will be generated.

AMPLIFIER FAILURE (XX2X)

Before each card read operation, a check is made on the light amplifiers. If any amplifier is not functioning properly an input disconnect signal is generated. Turn on the row 12 "0" check disable switch if cut-corner cards are used.

MOTOR POWER OFF (XX4X)

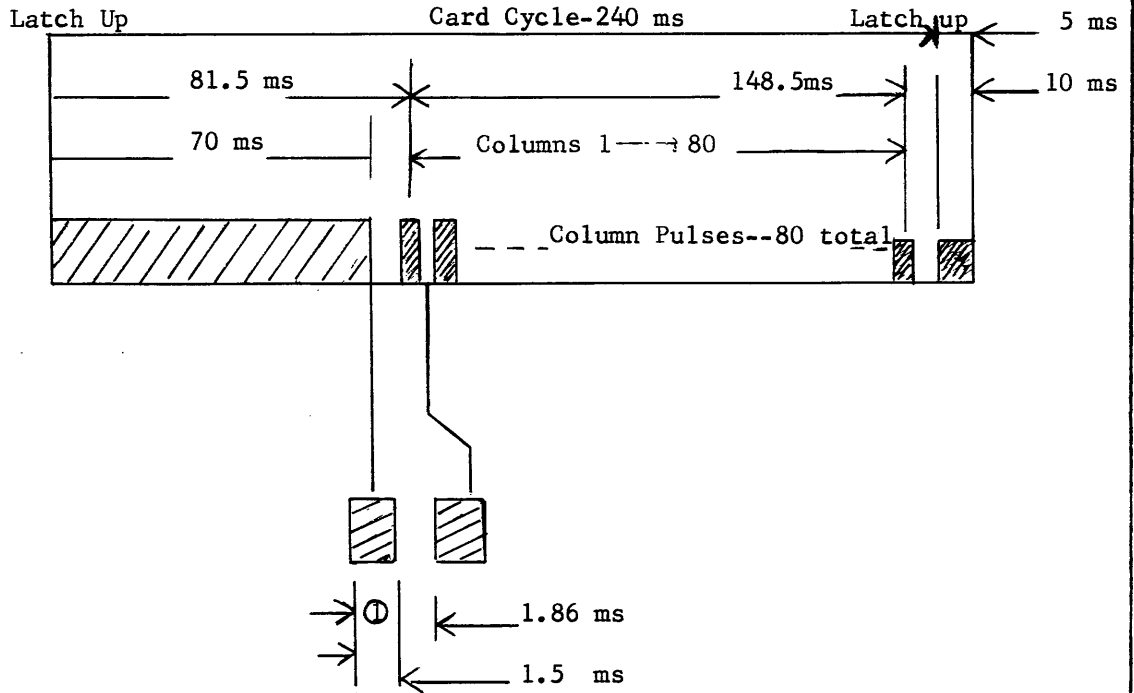
This response indicates the motor power switch had not been turned on.

THE INPUT DISCONNECT SIGNAL

The input instruction may establish a storage field block of greater capacity than the anticipated input block. This signal indicates to the computer that the card reader has no more data to deliver, so the computer may return to its main program with no further delay.

PROGRAM TIMING

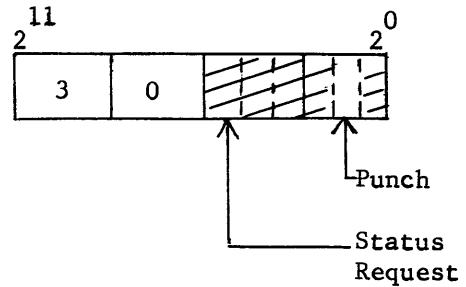
Each card requires 240 ms to pass through the read station. The card read may send up to 40 translated or 80 untranslated 12-bit words to the interrogating device.



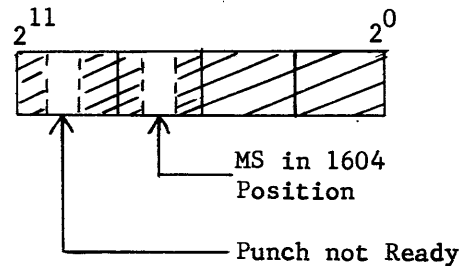
1. Shaded areas indicate computer program time available between column input requests.

J. 170 Card Punch Control Unit (8094 required)

External Function	Code
EF Clear	3000
Punch	3002
Check Status	3040



Status Response	Code
170 Ready	0000
MS in 1604 Position	X2XX
Punch not ready	2XXX



FUNCTION CODES

PUNCH (3002)

This code connects the 170 to the computer output channel. It also prepares the 170 to accept and transform data from computer into 80-column simulated card-row and send it to the 415 punch station at punching time for each of the 12 rows on one card. The punch selection is cleared after all 12 rows are processed (one complete card cycle).



CHECK STATUS (3040)

Allows the computer to obtain a status response from the 170 card punch controller. The code must be followed by an input instruction.

EF CLEAR (3000)

This code initializes the hardware in the 170.

STATUS RESPONSES

170 READY (0000)

This code indicates that the manual switch is in the 160 position and the controller is ready to accept data.

MS IN 1604 POSITION (X2XX)

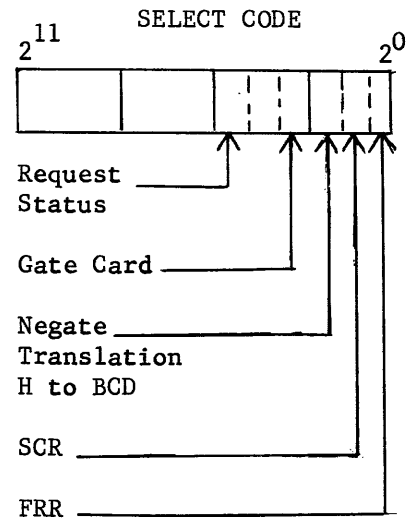
This code indicates that the manual switch is in the 1604 computer position and must be changed before the 160 may use it.

PUNCH NOT READY (2XXX)

This code indicates the 415 punch is not ready, in other words it is in the manual state, and must be set ready to put it under control of the computer.

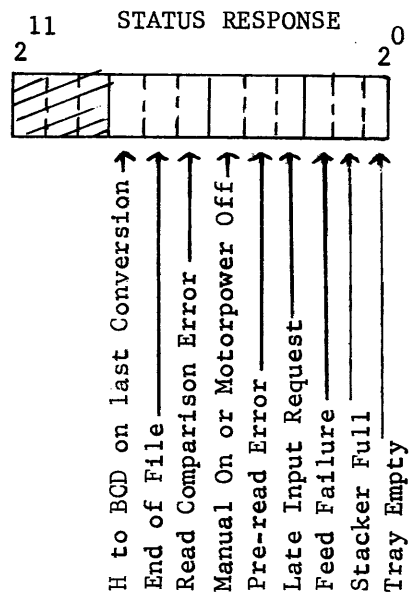
K. 177 Card Reader Controller (8094 required)

External Function	Code
EF Clear	4500
Free Run Read	4501
Single Cycle Read	4502
Negate Translation, H to BCD	4504*
Gate Card	4510
Status Request	4540



\* This code is used with the 4501 or 4502 code to choose the type of read. The codes become 4505 and 4506.

Status Response	Code
Input Tray Empty	XX1
Primary or Secondary Stacker Full	XX2
Feed Failure	XX4
Late Input Request	X1X
Pre-Read Error	X2X
Manual on or Motor Off	X4X
Read Comparison Error	1XX
End of File	2XX
H to BCD Conversion	4XX



## EXTERNAL FUNCTION CODES

### EF CLEAR (4500)

This code clears the function select FF's and the error circuits in the 177 logic.

### FREE RUN READ (FRR) (4501)

This code allows the cards to be read continuously without the need to reselect between cards. The FRR is terminated and an input disconnect generated if one or more of the following conditions occur:

1. Input tray empty
2. Feed failure
3. Late input request
4. Pre-read error
5. Read comparison error
6. Stacker full (either stacker)

### SINGLE CYCLE READ (SCR) (4502)

This code will cycle one card. The reading of the card may be terminated at any point. The 177 will operate at full speed if the SCR instruction is issued within 1.5 ms after each card is read. An input disconnect is generated if the input area length exceeds the number of words available from the card reader.

### NEGATE TRANSLATION H to BCD (4505, 4506)

In normal operation, the 177 checks column one of each card for a 7 and 9 punch. If a 7 and 9 punch exists, the card is read as binary card. If the 7 and 9 punch does not exist, the card is read as a

binary card. If the 7 and 9 punch does not exist, the card is considered Hollerith and the data is translated into BCD. These codes allow cards without the 7 and 9 punch to be read as binary. The 4505 is the FRR and negate translation, and the 4506 is the SCR and negate translation.

#### GATE CARD (4510)

This code allows a limited card sorting operation. The selected cards are directed to the secondary stacker. A gate card instruction must be issued for each card to be gated. The instruction must arrive with 1.5 ms after the last column of the selected card has been read.

#### STATUS REQUEST (4540)

A Status Request instruction followed by an input instruction provides the computer with the reader status. This code may be issued any time previous to a read operation or after the last word desired from a card. This code clears all select circuits.

#### STATUS RESPONSES

##### INPUT TRAY EMPTY (XXX1)

This code indicates the input tray is empty, and is used with the feed failure response to indicate the reader has read all the cards in the input tray on a FRR operation.

##### PRIMARY OR SECONDARY STACKER FULL (XXX2)

This code indicates either the primary or the secondary stacker is full. Before another card can be read the condition must be corrected.

FEED FAILURE (XXX4)

This code indicates that an input instruction was given, and because of some mechanical reasons, the card failed to move.

LATE INPUT REQUEST (XX1X)

This code will be issued if an input request (INA or INP) is not present as each column pulse rises. An input disconnect signal will be generated.

PRE-READ ERROR (XX2X)

A pre-read error indicates that a read amplifier was not functioning properly previous to the reading of the last card.

MANUAL ON OR MOTOR POWER OFF (XX4)

This code is used to indicate that the card reader is not in the automatic state, or the motor power switch is off.

READ COMPARISON ERROR (X1XX)

This condition exists if the data read at the second read station did not agree with the data read at the first read station.

END OF FILE (X2XX)

The end of file condition exists when the End of File switch is on. The End of File switch should be in the OFF position unless the input tray contains the last card of a file.

H to BCD CONVERSION (X4XX)

This response indicates the last card read was converted from Hollerith to BCD.

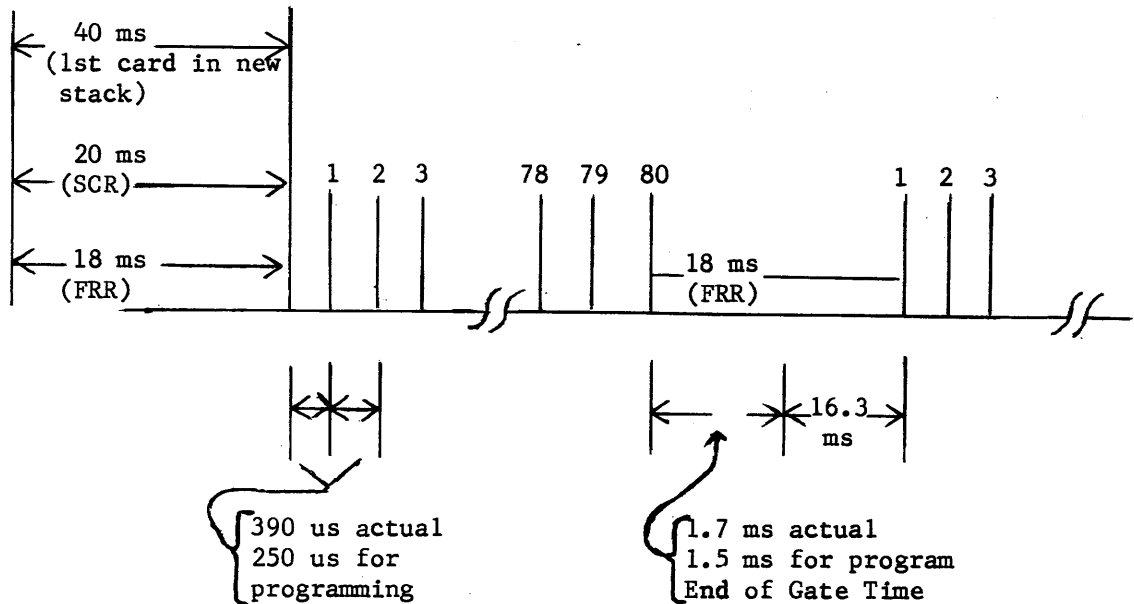
An input disconnect is generated if one of the following conditions exist:

1. Input tray empty
2. Feed failure
3. Late input request
4. Pre-read error
5. Read comparison error
6. Primary or Secondary Stacker Full

Also excluding the late input request, and including the manual switch on, another card cannot be read until the conditions are cleared out.

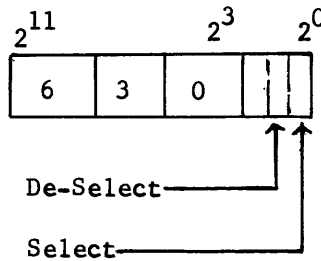
PROGRAMMING TIME.

The reader may send up to 40 translated and packed or 80 untranslated 12-bit words to the computer from each card.



L. 8094 Peripheral Adaptor

External Function	Code
Select	6301
De-Select	6302



NO STATUS RESPONSE CODE

FUNCTION CODES

SELECT (6301)

This code allows the Teleprogrammer to communicate with equipment that ordinarily would be impossible because of the difference of the 12-bit interface logic. This code selects the peripheral adaptor, then another select code is used to select another piece of equipment to carry out the communication.

DE-SELECT (6302)

Most peripheral equipment when selected stays selected until another piece of equipment is selected. This is not the case of the peripheral adaptor, once selected it stays selected until its own unique DE-SELECT CODE is sent out.

M. 8060 Series Digital Communications Terminals

External Function	Code
Select Stop Send	36X0
Select Send	36X1
Select Data	36X2
Select Status	36X3

Status Response	Code
Sending	000
Receiving	001
Word Request	002
Word Ready	003

EXTERNAL FUNCTION CODE

SELECT STOP SEND (36X0)

This code clears the send conditions and switches the status of data condition. This code also turns off the Interrupt System.



SELECT SEND (36X1)

This code places the DCU in a transmitting mode and causes a Request Send Output to the modem. Then when the modem is prepared to accept data from the computer, it generates a signal.

SELECT DATA (36X2)

This code selects the receive mode for the DCU and allows data to be passed to the computer.

SELECT STATUS (36X3)

Status is enabled to the computer when the computer follows with an input instruction. It is not necessary to examine status in order to send, or receive information. For the computer is interrupted automatically every seven milliseconds.

N. 8095 Record Transmission Control Panel

External Function	Code
Select to Input Data	35XX
Status Request 1	3501
Status Request 2	3502

Status Response	Code
Send End of Message	001
Send Come to Phone	002
End of Message not Acknowledged	004
Come to Phone not Acknowledged	010
Stop (Not Ready)	020
Translate	100
Send Move (Not Receive)	200

Status Response 2 "Sending Equipment"	Code	Bit Positions Containing "1"
Magnetic Tape	0X0	None
Punched Cards	0X1	00
Paper Tape	0X2	01
I/O Writer	0X4	02
A (Unassigned)	1X1	00 and 06
B (Unassigned)	1X2	01 and 06
C (Unassigned)	1X4	02 and 06
Status Response 2 "Receiving Equipment"		
Off	00X	None
Magnetic Tape	01X	03
Punched Cards	02X	04
Line Printer	04X	05
Paper Tape	21X	03 and 07
I/O Writer	22X	04 and 07
A (Unassigned)	24X	05 and 07

FUNCTION CODE

SELECT TO INPUT DATA (35XX)

The 8095 must be in a selected condition before it can accept a data word from the 8092. Once selected it remains selected until another piece of equipment is selected or cleared by a Master Clear Signal.

The following data words activate the 8095:

Bit Position Containing "1"	
00	Light End of Message and Stop lamps
01	Light Come to Phone Lamp
02	Extinguish End of Message lamp and light
03	Extinguish Come to Phone lamp
04	Light Remote Trouble lamp
05	Extinguish Remove Trouble lamp
06	Light Local Trouble and Stop lamps
07	Light Three Attempt Failure and Stop lamps
NOTE: A "1" in bit positions 0, 1, 6, 7 cause the audible alarm	

STATUS REQUEST 1 (3501)

Status Response 1 tells the 8092 which push buttons have been pressed. A "1" in each bit position has the unique meaning listed in status response 1's table. The status response can have a "1" in more than one bit position to convey more than one message.

STATUS RESPONSE 2 (3502)

Status Response 2 tells the 8092 the settings of the Select Send and Select Receive switches. The response word contains 8 bits. The Select Receive switch and the Select Send switch settings are conveyed in each status response 2 word. The Status Response 2 table lists the bit positions which are a "1" for each of the Select Receive and Select Send switch positions.

Select Send & Receive

Function

Select Send, Switch

This Response indicates which peripheral equipment will furnish data to be transferred to other peripheral equipment or sent to a Records Transmission Terminal.

Select Receive, Switch

This response tells the 8092 which type of peripheral equipment will receive transferred or incoming data.

O. 8022/915 Page Reader Function Codes

FUNCTION	SYMBOLIC	OCTAL	FUNCTION	SYMBOLIC	OCTAL
Select Page Advance	SPA	6701	Request Status #1	RS1	6740
Select Forward Mirror Position	SMF	6702	Request Mirror Status	RMS	6741
Select Reverse Mirror Position	SMR	6703	Request Status #2	RS2	6742
Start Coordinate Search	SCS	6704	Start Read	LR	6744
Zero Mirror	ZM	6705	Stop Read	STP	6747
Line Locator	LL	6707	Sort to Primary	PS	6750
Alpha-Numeric Read	ANM	6710	Sort to Secondary	SS	6751
Alpha Read	ALF	6711	Program Error Alarm	APR	6752
Numeric Read	NUM	6712	Header Reject Alarm	AHR	6753
Circle Filled/Unfilled Read	MKS	6713	Mark Document	MK	6757
Scan 3 Character Heights	R3P	6714	Select Partial	PAR	6761
Scan 2 Character Heights	R6P	6715	Select Sub Total	SUB	6762
Advance Counter 1	CT1	6730	Select Total	TOT	6764
Clear Counter 1	CL1	6731	Clear Partial	CPR	6771
Advance Counter 2	CT2	6732	Clear Sub Total	GST	6772
Clear Counter 2	CL2	6733	Clear Total	CTT	6774
Advance Counter 3	CT3	6734	Clear Entry	GEN	6777
Clear Counter 3	CL3	6735			

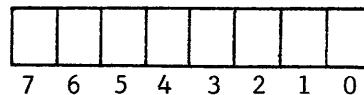
915 OCR STATUS RESPONSE CODES

REQUEST STATUS 1 (6740)		REQUEST STATUS 2 (6742)	
<u>OCTAL</u>		<u>OCTAL</u>	
2XX	Ready	2XX	Ready
1XX	Busy	1XX	Busy
X4X	Program Error	X4X	End of File
X2X	Compare	X2X	Partial
X1X	Reject	X1X	Subtotal
XX4	Mirror Far Reverse	XX4	Total
XX2	Mirror Far Forward		
XX1	Data Ready		

## 8022 CONTROLLER STATUS RESPONSE CODES

The 8022 Controller provides three status response words: one for each of the Status Requests 1, 2, and Mirror. The response to Request Mirror Status (RMS) provides the computer with the horizontal coordinate number of the current mirror position. The responses to Request Status 1 and 2 (RS1, RS2) contain all other status information which the Reader can pass to the computer.

The response words to Request Status 1 and 2 are eight bit words in which each bit position presents one item of status information. In the following explanation the bit positions will be referred to in accordance with the power of two which they represent:



8 bit word  
bit position number

Response to Request Status 1 (RS1)

Bit #	OCTAL	STATUS
7	2XX	Ready

When bit #7 is set to "1", the Reader and Controller are in "ready" status.

When bit #7 = "0", Reader and Controller are not "ready".

When status is "ready":

1. No malfunctions exist.
2. A document is at "document ready position".
3. "Start" button has been depressed.

Bit #	OCTAL	STATUS
6	1XX	Busy

1 = Reader and Controller "busy"

0 = Not busy

The Reader and Controller are busy only when a CLASS 1b function is being performed. (SCS, ZM, LL, PS, SS, MK) When status changes from busy to not busy, the Controller generates an Interrupt 40 to the computer.

Bit #	OCTAL	STATUS
5	X4X	Program Error

1 = Program Error

Program Error status occurs when the Reader fails to find a Line Locator symbol in response to a Line Locator search function.

Bit #	OCTAL	STATUS
4	X2X	Compare

1 = Compare

Compare is indicated when mirror position equals the coordinate to which the mirror has been directed, i.e., the mirror has completed the action last requested of it.

When the mirror is in motion in a forward direction the compare also generates a note "busy" status which lasts for 400u before the mirror is stopped. This change of status from "busy" to not "busy" causes an Interrupt 40.

Bit #	OCTAL	STATUS
3	X1X	Reject

1 = Reject



The reject status indicates that a reject code ( $35_8$ ) was generated for an unreadable character during the last read operation.

Bit #	OCTAL	STATUS
2	XX4	Mirror Far Reverse

1 = Mirror Far Reverse

This function indicates the mirror is positioned against the left most limits switch.

Bit #	OCTAL	STATUS
1	XX2	Mirror Far Forward

1 = Mirror Far Forward

Indicates the mirror is positioned at the right most limits switch.

Bit #	OCTAL	STATUS
0	XX1	Data Character Ready

1 = Character Ready

Indicates the Reader has recognized a character and is waiting to be inputed.

Program Error, Compare and Reject are reset to zero with the initiation of a CLASS 1b function.

Response to Request Status 2 (RS2)

Bit #	OCTAL	STATUS
7	2XX	Ready
6	1XX	Busy

Ready and Busy Status indicators are available in both response words for convenience in programming.

5	X4X	End of File
4	X2X	Enter Partial
3	X1X	Enter Subtotal
2	XX4	Enter Total

1 = Entry  
0 = No Entry

Entry of any of this group of status indicators is accomplished by depressing the corresponding entry switch on the operator's control panel, while the Reader is in a stopped condition, and an appropriate select function has enabled the switch. On depressing the Start button to permit the computer to receive these indicators the controller will generate an Interrupt 30 if entries have been made.

Bits #1 and #0 are not used in Request Status 2 response word.

APPENDIX A

TAS INSTRUCTION REPERTOIRE

The following symbols are used to describe the operation of the instructions in the following lists.

- t = tag register designation (t - 1, 2, 3 or blank)
- f = function code
- m = 8-bit word execution address (may include address 2 as additive unless n specified)
- n = same as m, but second address
- y = 8-bit operand (may include address 2 as additive unless z specified)
- z = same as y, but second address, no additive allowed

A. 8092 Codes

Loads		Function	Octal Code	Number Words	Cycle Time
LDN	y	Load A, no addr	20	2	2
LDM,t	m	Load A, memory	21	2	3
LDI,t	m	Load A, indirect	22	2	4
LCM,t	m	Load A, compl., memory	25	2	3
LCI,t	m	Load A, compl., indirect	26	2	4
TTA,t		Tag register to A	03	1	1
CLA		Clear A to zero	03	1	1
BER		Buffer Entrance Reg to A	06	1	1

Stores

STM,t	m	Store A, memory	41	2	3
STI,t	m	Store A, indirect	42	2	4
ATT,t		A to Tag register	02	1	1
ABR,t	m	A to Buffer Entrance Reg	04	2	1/2 *
ABX,t	m	A to Buffer Exit Reg	05	2	1/2 *

8092 Codes (continued)

Jumps		Function	Octal Code	Number Code	Cycle Code
ZJP,t	m	Jump, if A=0 (positive)	60	2	1/2 *
NZP,t	m	Jump, if A≠0 (positive)	61	2	1/2 *
PJP,t	m	Jump, if A positive	62	2	1/2 *
NJP,t	m	Jump, if A negative	63	2	1/2 *
UJP,t	m	Unconditional jump	64	2	2

Shift

SHA		A left 1 bit, circular	01	1	1
-----	--	------------------------	----	---	---

Arithmetics

ADN	y	Add, no addr	30	2	2
ADM,t	m	Add, memory	31	2	3
ADI,t	m	Add, indirect	32	2	4
SBN	y	Subtract, no addr	34	2	2
SBM,t	m	Subtract, memory	35	2	3
SBI,t	m	Subtract, indirect	36	2	4
RAM,t	m	Replace Add, memory	51	2	4
RAO,t	m	Replace Add One, memory	55	2	4

Logicals

LPN	y	Logical Product, no addr	10	2	2
LPM,t	m	Logical Product, memory	11	2	3
LPI,t	m	Logical Product, indirect	12	2	4
LSN	y	Logical Sum, no addr	14	2	2
LSM,t	m	Logical Sum, memory	15	2	3
LSI,t	m	Logical Sum, indirect	16	2	4

8092 Codes (continued)

I/O		Function	Octal Code	Number Words	Cycle Time
INN,t	m n	Input, normal channel	72	3	3+2(n+1)
OUT,t	m n	Output, normal channel	73	3	3+2(n+1)
IBI,t	m	Input, buffer channel	70	2	1/2 *
IBO,t	m	Output, buffer channel	71	2	1/2 *
INA		Input to A	76	1	2
OTN	y	Output, no address	74	2	2
EXF	y z	External select	75	3	3

Other

CIL		Clear interrupt lockout	13	1	1
CIR		Clear interrupt, return	113	1	1
CBC		Clear buffer control	07	1	1
ERR		Program halt	00	1	1
HLT		Program halt	77	1	1
DON		Do Nothing	02	1	1

\* 2 cycles are required if a jump is made.

B. 8092 Pseudo-Ops

SHA	y	Generate y shifts (1 - 7)
SPL	y	Space y lines on listing
REM		Remarks only follow
EQU	y	Equate tag to tag, or tag to constant
ORG	y	Set address counter to y
BSS	y	Increment address counter by y
BCD	y ...	Preset storage to following characters

8092 Pseudo-Ops (continued)

TTY	y ...	Preset storage to following teletype (Model 33) codes
(blank)	y	One-word constants can be generated by leaving the Op-field blank and mnemonics, decimal or octal constants in the address fields.
TAS		Initiate assembly, loc. field contains Prog. ident.
END		Terminate assembly pass

DETAILED INSTRUCTION DESCRIPTIONS

8092 PSEUDO-OPS

- SHA      If the y-field for this instruction is not blank, it will be interpreted and the number of shifts specified, up to 7, will be generated. If interpretation of the y-field results in a value greater than 7, seven shifts will be generated.
- SPL      The y-field is interpreted and the result specifies the number of lines to be skipped on the listing. If the end of a page is reached, any remaining skips are discarded. In this connection, a y-field value of 60D or more is equivalent to a "skip page" function.
- REM      A line containing this pseudo-op is printed without modification.
- EQU\*     The y-term is interpreted and the result equated to the tag found in the location field.
- ORG\*     The y-term is interpreted and the result substituted in the location counter.
- BSS\*     The y-term is interpreted and the result added to the location counter.

8092 Pseudo-Ops (continued)

- BCD\*        The y-term is interpreted and the result indicates the number of BCD characters in the comments field. The 6-bit octal equivalents of these characters are generated and stored in consecutive locations. If the y-field value exceeds the Comments field (39 chars. max.), 39 characters will be interpreted and an error indicated on the listing.
- TTY\*        Same function as BCD except 8-bit teletype codes are generated.
- (blank)     Constants are indicated by a blank op-code. (If the entire card is blank, it will be ignored.) The y-term is interpreted and the result becomes the contents of the current location.
- TAS         The first line accepted by the assembler must be a TAS pseudo-op. The first four characters in the location field serve to identify the binary output of the assembly.
- END         The last line processed for an assembly must be an END pseudo-op.



C. 915 Codes

Class 1a	Function	Cycle Time	Octal Code	Number Words
SPA	Select page advance	3	756701	3
SMF	Select mirror forward	3	756702	3
SMR	Select mirror reverse	3	756703	3
ANM	Select alpha-numeric	3	756710	3
ALF	Select alpha read	3	756711	3
NUM	Select numeric read	3	756712	3
MKS	Select mark sense read	3	756713	3
R3P	Read 3 per inch (double space)	3	756714	3
R6P	Read 6 per inch (single space)	3	756715	3

Class 1b

SCS	Start coordinate search	3	756704	3
ZM	Position mirror to coordinate zero	3	756705	3
LL	Line locator	3	756707	3
PS	Sort to primary	3	756750	3
SS	Sort to secondary	3	756751	3
MK	Mark document	3	756757	3

Class 1c

APE	Alarm prog. error	3	756752	3
AHR	Alarm header reject	3	756753	3

915 Codes (continued)

Class 2	Function	Cycle Time	Octal Code	Number Words
GEN	Clear Entry, Partial, total, subtotal, EOF	3	756777	3
CT1	Advance counter 1	3	756730	3
CT2	Advance counter 2	3	756732	3
CT3	Advance counter 3	3	756734	3
CL1	Clear counter 1	3	756731	3
CL2	Clear counter 2	3	756733	3
CL3	Clear counter 3	3	756735	3
PAR	Select partial	3	756761	3
SUB	Select subtotal	3	756762	3
TOT	Select total	3	756764	3
CPR	Clear partial	3	756771	3
CST	Clear subtotal	3	756772	3
CTT	Clear total	3	756774	3

Class 3

RS1	Request status 1	3	756740	3
RS2	Request status 2	3	756742	3
RMS	Request mirror coord.	3	756741	3
LR	Line Read	3	756744	3
STP	Stop read and mirror	3	756747	3

D. 915 MACROS

		Function	Number Words
LMV	y	Start mirror moving left to coordinate y	8
RMV	y	Start mirror moving right to coordinate y	8
SKP	y	Start mirror skipping y lines	8
SKL	y z	Start mirror moving left to coordinate y and skipping z lines	13
EMV,t		Wait for LMV, RMV or SKL operation to complete. <u>t</u> must indicate tag register referencing the EMV instruction location	8
POS,t	y	Wait until mirror position y has been reached. <u>t</u> must indicate tag register referencing the POS instruction location.	8
LIN,t	y z	Input a line to the buffer area referenced by tag register <u>t</u> , starting bank address y, ending address plus one z. (Right mirror motion must previously have been initiated.)	6
RNB,t	y	Test Page Reader status for "READY" and "NOT BUSY". If both conditions are not met, jump to location specified by <u>t</u> and y.  (y = * for wait function)	9
RDY,t	y	Test Page Reader status for "READY". Jump to location specified by <u>t</u> and y if "NOT READY".  (y = * for wait function)	6

## 915 MACROS (continued)

	Function	Number Words
NBY,t      y	Test Page Reader status for "NOT BUSY". Jump to location specified by <u>t</u> and y if "BUSY".  (y = * for wait function)	7
LOC,t	Wait up to 100 ms attempting to find line locator symbol. If found, A = positive, on exit. <u>t</u> must indicate tag register referencing the LOC instruction location.	11
RJC,t      y	Test Page Reader status for "Line Reject". Jump to location specified by <u>t</u> and y if "Line Reject" is set.	8

APPENDIX B

APPENDIX B

MOD 33 CODES

<u>Character</u>	<u>Octal Code</u>	<u>Character</u>	<u>Octal Code</u>	<u>Character</u>	<u>Octal Code</u>
A	301	{	333	b	266
B	302	\	334	?	267
C	303	}	335	B	270
D	304	↑	336	9	271
E	305	—	337	:	272
F	306	Sp	240	:	273
G	307	!	241	<	274
H	310	"	242	=	275
I	311	#	243	>	276
J	312	¢	244	?	277
K	313	%	245	EOT	204
L	314	&	246	WRU	205
M	315	†	247	RU	206
N	316	(	250	BELL	207
O	317	)	251	TAB	211
P	320	*	252	LINE FEED	212
Q	321	+	253	VT	213
R	322	,	254	FORM	214
S	323	-	255	RETURN	215
T	324	.	256	X ON	221
U	325	/	257	TAPE	222
V	326	0	260	X OFF	223
W	327	1	261	ACK	374
X	330	2	262	ALT MODE	375
Y	331	3	263	RUB OUT	377
Z	332	4	264		
@	300	5	265		

APPENDIX C

APPENDIX C

EXTERNAL BCD CODES

<u>Character</u>	<u>Octal Code</u>	<u>Character</u>	<u>Octal Code</u>
Blank	20	L	43
0	12	M	44
1	01	N	45
2	02	O	46
3	03	P	47
4	04	Q	50
5	05	R	51
6	06	S	22
7	07	T	23
8	10	U	24
9	11	V	25
.	73	W	26
-	40	X	27
+	60	Y	30
=	13	Z	31
Δ	14	%	34
/	21	}	74
#	37	,	33
&	52	?	76
⊕	53	:	77
⊗	54	;	16
A	61	[	32
B	62	REJ	35
C	63	▽	36
D	64	↑	55
E	65	↓	56
F	66	⊚	57
G	67	┌	72
H	70	└	75
I	71	:	15
J	41	—	17
K	42		