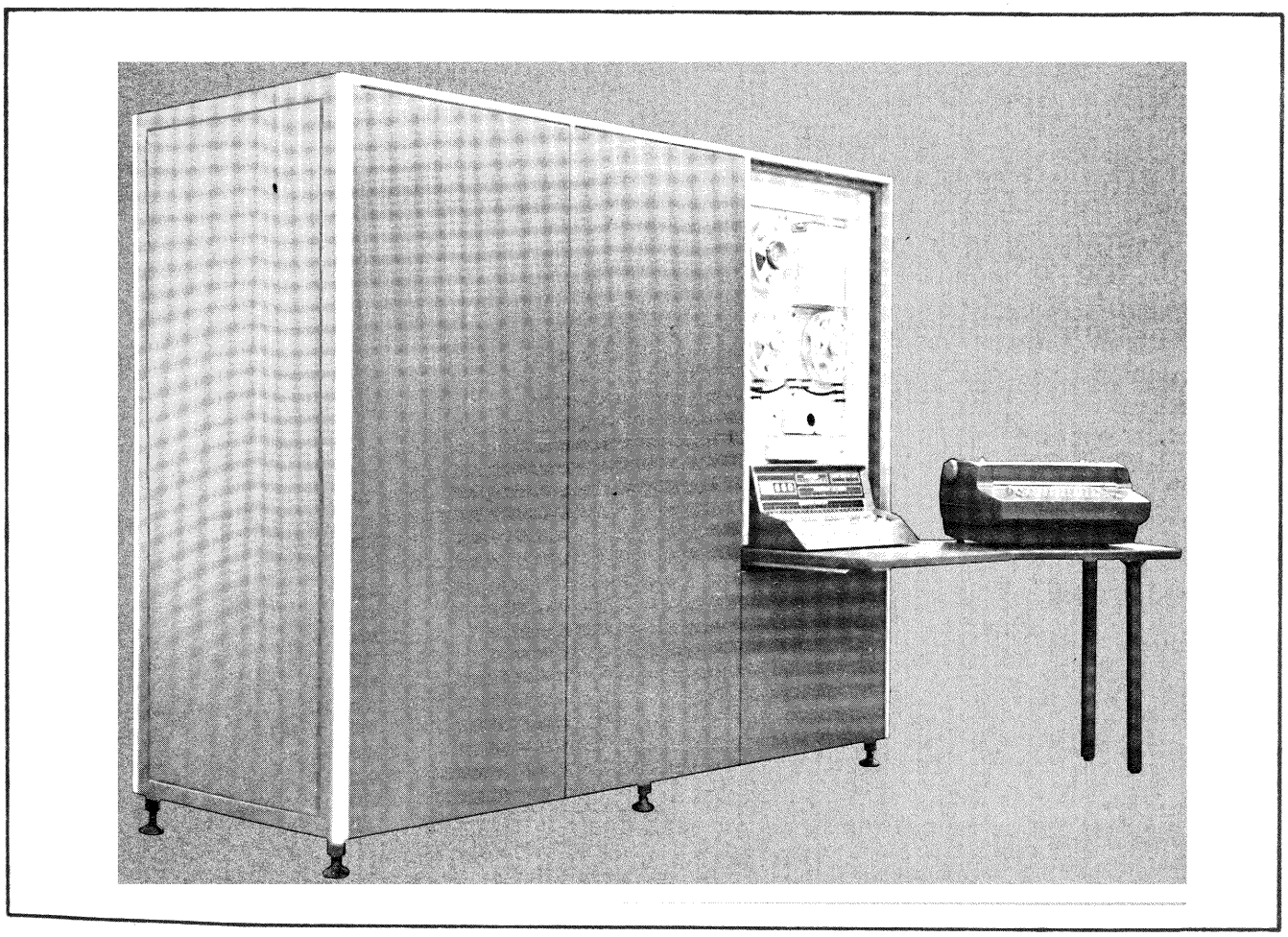
SCIENTIFIC CONTROL CORPORATION

SCC 660 COMPUTER reference manual



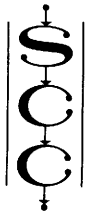
SCC 660 BASIC INSTRUCTIONS

Mnemonic	Code	Instruction Name
MEMORY/REGISTER		
LDA	27	Load AC
LDE	75	Load EA
LDX	71	Load XR
LAC	45	Load AC with Complement
LXC	44	Load XR with Complement
EAX	77	Effective Address into XR
STA	35	Store AC
STE	36	Store EA
STX	37	Store XR
SAF	42	Store Address Field of AC
XXM	76	Exchange XR and Memory
REGISTER/REGISTER		
CLR	0 46 30000	Clear AC and EA
AEC	0 46 20000	AC to EA, Clear AC
EAC	0 46 10000	EA to AC, Clear EA
XAE	0 46 00000	Exchange AC and EA
ARITHMETIC		
ADD	55	Add Memory to AC
SUB	54	Subtract Memory from AC
MPY	64	Multiply AC by Memory
DIV	65	Divide AC,EA by Memory
MIN	61	Memory Increment
MDC	60	Memory Decrement
LOGICAL		
AND	14	AND Memory with AC
ORA	16	OR Memory with AC
EOR	17	Exclusive OR Memory with AC
SHIFT/CYCLE		
LSH	0 67 000xx	Left Shift AC,EA
ALS	0 67 040xx	Left Shift AC
ELS	0 67 020xx	Left Shift EA
LCY	0 67 200xx	Left Cycle AC,EA
ALC	0 67 240xx	Left Cycle AC
ELC	0 67 220xx	Left Cycle EA
NDX	0 67 100xx	Normalize and Decrement XR
SND	0 67 140xx	Short Normalize and Decrement XR
RSH	0 66 000xx	Arithmetic Right Shift AC,EA
ARS	0 66 040xx	Arithmetic Right Shift AC
ERS	0 66 020xx	Right Shift EA
RCY	0 66 200xx	Right Cycle AC,EA
ARC	0 66 240xx	Right Cycle AC
ERC	0 66 220xx	Right Cycle EA
BRANCH		
BRA	01	Branch
BAP	26	Branch if AC Positive
BAN	24	Branch if AC Negative
BAZ	25	Branch if AC Zero
BEN	22	Branch if EA Negative
BIX	41	Increment XR and Branch if Negative
BSL	43	Store Location Counter and Branch
BRT	51	Return Branch
BRI	11	Branch and Clear Interrupt
SKIP		
SMN	53	Skip if Memory Negative
SAG	73	Skip if AC Greater Than Memory
SAE	70	Skip if AC Equals Memory with EA Mask
SAM	72	Skip if AND of AC With Memory Equals Zero
SNS	40	Skip if Signal Not Set
SBK1	0 40 20400	Skip if Breakpoint 1 Reset
SBK2	0 40 20200	Skip if Breakpoint 2 Reset
SBK3	0 40 20100	Skip if Breakpoint 3 Reset
SBK4	0 40 20040	Skip if Breakpoint 4 Reset
SNC	0 40 24000	Skip if CRO Reset
SOV	0 40 20001	Skip if OVF Reset
SIE	0 40 20004	Skip if Interrupts Enabled
SPC	0 40 20002	Skip if Interrupts Enabled by Program
BER	0 40 20010	Skip if No Error, B Buffer
CER	0 40 20020	Skip if No Error, C Buffer
BRD	0 40 21000	Skip if B Buffer Ready
CRD	0 40 22000	Skip if C Buffer Ready
PRDY	0 40 10060	Printer Ready
PLRDY	0 40 10062	Plotter Ready
CRDY	0 40 10006	Card Reader Ready
SNCRE	0 40 11006	Skip If No Card Reader Error
SRHNE	0 40 14006	Skip If Reader Hopper Not Empty
TRDY	0 40 1041x	Skip If Tape Ready
SNR	0 40 1401x	Skip If No Ring
SLP	0 40 1201x	Skip If At Load Point
SET	0 40 1101x	Skip If At End of Tape
SDEN5	0 40 1021x	Skip If Density 556
SDEN8	0 40 1061x	Skip If Density 800
SNG	0 40 10010	Skip If No Gap
MISCELLANEOUS		
HLT	00	Halt
NOP	20	No Operation
XEC	23	Execute
INPUT/OUTPUT		
TMB	12	Transfer Memory to B Buffer
TBM	32	Transfer B Buffer to Memory
TMC	10	Transfer Memory to C Buffer
TCM	30	Transfer C Buffer to Memory
WTP	13	Write Parallel
RDP	33	Read Parallel
ACT	02	Activate
DSC	0 02 00000	Disconnect B Buffer
DSCC	0 02 00100	Disconnect C Buffer
TOP	0 02 14000	Terminate Output, B Buffer
TOPC	0 02 14100	Terminate Output, C Buffer
ARMB	0 02 12000	Arm B Buffer Interrupt
DRMB	0 02 11000	Disarm B Buffer Interrupt
ARMC	0 02 12100	Arm C Buffer Interrupt
DRMC	0 02 11100	Disarm C Buffer Interrupt
RTY	0 02 01601	Read Typewriter
WTY	0 02 01641	Write Typewriter
RPT	0 02 01604	Read Paper Tape
WPT	0 02 01644	Write Paper Tape
PRN	0 02 01660	Print
PLOT	0 02 01662	Activate Plotter
RCB	0 02 03606	Read Card Binary
CRO	0 02 12006	Card Reader Offset
REW	0 02 1401x	Rewind
RUN	0 02 1201x	Rewind and Unload
CRS	0 02 10400	Convert Read to Scan
RTB	0 02 0361x	Read Tape Binary
RTD	0 02 0261x	Read Tape Decimal
SFB	0 02 0363x	Scan Forward Binary
SFD	0 02 0263x	Scan Forward Decimal
SRB	0 02 0763x	Scan Reverse Binary
SRD	0 02 0663x	Scan Reverse Decimal
WTB	0 02 0365x	Write Tape Binary
WTD	0 02 0265x	Write Tape Decimal
ETF	0 02 0367x	Erase Tape Forward
ETR	0 02 0767x	Erase Tape Reverse
INTERNAL CONTROL		
ARMI	0 02 20040	Arm Internal Interrupt
DRMI	0 02 20020	Disarm Internal Interrupt
ENA	0 02 20002	Enable Interrupts
DIS	0 02 20004	Disable Interrupts
TOV	0 02 20001	Turn Off OVF
TOC	0 02 20010	Turn Off CRO

Price: $3.00

SCC 660 COMPUTER
REFERENCE MANUAL

January 1968



SCIENTIFIC CONTROL CORPORATION/14008 Distribution Way/Dallas, Texas/(214) CH1-2111

Printed in U.S.A.

CONTENTS

I SYSTEM DESCRIPTION ____ 1

General Characteristics ____ 1
Central Processor ____ 2
Organization ____ 2
Instruction Word Format ____ 2
Indexing and Indirect Addressing ____ 3
Fixed Point Format ____ 3
Floating Point Format ____ 3
Memory ____ 4
Organization ____ 4
Memory Protection ____ 5
Input/Output ____ 5
Organization ____ 5
Transfer Rates ____ 6
Buffer Input/Output ____ 6
Operation ____ 6
Data Flow ____ 6
Data Flow Control ____ 7
Computer Control ____ 7
Channel Controller Control ____ 8
C Register Input/Output (Parallel I/O) ____ 9
Operation ____ 9
Data Flow ____ 9
Interrupts ____ 9
Operation ____ 9
Processing ____ 9
Priority ____ 9
Arm/Disarm ____ 10
Enable/Disable ____ 10
Programmed Operators ____ 10
Operation ____ 10
Invalid Operation Codes ____ 10

II MACHINE INSTRUCTIONS ____ 11

General ____ 11
Memory/Register ____ 11
LDA --- Load AC ____ 11
LDE --- Load EA ____ 11
LDX --- Load XR ____ 12
LAC --- Load AC With Complement ____ 12
LXC --- Load XR With Complement ____ 12
EAX --- Effective Address Into XR ____ 12
STA --- Store AC ____ 12
STE --- Store EA ____ 13
STX --- Store XR ____ 13
SAF --- Store Address Field of AC ____ 13
XXM --- Exchange XR With Memory ____ 13
Register/Register ____ 13
CLR --- Clear AC and EA ____ 13
AEC --- AC to EA, Clear AC ____ 13
EAC --- EA to AC, Clear EA ____ 14
XAE --- Exchange AC and EA ____ 14
Arithmetic ____ 14
ADD --- Add Memory to AC ____ 14
SUB --- Subtract Memory from AC ____ 15
MPY --- Multiply AC by Memory ____ 15
DIV --- Divide AC, EA by Memory ____ 16
MIN --- Memory Increment ____ 16
MDC --- Memory Decrement ____ 16
Logical ____ 16
AND --- AND Memory With AC ____ 16
ORA --- OR Memory With AC ____ 17
EOR --- Exclusive OR Memory With AC ____ 17
Shift Cycle ____ 17
Direction of Shift ____ 17
Type of Shift ____ 18
Places to Shift ____ 19
Execution Time of Shifts ____ 19
LSH --- Left Shift AC, EA ____ 19
ALS --- Left Shift AC ____ 20
ELS --- Left Shift EA ____ 20
LCY --- Left Cycle AC, EA ____ 20
ALC --- Left Cycle AC ____ 20
ELC --- Left Cycle EA ____ 21
NDX --- Normalize and Decrement XR ____ 21
SND --- Short Normalize and Decrement XR ____ 22
RSH --- Arithmetic Right Shift AC, EA ____ 23
ARS --- Arithmetic Right Shift AC ____ 23
ERS --- Right Shift EA ____ 23
RCY --- Right Cycle AC, EA ____ 24
ARC --- Right Cycle AC ____ 24
ERC --- Right Cycle EA ____ 24
Branch ____ 24
BRA --- Branch ____ 25
BAP --- Branch If AC Positive ____ 25
BAN --- Branch If AC Negative ____ 25
BAZ --- Branch If AC Zero ____ 25
BEN --- Branch If EA Negative ____ 25
BIX --- Increment XR and Branch If Negative ____ 25
BSL --- Store Location Counter and Branch ____ 25
BRT --- Return Branch ____ 25
BRI --- Branch and Clear Interrupt ____ 25
BSL, BRT Operation ____ 26
BSL, BRI Operation ____ 27
Skip ____ 28
SMN --- Skip if Memory Negative ____ 28
SAG --- Skip if AC Greater Than Memory ____ 28
SAE --- Skip If AC Equals Memory Thru EA Mask ____ 28
SAM --- Skip If AND of AC With Memory Equals Zero ____ 28
SNS --- Skip If Signal Not Set ____ 29
Internal SNS Instructions ____ 29
SBK1 --- Skip if Breakpoint 1 Reset ____ 30
SBK2 --- Skip if Breakpoint 2 Reset ____ 30
SBK3 --- Skip If Breakpoint 3 Reset ____ 30
SBK4 --- Skip If Breakpoint 4 Reset ____ 30
SNC --- Skip If CRO Reset ____ 30
SOV --- Skip If OVF Reset ____ 30

SIE — — — Skip If Interrupts Enabled 31
SPC — — — Skip If Interrupts Enabled By Program 31
BER — — — Skip If No Error, B Buffer 31
CER — — — Skip If No Error, C Buffer 31
BRD — — — Skip If B Buffer Ready 31
CRD — — — Skip If C Buffer Ready 31
Miscellaneous 31
HLT — — — Halt 31
NOP — — — No Operation 31
XEC — — — Execute 32
Executing an ADD 32
Executing a Branch 33
Executing a Skip 34
Input/Output 35
TMB — — — Transfer Memory To B Buffer 35
TBM — — — Transfer B Buffer To Memory 35
TMC — — — Transfer Memory To C Buffer 35
TCM — — — Transfer C Buffer To Memory 36
WTP — — — Write Parallel 36
RDP — — — Read Parallel 37
Activate Instructions 37
ACT — — — Activate 37
Buffer Control Activate 37
Input/Output Device Codes 41
DSC — — — Disconnect B Buffer 42
DSCC — — — Disconnect C Buffer 42
Device Control Activate 43
TOP — — — Terminate Output, B Buffer 44
TOPC — — — Terminate Output, C Buffer 44
ARMB — — — Arm B Buffer Interrupt 44
DRMB — — — Disarm B Buffer Interrupt 44
ARMC — — — Arm C Buffer Interrupt 45
DRMC — — — Disarm C Buffer Interrupt 45
External Control Activate 45
Internal Control Activate 45
ARMI — — — Arm Internal Interrupt 46
DRMI — — — Disarm Internal Interrupt 46
ENA — — — Enable Interrupts 46
DIS — — — Disable Interrupts 46
TOV — — — Turn Off (Reset) OVF 47
TOC — — — Turn Off (Reset) CRO 47

III INTERRUPT SYSTEM 48

General 48
Single Instruction Interrupt Program 48
Multi-Instruction Interrupt Program 49
Interrupt Locations 49
Privileged Instructions 49
Priority 49
Arming/Enabling 49
Console Enable Switch 50
Internal Interrupt 50
Buffer Interrupt 50

IV PERIPHERALS 51

Typewriter 51
Typewriter Instructions 51
RTY — — — Read Typewriter 51
WTY — — — Write Typewriter 51
Typewriter I/O Examples 51
Typewriter Output 51
Typewriter Input 52
Output Under Interrupt Control 53
Paper Tape 54
Paper Tape Instructions 55
RPT — — — Read Paper Tape 55
WPT — — — Write Paper Tape 55
Paper Tape I/O Examples 55
Input of Variable Length Record 55
Input of Fixed Length Record 56
Input of Variable Length Record Under Interrupt Control 57
Output Under Interrupt Control 58
Line Printer 59
Printer Controls 59
Format Loop 59
Line Printer Instructions 59
PRN — — — Print 59
PRDY — — — Printer Ready 59
Carriage Control 59
Printer Examples 61
Single Line Output 61
Multi-Line Output 62
Plotter 67
Plotter Instructions 67
PLOT — — — Plot 67
PLRDY— — — Plotter Ready 67
Plotter Commands 67
Card Reader 67
Operating Procedures 68
Reader Control Panel 68
Card Reader Instructions 69
RCB — — — Read Card Binary 69
CRO — — — Card Reader Offset 69
CRDY — — — Card Reader Ready 69
SNCRE— — — Skip If No Card Reader Error 69
SHRNE— — — Skip If Read Hopper Not Empty 69
Magnetic Tape 69
Magnetic Tape Test Instructions 69
TRDY — — — Tape Ready 69
SNR — — — Skip If No Ring 69
SLP — — — Skip If At Load Point 69
SET — — — Skip If At End of Tape 69
SDEN5— — — Skip If Density 556 69
SDEN8— — — Skip If Density 800 70
SNG — — — Skip If No Gap 70
Magnetic Tape Control Instructions 70
REW — — — Rewind 70
RUN — — — Rewind And Unload 70
CRS — — — Convert Read To Scan 70
Read Operation 70
RTB — — — Read Tape Binary 70
RTD — — — Read Tape Decimal 70
Scan Operation 70
SFB — — — Scan Forward Binary 70
SFD — — — Scan Forward Decimal 70
SRB — — — Scan Reverse Binary 70
SRD — — — Scan Reverse Decimal 70
Write Operation 70
WTB — — — Write Tape Binary 70

WTD – – – Write Tape Decimal 70
Erase Operation 70
ETF – – – Erase Tape Forward 71
ETR – – – Erase Tape Reverse 71

V CONTROL CONSOLE 72

Console Switches 72
Control Panel 72
Memory Protect 72
Breakpoint 72
Start 72
Clear 72
Register 72
Bootstrap 72
Store 73
I/O Display 73
Interrupt 73
Run 73
Test 73
Step 73
Power 73
Displays 73
Register Display 73
Program Location 73
Input/Output Unit 73
Control 73
SCC 660 Console 74

APPENDIXES

APPENDIX A NUMBER SYSTEMS 75

APPENDIX B LOGICAL OPERATIONS 78

APPENDIX C POWERS OF TWO 80

APPENDIX D POWERS OF TEN EXPRESSED IN OCTAL 81

APPENDIX E MATHEMATICAL CONSTANTS 82

APPENDIX F OCTAL – DECIMAL TABLE 83

APPENDIX G OCTAL – DECIMAL FRACTION CONVERSION TABLE 86

APPENDIX H BAUDOT CODE 89

APPENDIX I CHARACTER CODES 90

SECTION I

SYSTEM DESCRIPTION

SYSTEM DESCRIPTION

The SCC 660 is a general purpose, high speed, binary computer with a single address structure allowing indexing and indirect addressing. Salient features such as hardware multiply and divide, fully parallel internal processing, memory protection, a priority interrupt system and automatic subroutine linkage provide a high system performance capability. Optional features include two character buffer units, channel controllers for controlling buffer operation and a memory access system permitting fully simultaneous data transfer over four input/output channels to any four of eight memory modules.

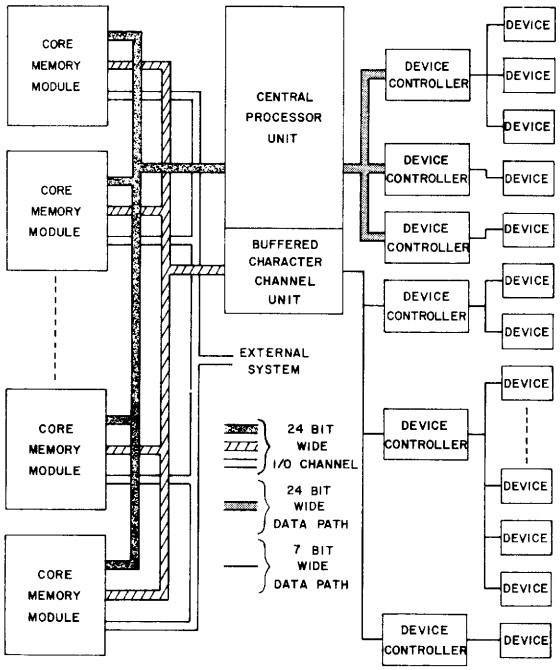
The entire magnetic core memory of the SCC 660 is directly addressable and consists of one to eight independently addressable memory modules. Each memory module has a 1.75 usec memory cycle and contains an address register, a data register and 4096 twenty-four bit words of core storage. The memory accessing structure may vary from a single input/output channel communicating with a single memory module to a sophisticated fully overlapped system allowing simultaneous communication by four input/output channels with any four of eight memory modules.

The SCC 660 computer system includes a wide range of standard and special purpose peripheral equipment. Four models of 7 and 9 track, IBM compatible magnetic tape units are available with data transfer rates to 120,000 characters per second on the high speed units and data transfer rates of 20,800 characters per second on the low cost units. Magnetic disc units with capacities to 1,048,574 characters per unit, 64 tracks containing 16,384 characters per track and an average access time of 16.7 milliseconds due to a fixed read/write head for each track are available for systems requiring additional rapid-access data storage.

The card equipment reads and punches data in binary and card codes with read speeds up to 1000 cards per minute and punch speeds up to 300 cards per minute. The paper tape equipment reads with speeds up to 300 characters per second and punches up to 110 characters per second. The line printer is fully buffered with speeds up to 1000 lines per minute, 132 characters per line and 64 different characters. Low speed input/output devices offered as standard equipment are keyboard typewriter/printers, teletypewriters and plotters.

Software for the SCC 660 includes a FORTRAN compiler and an assembler for SPL, Symbolic Programming Language. Mathematical subroutines and plotter subroutines are provided to assist the programmer in scientific applications. Utility programs and diagnostic aids are provided to assist the programmer in such areas as media conversion and program debugging.

TYPICAL SCC 660 SYSTEM



CENTRAL PROCESSOR

Organization

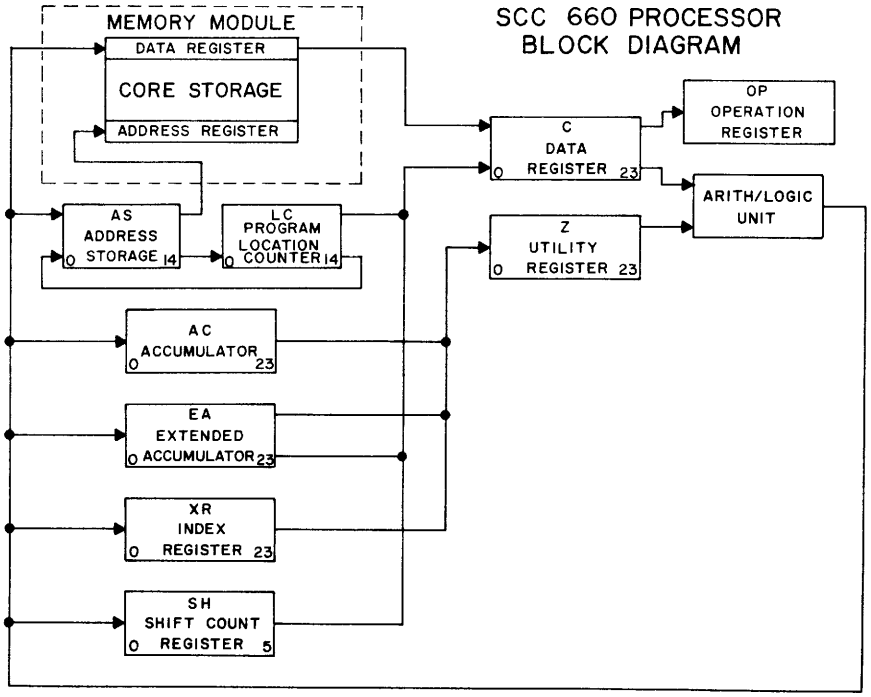
Parallel data transfer, a repertoire of 59 instructions including hardware multiply and divide, automatic subroutine linkage, indirect addressing, indexing, memory protection and a priority interrupt system provide the SCC 660 with a powerful, high speed Central Processor Unit.

The CPU (Central Processor Unit) contains two accumulators (AC, EA) an index register (XR), a data register (C), a memory address register (AS), a program location counter (LC), a shift count register (SH), an operation register (OP), a utility register (Z) and an arithmetic/logic unit. Indicators include an overflow indicator (OVF) and an adder carry out indicator (CRO). The accumulators and the index register are directly programmable. Arithmetic/logic unit operation is programmable through a microinstruction capability.

The AC and EA are 24 bit accumulators for arithmetic and logic operations with the EA serving as an extension of the AC in operations involving a double word operand. XR, the 24 bit index register, serves as an address modifier and as an auxiliary register. The C register accepts data from the memory unit and serves as a utility register for the ALU (arithmetic/logic unit).

The AS register is used by the CPU to address the memory unit. LC, the Location Counter, contains the address of the instruction being executed. During execution of each instruction, the LC is incremented by one (normal instruction progression), incremented by twos (skip instructions) or set to a new value (branching instruction). The SH register holds the shift count for operations involving shifting or rotating of the AC or EA. The OP register contains the operation code of the instruction being executed. The Z register functions strictly as a utility register for the ALU.

SCC 660 PROCESSOR
BLOCK DIAGRAM

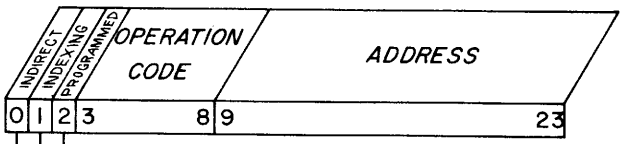


Instruction Word Format

Instructions are single address and occupy one word, resulting in a basic instruction execution time of 2 memory cycles for instructions requiring an operand fetch from memory. (One cycle to fetch the instruction and one cycle to fetch the operand and execute the instruction.)

The instruction word utilizes bits 9-23 to specify a 15 bit address, bits 3-8 to specify an operation, bit 2 to designate a programmed operation, bit 1 to specify indexing, and bit 0 to specify indirect addressing.

INSTRUCTION WORD



0 BITS 3–8 SPECIFY MACHINE OPERATION
1 BITS 2–8 SPECIFY CORE LOCATION WHICH CONTAINS ADDRESS OF PROGRAMMED OPERATION

0 NO INDEXING
1 INDEXING; ADD CONTENTS OF XR AND ADDRESS FIELD

0 NO INDIRECT ADDRESSING
1 INDIRECT ADDRESSING; AFTER INDEXING, EXAMINE RESULTING LOCATION.

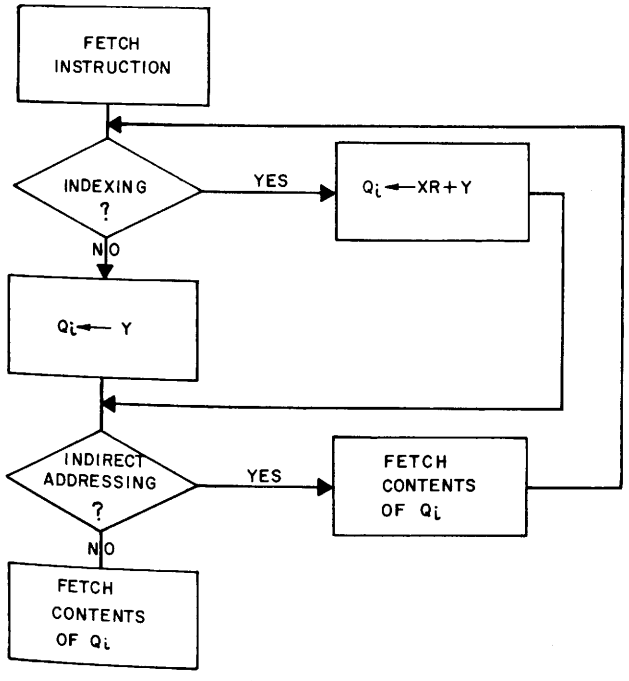
Indexing and Indirect Addressing

Bits 3-8 of a machine instruction word specify the operation code and bits 0, 1, 9-23 specify the core memory location of the data. The actual memory location is determined by examining the address field (bits 9 thru 23), the indexing field (bit 1) and the indirect address field (bit 0). The actual core location is referred to as the effective address or effective location and its contents is referred to as the operand.

If bit 0 and bit 1 are zero, the address field (bits 9 thru 23) specifies the core memory location to be used. If bit 1 is a one (indexing) and bit 0 is a zero, the contents of XR (index register) and the address field are added together. This sum specifies the core location to be used. The contents of XR and the address field are not altered.

If bit 0 is a one (indirect address) and bit 1 is a zero (no indexing), the CPU will fetch the word at the location specified by the address field. The CPU will then examine the address field, indexing field and indirect addressing field of this word to determine the actual core location to be used. If bit 0 and bit 1 are ones (indirect addressing and indexing), the contents of XR and the address field are added together and the CPU examines the word at the location specified by this sum to determine the actual core location to be used. There are no restrictions upon this procedure. In other words, indexing and indirect addressing may be specified in each word examined.

EFFECTIVE OPERAND FETCH

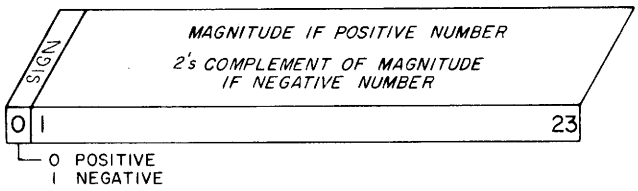


Q_i denotes intermediate address
Y denotes contents of bits 9-23
← denotes "is specified by"

Fixed Point Format

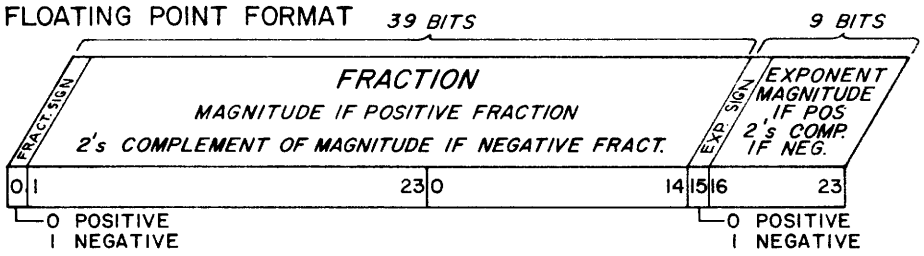
The SCC 660 fixed point format allows the representation of signed integers ranging in size from -8,388,608 to +8,388,607. Integers are represented internally as a sign followed by 23 binary digits. Bit 0 is the sign bit with 0 denoting positive and 1 denoting negative. If the sign is positive, bits 1 thru 23 contain the binary representation of the magnitude of the number. If the sign is negative, bits 1 thru 23 contain the two's complement of the magnitude of the number.

FIXED POINT FORMAT



Floating Point Format

The SCC 660 floating point format allows a representation of numbers with up to 12 significant digits and an exponent range of 10^{-72} to 10^{+72}. Floating point numbers are represented internally in two words with a 39 bit fraction portion and a 9 bit exponent portion. The fraction portion consists of a sign bit and a 38 bit binary fraction. The exponent portion consists of a sign bit and an 8 bit binary exponent. Bit 0 of word 1 is the sign of the fraction with 0 denoting positive and 1 denoting negative. If the fraction is positive, the magnitude of the fraction is contained in bits 1 of word 1 thru 14 of word 2. If the fraction is negative, the 2's complement of the magnitude is contained in bit 1 of word 1 thru bit 14 of word 2. Bit 15 of word 2 contains the sign of the exponent with 0 denoting a positive exponent and 1 denoting a negative exponent. If the exponent is positive, bits 16-23 contain the magnitude of the exponent. If the exponent is negative, bits 16-23 contain the 2's complement of the magnitude.



MEMORY

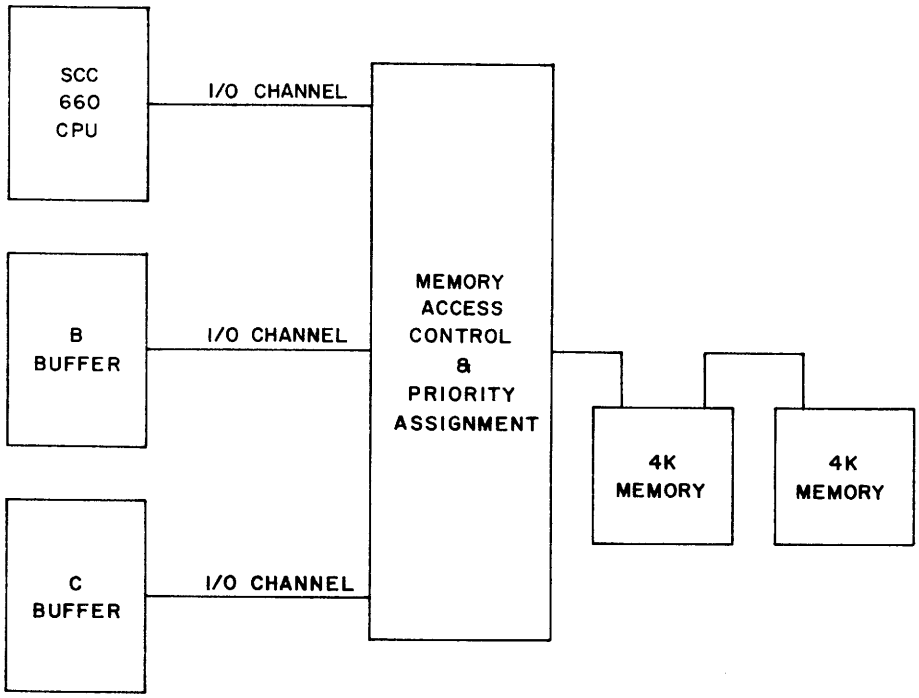
Organization

The core memory of a SCC 660 system consists of 1 to 8 modules of 4096 twenty-four bit words with a 1.75 usec memory cycle time. Each module contains an address and data register allowing independent addressing of the modules. The memory accessing structure allows a wide range of configurations from three channels accessing a single module to four channels independently and simultaneously accessing any four of eight modules. Simultaneous requests of the same module are handled on a priority basis with the priority assignment being a system variable. In a typical configuration using three I/O channels, the priority assignment is external system, B buffer, and CPU with the external system having the highest priority.

Two systems are illustrated, one system permits one of three I/O channels to access one of two memory modules. In the other system, three I/O channels are allowed to simultaneously access any three of seven memory modules. If the C buffer and CPU request the same module, access is given to the C buffer since the designated priority assignment is C buffer, B buffer, CPU in descending priority.

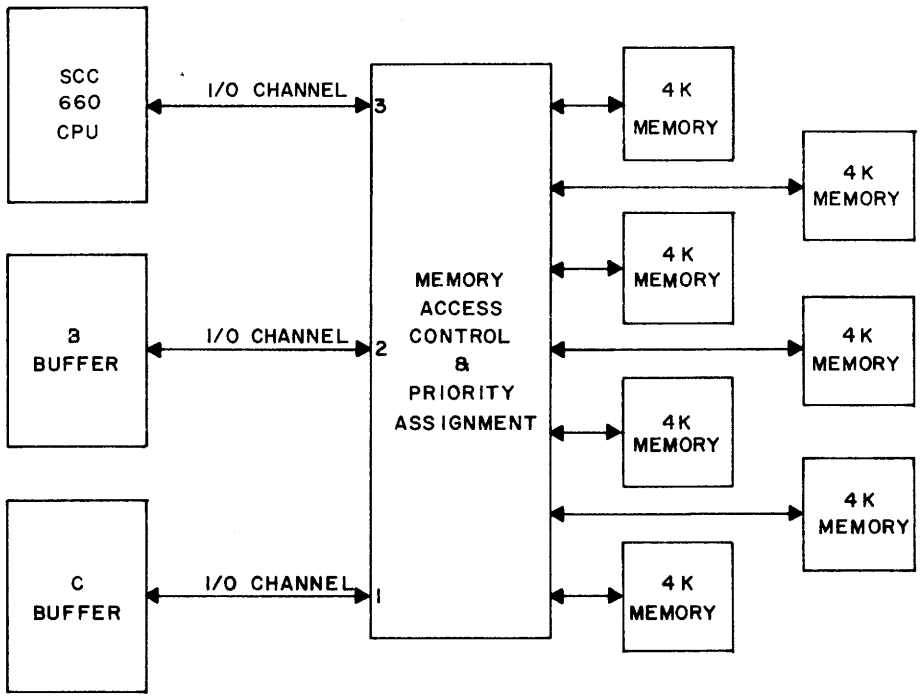
SINGLE MEMORY ACCESS SYSTEM

3 I/O CHANNELS—2 MEMORY MODULES



MULTIPLE MEMORY ACCESS SYSTEM

3 I/O CHANNELS — 7 MEMORY MODULES



Memory Protection

Program integrity is maintained through a hardware memory protection capability which prevents writing in memory locations 0_8 thru 777_8. Memory protection may be enabled or disabled from the control console.

INPUT/OUTPUT

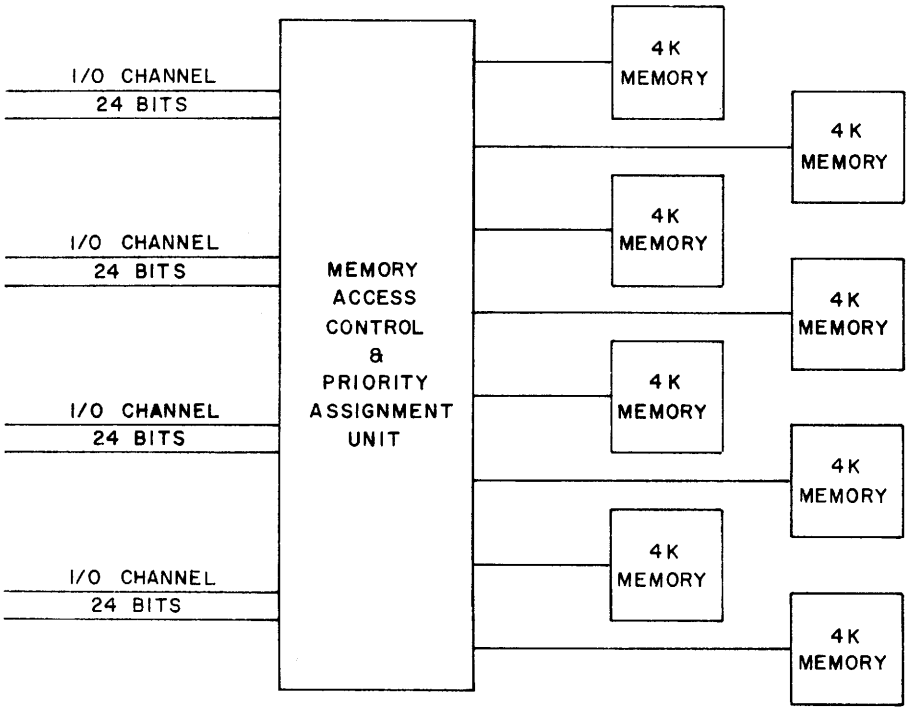
Organization

The SCC 660 input/output structure allows complete overlapping of input, processing and output operations and permits systems external to the SCC 660 to directly access the SCC 660 memory modules. Thus an external system may provide information to a SCC 660 system and obtain information from the system efficiently with a minimum of software and hardware intervention.

A SCC 660 has from one to four 24 bit wide input/output channels for transferring information to and from memory. Data is transferred to and from memory through the CPU C register, through a character buffer (B or C buffer) or by a system external to the SCC 660.

In a SCC 660 system, one channel is assigned to the CPU with any additional channels being assigned to either character buffers or to systems external to the SCC 660 system. With four channels, the possible assignments are: 1) CPU, 2 character buffers, 1 external; 2) CPU, 1 character buffer, 2 external; 3) CPU, 3 external. With three channels, the possible assignments are: 1) CPU, 2 character buffers; 2) CPU, 1 character buffer, 1 external; 3) CPU, 2 external. With two channels the possible assignments are: 1) CPU, 1 character buffer; 2) CPU, 1 external.

MAXIMUM INPUT/OUTPUT CONFIGURATION



Transfer Rates

With a memory cycle time of 1.75 usec, the maximum data transfer rate of the memory is slightly greater than 570,000 words per second. The maximum data transfer capability of the 24 bit wide input/output channels which carry the data to and from the memory is also slightly greater than 570,000 words per second. The maximum data transfer capability of the B or C buffer with a channel controller approaches 190,000 words per second. The data transfer rate for input/output through the CPU C register approaches 190,000 words per second.

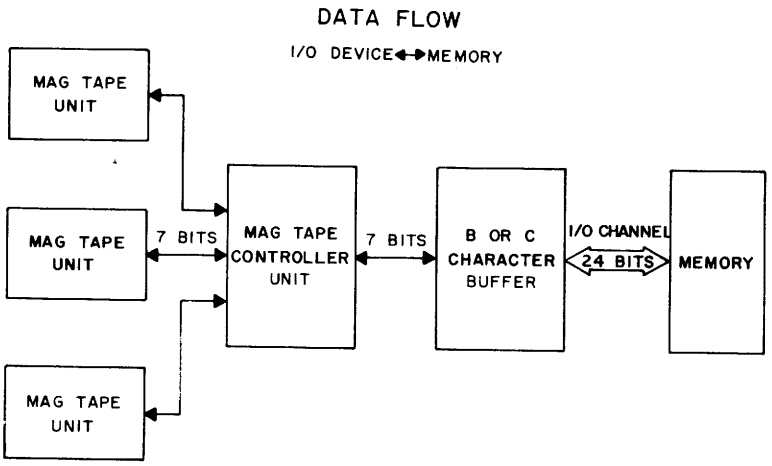
Buffer Input/Output

Operation

During input operations, the B buffer receives from 1 to 4 seven bit characters consisting of 6 data bits and 1 parity bit. The characters are assembled into a word and sent over the I/O channel to a memory module. During output operations, the procedure is reversed. Notification of the CPU by the B buffer of completion of an input/output operation may be by means of an interrupt or a buffer ready line which may be tested. This interrupt may be enabled or disabled under program control or from the control console. In the single word transfer mode, an interrupt occurs when the buffer is ready for the next word. In the block transfer mode, an interrupt occurs upon termination of transfer of the data block.

Data Flow

The flow of data between an input/output device and memory is: Device ⟷ Device Controller ⟷ Buffer ⟷ Memory.



Data Flow Control

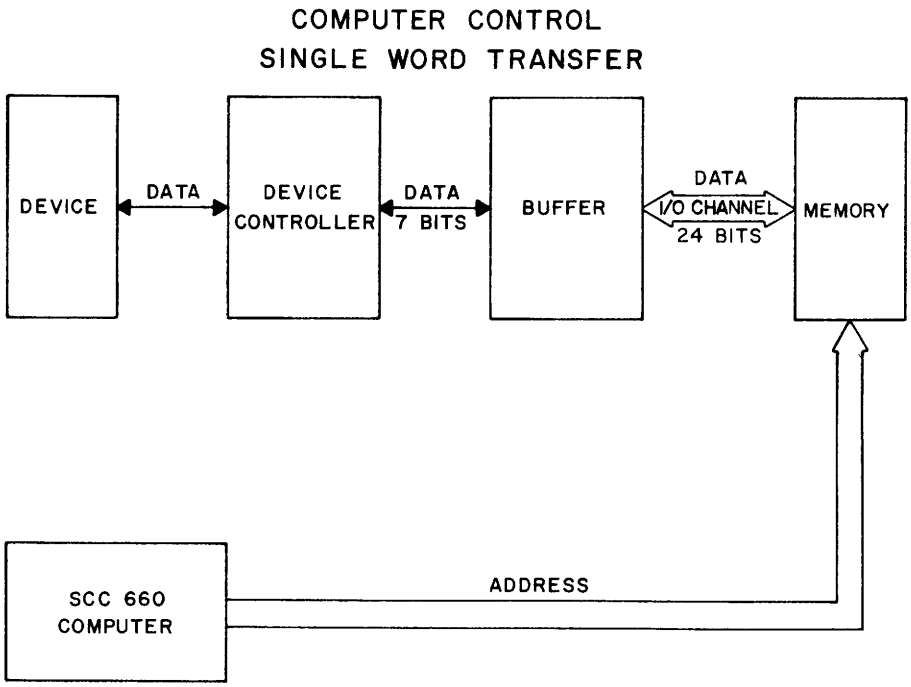
The control of data flow is by the computer or an optional channel controller. Under computer control each word transferred between the memory and the buffer requires computer action. With a channel controller many words may be transferred between the memory and the buffer with computer action required only to initiate the channel controller.

Computer Control

Under computer control, the computer supplies the address to the memory and waits until the data is transferred between the buffer and memory before continuing instruction execution. To transfer another word between the memory and the buffer, the pro- procedure is repeated.

For output operation, the computer sends an address to the memory and a word is sent from memory to the buffer. After the buffer receives the word from the memory, the computer continues executing instructions and the buffer disassembles the word into 6 bit characters which are transmitted to the device controller which in turn transmits the character to a device.

For input operations, the computer sends an address to the memory and the buffer receives 6 bit characters from the device controller which the buffer assembles into a word. After assembly, the word is sent from the buffer to the memory unit and the computer continues executing instructions.



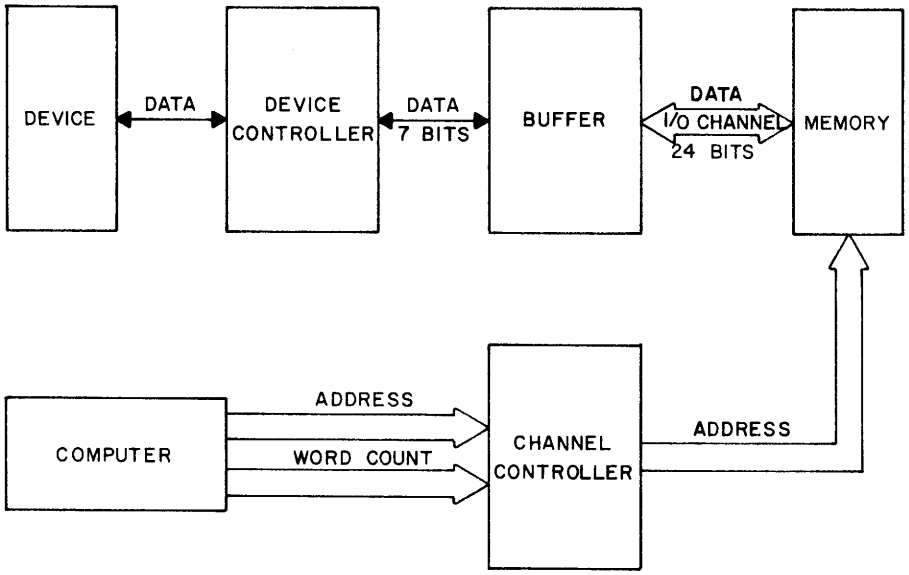
Channel Controller Control

With a channel controller, the computer supplies the channel controller with an address and a count. The address is the address of the first word to be transferred and the count is the number of words to be transferred. After sending the start address and the word count to the channel controller, the computer continues instruction execution and the channel controller controls the transfer of data. With a channel controller only one memory cycle is required to transfer a word between the memory and the buffer.

For output operations, the computer sends the start address and the word count to the channel controller and continues instruction execution. The channel controller places the start address and word count into its address and word count register, respectively and sends the address to the memory. The buffer receives the word from memory and disassembles the word into 6 bit characters which are sent to the device controller which in turn sends the characters to the device. During this time the word count register is decremented by one and the address register is incremented by one. After the buffer sends the word to the device controller, the channel controller tests the word count register for zero. If the word count is zero, the output operation is terminated, otherwise, the address in the address register is sent to the memory and output continues.

For input operations, the computer sends the start address and the word count to the channel controller and continues instruction execution. The channel controller places the start address and word count into its address and word count register, respectively and sends the address to the memory. The buffer receives 6 bit characters from the device controller and assembles the characters into a word. After assembly, the word is sent from the buffer to the memory unit. During this time the word count register is decremented by one and the address register is incremented by one. After the buffer sends the word to the memory, the channel controller tests the word count register for zero. If the word count is zero, the input operation is terminated, otherwise the address in the address register is sent to the memory and input continues.

CHANNEL CONTROLLER
MULTI-WORD TRANSFER



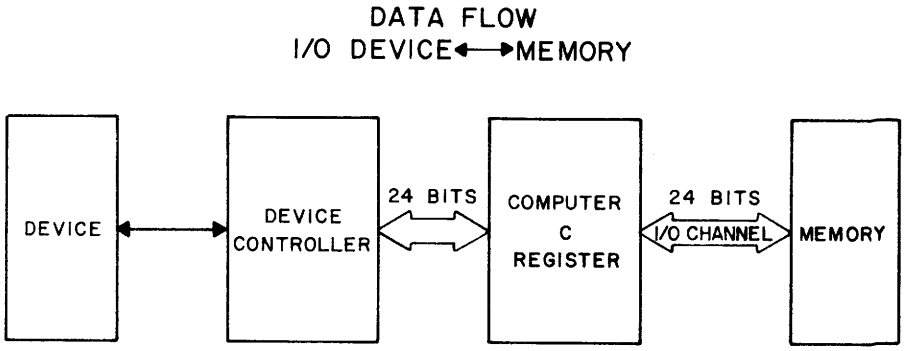
C Register Input/Output (Parallel I/O)

Operation

The CPU C Register sends and receives data in a full word parallel mode. The data is conveyed between the CPU C Register and the memory modules one word at a time over a full word parallel data channel.

Data Flow

The flow of data between an input/output device and memory is: Device ⟷ Device Controller ⟷ C Register ⟷ Memory.



INTERRUPTS

Operation

The occurrence of certain conditions will result in the normal instruction execution sequence being interrupted and the computer executing the instruction stored in a fixed memory location assigned to the particular condition. If an attempt is made to store information in locations 0 thru 777_8 (protected memory area) when the memory protect feature is enabled, the instruction being executed is completed and the instruction in location 27_8 is executed. Location 32_8 is assigned to a B buffer end-of-word and end-of-record conditions. When the B buffer is ready to send another word from memory to a device or when the B buffer has received a word from an external device, the instruction being executed is finished and the instruction in location 32_8 is executed.

Since a memory violation and B buffer end-of-word or end-of-record results in an interruption of the normal instruction execution, they are referred to as interrupts and locations 27_8 and 32_8 are referred to as interrupt locations.

If an optional C buffer is present, an additional interrupt for the C buffer end-of-word and end-of-record conditions is assigned to location 30_8. Any number of additional interrupts may be added to a SCC 660 system to fulfill varied system requirements.

Processing

Execution of the instruction stored in the interrupt location completes processing of the interrupt unless the instruction is a BSL (Store Location Counter and Branch). If the instruction is a BSL, interrupt processing is completed by execution of a BRI (Branch and Clear Interrupt) instruction. Thus the processing of an interrupt may be simply the interruption of the normal sequence to perform a single instruction or may involve the interruption of the normal sequence to perform a series of instructions through use of the BSL and BRI instructions.

Priority

The SCC 660 interrupt system is a priority interrupt system with the lower-numbered location normally having the high priority. For example, if interrupt 27_8 and 32_8 occur simultaneously, interrupt 27_8 is processed then interrupt 32_8 is honored. If an interrupt occurs while another interrupt is being processed, the interrupt will be executed immediately if assigned to a lower-numbered location than the interrupt being executed. If an interrupt condition occurs which is assigned to the same location or a higher location, the interrupt will be honored after completion of the interrupt being processed. For example, interrupt 27_8 can interrupt processing of interrupt 32_8 but interrupt 32_8 cannot interrupt processing of interrupt 27_8.

Arm/Disarm

The memory protection, C buffer and B buffer interrupts may be prevented from occurring under program control. If one of these is prevented from occurring, the respective interrupt is said to be disarmed. If the interrupt is permitted, it is said to be armed.

Enable/Disable

Interrupt conditions may be acknowledged by the computer but restrained from causing an interrupt until the restraint is removed. If interrupt conditions are acknowledged but the interrupts are temporarily restrained, the interrupts are said to be disabled. If interrupt conditions are acknowledged and interrupts are not restrained the interrupts are said to be enabled. If an interrupt condition occurs while the interrupts are disabled, the corresponding interrupt will not occur until the interrupts are enabled, at which time the acknowledged interrupt will occur. All interrupts are enabled or disabled as a group and may be enabled either under program control or from the system console.

PROGRAMMED OPERATORS

Operation

Bit 2 of the instruction word designates a programmed operation, that is, the operation to be performed is a programmed routine. If bit 2 is set (a one), bits 2-8 designate a programmed operation. If bit 2 is reset (a zero), bits 2-8 designate a machine instruction. If bit 2 is set, the contents of bits 3-8 can designate up to 2^6 or 64 different routines. If bit 2 is set, the I and X bits are ignored, bits 2–8 are treated as a memory address and the instruction at this address is executed by the computer. This instruction is a BSL (Store Location and Branch) to the programmed routine.

The SCC 660 programmed operator feature effectively allows expansion of the instruction repertoire by permitting programmed routines to perform the additional instructions. For example, the SCC 660 cannot perform a floating point multiply but a routine to perform a floating point multiply can be written. The procedure for assigning an operation code to the routine, for defining the routine as a programmed operator, and for assigning a mnemonic to the routine is given in the SCC 660 Symbolic Programming Language manual.

Invalid Operation Codes

Invalid op codes are treated in the same manner as programmed operators. If the contents of bits 3-8 of an instruction word do not specify a valid op code, the contents is treated as a memory location and the instruction at that location is executed.

SECTION II

MACHINE INSTRUCTIONS

GENERAL

Instructions are classified as follows:

1. Memory/Register
2. Register/Register
3. Arithmetic
4. Logical
5. Shift/Cycle
6. Branch
7. Skip
8. Miscellaneous
9. Input/Output
10. Internal Control

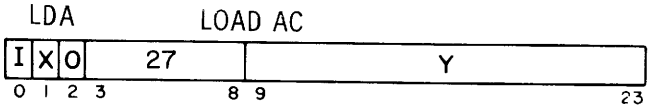
The meaning of symbols and terms appearing in this section are as follows:

Symbol/Term	Definition
AC	AC Accumulator
ALU	Arithmetic/Logic Unit
CRO	Carry Out Indicator
EA	EA Accumulator
LC	Location Counter
M	Bits 18-23 of Instruction Word
OVF	Overflow Indicator
Q	Effective Location
UIQ	Unindexed Effective Location
XR	Index Register
Y	Bits 9-23 of Instruction Word
Cycle	Memory Cycle Time; 1.75 usec
Effective Address	Effective Location
Effective Location	See Section 1, Indexing and Indirect Addressing
Operand	Contents of the Effective Location
Reset	0
Set	1
Unindexed Effective Address	Effective Address Formed by Ignoring Indexing at Last Level
125	125_8; Octal Number
125_8	Octal Number
125_{10}	Decimal Number
||	Magnitude
<	Less Than
>	Greater Than
=	Equal
≠	Not Equal
▨	Bit position is ignored. Contents are irrelevant.

MEMORY/REGISTER

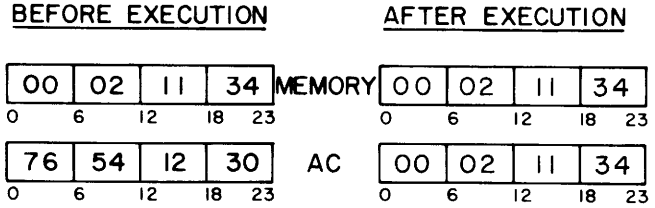
The Memory to Register and Register to Memory instructions are:

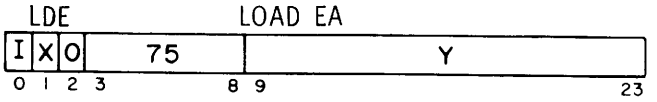
Mnemonic	Op Code	Name
LDA	27	Load AC
LDE	75	Load EA
LDX	71	Load XR
LAC	45	Load AC with Complement
LXC	44	Load XR with Complement
EAX	77	Effective Address into XR
STA	35	Store AC
STE	36	Store EA
STX	37	Store XR
SAF	42	Store Address Field of AC
XXM	76	Exchange XR and Memory



The operand is placed in the AC. The operand is not altered. The LC is incremented by 1.

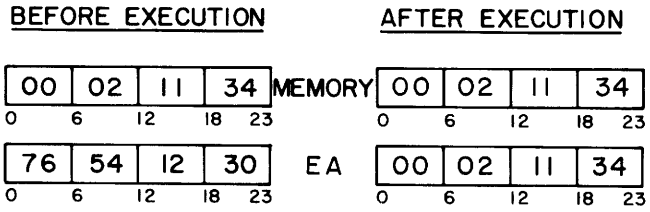
2 cycles



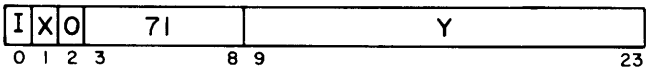


The operand is placed in the EA. The operand is not altered. The LC is incremented by 1.

2 cycles

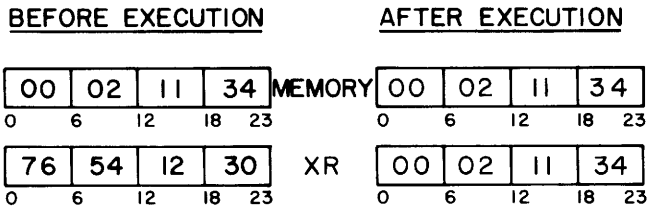


LDX LOAD XR

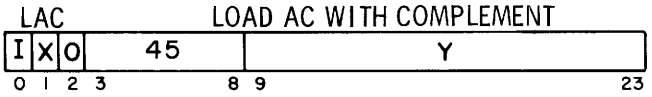


The operand is placed in XR. The operand is not altered. The LC is incremented by 1.

2 cycles



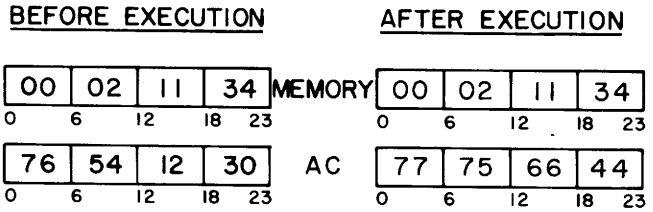
LAC LOAD AC WITH COMPLEMENT



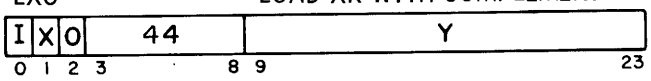
The two's complement of the operand is placed in the AC. The operand is not altered. The LC is incremented by 1.

2 cycles

If the operand is 40000000_8 (the largest negative number), the contents of the AC will be 40000000_8. The overflow indicator is not changed.



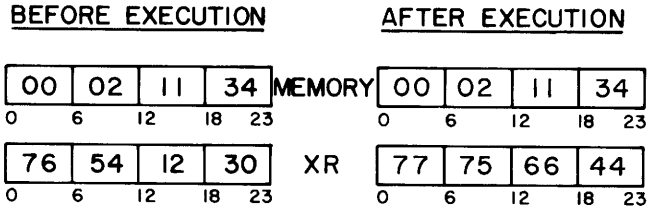
LXC LOAD XR WITH COMPLEMENT



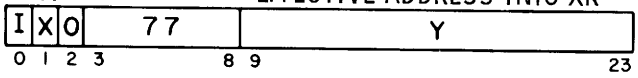
The two's complement of the operand is placed in XR. The operand is not altered. The LC is incremented by 1.

2 cycles

If the operand is 40000000_8 (the largest negative number), the contents of the XR will be 40000000_8. The overflow indicator is not changed.

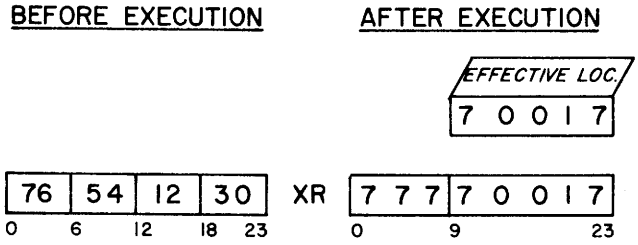


EAX EFFECTIVE ADDRESS INTO XR

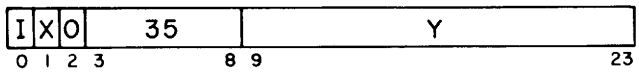


The effective location is placed in XR_{9-23}. XR_9 is then copied into XR_{0-8}. The LC is incremented by 1.

1 cycle

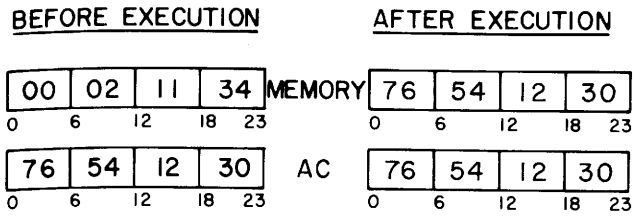


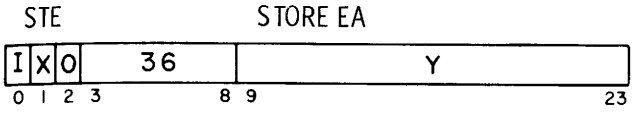
STA STORE AC



The AC is placed in the effective location. The AC is not altered. The LC is incremented by 1.

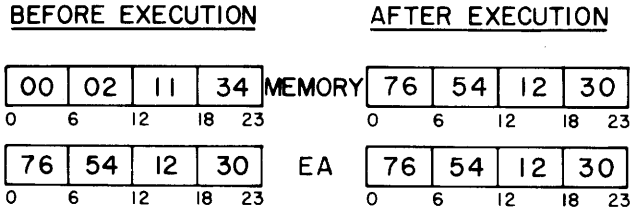
2 cycles

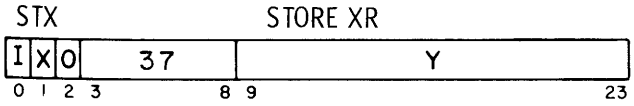




The EA is placed in the effective location. The EA is not altered. The LC is incremented by 1.

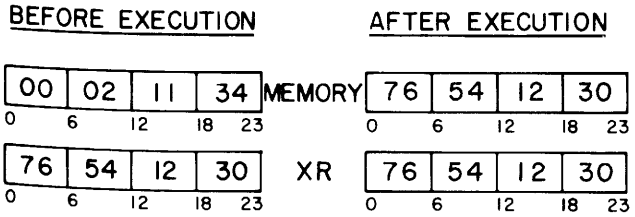
2 cycles

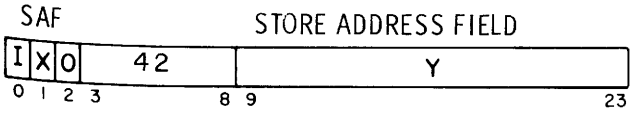




The XR is placed in the effective location. The XR is not altered. The LC is incremented by 1.

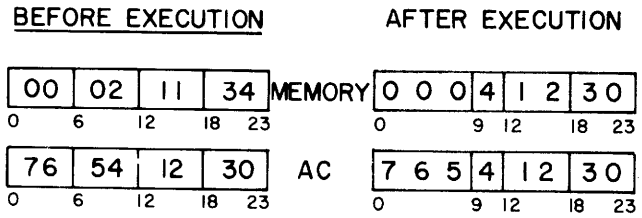
2 cycles

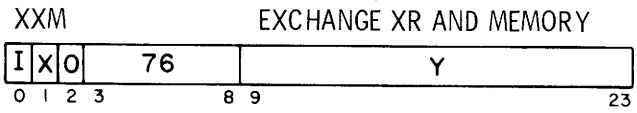




$AC_{9\text{-}23}$ is placed in bits 9-23 of the effective location. The AC is not altered. The contents of bits 0-8 of the effective location are not altered. The LC is incremented by 1.

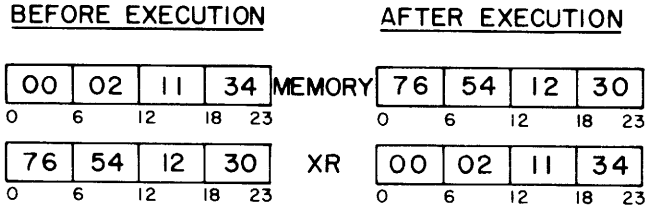
2 cycles





The operand is placed in XR. The original contents of XR is placed in the effective location. The LC is incremented by 1.

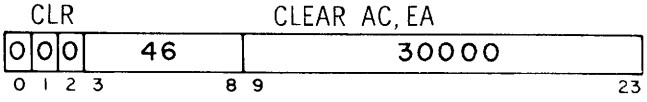
3 cycles



REGISTER/REGISTER INSTRUCTIONS

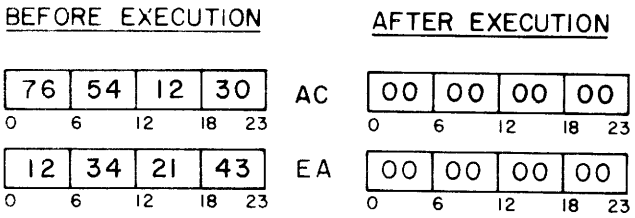
The register/register instructions are:

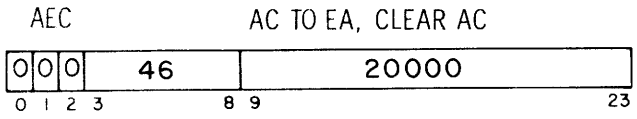
Mnemonic	Octal Code	Name
CLR	0 46 30000	Clear AC and EA
AEC	0 46 20000	AC to EA, Clear AC
EAC	0 46 10000	EA to AC, Clear EA
XAE	0 46 00000	Exchange AC and EA



The AC and EA are reset to zero. The LC is incremented by 1.

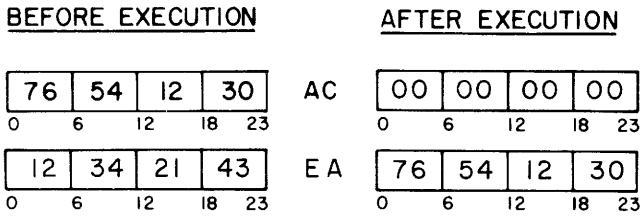
1 cycle

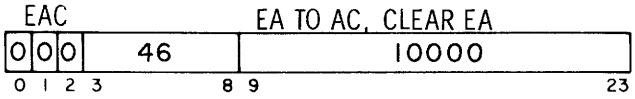




The AC is placed in the EA. The AC is reset to zero. The LC is incremented by 1.

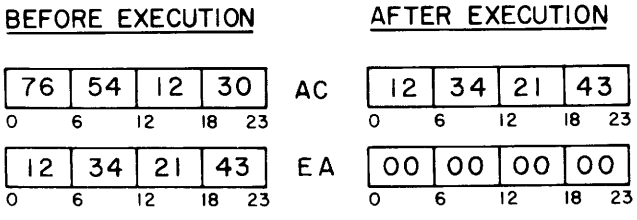
1 cycle

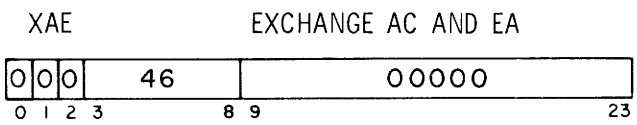




The EA is placed in the AC. The EA is reset to zero. The LC is incremented by 1.

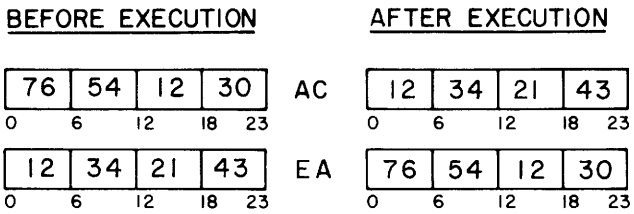
1 cycle





The AC is placed in the EA. The original contents of EA is placed in the AC. The LC is incremented by 1.

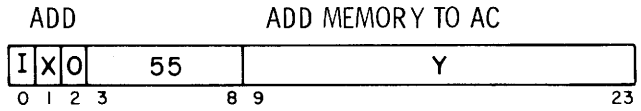
1 cycle



ARITHMETIC INSTRUCTIONS

The arithmetic instructions are:

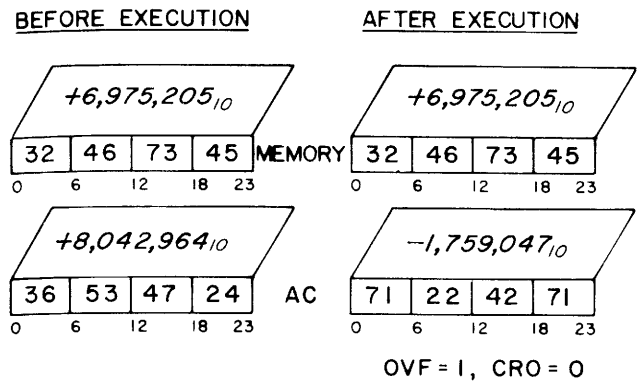
Mnemonic	Op Code	Name
ADD	55	Add Memory to AC
SUB	54	Subtract Memory from AC
MPY	64	Multiply AC by Memory
DIV	65	Divide AC,EA by Memory
MIN	61	Memory Increment
MDC	60	Memory Decrement



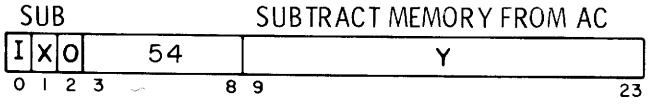
The CRO is cleared. The operand is added to the AC. The sum is placed in the AC. The operand is not altered. If a carry out of bit position 0 occurs, the CRO is set. If the result is algebraically greater than $2^{23}-1$ or less than -2^{23}, the overflow indicator is set. If the result is within range, the overflow indicator is not altered. The LC is incremented by 1.

2 cycles

If both numbers have the same sign, but the sign of the result is different, an overflow has occurred. 2^{23} is 8,388,608.



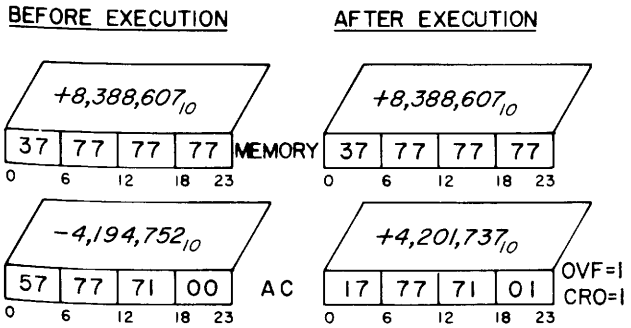
In the preceding example, the sum of $+6,975,205_{10}$ and $+8,042,964_{10}$ is $+15,018,169_{10}$. This sum is too large to be represented in the 24 bit AC. The binary addition of the AC and memory produces $71,224,271_8$ which is the two's complement of $-1,759,047_{10}$. Notice that bit 0 of the result contains a 1 and bit 0 of the AC and memory contained a 0 before the addition. Both numbers were of the same sign, positive, and the result is negative, therefore an overflow occurred and the overflow indicator is set. However, a carry out of bit 0 did not occur, therefore the carry out indicator is reset.



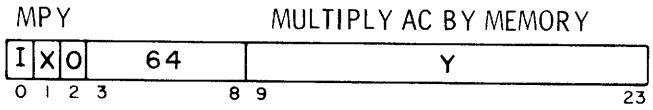
The CRO is cleared. The operand is subtracted from the AC. The result is placed in the AC. The operand is not altered. If a carry out of bit position 0 occurs, the CRO is set. If the result is algebraically greater than $2^{23}-1$ or less than -2^{23}, the overflow indicator is set. If the result is within range, the overflow indicator is not altered. The LC is incremented by 1.

2 cycles

If both numbers have different signs and the sign of the result is different from the sign of the AC, an overflow has occurred. 2^{23} is $8,388,608_{10}$



In the preceding example, a negative $4,194,752_{10}$ minus a positive $6,975,205_{10}$ yields $-12,583,359_{10}$. This difference is too large to be represented in the 24 bit AC. The binary subtraction of the memory contents from the AC produces 17777101_8 is $+4,201,737_{10}$. (The computer subtracts by adding the two's complement of the memory contents to the AC contents. The two's complement of 37777777_8 is 400000001_8. 400000001_8 + 57777100_8 is 117777101_8. However, 117777101_8 is 25 binary digits and the AC is 24 bits in size. The AC retains 17777101_8 and the high order bit sets the carry out indicator.) Notice that bit 0 of the memory contents is a 0 and that bit 0 of the AC before subtraction was a 1. Also, notice that bit 0 of the result in the AC is a 0. Therefore, an overflow has occurred since the memory and the AC were of different sign and the sign of the AC has changed. The overflow indicator is set and the carry out indicator is set since a carry out of bit 0 occurred.



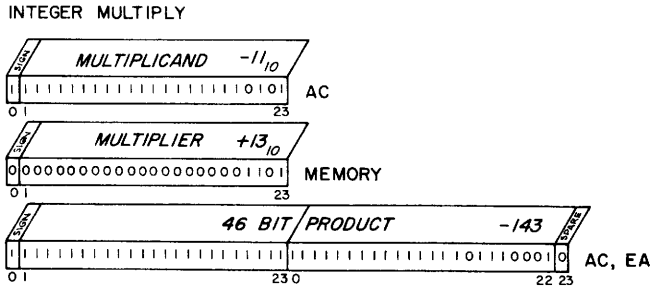
The AC is multiplied by the operand. The most significant part of the result is placed in the AC. The least significant part of the result is placed in the EA. CRO is not altered. The LC is incremented by 1.

If the AC and the effective location contain 40000000_8 (-2^{23}), the product will be zero and the overflow indicator will be set. All other cases are valid and do not alter the overflow indicator.

8 cycles

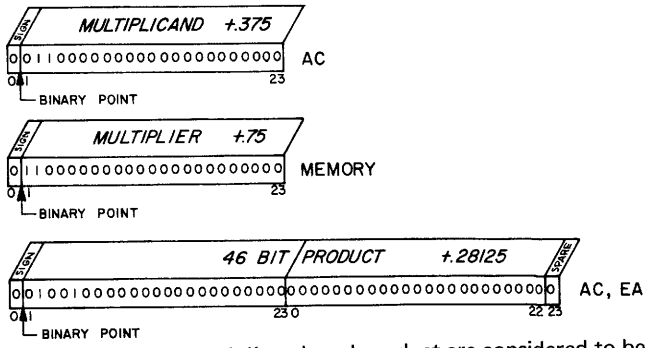
The operand and the contents of the AC are treated as a sign bit and 23 data bits with multiplication producing a sign bit followed by 46 data bits, followed by a zero.

AC	Sign bit, 23 data bits	= 24 bits
Memory	Sign bit, 23 data bits	= 24 bits
AC,EA	Sign bit, 46 data bits, 1 spare bit	= 48 bits



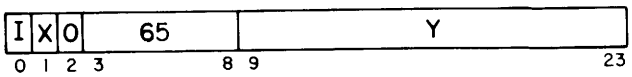
If the multiplier and multiplicand are considered to be integers, the result must be shifted one position to the right for the AC,EA to contain the mathematically correct result.

FRACTION MULTIPLY



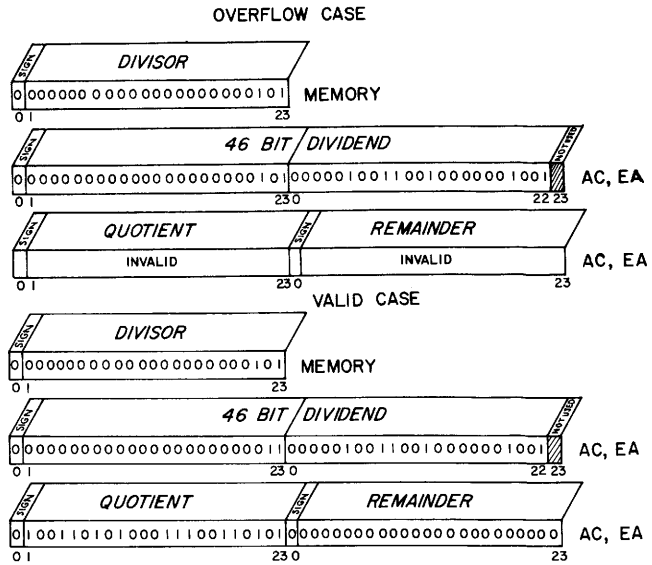
If the multiplier, multiplicand, and product are considered to be binary fractions with the binary point between bit 0 and bit 1, the result is mathematically correct. The contents of AC is $+.011_2(+.375_{10})$; the contents of memory is $+.11_2$ $(+.75_{10})$; the AC,EA contents is $+.010001_2$ $(+.28125_{10})$.

DIV DIVIDE AC, EA BY MEMORY

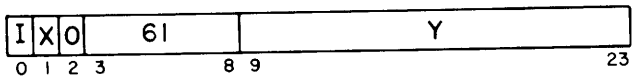


Bit 0 of AC thru bit 22 of EA are treated as a signed 46 bit dividend. Bit 23 of EA is ignored. The operand is treated as a signed 23 bit divisor. The signed 23 bit quotient is placed in the AC. The signed 23 bit remainder is placed in the EA. The sign of the remainder is the sign of the dividend. The sign of the quotient obeys the rules of algebra. CRO is not altered. Division by zero sets the overflow indicator and yields invalid results. The signed 23 bit quotient is placed in the AC. If the quotient cannot be contained in 23 bits, the overflow indicator is set and the contents of the AC and EA are invalid and unpredictable. By comparing AC and memory prior to division, overflow situations may be detected. If AC and memory are of like signs and the $|\text{memory}| \leq |AC|$, division will result in overflow. If AC and memory are of unlike signs and the $|\text{memory}| < |AC|$, division will result in overflow. The LC is incremented by 1.

9 cycles



MIN MEMORY INCREMENT



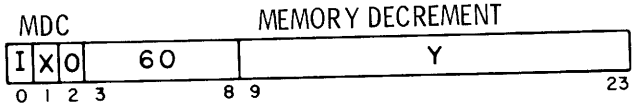
The operand is incremented by 1. OVF is not altered. CRO is not altered. The LC is incremented by 1.

3 cycles

If the operand is 37777777_8 $(+2^{23}-1)$, the result is 40000000_8 (-2^{23}).

BEFORE EXECUTION AFTER EXECUTION

MDC MEMORY DECREMENT



The operand is decremented by 1. OVF is not altered. CRO is not altered. The LC is incremented by 1.

3 cycles

If the operand is 40000000_8 (-2^{23}), the result is 37777777_8 $(+2^{23}-1)$.

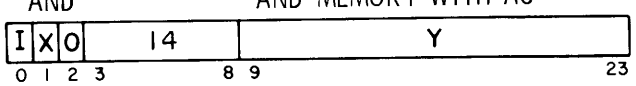
BEFORE EXECUTION AFTER EXECUTION

LOGICAL INSTRUCTIONS

The logical instructions are:

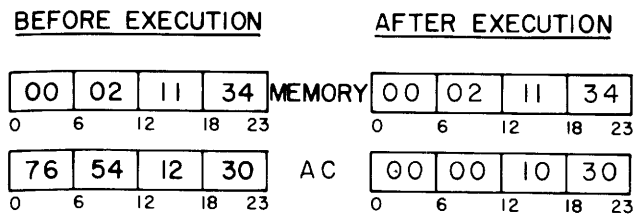
Mnemonic	Op Code	Name
AND	14	AND Memory with AC
ORA	16	OR Memory with AC
EOR	17	Exclusive OR Memory with AC

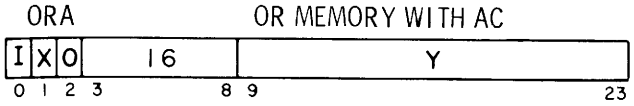
AND AND MEMORY WITH AC



The operand is anded with the AC. The result is placed in the AC. The operand is not altered. The LC is incremented by 1.

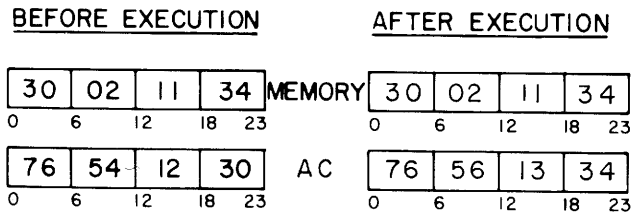
2 cycles

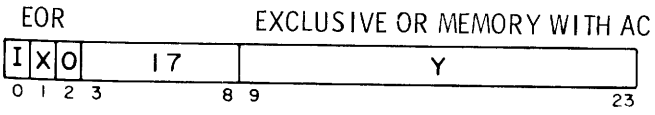




The operand is ored with the AC. The result is placed in the AC. The operand is not altered. The LC is incremented by 1.

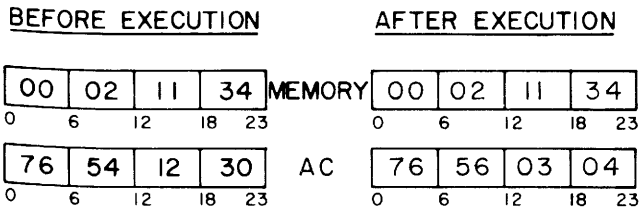
2 cycles





The operand is exclusive ored with the AC. The result is placed in the AC. The operand is not altered. The LC is incremented by 1.

2 cycles



SHIFT/CYCLE INSTRUCTIONS

The Shift and Cycle instructions are:

Mnemonic	Octal Code	Name
LSH	0 67 0 00 XX	Left Shift AC,EA
ALS	0 67 0 40 XX	Left Shift AC
ELS	0 67 0 20 XX	Left Shift EA
LCY	0 67 2 00 XX	Left Cycle AC,EA
ALC	0 67 2 40 XX	Left Cycle AC
ELC	0 67 2 20 XX	Left Cycle EA
NDX	0 67 1 00 XX	Normalize and Decrement XR
SND	0 67 1 40 XX	Short Normalize and Decrement XR
RSH	0 66 0 00 XX	Arithmetic Right Shift AC,EA
ARS	0 66 0 40 XX	Arithmetic Right Shift AC
ERS	0 66 0 20 XX	Right Shift EA
RCY	0 66 2 00 XX	Right Cycle AC,EA
ARC	0 66 2 40 XX	Right Cycle AC
ERC	0 66 2 20 XX	Right Cycle EA

Direction of Shift

The direction of a shift is designated by the op code with all shifts to the left having a 67_8 op code and all shifts to the right having a 66_8 op code.

LEFT SHIFTS

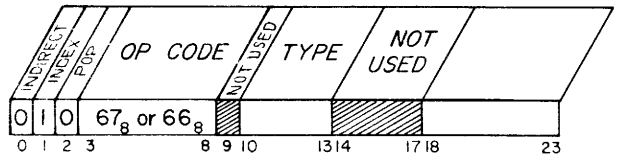
Name	Mnemonic	Op Code
Left Shift AC,EA	LSH	67
Left Shift AC	ALS	67
Left Shift EA	ELS	67
Left Cycle AC,EA	LCY	67
Left Cycle AC	ALC	67
Left Cycle EA	ELC	67
Normalize and Decrement XR	NDX	67
Short Normalize and Decrement XR	SND	67

RIGHT SHIFTS

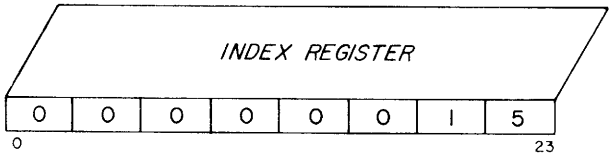
Arithmetic Right Shift AC,EA	RSH	66
Airthmetic Right Shift AC	ARS	66
Right Shift EA	ERS	66
Right Cycle AC,EA	RCY	66
Right Cycle AC	ARC	66
Right Cycle EA	ERC	66

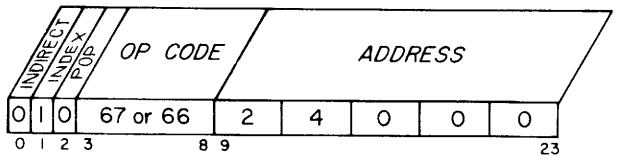
Type of Shift

The type of shift is designated by the high order bits of the unindexed effective address. The unindexed effective address is the effective address prior to indexing at the final level. In the vast majority of cases, the shifts are coded without specifying indirect addressing in which case the unindexed effective address is simply bits 9-23 of the instruction word.



The type of shift is specified by bits 1-4 of the unindexed effective address. If indirect addressing is not specified, as in the above example, bits 1-4 of the unindexed effective address are bits 10-13 of the instruction word.





EFFECTIVE ADDRESS = 24015

UNINDEXED EFFECTIVE ADDRESS = 24000

Since indirect addressing is not specified, the effective address is the contents of bits 9-23 of the instruction plus the contents of XR, thus, the effective address is 24015_8 (24000_8 + 15_8).

Since indirect addressing is not specified, the unindexed effective address is simply the address field of the instruction, 24000_8. In order to reflect common usage and simplify the explanations, all examples and descriptions are given with indirect addressing not being specified.

LEFT SHIFTS

Name	Mnemonic	Op Code	Bits 10-13
Left Shift AC,EA	LSH	67	0 0 0 0
Left Shift AC	ALS	67	0 0 1 0
Left Shift EA	ELS	67	0 0 0 1
Left Cycle AC,EA	LCY	67	1 0 0 0
Left Cycle AC	ALC	67	1 0 1 0
Left Cycle EA	ELC	67	1 0 0 1
Normalize and Decrement XR	NDX	67	0 1 0 0
Short Normalize and Decrement XR	SND	67	0 1 1 0

RIGHT SHIFTS

Arithmetic Right Shift AC,EA	RSH	66	0 0 0 0
Arithmetic Right Shift AC	ARS	66	0 0 1 0
Right Shift EA	ERS	66	0 0 1 0
Right Cycle AC,EA	RCY	66	1 0 0 0
Right Cycle AC	ARC	66	1 0 1 0
Right Cycle EA	ERC	66	1 0 0 1

All examples and listings of the shift instruction show bits 9 and 14-17 which are not used as zero. A listing of the shift instruction reflecting the contents of bits 3-14, assuming no indirect addressing follows.

LEFT SHIFTS

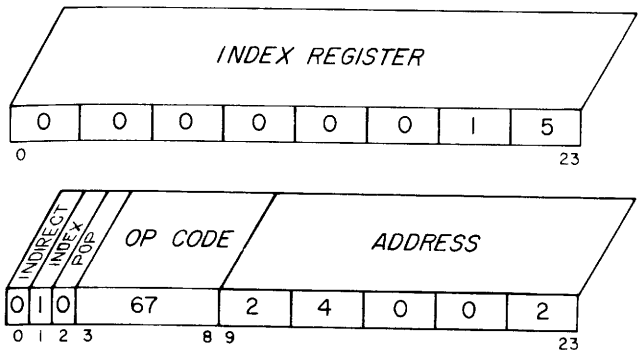
Name	Mnemonic	Octal Code
Left Shift AC, EA	LSH	67 000
Left Shift AC	ALS	67 040
Left Shift EA	ELS	67 020
Left Cycle AC, EA	LCY	67 200
Left Cycle AC	ALC	67 240
Left Cycle EA	ELC	67 220
Normalize and Decrement XR	NDX	67 100
Short Normalize and Decrement XR	SND	67 140

RIGHT SHIFTS

Arithmetic Right Shift AC, EA	RSH	66 000
Arithmetic Right Shift AC	ARS	66 040
Right Shift EA	ERS	66 020
Right Cycle AC, EA	RCY	66 200
Right Cycle AC	ARC	66 240
Right Cycle EA	ERC	66 220

Places to Shift

The number of places to shift the register contents is designated by the six low order bits of the effective address.



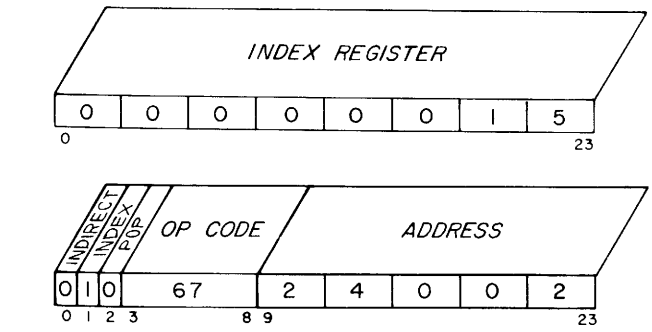
EFFECTIVE ADDRESS = 24017_8

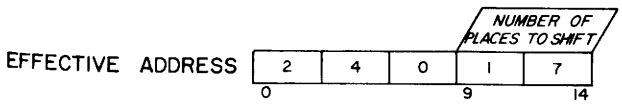
UNINDEXED EFFECTIVE ADDRESS = 24002_8

TYPE OF SHIFT = LEFT CYCLE AC

PLACES TO SHIFT = $17_8 = 15_{10}$

Since indirect addressing is not specified, the effective address is the contents of bits 9-23 of the instruction word plus the contents of XR. The effective address is 24017_8 (24002_8 + 15_8).





EFFECTIVE ADDRESS = 24017_8

NO. OF PLACES TO SHIFT = LOW ORDER 6 BITS = 17_8

TYPE OF SHIFT = LEFT CYCLE AC

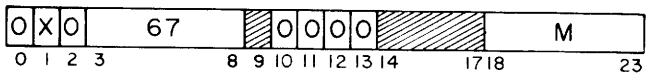
In the above example, the number of places to shift is specified by bits 9-14 (low order 6 bits) of the effective address.

Execution Time of Shifts

The execution time of all shifts is 2+N/4 where N is the number of places the register is shifted and fractions of a cycle are rounded to the next cycle. If N is 8, 2+N/4 is 2+8/4 or 4 cycles. If N is 5, 2+N/4 is 2+5/4 or 3-1/4 cycles or 4 cycles. If N is 24 2+N/4 is 2+24/4 or 8 cycles. There is one exception to this timing formula. The exception is the case of 0 shifts. Any shift with a shift count of 0 requires 3 cycles and does not alter the register involved.

Number of Shifts	Timing (Cycles)	Number of Shifts	Timing (Cycles)
0 – 4	3	33 – 36	11
5 – 8	4	37 – 40	12
9 – 12	5	41 – 44	13
13 – 16	6	45 – 48	14
17 – 20	7	49 – 52	15
21 – 24	8	53 – 56	16
25 – 28	9	57 – 60	17
29 – 32	11	61 – 63	18

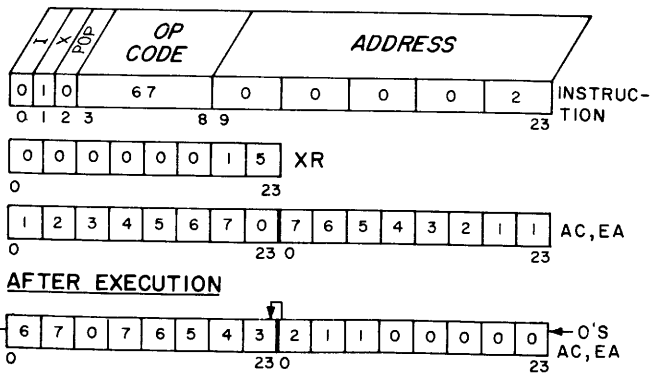
LSH LEFT SHIFT AC, EA



The AC and EA are treated as a single 48 bit register and shifted left N positions where N = M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of AC_0 are lost. Bits shifted out of EA_0 are shifted into AC_{23}. Zeros fill the positions vacated on the right. If during execution the contents of AC_0 changes, OVF is set; otherwise OVF is not altered. The LC is incremented by 1.

2 + N/4 cycles

BEFORE EXECUTION



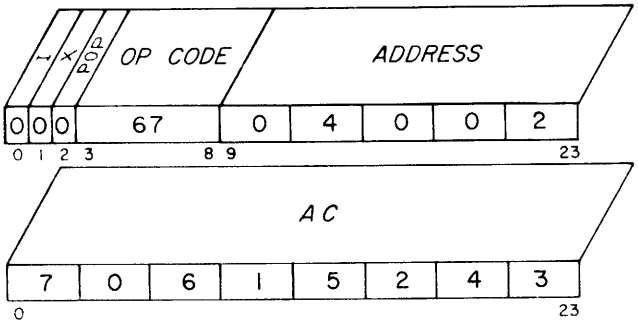
$N=M+XR_{18\text{-}23}=15_8+2_8=17_8=15_{10}$ Time = 2+15/4 = 6 cycles

ALS LEFT SHIFT AC

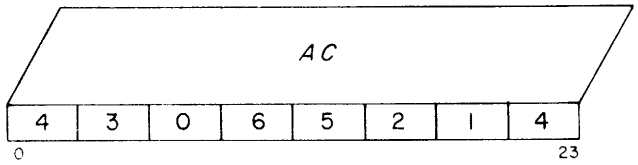
The AC is shifted to the left N positions where N = M if no indexing and $N=M+XR_{18\text{-}23}$ if indexing is specified. Bits shifted out of AC_0 are lost. Zeros fill the positions vacated on the right. If during execution the contents of AC_0 changes, OVF is set; otherwise OVF is not altered. The LC is incremented by 1.

2 + N/4 Cycles

BEFORE EXECUTION

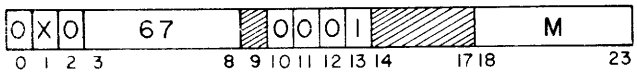


AFTER EXECUTION



N=M; N=2 Time = 2+N/4=2+2/4 = 3 cycles

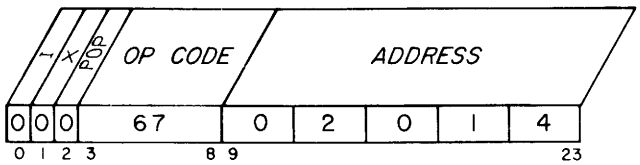
ELS LEFT SHIFT EA



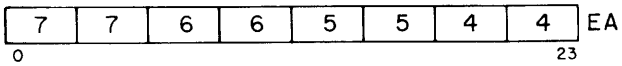
The EA is shifted to the left N positions where N = M if no indexing and N = $M+XR_{18\text{-}23}$ if indexing is specified. Bits shifted out of EA_0 are lost. Zeros fill the positions vacated on the right. The LC is incremented by 1.

2 + N/4 cycles

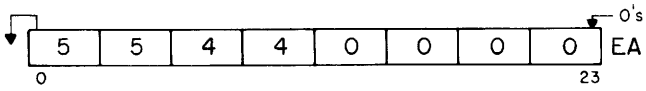
INSTRUCTION



BEFORE EXECUTION

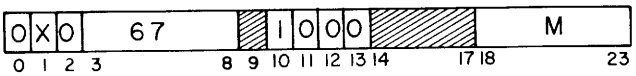


AFTER EXECUTION



N=M; $N=14_8=12_{10}$ Time = 2+12/4 = 5 cycles

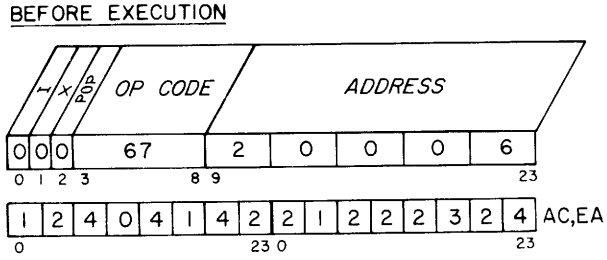
LCY LEFT CYCLE AC, EA



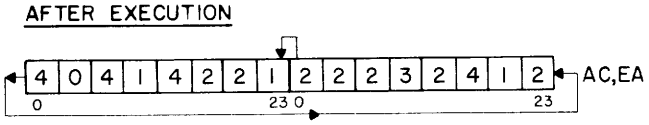
The AC and EA are treated as a single 48 bit register and rotated to the left N positions where N = M if no indexing and $N+M+XR_{18\text{-}23}$ if indexing is specified. Bits shifted out of AC_0 are shifted into EA_{23}. Bits shifted out of EA_0 are shifted into AC_{23}. No bits are lost. The LC is incremented by 1.

2 + N/4 cycles

BEFORE EXECUTION

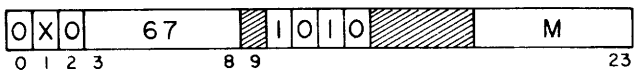


AFTER EXECUTION



N=M; N=6 Time = 2+N/4 = 2+6/4 = 4 cycles

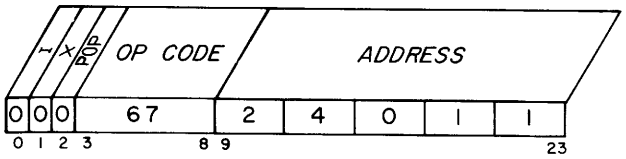
ALC LEFT CYCLE AC



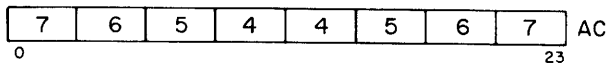
The AC is rotated left N positions where N = M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of AC_0 are shifted into AC_{23}. No bits are lost. The LC is incremented by 1.

2 + N/4 cycles

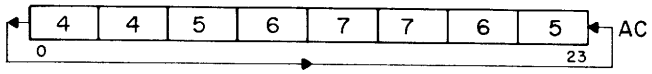
INSTRUCTION



BEFORE EXECUTION

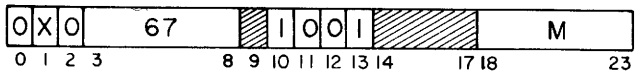


AFTER EXECUTION



N=M; $N=11_8=9_{10}$ Time = 2+9/4 = 4 1/4 = 5 cycles

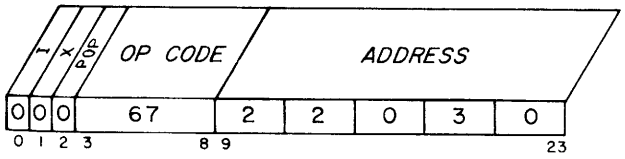
ELC LEFT CYCLE EA



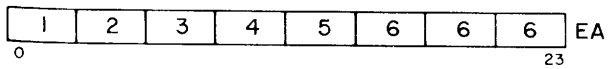
The EA is rotated left N positions where N = M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of EA_0 are shifted into EA_{23}. No bits are lost. The LC is incremented by 1.

2 + N/4 cycles

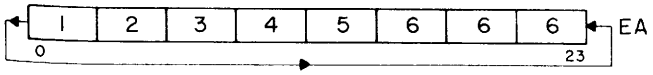
INSTRUCTION



BEFORE EXECUTION

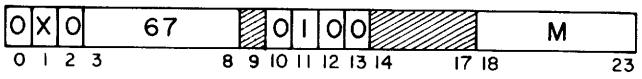


AFTER EXECUTION



N=M; $N=30_8=24_{10}$ Time = 2+N/4 = 2+24/4 = 8 cycles

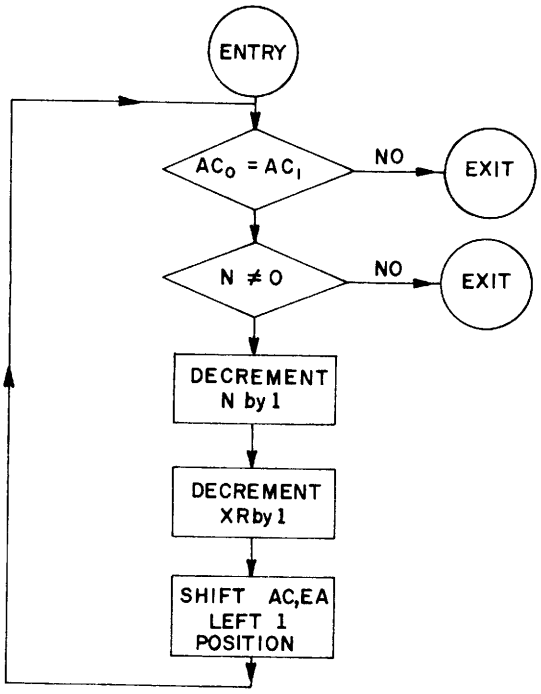
NDX NORMALIZE AND DECREMENT XR



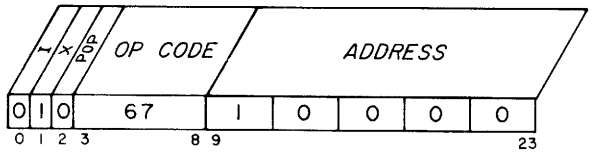
The AC and EA are treated as a single 48 bit register. If AC_0 = AC_1 and N ≠ 0, then N is decremented by 1, the AC,EA is shifted left 1 position, and XR is decremented by 1. This procedure is repeated until N = 0 or $AC_0 \neq AC_1$. Initially, N = M if no indexing and N = M + XR_{18-23} if indexing is specified. Bits shifted out of AC_0 are lost. Bits shifted out of EA_0 are shifted into AC_{23}. Zeros fill the vacated positions on the right. The LC is incremented by 1.

2 + Number of Shifts /4 cycles

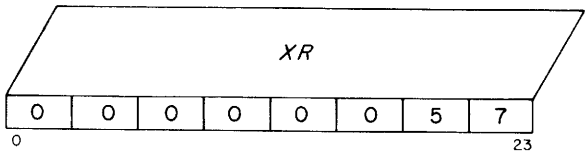
NDX OPERATION

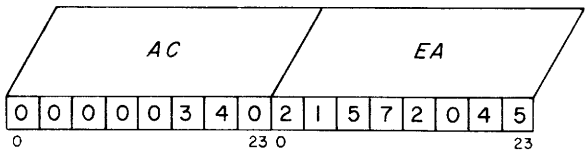


INSTRUCTION

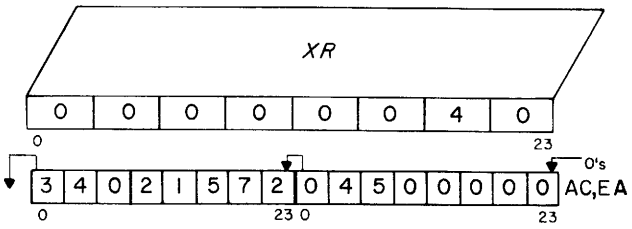


BEFORE EXECUTION





AFTER EXECUTION

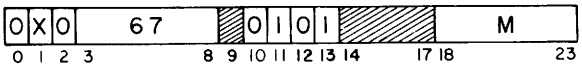


$N=M+XR_{18-23}$; $N=0+57_8 = 47_{10}$

Time = 2+15/4 = 5 3/4 = 6 cycles

The AC,EA was shifted 15_{10} positions (57_8 - 40_8). The number in the AC,EA is normalized and has 32_{10} (40_8) significant digits.

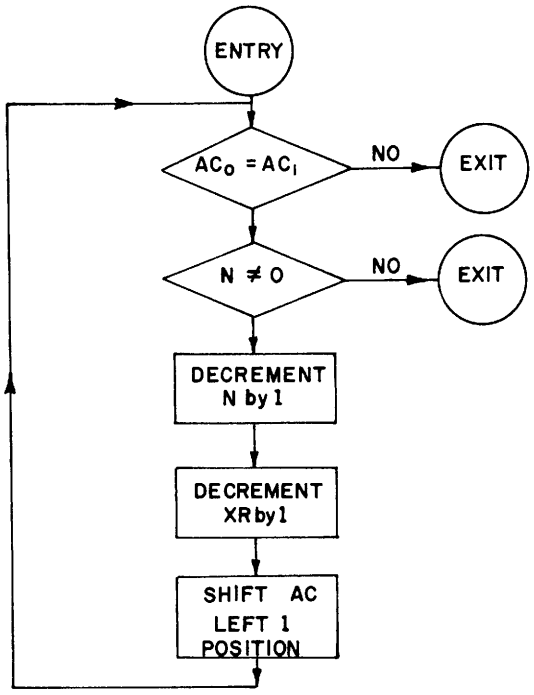
SND SHORT NORMALIZE AND DECREMENT XR



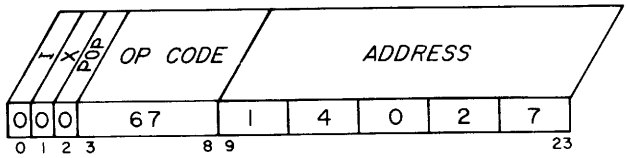
If AC_0 - AC_1 and N ≠ 0, then N is decremented by 1, XR is decremented by 1, and the AC is shifted left 1 position. This procedure is repeated until N = 0 or AC_0 = AC_1. Initially, N = M is no indexing and N = M + XR_{18-23} if indexing is specified. Bits shifted out of AC_0 are lost. Zeros fill the vacated positions on the right. The LC is incremented by 1. The operation of SND is identical to that of NDX except SND shifts AC only and NDX shifts AC,EA.

2 + Number of Shifts / 4 cycles

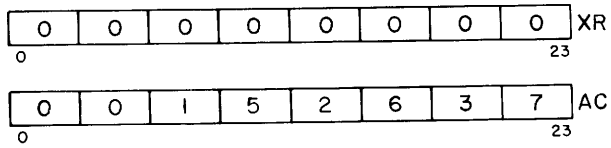
SND OPERATION



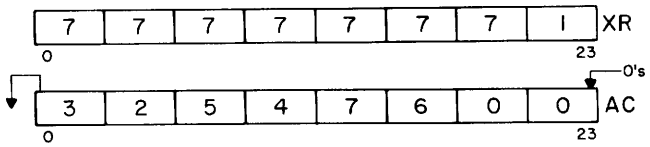
INSTRUCTION



BEFORE EXECUTION



AFTER EXECUTION



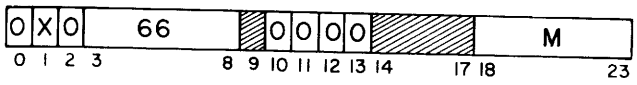
XR contains -7. Number of Shifts = 7

Number of significant digits = -7+23 = 16

N=M; $N=27_8=23_{10}$

Time=2+Number of Shifts/4=2+7/4=3 3/4=4 cycles

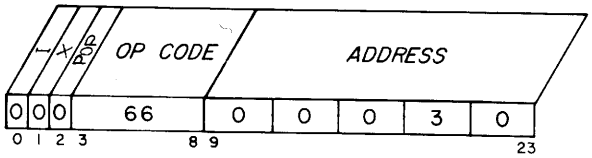
RSH ARITHMETIC RIGHT SHIFT AC, EA



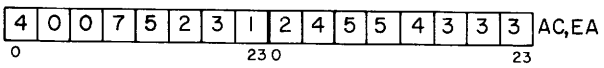
The AC and EA are treated as a single 48 bit register and shifted right N positions where N=M if no indexing and $N=M+XR_{18-23}$ if indexing is specified. Bits shifted out of EA_{23} are lost. Bits shifted out of AC_{23} are shifted into EA_0. Vacated positions on the left are filled with the original contents of AC_0, i.e., the sign is propagated. The LC is incremented by 1.

2 + N/4 cycles

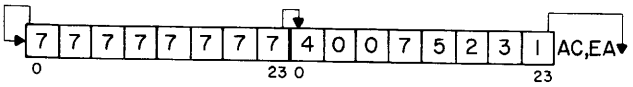
INSTRUCTION



BEFORE EXECUTION

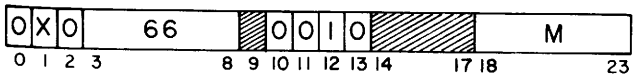


AFTER EXECUTION



N=M; $N=30_8=24_{10}$ Time=2+N/4=2+24/4 = 8 cycles

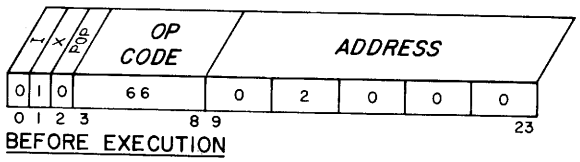
ARS ARITHMETIC RIGHT SHIFT AC



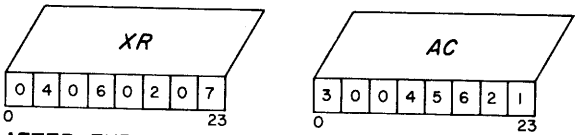
The AC is shifted right N positions where N=M if no indexing and $N=M+XR_{18-23}$ if indexing is specified. Bits shifted out of AC_{23} are lost. Vacated positions on the left are filled with the original contents of AC_0, i.e., the sign is propagated. The LC is incremented by 1.

2 + N/4 cycles

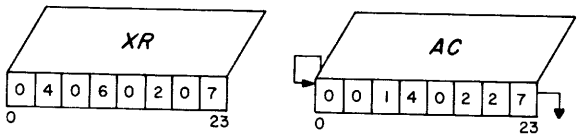
INSTRUCTION



BEFORE EXECUTION

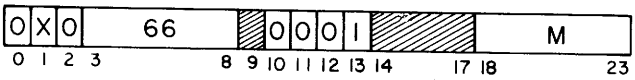


AFTER EXECUTION



$N=M+XR_{18-23}$;N=0+7 = 7 Time 2+N/4 = 2+7/4 = 4 cycles

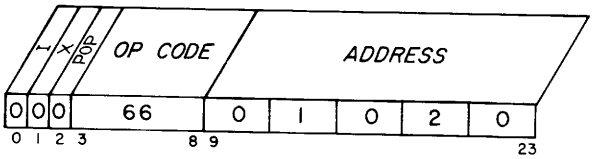
ERS RIGHT SHIFT EA



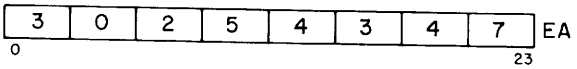
The EA is shifted right N positions where N = M if no indexing and $N = M+XR_{18-23}$ if indexing is specified. Bits shifted out of EA_{23} are lost. Zeros fill the vacated positions on the left. The LC is incremented by 1.

2 + N/4 cycles

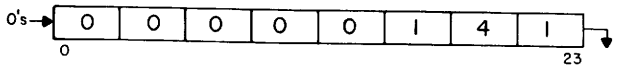
INSTRUCTION



BEFORE EXECUTION

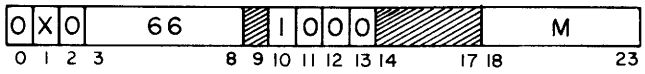


AFTER EXECUTION



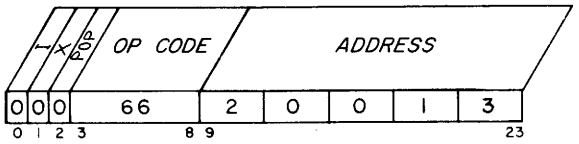
N=M;$N=20_8 = 16_{10}$ Time = 2+N/4 = 2+16/4 = 6 cycles

RCY RIGHT CYCLE AC, EA

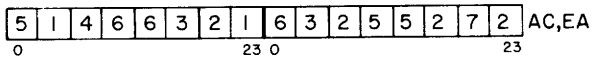


The AC and EA are treated as a single 48 bit register and rotated to the right N positions where N=M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of EA_{23} are shifted into AC_0. Bits shifted out of AC_{23} are shifted into EA_0. No bits are lost. The LC is incremented by 1.

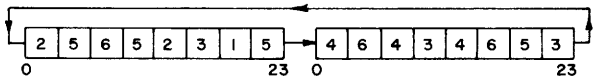
INSTRUCTION



BEFORE EXECUTION



AFTER EXECUTION



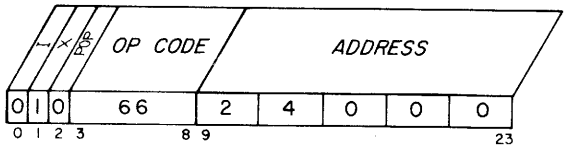
N=M; $N=13_8 = 11_{10}$ Time = 2+N/4 = 2+11/4 = 4 3/4 = 5 cycles

ARC RIGHT CYCLE AC

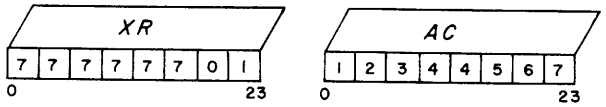
The AC is rotated right N positions where N=M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of AC_{23} are shifted into AC_0. No bits are lost. The LC is incremented by 1.

2 + N/4 cycles

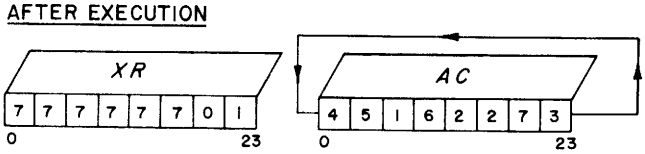
INSTRUCTION



BEFORE EXECUTION



AFTER EXECUTION



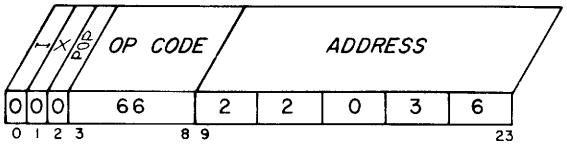
N=M+XR_{18-23};N=0+1=1 Time = 2+N/4 = 2 1/4 = 3 cycles

ERC RIGHT CYCLE EA

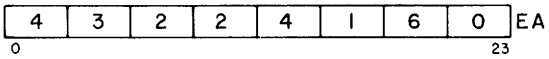
The EA is rotated right N positions where N=M if no indexing and N=M+XR_{18-23} if indexing is specified. Bits shifted out of EA_{23} are shifted into EA_0. No bits are lost. The LC is incremented by 1.

2 + N/4 cycles

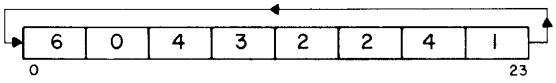
INSTRUCTION



BEFORE EXECUTION



AFTER EXECUTION



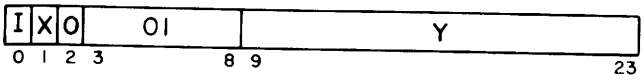
N=M;$N=36_8 = 30_{10}$ Time = 2+N/4 = 2+30/4 = 9 1/2 = 10 cycles

BRANCH INSTRUCTIONS

The branch instructions are:

Mnemonic	Op Code	Name
BRA	01	Branch
BAP	26	Branch if AC Positive
BAN	24	Branch if AC Negative
BAZ	25	Branch if AC Zero
BEN	22	Branch if EA Negative
BIX	41	Increment XR and Branch if Negative
BSL	43	Store Location Counter and Branch
BRT	51	Return Branch
BRI	11	Branch and Clear Interrupt

BRA BRANCH



The LC is set to the effective location. Instruction execution proceeds with the instruction at the effective location.

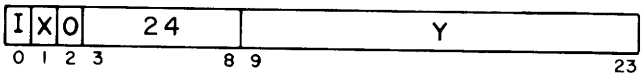
1 cycle

BAP BRANCH IF AC POSITIVE

If $AC_0 = 0$, instruction execution proceeds with the instruction at the effective location. (The LC is set to the effective location.) Otherwise, instruction execution proceeds in the normal sequence. (The LC is incremented by 1.)

1 cycle

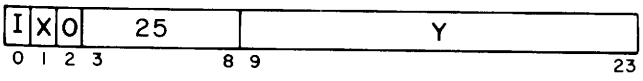
BAN BRANCH IF AC NEGATIVE



If $AC_0 = 1$, instruction execution proceeds with the instruction at the effective location. (The LC is set to the effective location.) Otherwise, instruction execution proceeds in the normal sequence. (The LC is incremented by 1.)

1 cycle

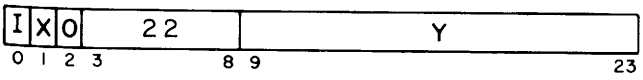
BAZ BRANCH IF AC ZERO



If the AC contains all zeros, instruction execution proceeds with the instruction at the effective location. (The LC is set to the effective location.) Otherwise, instruction execution proceeds in the normal sequence. (The LC is incremented by 1.)

1 cycle

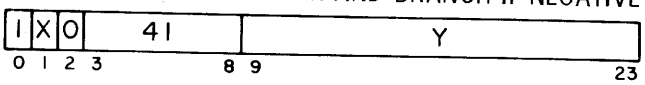
BEN BRANCH IF EA NEGATIVE



If $EA_0 = 1$, instruction execution proceeds with the instruction at the effective location. (The LC is set to the effective location.) Otherwise, instruction execution proceeds in the normal sequence. (The LC is incremented by 1.)

1 cycle

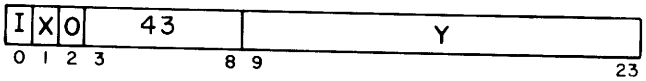
BIX INCREMENT XR AND BRANCH IF NEGATIVE



The contents of XR is incremented by one. If $XR_8 = 1$ after incrementing, instruction execution proceeds with the instruction at the effective location. (The LC is set to the effective location.) Otherwise, instruction execution proceeds in the normal sequence. (The LC is incremented by 1.)

1 cycle

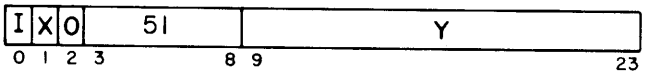
BSL STORE LOCATION COUNTER AND BRANCH



The operand is reset to zero. Bit 0 of the operand is set; bit 3 of the operand is specified by OVF; bit 4 of the operand is specified by CRO; the LC is placed in bits 9-23 of the operand; OVF and CRO are reset; the LC is set to the effective location plus one. Instruction proceeds at the effective location plus one.

2 cycles

BRT RETURN BRANCH



The contents of the effective location is treated as follows:

the OR of bit 3 and OVF is placed in OVF,

the OR of bit 4 and CRO is placed in CRO,

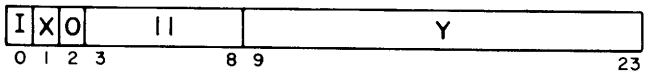
bits 9-23 are placed in the LC.

The contents of the effective location is not altered. The LC is incremented by 1. Instruction execution proceeds at the location now specified by the LC.

2 cycles

The BRT in conjunction with BSL facilitates getting to and returning from a subroutine.

BRI BRANCH AND CLEAR INTERRUPT



The highest priority active interrupt is cleared. The contents of the effective location is treated as follows:

bit 3 is placed in OVF,

bit 4 is placed in CRO,

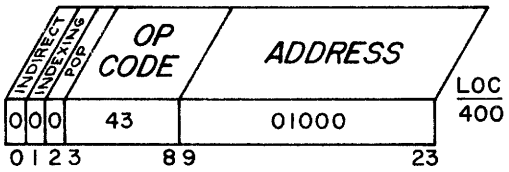
bits 9-23 are placed in the LC.

The contents of the effective location is not altered. Instruction execution continues at the location now specified by the LC.

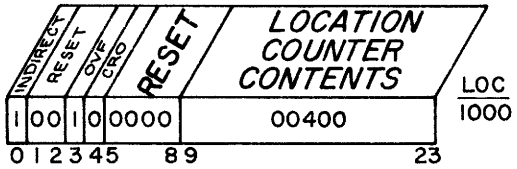
2 cycles

Interrupts are inhibited until at least one instruction following the BRI has been executed. The BRI in conjunction with BSL facilitates getting to and returning from interrupt subroutines.

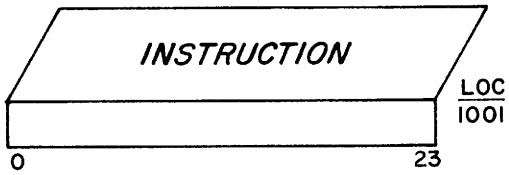
BSL, BRT OPERATION



Loc 400 contains a BSL instruction. Assume OVF is set, CRO is reset. The instruction at 377 has been executed. The instruction at 400 is executed.

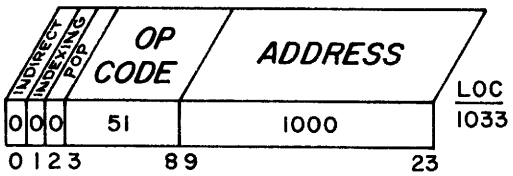


Loc 1000 is reset. Indirect bit is set. OVF and CRO are placed in bits 3 and 4. The LC is placed in bits 9-23. OVF and CRO are reset.



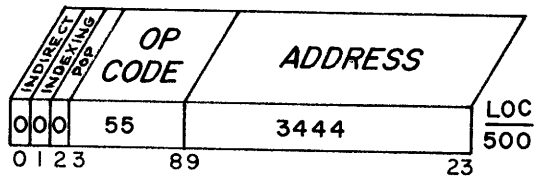
Instruction execution proceeds with the instruction at loc 1001.

Assume instruction execution continues sequentially.

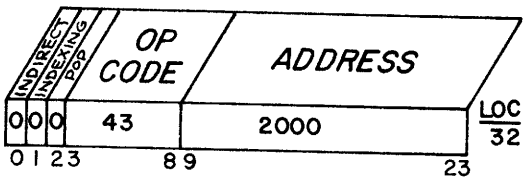


Loc 1033 contains a BRT instruction. Bits 3 and 4 of 1000 are ORed with OVF and CRO and placed in OVF and CRO. Bits 9-23 of 1000 are placed in LC. The LC is incremented. The contents of 1000 are not altered and instruction execution proceeds at 401_8.

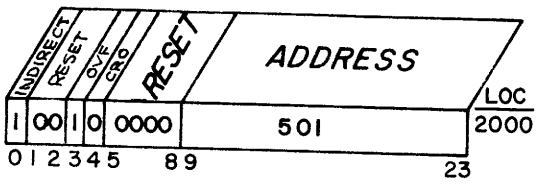
BSL, BRI OPERATION



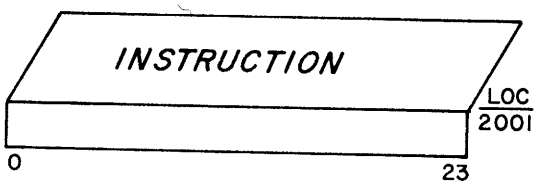
Loc 500 contains an ADD instruction. During execution of this instruction, interrupt 32_8 occurs. The LC is incremented to 501_8. The instruction at 32_8 is executed.



Loc 32 contains a BSL instruction. Assume OVF is set, CRO is reset.

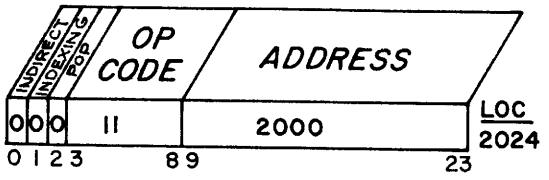


Loc 2000 is reset. Indirect bit is set. OVF and CRO are placed in bits 3 and 4. The LC is placed in bits 9-23. OVF and CRO are reset.



Instruction execution proceeds with the instruction at loc 2001_8.

Assume instruction execution continues sequentially.



Loc 2024 contains a BRI instruction. Bits 3 and 4 of 2000 are placed in OVF and CRO, respectively. Bits 9-23 of 2000 are placed in LC. Interrupt 32_8 is cleared. Instruction execution proceeds at 501. The contents of 2000 are not altered.

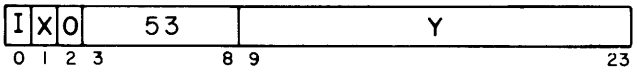
SKIP INSTRUCTIONS

The skip instructions are:

Mnemonic	Op Code	Name
SMN	53	Skip if Memory Negative
SAG	73	Skip if AC Greater Than Memory
SAE	70	Skip if AC Equals Memory Thru EA Mask
SAM	72	Skip if AND of AC With Memory Equals Zero
SNS	40	Skip if Signal Not Set

Skip instructions test the presence of a condition. If the condition is present, the location counter is incremented by two, thus skipping the instruction that would have been next. If the condition is not present, the location counter is incremented by one as normal. For example, if location 2005 contains a Skip if Overflow Indicator Reset instruction and OVF is reset, the instruction execution proceeds at location 2007. If OVF had been set, the instruction at 2006 would have been executed.

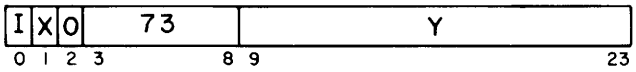
SMN SKIP IF MEMORY NEGATIVE



The LC is incremented by 2 (skip) if bit 0 of the operand is set. The LC is incremented by 1 (no skip) if bit 0 of the operand is reset. The operand is not altered.

2 cycles

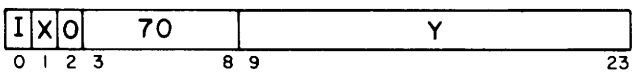
SAG SKIP IF AC GREATER THAN MEMORY



The LC is incremented by 2 (skip) if the AC contents is algebraically greater than the operand. The LC is incremented by 1 (no skip) if the AC contents is algebraically less than or equal to the operand. The AC and operand are not altered.

3 cycles

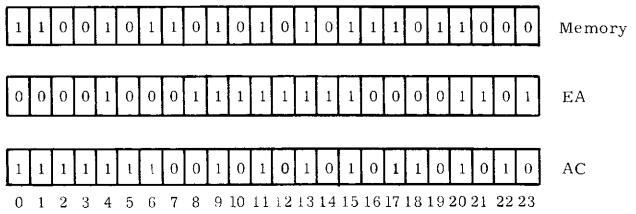
SAE SKIP IF AC EQUALS MEMORY THRU EA MASK



For each bit position in the EA containing a 1, the corresponding bit position in the AC and the operand are compared. The LC is incremented by 2 (skip) if every specified position compares equal. The LC is incremented by 1 (no skip) if any of the specified positions do not compare equal. The AC and the operand are not altered.

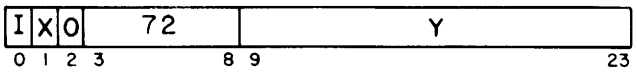
3 cycles

If the EA contains all ones (7777777_8), SAE is effectively an AC equal to memory test. If EA contains all zeros, a skip will occur.



EA specifies a comparison of bits 4, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 23. A skip would result since Memory and AC are identical in these positions.

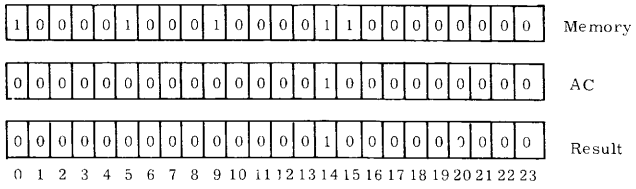
SAM SKIP IF AND OF AC AND MEMORY EQUALS ZERO



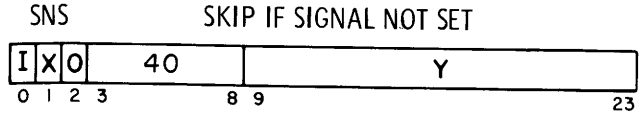
The LC is incremented by 2 (skip) if the AND of the operand and AC produces all zeros. The LC is incremented by 1 (no skip) if the AND of the operand and AC does not produce all zeros. The AC and operand are not altered.

2 cycles

If the AC and operand both contain a 1 in the same position, a skip does not occur.



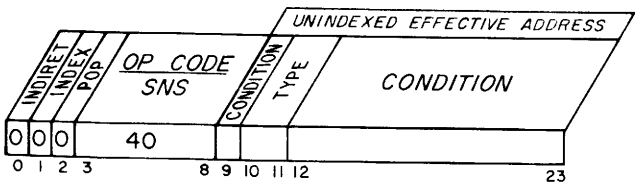
In the preceding example a skip will not occur since the AND of AC with Memory is not zero.



Input/output conditions, internal conditions and external conditions are tested with the SNS instruction. The second and third high order bits of the unindexed effective address designate the SNS as an input/output, internal or external condition test. The high order bit and the fourth thru fifteenth bit of the unindexed effective address designate the condition to be tested. Since the unindexed effective address is the effective address determined by ignoring indexing at the last level, bits 9-23 of the instruction word are the unindexed effective address if indexing and indirect addressing are not specified. In order to reflect common usage and simplify the explanations, all examples and descriptions are given without indexing and indirect addressing.

2 cycles

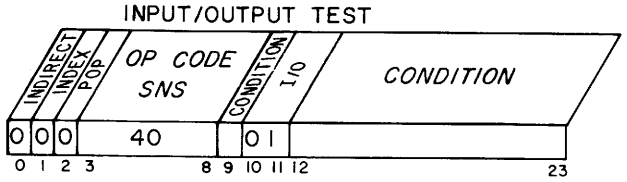
The following example illustrates the interpretation of bits 9-23 of the SNS instruction if indexing and indirect addressing are not specified.

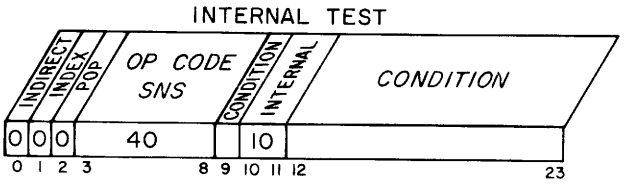


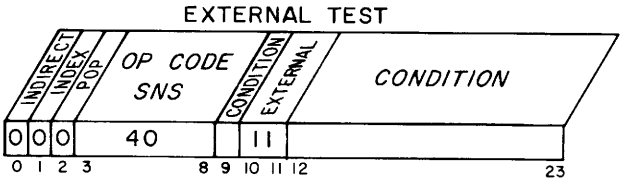
Bits 10-11 designate the SNS as an input/output, internal or external condition test as follows:

	10	11
Input/output condition test	0	1
Internal condition test	1	0
External condition test	1	1

The following examples illustrate the interpretation of bits 10-11 of the SNS instruction if indexing and indirect addressing are not specified.

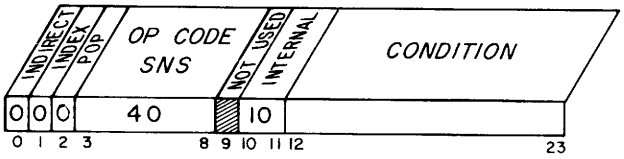






Internal SNS Instructions

The internal test instructions are SNS 2 XXXX (0 40 2 XXXX) instructions where XXXX specifies the condition to be tested.



Each of the twelve bits in bits 12-23 specifies a condition to be tested. Normally a skip tests only one of the twelve conditions. For example, if bit 15 is set, breakpoint switch 1 is tested for being reset. If bit 16 is set, breakpoint switch 2 is tested for being reset. A test of more than one condition may be specified. In this case, if any of the conditions tested are present, a skip will occur. For example, if bits 15 and 16 are set, a skip would

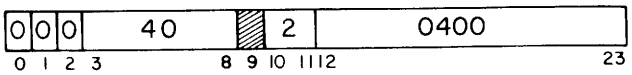
occur if breakpoint switch 1 OR 2 is reset. If bits 12-23 are reset, then no test is specified and a skip will occur. The interpretation of bits 12-23 is:

Bit	Skip Condition
12	Carry Out Indicator Reset
13	C Buffer Ready
14	B Buffer Ready
15	Breakpoint 1 Reset
16	Breakpoint 2 Reset
17	Breakpoint 3 Reset
18	Breakpoint 4 Reset
19	C Buffer Error Indicator Reset
20	B Buffer Error Indicator Reset
21	Interrupt System Enabled
22	Interrupt System Enabled by Program
23	Overflow Indicator Reset

The assembler defined internal test instructions are:

Mnemonic	Octal Code	Name
SBK1	0 40 2 0400	Skip if Breakpoint 1 Reset
SBK2	0 40 2 0200	Skip if Breakpoint 2 Reset
SBK3	0 40 2 0100	Skip if Breakpoint 3 Reset
SBK4	0 40 2 0040	Skip if Breakpoint 4 Reset
SNC	0 40 2 4000	Skip if CRO Reset
SOV	0 40 2 0001	Skip if OVF Reset
SIE	0 40 2 0004	Skip if Interrupts Enabled
SPC	0 40 2 0002	Skip if Interrupts Enabled by Program
BER	0 40 2 0010	Skip if No Error, B Buffer
CER	0 40 2 0020	Skip if No Error, C Buffer
BRD	0 40 2 1000	Skip if Ready, B Buffer
CRD	0 40 2 2000	Skip if Ready, C Buffer

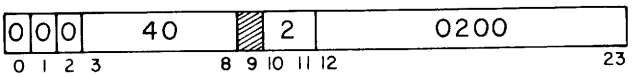
SBK1 SKIP IF BREAKPOINT 1 RESET



The LC is incremented by 2 (skip) if breakpoint switch 1 on the system console is reset. The LC is incremented by 1 (no skip) if breakpoint switch 1 on the system console is set.

2 cycles

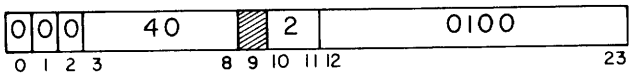
SBK2 SKIP IF BREAKPOINT 2 RESET



The LC is incremented by 2 (skip) if breakpoint switch 2 on the system console is reset. The LC is incremented by 1 (no skip) if breakpoint switch 2 on the system console is set.

2 cycles

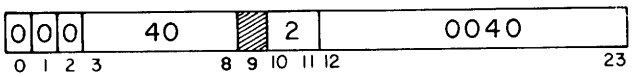
SBK3 SKIP IF BREAKPOINT 3 RESET



The LC is incremented by 2 (skip) if breakpoint switch 3 on the system console is reset. The LC is incremented by 1 (no skip) if breakpoint switch 3 on the system console is set.

2 cycles

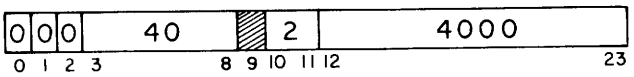
SBK4 SKIP IF BREAKPOINT 4 RESET



The LC is incremented by 2 (skip) if breakpoint switch 4 on the system console is reset. The LC is incremented by 1 (no skip) if breakpoint switch 4 on the system console is set.

2 cycles

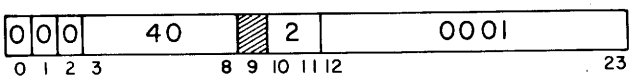
SNC SKIP IF CRO RESET



The LC is incremented by 2 (skip) if the carry out indicator is reset. The LC is incremented by 1 (no skip) and the carry out indicator is reset if the carry out indicator is set.

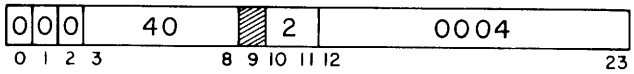
2 cycles

SOV SKIP IF OVF RESET



The LC is incremented by 2 (skip) if the overflow indicator is reset. The LC is incremented by 1 (no skip) and the overflow indicator is reset if the overflow indicator is set.

SIE SKIP IF INTERRUPTS ENABLED

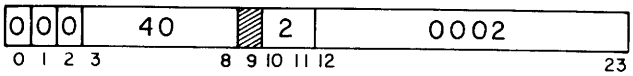


The LC is incremented by 2 (skip) if the interrupt system is enabled. The LC is incremented by 1 (no skip) if the interrupt system is disabled.

2 cycles

The interrupt system may be enabled by the program (ENA Instruction) or by the operator from the system console.

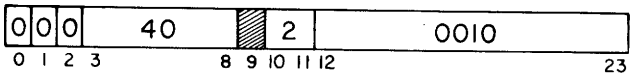
SPC SKIP IF INTERRUPTS ENABLED BY PROGRAM



The LC is incremented by 2 (skip) if the interrupt system was enabled by the execution of the ENA instruction. The LC is incremented by 1 (no skip) if the interrupt system is disabled or if the interrupt system is presently enabled from the enable switch on the system console and not by the ENA instruction.

2 cycles

BER SKIP IF NO ERROR, B BUFFER

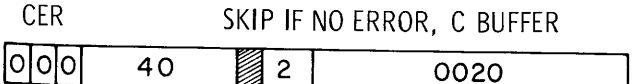


The LC is incremented by 2 (skip) if the B buffer error indicator is off. The LC is incremented by 1 (no skip) if the B buffer error indicator is on.

2 cycles

The error indicator is set by a buffer rate error (hardware malfunction) or detection of a parity error during input. The error indicator is reset by any buffer control Activate instruction which addresses the B buffer.

CER SKIP IF NO ERROR, C BUFFER

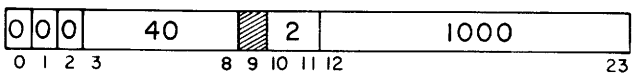


The LC is incremented by 2 (skip) if the C buffer error indicator is off. The LC is incremented by 1 (no skip) if the C buffer error indicator is on.

2 cycles

The buffer error indicator is set by a buffer rate error (hardware malfunction) or detection of a parity error during input. The error indicator is reset by any buffer control Activate instruction which addresses the C buffer.

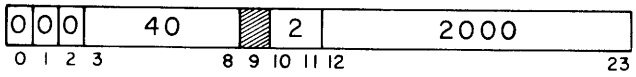
BRD SKIP IF READY, B BUFFER



The LC is incremented by 2 (skip) if the B buffer is ready (not busy). The LC is incremented by 1 (no skip) if the B buffer is not ready (busy).

2 cycles

CRD SKIP IF READY, C BUFFER



The LC is incremented by 2 (skip) if the C buffer is ready (not busy). The LC is incremented by 1 (no skip) if the C buffer is not ready (busy).

2 cycles

MISCELLANEOUS INSTRUCTIONS

The miscellaneous instructions are:

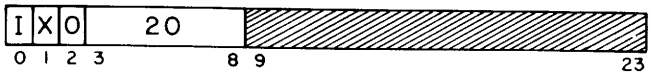
Mnemonic	Op Code	Name
HLT	00	Halt
NOP	20	No Operation
XEC	23	Execute

HLT HALT

The computer halts. If the RUN switch is in the RUN position, the location counter contains the address of the next instruction (HALT instruction plus one). Instruction execution continues by putting the RUN switch in IDLE then RUN.

2 cycles

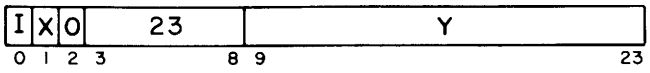
NOP NO OPERATION



The LC is incremented by 1. The NOP has no other effect.

1 cycle

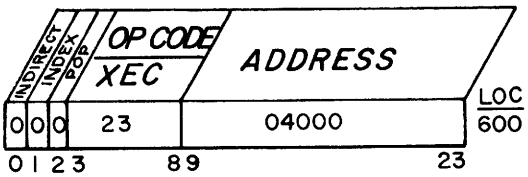
XEC EXECUTE



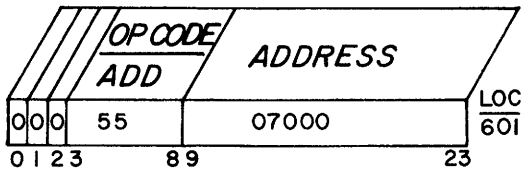
The instruction at the effective location is executed. The LC is incremented by 1 if the instruction at the effective location is not a skip or branch instruction. If the instruction at the effective location is a branch instruction and the condition tested by the branch instruction is true, the branch is taken (the LC is set to the effective location specified by the branch instruction). If the branch is not taken, the LC is incremented by 1 (the instruction following XEC is next). If the instruction at the effective location is a skip instruction, the LC is incremented by 2 or by 1 and the skip takes place relative to the XEC rather than the skip instruction. If an interrupt condition is acknowledged during an XEC, the interrupt does not occur until after the instruction at the effective location is executed.

1 cycle + time of instruction executed

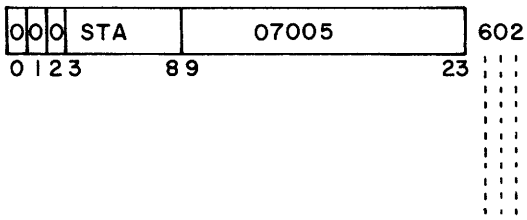
EXECUTING AN ADD

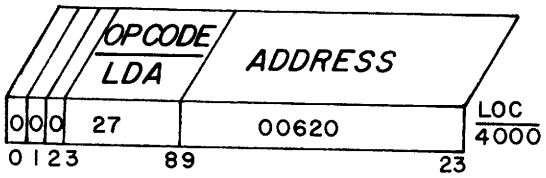


Loc 600 contains an XEC instruction. The instruction at loc 4000 is to be executed.



Loc 601 contains an ADD instruction. The instruction at loc 4000 has been executed. The instruction at 601 is now executed. The next instruction is at loc 602.



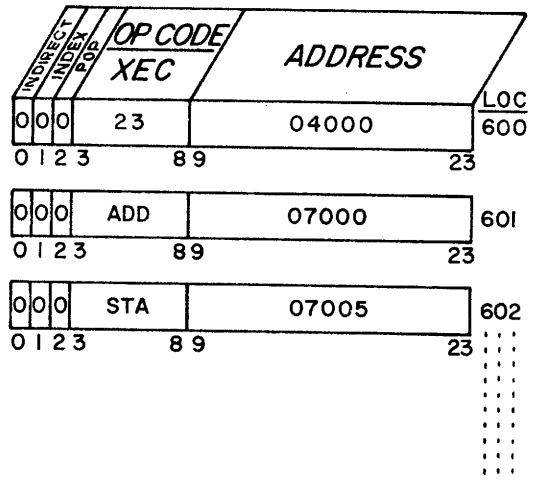


Loc 4000 contains a LDA instruction. This instruction is executed by the XEC at 600 and the LC is incremented to 601.

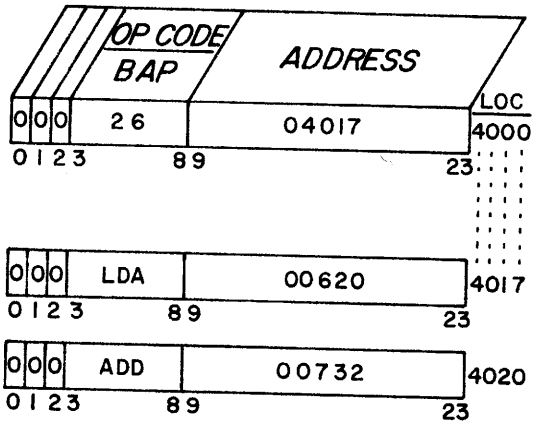
INSTRUCTION EXECUTION SEQUENCE

600	XEC	04000
4000	LDA	00620
601	ADD	07000
602	. . . . .	

EXECUTING A BRANCH



Loc 600 contains an XEC instruction. The instruction at loc 4000 is to be executed.

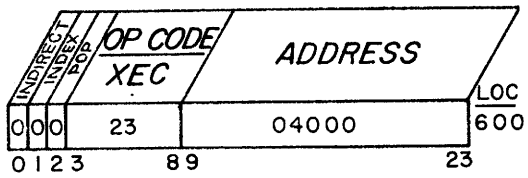


Loc 04000 contains a BAP instruction. If the AC is positive, 4017 is placed in the LC and instruction execution continues at 4017. If the AC is negative, the LC is incremented by 1 to 601 and instruction execution continues at 601.

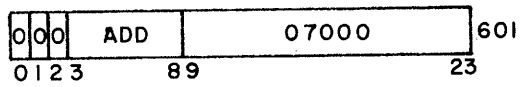
INSTRUCTION EXECUTION SEQUENCE

AC Positive			AC Negative		
600	XEC	04000	600	XEC	04000
4000	BAP	04017	4000	BAP	04017
4017	LDA	00620	601	ADD	07000
4020	ADD	00732	602	STA	07005
4021	. . . . .		. . . . .		

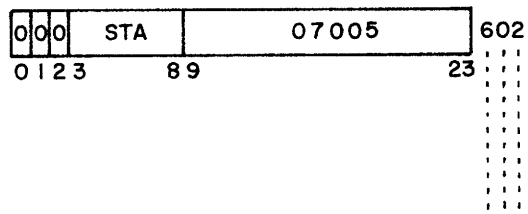
EXECUTING A SKIP



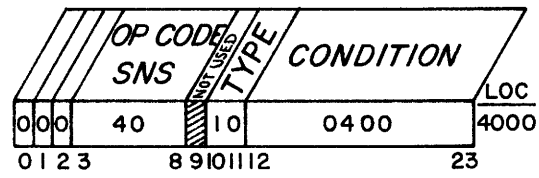
Loc 600 contains an XEC instruction. The instruction at loc 4000 is to be executed.



Instruction execution resumes here if No Skip occurs.



Instruction execution resumes here if a Skip occurs.



Loc 4000 contains a Skip If Breakpoint 1 is Reset instruction. If Breakpoint 1 is Reset, the LC is incremented to 602 and instruction execution continues at 602. If Breakpoint 1 is set, the LC is incremented to 601 and instruction execution continues at 601.

INSTRUCTION EXECUTION SEQUENCE

No Skip			Skip		
600	XEC	04000	600	XEC	04000
4000	SNS	20400	4000	SNS	20400
601	ADD	07000	602	STA	07005
602	. . . . .		. . . . .		

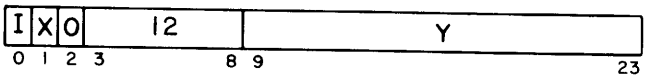
INPUT/OUTPUT INSTRUCTIONS

The Input/Output instructions are:

Mnemonic	Op Code	Name
TMB	12	Transfer Memory to B Buffer
TBM	32	Transfer B Buffer to Memory
TMC	10	Transfer Memory to C Buffer
TCM	30	Transfer C Buffer to Memory
WTP	13	Write Parallel
RDP	33	Read Parallel
ACT	02	Activate

The TMB, TBM, TMC, TCM, WTP and RDP instructions transfer a word between the memory and the B buffer, C buffer or C register whereas buffer control, device control and control of external devices is accomplished with the ACT instruction.

TMB TRANSFER MEMORY TO B BUFFER



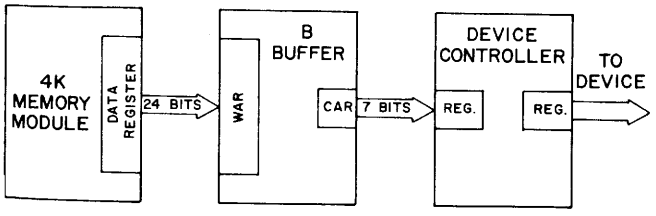
The operand is placed in the Word Assembly Register of the B buffer. The operand is not altered. The LC is incremented by 1.

2 + Wait cycles

Detailed Operation

If the Word Assembly Register, WAR, of the B buffer is ready to receive a word from memory, the operand is transferred immediately and execution time is 2 cycles. If the WAR is not ready to receive the operand, the computer waits until the WAR is ready before transferring the operand and the execution time is 2 cycles plus the wait time.

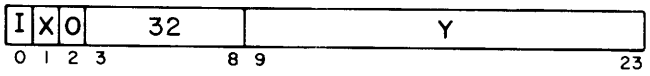
TMB
DATA FLOW



WAR	Word Assembly Register
CAR	Character Assembly Register
Reg.	Register

The TMB transfers a word from memory to the B buffer and the computer continues instruction execution. The B buffer disassembles the word into characters and sends 7 bit characters (6 data bits + 1 parity bit) to the device controller which sends the data to the device.

TBM TRANSFER B BUFFER TO MEMORY



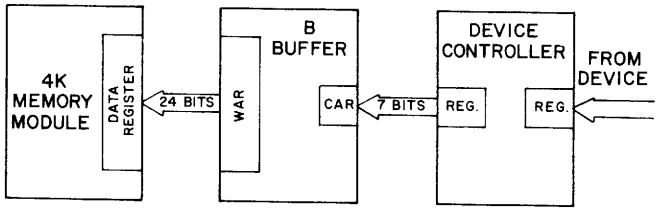
The contents of the Word Assembly Register of the B buffer is placed in the effective location. The LC is incremented by 1.

2 + Wait cycles

Detailed Operation

If the WAR, Word Assembly Register, of the B buffer is ready to be read into memory, the contents of the WAR is transferred immediately and the execution time is 2 cycles. If the WAR is not ready to be read into memory, the computer waits until the WAR is ready to be read into memory before transferring the WAR contents into memory and execution time is 2 cycles plus the wait time.

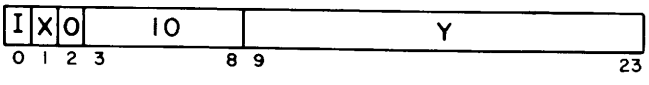
TBM
DATA FLOW



WAR	Word Assembly Register
CAR	Character Assembly Register
Reg.	Register

The device sends data to the device controller which sends 7 bit (6 data bits + 1 parity bit) characters to the B buffer. The B buffer assembles the characters into a 24 bit data word in the Word Assembly Register. The TBM transfers the word from the B buffer WAR to the memory module.

TMC TRANSFER MEMORY TO C BUFFER



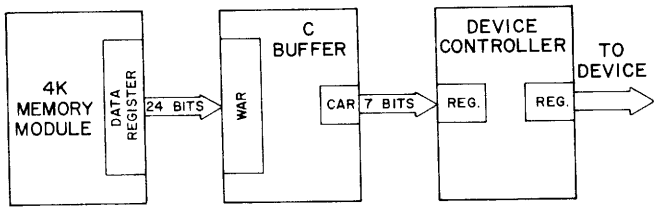
The operand is placed in the Word Assembly Register of the C buffer. The operand is not altered. The LC is incremented by 1.

2 + Wait cycles

Detailed Operation

If the Word Assembly Register of the C buffer is ready to receive a word from memory, the operand is transferred immediately and execution time is 2 cycles. If the WAR is not ready to receive the operand, the computer waits until the WAR is ready before transferring the operand and the execution time is 2 cycles plus the wait time.

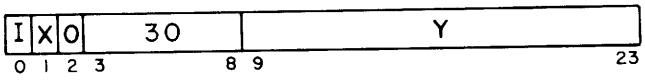
TMC
DATA FLOW



WAR	Word Assembly Register
CAR	Character Assembly Register
Reg.	Register

The TMC transfers a word from memory to the C buffer and the computer continues instruction execution. The C buffer disassembles the word into characters and sends 7 bit characters (6 data bits and 1 parity bit) to the device controller which sends the data to the device.

TCM TRANSFER C BUFFER TO MEMORY

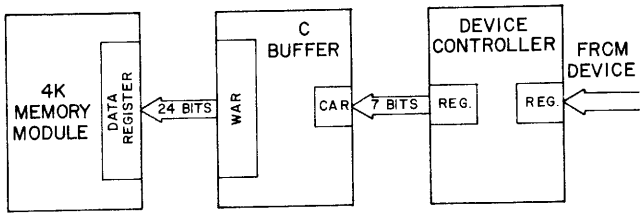


The contents of the Word Assembly Register of the C buffer is placed in the effective location. The LC is incremented by 1.

2 + wait cycles

If the Word Assembly Register, WAR, of the C buffer is ready to be read into memory, the contents of the WAR is transferred immediately and the execution time is 2 cycles. If the WAR is not ready to be read into memory, the computer waits until the WAR is ready to be read into memory before transferring the WAR contents into memory and execution time is 2 cycles plus the wait time.

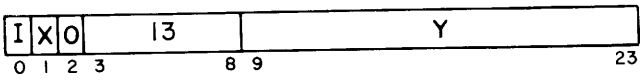
TCM
DATA FLOW



WAR	Word Assembly Register
CAR	Character Assembly Register
Reg.	Register

The device sends data to the device controller which sends 7 bit (6 data bits + 1 parity bit) characters to the C buffer. The C buffer assembles the characters into a 24 bit data word in the Word Assembly Register. The TCM transfers the word from the C buffer WAR to the memory module.

WTP WRITE PARALLEL



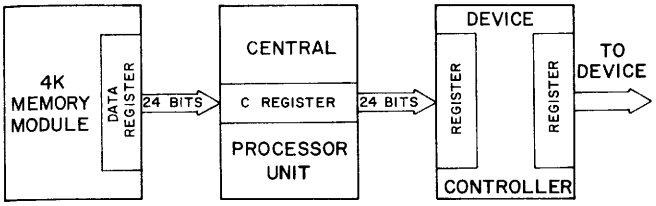
The operand is transferred to the word register of the active device controller via the Central Processor Unit C Register. The operand is not altered. The LC is incremented by 1.

2 + Wait cycles

Detailed Operation

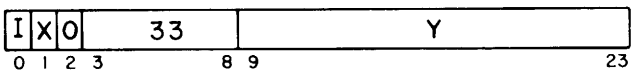
The operand is transferred from memory to the C register of the CPU. If the word register of the device controller is ready to receive the operand from the C register, the operand is transferred from the C register immediately and execution time is 2 cycles. If the word register of the device controller is not ready to receive the operand from the C register, the computer waits until the word register is ready before transferring the operand from the C register and execution time is 2 cycles plus the wait time.

WTP
DATA FLOW



The WTP transfers a word from memory to the C register of the CPU. When the device controller is ready to receive the word from the C register, the word is sent to the device controller register and instruction execution continues. The device controller sends the word to the device.

RDP READ PARALLEL



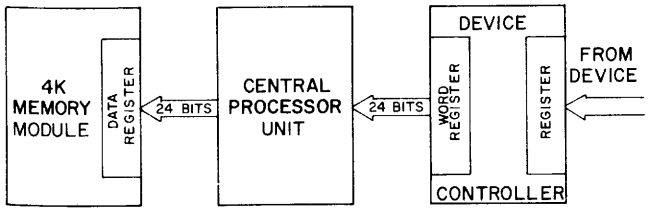
The contents of the word register of the device controller is placed via the CPU in the effective location. The LC is incremented by 1.

2 + Wait cycles

Detailed Operation

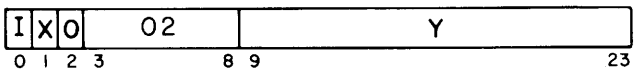
If the word register of the device controller is ready to be read into memory, the contents of the device controller's word register is transferred immediately and the execution time is 2 cycles. If the device controller's word register is not ready to be read into memory, the computer waits until the register is ready to be read into memory before transferring the register contents into memory, consequently, the execution time is 2 cycles plus the wait time.

RDP
DATA FLOW



The device sends data to the device controller which places the data in its word register. When the word register is ready to be read, the RDP transfers the data word in the device controller's word register to the memory module via the CPU.

ACT ACTIVATE

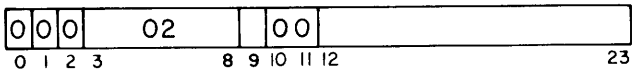


Buffer control, device control, internal control and external system control are accomplished with the ACT instruction. The second and third high order bits of the unindexed effective address designate the ACT as buffer control, device control, internal control or external system control. The high order bit and the fourth thru fifteenth bit of the unindexed effective address contain the control information. Since the unindexed effective address is the effective address determined by ignoring indexing at the last level, bits 9-23 of the instruction word are the unindexed effective address if indexing and indirect addressing are not specified. In order to reflect common usage and simplify explanations, all examples and descriptions are given without indexing and indirect addressing.

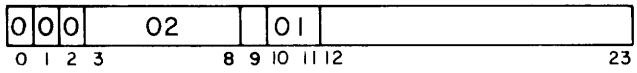
1 cycle

The following examples illustrate the instruction word format for the buffer control, device control and external system control Activate.

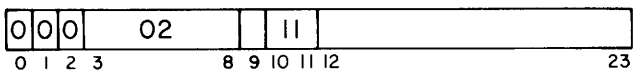
BUFFER CONTROL



DEVICE CONTROL



EXTERNAL SYSTEM CONTROL



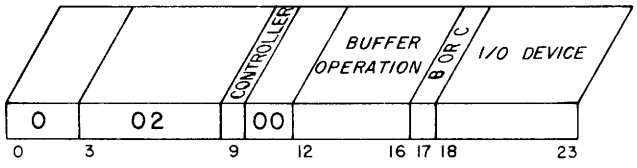
Buffer Control Activate

The buffer control Activate specifies:

	Bit Position
1. channel controller operation	9
2. buffer operation	12-16

3. B or C buffer	17
4. an I/O device	18-23

The format of the buffer control Activate is illustrated below:



Bit 9 Channel Controller Operation

The interpretation of bit 9 is:

Bit 9	Meaning
1	Enable Channel Controller
0	Disable Channel Controller

If bit 9 is set, the channel controller controls the transfer of data between memory and the buffer specified by bit 17. If bit 9 is reset, the channel controller is deactivated and the transfer of data between memory and the buffer specified by bit 17 is accomplished by the single word I/O instructions (TBM, TMB, TCM, TMC). To control data transfer, the channel controller must have a 9 or 15 bit word count specifying the number of words to be transferred and a 15 bit memory address specifying the address of the first word to be transferred. A 9 bit word count allows a transfer of up to 2^9-1 or 511_{10} words. A 15 bit count allows a transfer of up to 2^{15}-1 or $32{,}767_{10}$ words. The WTP (Write Parallel) instruction is used to transfer a 15 bit address and 9 bit word count to the channel controller. If a 15 bit word count is required, the high order 6 bits of the word count are transferred to the channel controller by a device control Activate which must follow immediately after the buffer control Activate. The low order 9 bits of the word count and the 15 bit address are transferred to the channel controller by a WTP instruction. The sequence is:

1. buffer control Activate to enable channel controller
2. ACT to transfer high order 6 bits of word count
3. WTP to transfer low order 9 bits of word count and 15 bit address.

If 511_{10} words or less are to be transferred, the ACT following the buffer control Activate is unnecessary and the sequence is:

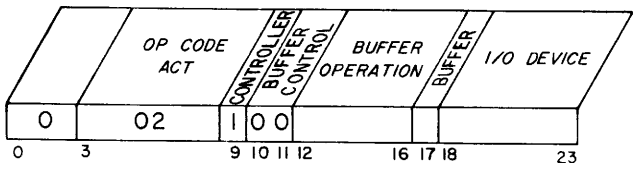
1. buffer control Activate to enable channel controller
2. WTP to transfer 9 bit word count and 15 bit address.

The WTP instruction transfers a word from memory to the C register of the CPU. The contents of the C register are transferred to the channel controller of the buffer specified in the preceding ACT instruction. For example, if a buffer control ACT has enabled the channel controller for the B buffer, then the instruction WTP 03000 will transfer the contents of memory location 03000_8 to the channel controller for the B buffer. Bits 0-8 of the word transferred to the channel controller specify the 9 bit word count. Bits 9-23 of the word transferred to the channel controller specify the 15 bit address.

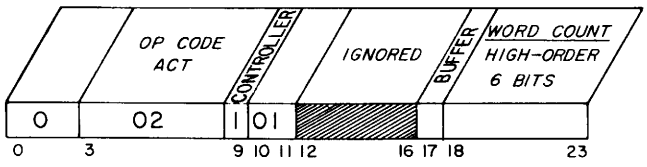
CHANNEL CONTROLLER WORD

The following example illustrates the format of the Activate instructions to enable the channel controller and specify the high order 6 bits of the word count.

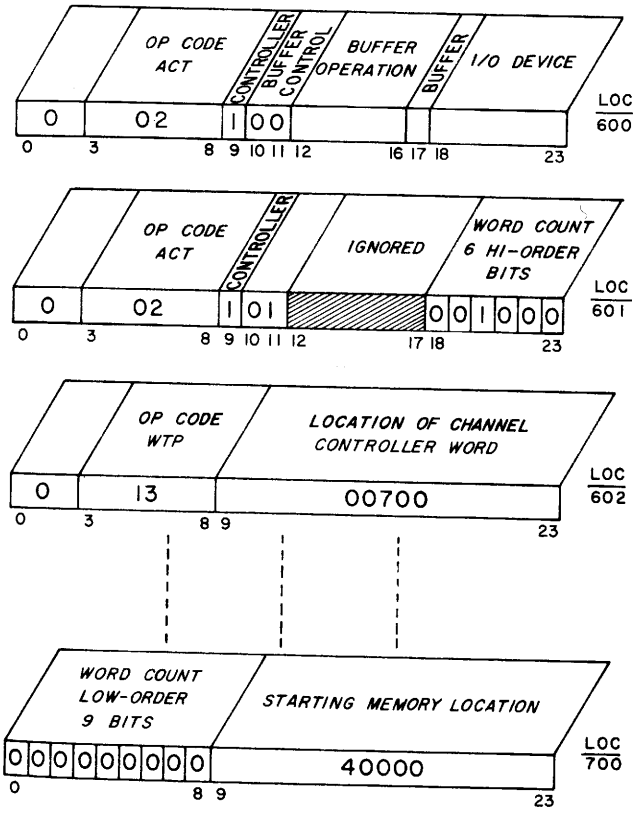
BUFFER CONTROL ACTIVATE



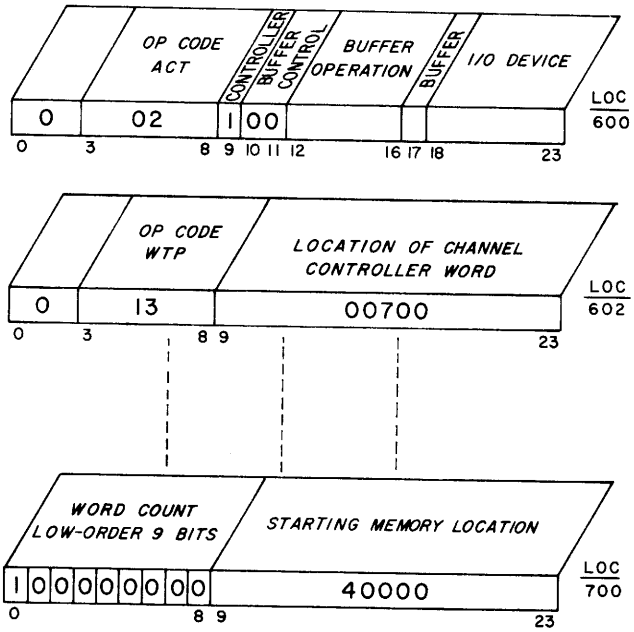
DEVICE CONTROL ACTIVATE



The instruction sequence to enable the channel controller and transfer a word count of 4096_{10} words and a starting address of 4000_8 would be:



The instruction sequence to enable the channel controller, transfer a word count of 256_{10} words and a starting address of 40000_8 would be



Bits 12-16 Buffer Operation

The interpretation of bits 12-16 is:

Bits 12-16	Meaning
12	Direction of Movement
13	Leader
14	Parity
15-16	Characters per Word

Bit 12 Direction of Movement

The interpretation of bit 12 is:

Bit 12	Meaning
1	Reverse Direction
0	Forward Direction

If bit 12 is set, the data medium's direction is backward or reverse. If bit 12 is reset, the direction is forward. For example, if magnetic tape were to be read in the forward direction, bit 12 would be reset. If magnetic tape were to be read in the reverse direction, bit 12 would be set.

Bit 13 Leader

The interpretation of bit 13 is:

Bit 13	Meaning
1	No Leader
0	Leader

For example, bit 13 would be set if it were desired to punch a paper tape with leader. If a leader is not desired, bit 13 would be reset.

Bit 14 Parity

The interpretation of bit 14 is:

Bit 14	Meaning
1	Odd Parity
0	Even Parity

If bit 14 is set, the data parity is odd (binary). If bit 14 is reset, the data parity is even (BCD).

Bits 15-16 Characters per Word

The interpretation of bits 15-16 is:

Bits 15-16	Meaning
00	1 Character per Word
01	2 Characters per Word
10	3 Characters per Word
11	4 Characters per Word

For output operations, the interpretation of bits 15-16 of the buffer control Activate is:

Bits 15-16	Meaning
00	1st CHAR | IGNORED | IGNORED | IGNORED (0, 5 6, 11 12, 17 18, 23)
01	1st CHAR | 2 nd CHAR | IGNORED | IGNORED (0, 5 6, 11 12, 17 18, 23)
10	1st CHAR | 2 nd CHAR | 3rd CHAR | IGNORED (0, 5 6, 11 12, 17 18, 23)
11	1st CHAR | 2 nd CHAR | 3rd CHAR | 4th CHAR (0, 5 6, 11 12, 17 18, 23)

For example, if bits 15-16 contain 11_2, all four 6 bit characters of each word will be outputted with bits 0-5 of each word being the first character and bits 18-23 being the fourth character. If bits 15-16 contain 01_2, only the first two 6 bit characters of each word will be outputted, that is, bits 0-5 and bits 6-11 of each word will be outputted.

The interpretation of bits 15-16 of the buffer control Activate for input operations is:

Bits	Meaning
00	00 | 00 | 00 | 1st CHAR. (0, 5 6, 11 12, 17 18, 23)
01	00 | 00 | 1st CHAR. | 2nd CHAR. (0, 5 6, 11 12, 17 18, 23)
10	00 | 1st CHAR. | 2nd CHAR. | 3rd CHAR. (0, 5 6, 11 12, 17 18, 23)
11	1st CHAR. | 2nd CHAR. | 3rd CHAR. | 4th CHAR. (0, 5 6, 11 12, 17 18, 23)

During input, characters are received by the buffer and assembled into a word. Bits 15-16 of the buffer control Activate specify the number of data characters to be assembled per word. If bits 15-16 contain 11_2, four data characters are assembled into each word with the first contained in bits 0-5 and the fourth contained in bits 18-23. If bits 15-16 contain 01_2, two data characters are assembled per word into bits 12-17 and 18-23. The unused bit positions, bits 0-11 will contain zeros.

Bit 17 Buffer

The interpretation of bit 17 is:

Bit 17	Meaning
1	C Buffer
0	B Buffer

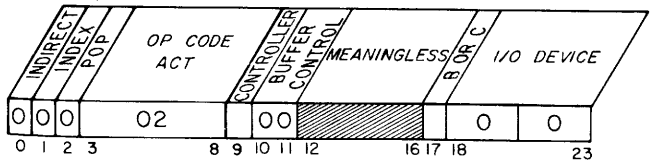
Bits 18-23 I/O Device

The interpretation of bits 18-23 is:

Bits 18-23	Meaning
00	Disconnect device from buffer
$01\text{-}77_8$	Connect device to buffer

If bits 18-23 of the buffer control Activate contain 00_8, and a device is connected to the buffer specified by bit 17, the device is immediately disconnected from the buffer. The device is disconnected regardless of any input/output operation which may be in progress. Thus, if the buffer is transferring data and a disconnect is specified, data transfer terminates immediately. If a disconnect is specified, bits 12-16 are ignored. The instruction word format of a disconnect is illustrated.

DISCONNECT



If bits 18-23 of the buffer control Activate do not contain 00_8, the designated input/output device is connected to the buffer specified by bit 17. The device code for each input/output device is:

INPUT

Bits 18-23	Device	Bits 18-23	Device
01	Typewriter 1	20	Unassigned
02	Typewriter 2 or Teletypewriter 2	21	Unassigned
		22	Unassigned
03	Teletypewriter 1	23	Unassigned
04	Paper Tape Reader 1	24	Unassigned
05	Paper Tape Reader 2	25	Unassigned
06	Card Reader 1	26	Disc File 1
07	Card Reader 2	27	Disc File 2
10	Magnetic Tape 0	30	Scan Magnetic Tape 0
11	Magnetic Tape 1	31	Scan Magnetic Tape 1
12	Magnetic Tape 2	32	Scan Magnetic Tape 2
13	Magnetic Tape 3	33	Scan Magnetic Tape 3
14	Magnetic Tape 4	34	Scan Magnetic Tape 4
15	Magnetic Tape 5	35	Scan Magnetic Tape 5
16	Magnetic Tape 6	36	Scan Magnetic Tape 6
17	Magnetic Tape 7	37	Scan Magnetic Tape 7

OUTPUT

Bits 18-23	Device	Bits 18-23	Device
41	Typewriter 1	55	Magnetic Tape 5
42	Typewriter 2 or Teletypewriter 2	56	Magnetic Tape 6
		57	Magnetic Tape 7
43	Teletypewriter 1		
		60	Line Printer 1
44	Paper Tape Punch 1		
45	Paper Tape Punch 2	61	Line Printer 2
46	Card Punch 1		
47	Card Punch 2	62	Plotter 1
		63	Plotter 2
50	Magnetic Tape 0	64	Plotter 3
51	Magnetic Tape 1	65	Plotter 4
52	Magnetic Tape 2		
53	Magnetic Tape 3	66	Disc File 1
54	Magnetic Tape 4	67	Disc File 2

OUTPUT

Bits 18-23	Device	Bits 18-23	Device
70	Erase Magnetic Tape 0	74	Erase Magnetic Tape 4
71	Erase Magnetic Tape 1	75	Erase Magnetic Tape 5
72	Erase Magnetic Tape 2	76	Erase Magnetic Tape 6
73	Erase Magnetic Tape 3	77	Erase Magnetic Tape 7

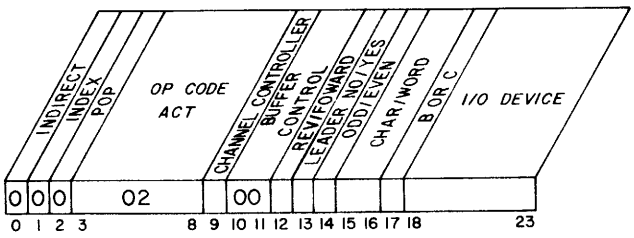
Summary

The buffer control Activate specifies:

1. channel controller operation
2. buffer operation
3. B or C buffer
4. an I/O device.

The format and interpretation of the buffer control Activate instruction word is as follows:

BUFFER CONTROL ACTIVATE



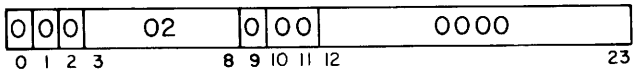
Bit Position	Interpretation
9	Channel Controller 1=Enable 0=Disable
10-11	Type of Activate 01_2=Buffer Control
12	Direction of Movement 1=Reverse 2=Forward
13	Tape Leader 1=No Leader 0=Leader
14	Parity 1=Odd 0=Even
15-16	Characters/Word 00=1 character 01=2 characters 10=3 character 10=4 characters
17	Buffer 1=C buffer 0=B buffer
18-23	I/O Device 000000=Disconnect 000001 . . . 111111 =I/O Device

Assembler Defined Buffer Control Activates

The assembler defined buffer control Activates for connecting I/O devices are discussed in the peripheral section of this manual. The assembler defined buffer control Activate for disconnecting I/O devices are:

Mnemonic	Octal Code	Name
DSC	0 02 0 0000	Disconnect B Buffer
DSCC	0 02 0 0100	Disconnect C Buffer

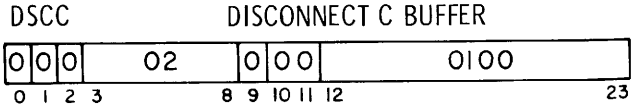
DSC DISCONNECT B BUFFER



Any device connected to the B Buffer is disconnected. If data transfer is in progress, the disconnect occurs after transfer of the character presently being sent or received.

1 cycle

DSCC DISCONNECT C BUFFER



Any device connected to the C Buffer is disconnected. If data transfer is in progress, the disconnect occurs after transfer of the character presently being sent or received.

Device Control Activate

The device control Activate specifies:

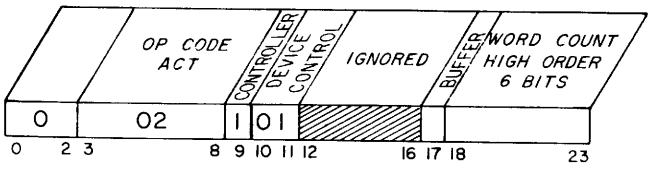
1. the high order six bits of the channel controller word count or
2. buffer information or
3. device control information.

High Order Six Bits of Word Count

If bit 9 is set, the interpretation of bits 12-23 is:

Bit	Meaning
12-16	Ignored
17	Buffer 1=C Buffer 0=B Buffer
18-23	High Order 6 bits of Word Count

The instruction word format is:



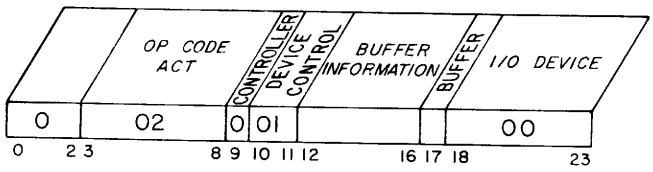
If the channel controller is not conditioned to accept the word count by a buffer control Activate, the channel controller will ignore this Activate. In such a case, the Activate is effectively a NOP and has no effect on the channel controller, buffer or I/O device.

Buffer Information

If bit 9 is reset and bits 18-23 contain 00_8, the interpretation of bits 12-17 is:

Bit	Meaning
12	Terminate Output 1=Terminate Output 0=No effect
13	Arm Buffer Interrupt 1=Arm 0=No effect
14	Disarm Buffer Interrupt 1=Disarm 0=No effect
15	Read to Scan 1=Convert Read to Scan 0=No effect
16	Not Used
17	Buffer 1=C Buffer 0=B Buffer

The instruction word format is:



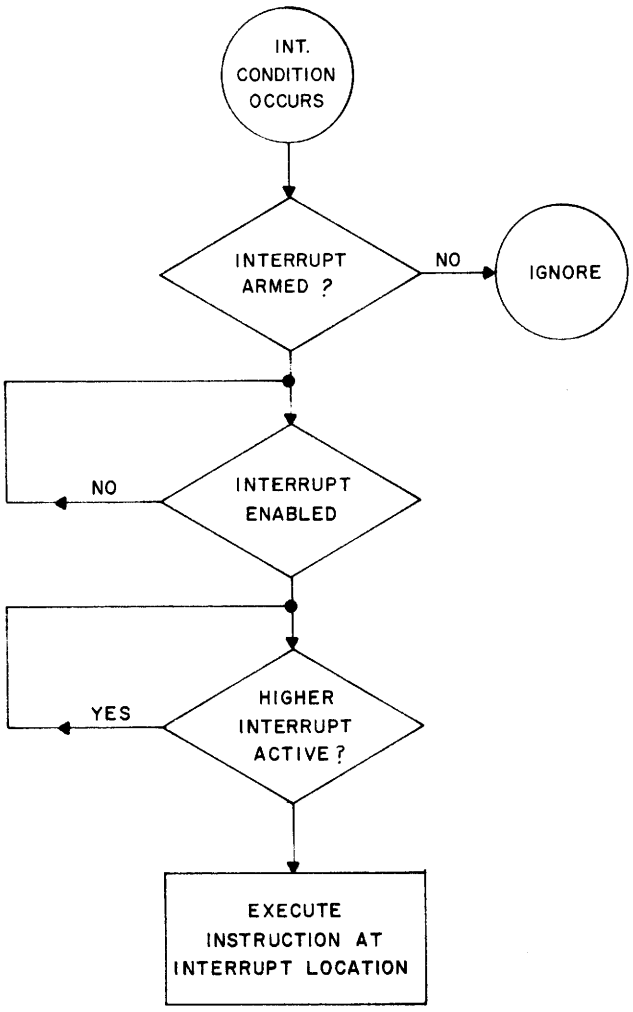
Bit 12 Terminate Output

If bit 12 is set and the buffer is ready (not busy), the buffer will disconnect the I/O device. If bit 12 is set and the buffer is busy (not ready), the buffer will disconnect the I/O device when the buffer becomes ready (not busy). If the buffer is disassembling a word into characters and outputting the characters to a device, then the buffer is busy and the device will not be disconnected until the buffer has completed outputting the word to the device. It is important to note the difference between this Activate which disconnects upon termination of output of a word and the buffer control disconnect Activate which disconnects upon output of the character presently being outputted. A buffer control Activate disconnecting the output device should not be used if the buffer is busy outputting since output terminates with the character being outputted whereas a device control Activate will disconnect the output device only after the word is outputted. For this reason, it is recommended that the device control Activate be used for terminating output operation instead of the buffer control Activate.

Bits 13-14 Buffer Interrupts

If bit 13 is set, the interrupt assigned to the buffer specified by bit 17 is armed. If bit 14 is set, the interrupt assigned to the buffer specified by bit 17 is disarmed. The following flow chart illustrates the sequence of events from occurrence of an interrupt condition to the corresponding interrupt.

INTERRUPT SEQUENCE



Bit 15 Convert Read to Scan

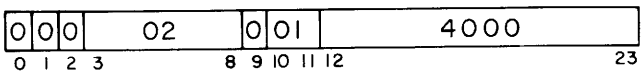
If Bit 15 is set and a magnetic tape read operation is in progress on the tape unit specified by bits 18-23, the read operation is immediately converted to a scan operation.

Assembler Defined Device Control Activates

The assembler defined device control Activates which supply buffer information are:

Mnemonic	Octal Code	Name
TOP	0 02 1 4000	Terminate Output, B Buffer
TOPC	0 02 1 4100	Terminate Output, C Buffer
ARMB	0 02 1 2000	Arm B Buffer Interrupt
DRMB	0 02 1 1000	Disarm B Buffer Interrupt
ARMC	0 02 1 2100	Arm C Buffer Interrupt
DRMC	0 02 1 1100	Disarm C Buffer Interrupt

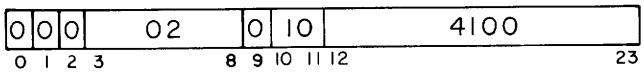
TOP TERMINATE OUTPUT, B BUFFER



Any device connected to the B buffer is disconnected when the buffer transfers the last specified character of the word in the word assembly register to the output device.

1 cycle

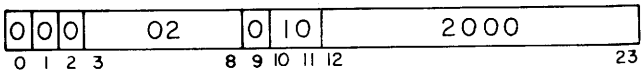
TOPC TERMINATE OUTPUT, C BUFFER



Any device connected to the C Buffer is disconnected when the buffer transfers the last specified character of the word in the word assembly register to the output device.

1 cycle

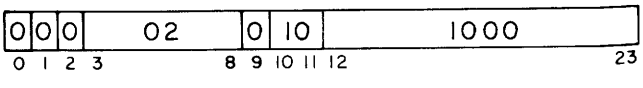
ARMB ARM B BUFFER INTERRUPT



The B buffer interrupt is armed. The B Buffer interrupt location is 32_8.

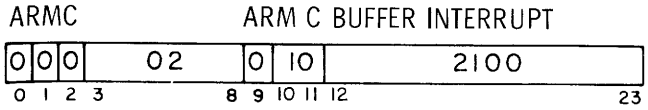
1 cycle

DRMB DISARM B BUFFER INTERRUPT



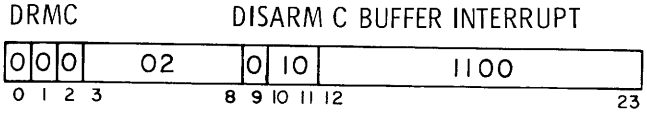
The B buffer interrupt is armed. The B buffer interrupt location is 32_8.

1 cycle



The C buffer interrupt is armed. The C buffer interrupt location is 30_8.

1 cycle



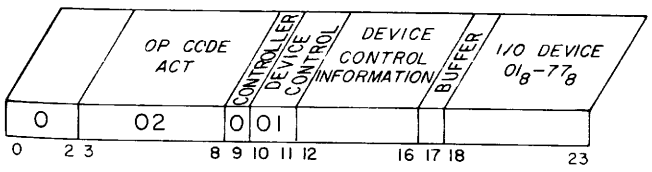
The C buffer interrupt is disarmed. The C buffer interrupt location is 30_8.

Device Control Information

If bit 9 is reset and bits 18-23 contain $01\text{-}77_8$, the interpretation of bits 12-17 is:

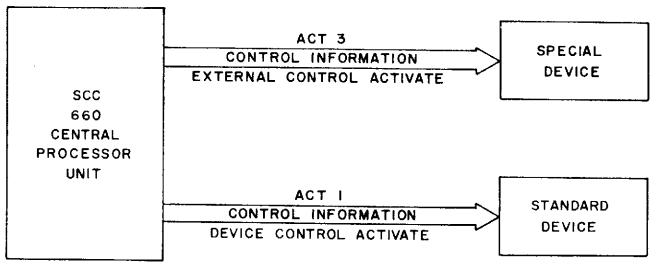
Bit	Meaning
12-16	Device Control Information
17	Buffer 1=C Buffer 0=B Buffer

A detailed description of the interpretation of bits 12-16 for each I/O device is given in the peripheral section of this manual. The instruction word format is illustrated.

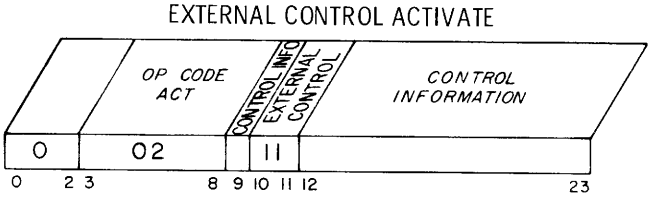


External Control Activate

The standard I/O devices receive control information via the device control Activate. Non-standard I/O devices or special devices receive control information via the external control Activate.



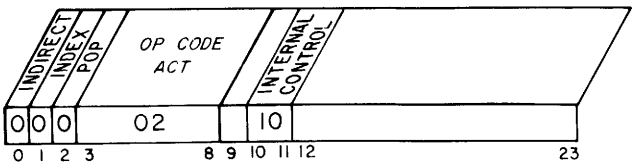
The external control Activate presents the contents of the C register to the special devices. The contents of the C register will be the instruction word if indexing and indirect addressing are not specified. In this case, bits 0-2 contain 0, bits 3-8 contain 02_8 and bits 10-11 contain 11_2. Thus bits 9, 12-23 (a total of 13 bits) are available to contain control information. The instruction word format is:



INTERNAL CONTROL INSTRUCTIONS

The internal control instructions are Activate (ACT) instructions with bits 10-11 of the unindexed effective address containing 10_2 to designate the ACT as an internal control Activate. Usually indexing and indirect addressing are not specified with an internal control Activate instruction in which case bits 9-23 of the instruction word are the unindexed effective address. In order to reflect common usage and simplify the explanations, all examples and descriptions are given without indexing and indirect addressing. The contents of bits 3-8, 10-11 for an internal control Activate is illustrated.

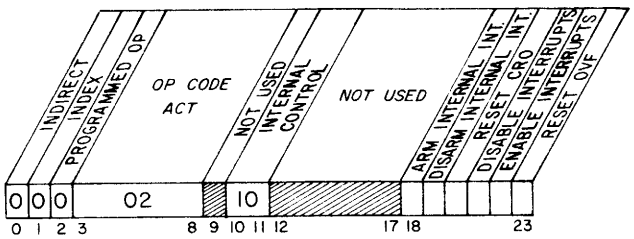
INTERNAL CONTROL ACTIVATE



The interpretation of bits 9, 12-23 is:

Bit	Meaning
9	Not used
12-17	Not used
18	Arm Internal Interrupt 1=Arm 0=No effect
19	Disarm Internal Interrupt 1=Disarm 0=No effect
20	Carry-out Indicator 1=Reset 0=No effect
21	Disable Interrupts 1=Disable 0=No effect
22	Enable Interrupts 1=Enable 0=No effect
23	Overflow Indicator 1=Reset 0=No effect

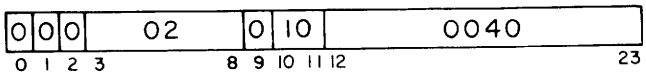
INTERNAL CONTROL ACTIVATE



The assembler defined internal control instructions are:

Mnemonic	Octal Code	Name
ARMI	02 2 0040	Arm Internal Interrupt
DRMI	02 2 0020	Disarm Internal Interrupt
ENA	02 2 0002	Enable Interrupts
DIS	02 2 0004	Disable Interrupts
TOV	02 2 0001	Turn Off (Reset) Overflow Indicator
TOC	02 2 0010	Turn Off (Reset) Carry Out Indicator

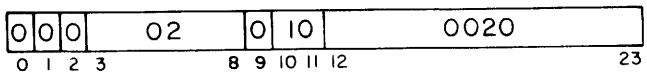
ARMI ARM INTERNAL INTERRUPT



The internal interrupt is armed. The internal interrupt is assigned to location 27_8.

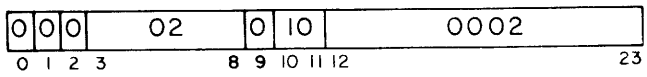
1 cycle

DRMI DISARM INTERNAL INTERRUPT



The internal interrupt is disarmed. The internal interrupt is assigned to location 27_8.

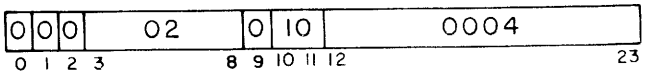
ENA ENABLE INTERRUPT SYSTEM



The interrupt system is enabled.

1 cycle

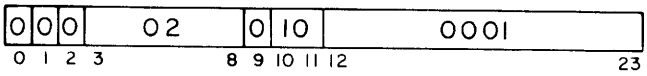
DIS DISABLE INTERRUPT SYSTEM



If the console Enable/Disable switch is in the Disable position, the interrupt system is disabled.

1 cycle

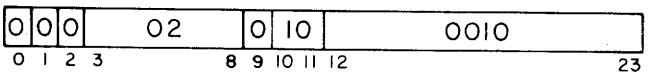
TOV TURN OFF OVERFLOW INDICATOR



The overflow indicator is reset.

1 cycle

TOC TURN OFF CARRY OUT INDICATOR



The carry out indicator is reset.

1 cycle

SECTION III

INTERRUPT SYSTEM

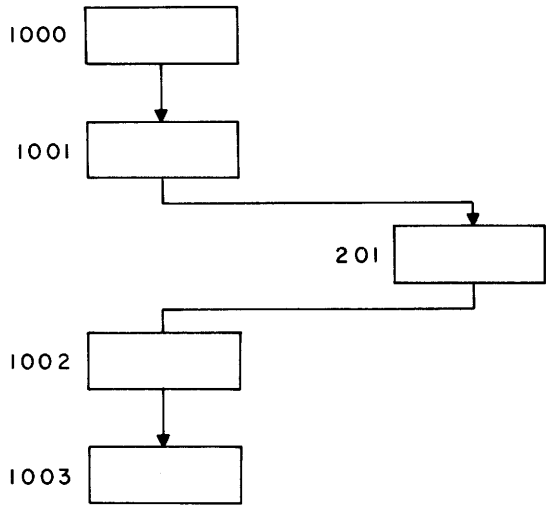
GENERAL

The SCC 660 interrupt capability allows the normal execution of a program to be interrupted in order to execute a program of higher priority. Conditions which may interrupt program execution so that a program of higher priority may be executed are referred to as interrupt conditions. After receiving notification of an interrupt condition, the SCC 660 interrupt system instructs the central processor to perform a program of higher priority. The central processor honors the request for an "interrupt" at the first available time and the interrupt system supplies a memory location to the central processor. The central processor fetches and executes the instruction stored in the memory location supplied by the interrupt system. If the instruction is not a BSL (Branch and Store Location Counter), the interrupt program consists of this single instruction. The central processor notifies the interrupt system of completion of the interrupt program and continues execution of the interrupted program. If the instruction is a BSL (Branch and Store Location Counter), the interrupt program is a multi-instruction program. Execution of the BSL transfers control to the interrupt program. The interrupt program maintains control until execution of a BRI (Branch and Clear Interrupt) instruction. The Branch and Clear Interrupt (BRI) instruction returns control to the interrupted program and the central processor notifies the interrupt system of completion of the interrupt program.

Single Instruction Interrupt Program

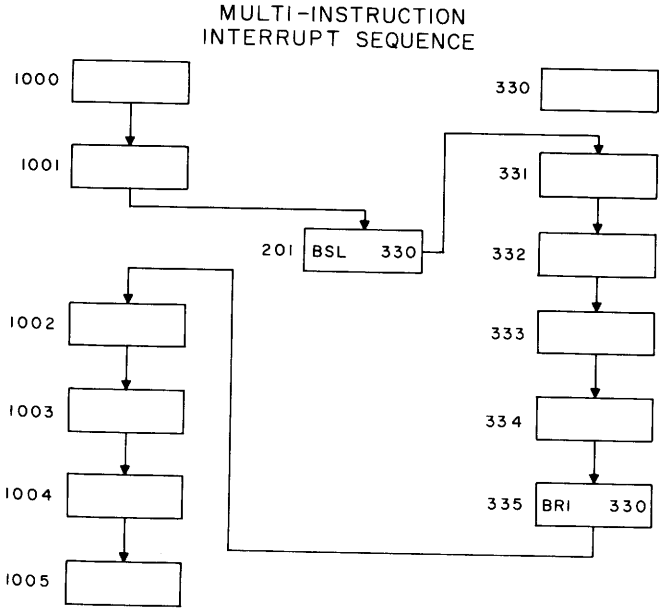
If the instruction in the interrupt location is not a "Branch and Store Location Counter" (BSL) instruction, the interrupt program is the single instruction stored in the interrupt location. If the instruction in the interrupt location does not alter the location counter, the instruction is executed and control is returned to the interrupted program at the location of the instruction which would have been executed had the interrupt not occurred. In the following diagram, the central processor receives a request from the interrupt system to process an interrupt. The interrupt request is received during execution of the instruction in location 1001 and the interrupt location supplied by the interrupt system is 201. The instruction in location 1001 is completed, the instruction in location 201 is executed, and control returns to location 1002.

SINGLE INSTRUCTION
INTERRUPT SEQUENCE



Multi-Instruction Interrupt Program

If the instruction in the interrupt location is a "Branch and Store Location Counter" (BSL) instruction, control is transferred to the location following the location designated by the BSL. Execution of the BSL transfers control from the interrupted program to an interrupt program which relinquishes control to the interrupted program and notifies the interrupt system of completion by executing a "Branch and Clear Interrupt" (BRI) instruction. The central processor returns control to the location of the instruction which would have been next had the interrupt not occurred and the interrupt system clears the interrupt's active indicator. In the following diagram, the central processor receives a request from the interrupt system to process an interrupt. The interrupt request is received during execution of the instruction in location 1001 and the interrupt location supplied by the interrupt system is 201. The instruction in location 1001 is executed, the BSL instruction in location 201 is executed, instruction execution continues at location 331, and the BRI instruction in location 335 returns control to the location specified by bits 9 thru 23 of location 330. Since execution of the BSL instruction placed 1002 in bits 9 thru 23 of location 330, control is returned to location 1002 which is the instruction which would have been executed had the interrupt not occurred.



Interrupt Locations

The memory address supplied by the interrupt system to the central processor is referred to as an "interrupt location". The interrupt locations for the internal interrupt, the B buffer interrupt, and the C buffer interrupt are 27, 32, and 30 respectively. The number of additional interrupt locations is not restricted and the assignment of memory locations for additional interrupts is arbitrary.

Privileged Instructions

An interrupt request from the interrupt system to the central processor is honored following completion of the instruction under execution unless the instruction is a privileged instruction. If the instruction is a privileged instruction, the interrupt request is not granted until completion of the instruction following the privileged instruction. The privileged instructions are Branch and Store Location Counter (BSL), Branch and Clear Interrupt (BRI), Activate (ACT), and Execute (XEC).

Priority

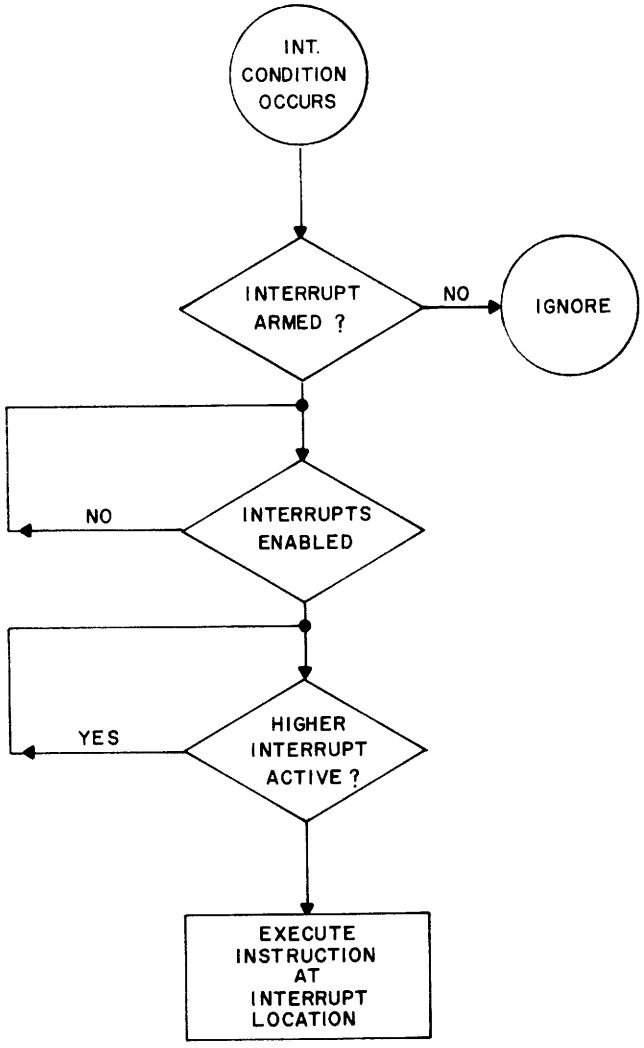
The priority structure of the SCC 660 interrupt system resolves contention problems arising from the simultaneous occurrence of interrupt conditions and permits an interrupt program to be interrupted by a request to service an interrupt of higher priority. The priority assignment for the internal interrupt, C buffer interrupt and B buffer interrupt in descending priority order is internal, C buffer and B buffer. The priority assignment for additional interrupts is arbitrary; however, a common scheme is to assign the higher priority interrupt to the lower-numbered interrupt location. For example, if memory locations 200 thru 237 are interrupt locations, the interrupt assigned to location 200 is of a higher priority than the interrupt assigned to location 201 which is of higher priority than the interrupt assigned to location 202 and so forth with the interrupt assigned to location 237 having the lowest priority.

ARMING/ENABLING

An interrupt is disarmed if occurrence of the corresponding interrupt condition is ignored by the interrupt system. If the interrupt is armed, the interrupt system recognizes the occurrence of the interrupt condition. At the completion of the processing of any interrupts of higher priority, the interrupt system issues an interrupt request to the central processor if the interrupt system is enabled. If the interrupt system is disabled at the present time, the interrupt system delays issuance of the interrupt request until the interrupt system is enabled. Arming and disarming of an interrupt condition permits the interrupt system to ignore or honor the occurrence of an interrupt condition, whereas enabling and disabling of the interrupt system allows interrupt processing to occur promptly or to be delayed.

The internal, C buffer and B buffer interrupts may be individually armed or disarmed. The arming and disarming of additional interrupts is an optional feature which permits interrupts to be selectively armed or disarmed. If the selective arming/disarming option is not present, the additional interrupts are armed if the interrupt system is enabled and disarmed if the interrupt system is disabled.

INTERRUPT SEQUENCE



Console Enable Switch

The system console has an Enable Interrupts switch and an indicator which is illuminated if the interrupt system is enabled. Depressing the START button on the console sets the state of the interrupt system to the state of the console Enable switch. If the interrupt system is disabled, the interrupt system may be enabled at any time by either an enable Activate or by the Enable switch. If the interrupt system is enabled, the interrupt system may be disabled by the disable Activate only if the console switch is not in the Enable position. After execution of an enable Activate, the interrupt system cannot be disabled by the Enable switch.

Internal Interrupt

An attempt to store information into protected memory when the Memory Protect Switch on the console is in the ON position results in an internal interrupt condition.

Buffer Interrupt

The B buffer interrupt and the C buffer interrupt function in an identical manner and are the result of either an end-of-word condition or end-of-record condition. During input operations which do not involve a channel controller, the buffer interrupt indicates that the buffer's word assembly register is ready to read into memory or that the physical end-of-record has been encountered. A buffer ready test may be used to determine which condition generated the buffer interrupt. If an end-of-record condition generated the interrupt, the buffer automatically disconnects the input device and becomes ready. Thus, an execution of a buffer ready test instruction will result in a skip. If the buffer interrupt was not due to end-of-record condition, the input device remains connected to the buffer and the buffer is busy (not ready). Thus, execution of a buffer ready test instruction will not result in a skip. During output operations which do not involve a channel controller, the buffer interrupt indicates that the buffer's word assembly register is ready to receive a word from memory. During input operations involving a channel controller, the buffer interrupt indicates that the controller's word count register has decremented to zero or that the physical end-of-record has been encountered. During output operations involving a channel controller, the buffer interrupt indicates that the channel controller's word count register has decremented to zero. Examples of interrupt routines associated with input/output programs are given in the description of paper tape input/output.

SECTION IV

PERIPHERALS

TYPEWRITER

Typewriter Instructions

RTY Read Typewriter 0 02 01601

Typewriter 1 is connected to the B buffer. The buffer is initialized to assemble four characters per word.

WTY Write Typewriter 0 02 01641

Typewriter 1 is connected to the B buffer. The buffer is initialized to output four characters per word.

Execution of a RTY or WTY connects the typewriter to a buffer. Execution of an Activate instruction which does not specify the typewriter disconnects the typewriter. Input operations which do not involve a channel controller must be terminated with a Disconnect (DSC) instruction. Output operations which do not involve a channel controller must be terminated with a Terminate Output (TOP) instruction. If typewriter input/output is accomplished using a channel controller, the channel controller automatically disconnects the typewriter when the word count register has been decremented to zero. The typewriter does not generate error signals, but if an input or output parity error or character rate error is detected by the buffer, the buffer error indicator is set and the ERROR indicator on the control panel is turned on. If the typewriter is connected to the buffer, the buffer ready test will not skip.

EXAMPLE: Typewriter Output

This routine types out the message:

A CHARACTER IS SOMEONE INTERESTED IN COMPUTERS.

Location	Instruction	Address	Comments
TYPOUT	PZE	**	Reserve entry location
	WTY		Connect Typewriter to B buffer for output with four characters per word.
	LDX	COUNT	Load XR with negative of number of words in message
	TMB	MESAG+12,X	Output a word
	BIX	*−1	If XR is negative, the XR is incremented by one and a branch to the TMB instruction occurs. If XR is zero, the next instruction is executed.
	TOP		Disconnect the typewriter after the buffer completes transfer of the word in the buffer to the typewriter.
	BRD		When the device is disconnected, the buffer will report ready and a skip will occur.
	BRA	*−1	The buffer is busy. Branch to test the buffer again.
	BRT	TYPOUT	Return to the main program.
**CONSTANTS			
MESAG	BCI	3,A CHARACTER	
	BCI	5,IS SOMEONE INTERESTE	
	BCI	4,D IN COMPUTERS.	
COUNT	DEC	−12	

EXAMPLE: Typewriter Input

This routine inputs characters from the typewriter until a carriage return is typed by the operator.

Location	Instruction	Address	Comments
TYPIN	PZE	**	Reserve entry location
	LDX	XZERO	Initialize XR_{9-23} to zero. Set XR_8 to provide for incrementing XR.
	LDA	CR	Initialize AC to code for carriage return.
	LDE	MASK	Initialize EA with mask which will result in SAE instruction comparing bits 18—23 of AC with memory
	RTY	0,1	Connect typewriter to B buffer for input with one character per word.
READ	TBM	INAREA,X	The computer "hangs up" on this instruction until a character enters the B buffer from the typewriter; then the word in the buffer is placed in the effective address (INAREA+XR). The input character is in bits 18-23 of the effective location. Bits 0-17 contain zero.
	SAE	INAREA,X	Bits 18-23 of AC are compared to bits 18-23 of the effective location. A skip occurs if equal.
	BIX	READ	Character is not a carriage return, go input another character.
	DSC		Character is a carriage return. Disconnect the typewriter from the buffer.
	BRT	TYPIN	Return to main program.
**CONSTANTS,ETC.			
XZERO	OCT	00100000	
CR	OCT	52	
MASK	OCT	00000077	
INAREA	BSS	100	

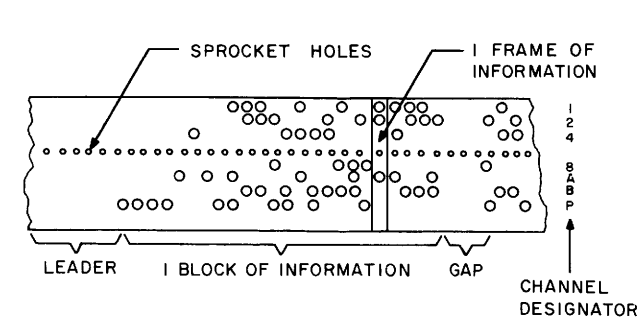
EXAMPLE: Typewriter Output under Interrupt Control

Location	Instruction	Address	Comments
** MAIN PROGRAM			
*** INITIALIZATION FOR TYPEWRITER			
*** OUTPUT UNDER INTERRUPT CONTROL			
**** INITIALIZE INTERRUPT LOCATION			
	LDA	WORD	Load the AC with the BSL instruction.
	STA	'32	Place the BSL instruction in the B buffer interrupt location
	LDA	NEGTEN	Load the AC with -10
	STA	CNTR	Store the AC in CNTR
	LDA	TEN	Load the AC with word count
	ADD	STARTR	Add the address of first word
	SAF	WRITE	Store AC_{9-23} in address field of TMB instruction
	ENA		Enable interrupt system
	ARMB		Arm B buffer interrupt
	WTY		Connect typewriter to B buffer for output with four char/word
*** MORE CODE			
	o		
	o		
	o		
*** INTERRUPT ROUTINE			
TYPOUT	PZE	**	Reserve entry location
	XXM	CNTR	Exchange contents of XR and CNTR. This saves contents of XR at the time of interrupt and puts a count into XR so TMB instruction will reference next location in output area.
WRITE	TMB	**,X	Output a word
	BIX	RESTOR	Increment XR. If XR=0, last word has been sent to buffer.

Location	Instruction	Address	Comments
	TOP		Disconnect typewriter when last character in buffer is sent to typewriter.
RESTOR	XXM	CNTR	Exchange contents of XR and CNTR. This restores XR to original condition and puts count back into CNTR
	BRI	TYPOUT	Return to main program
** CONSTANTS, ETC.			
STARTER	PZE	MESAG	
MESAG	BCI	3,A CHARACTER	
	BCI	5,READER IS A FORTUNE	
	BCI	2,TELLER.	
WORD	BSL	TYPOUT	
CNTR	BSS	1	
NEGTEN	DEC	-10	
TEN	DEC	10	

PAPER TAPE

The paper tape uses six hole positions for information and one for parity check in each frame. The paper tape is one inch wide, with ten frames of information per inch in the direction of travel. Information is organized on the tape in blocks. A block is any number of information frames set off by a gap (in which only the sprocket hole is punched) at either end. Gap in front of the first block of tape is called "leader".



The paper tape reader is primarily used for loading programs and/or data into memory. The reader is always ready for operation and no ready test is required. Before executing the ACT instruction to read a tape, the tape must be loaded into the reader. The loading procedure is:

1. Place the tape actuator in the LOAD position.
2. Insert the tape (from right to left) into the tape guide, with channel P toward the operator. (If a spool of tape is used, mount the spool on the spooler and thread the tape into the takeup spool.)
3. Place the tape actuator in the RUN position.

The paper tape punch is used primarily for punching programs and/or data to be loaded back into memory later. The punch is always ready to be loaded for operation and no ready test is required. Before executing the ACT to punch a tape, the operator should determine if there is enough tape on the supply reel for the punching operation and that the tape is properly threaded. For extensive punching operations, the tape should be threaded onto a takeup reel. After each roll of tape has been punched, the operator should empty the chad box and brush all loose chad from the tape guide. Otherwise, the punch may jam during a punching operation.

If the toggle switch on the punch panel is placed in the ON position, the punch motor runs continuously. If the switch is in the OFF position, the punch motor is turned on only when the punch is addressed by the buffer (with an automatic delay to allow the motor to reach punching speed) or when the FEED button on the punch panel is pressed. Tape leader may be punched manually by depressing the FEED button until the desired amount of leader is produced.

PAPER TAPE INSTRUCTIONS

RPT Read Paper Tape 0 02 01604

The paper tape reader is connected to the B buffer. The buffer is initialized to assemble four characters per word with odd parity. Leader is ignored and the first four characters are transferred to the buffer. A TBM instruction transfers the buffer's contents to memory and reads the next four characters into the buffer. If gap (blank tape) is encountered, the paper tape reader is disconnected from the buffer. When the reader disconnects, the end-of-record interrupt occurs if the interrupt system is enabled. Input operations which do not involve a channel controller are terminated by the Disconnect (DSC) instruction or by the reader encountering blank tape. If paper tape input is accomplished using a channel controller, the channel controller automatically disconnects the paper tape reader when the word count register has been decremented to zero. The reader will also disconnect if gap is encountered. The reader does not generate error signals, but if a parity error or character rate error is detected by the buffer, the buffer error indicator is set and the ERROR indicator on the control panel is turned on. If the reader is connected to the buffer, the buffer ready test (BRD) will not skip.

WPT Write Paper Tape 0 02 01644

The paper tape punch is connected to the B buffer. The punch motor is turned on (if not already on) and the buffer is initialized to output four characters/word without leader. Odd parity is specified.

Output operations which do not involve a channel controller must be terminated with a Terminate Output (TOP) instruction. If paper tape output is accomplished using a channel controller, the channel controller automatically disconnects the punch when the word count register has been decremented to zero. If a parity error occurs during a paper tape punch operation, the buffer error indicator is set and the ERROR indicator on the control panel is turned on. If the punch is connected to the buffer, the buffer ready test (BRD) will not skip.

EXAMPLE: Read a block of tape until a gap is encountered using the B buffer without a channel controller or interrupt.

This routine reads data into memory, at 4 characters/word, starting at location 02000.

Location	Instruction	Address	Comments
READ	PZE	**	Reserve the entry location
	LDX	LOC	Load XR_{9-23} with starting location. Set XR_8 to provide for incrementing XR
	RPT		Connect the reader to the B buffer for input with 4 char/word.
	TBM	0,X	Put the contents of the buffer in the location specified by XR.
	BRD		Test buffer for ready. If the reader has encountered a blank frame, the reader is disconnected and is ready. Thus, a skip occurs if gap is encountered. Otherwise, the reader is still connected and a skip does not occur.
	BIX	*−2	The XR is incremented by 1 and a branch to the TBM instruction occurs.

Location	Instruction	Address	Comments
	BRT	READ	Return to the main program. The address of the last word stored in memory is contained in XR_{9-23}.
**CONSTANTS			
LOC	OCT	102000	

If the number of characters on the tape is an integral mutliple of 4, the last word stored in memory is all zeros. If the number of characters is not an integral multiple of 4, the last word stored in memory contains zeros in the "missing" character positions. Thus, the number of words read into memory is one more than the number of 4 character words on the tape.

EXAMPLE: Read Paper Tape of Known Length

This routine reads a 64-character block from paper tape into memory beginning at location 02000. The routine uses the 4 char/word mode, making the input 16 words. The routine uses the channel controller and interrupts.

Location	Instruction	Address	Comments
RDPT	PZE	**	Reserve the entry location
	CLR		Reset AC and EA to zero. Store AC in SWITCH. These two instructions clear an "input-complete" indicator.
	STA	SWITCH	
	ENA		Enable the interrupt system
	ARMB		Arm the B buffer interrupt
	ACT	'40000	Enable channel controller for the B buffer
	WTP	CONTRL	The word count (20) and the start address (020000) is sent to the channel controller.
	RPT		The reader is connected to the B buffer for input with 4 char/word. The input operation begins.
	BRT	RDPT	Return to the main program.
**CONSTANTS			
CONTRL	OCT	02002000	
When the buffer has transferred 16 words to memory, the B buffer end-of-record interrupt occurs. The instruction in location 32 is executed.			
032	MIN	SWITCH	The "operation finished" indicator is set and the interrupt's active indicator is cleared. Control returns to the interrupted program.

EXAMPLE: Read a block of tape until a gap is encountered using the B buffer with interrupts and without a channel controller.

Location	Instruction	Address	Comments
**MAIN PROGRAM			
*** INITIALIZATION FOR PAPER TAPE INPUT			
*** UNDER INTERRUPT CONTROL.			
	EAX	INAREA	Put starting location of input area into XR.
	STX	CURNT	Store (XR) in CURNT.
	ENA		Enable interrupt system
	ARMB		Arm B buffer interrupt
	RPT		Connect the reader to the B buffer for input with 4 char/word
	o		
	o		
	o		
**INTERRUPT ROUTINE			
PTIN	PZE	**	Reserve Entry Location
	BRD		Test B buffer for ready. If ready, skip.
	BRA	READIT	Reader is connected, not end-of-record.
	TBM	LAST	Reader is disconnected, end-of-record (gap) encountered. Input buffers contents. If the number of characters in record was multiple of four, this word is extra word of all zeros. If no. of char not a multiple of 4, this word contains last character of record and zeros in unused portion of word.
	BRI	PTIN	Return to main program, clear interrupt 32 active indicator.
READIT	XXM	CURNT	Put (CURNT) which is next available input location into XR. Put (XR) into CURNT.
	TBM	0,X	Input a word into location specified by XR.
	BIX	*+1	Increment (XR) by 1.
	XXM	CURNT	Put (XR) into CURNT. Put (CURNT) into XR.
	BRI	PTIN	Clear interrupt 32 active indicator. Return to main program.
**CONSTANTS			
032	BSL	PTIN	Location 32 is the B buffer interrupt location.
CURNT	OCT	0	CURNT is used to point to next available input location.

EXAMPLE:

This routine punches a block of 100_{10} words. The 100 words are AREA thru AREA+99. The output is performed by an interrupt service routine.

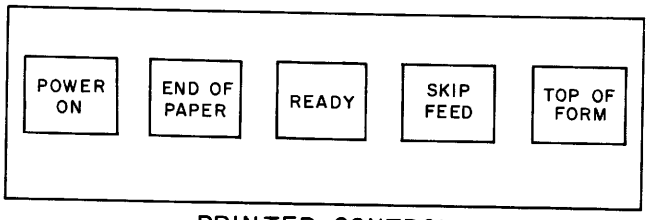
Location	Instruction	Address	Comments
** MAIN PROGRAM			
*** INITIALIZATION FOR PAPER TAPE			
*** OUTPUT UNDER PROGRAM CONTROL			
	LDA	ADDRESS	Fetch start address
	ADD	HUNDRD	Add 100
	SAF	OUT	Store in bits 9–23 of output instruction
	CLR		These two instructions reset an operation-complete switch
	STA	SWITCH	
	LAC	HUNDRD	Word count is initialized to neg. 100
	STA	COUNT	
	ENA		Enable interrupt system
	ARMB		Arm B buffer interrupt
	WPT		Activate paper tape punch
*** CODE CONTINUES			
	o		
	o		
	o		
** INTERRUPT ROUTINE			
PTOUT	PZE	**	Reserve interrupt location
	XXM	COUNT	Put negative of present word count into XR. Put contents of XR in count
OUT	TMB	**,X	Output a word
	BIX	*+2	Increment XR by 1. Test XR for zero. If zero, have outputted last word.
	TOP		XR is zero. Disconnect punch
	MIN	SWITCH	Set operation complete switch
	XXM	COUNT	Put contents of count into XR, thus restoring XR. Put negative of present word count into count.
	BRI	PTOUT	Clear int. 32 active indicator and return to main program
** CONSTANTS, ETC.			
032	BSL	PTOUT	Location 32 is the B buffer interrupt location
ADDRES	PZE	AREA	Area is start address of 100 words area
HUNDRD	DEC	100	
COUNT	OCT	0	Count is used to contain negative of word count

LINE PRINTER

SCC buffered line printers are capable of printing up to 1000 lines per minute at 132 characters per line, with a standard set of 64 characters. The characters are transmitted sequentially to the printer buffer for storage before printing. A program-controlled format tape loop provides fixed (or preselected) line space control. Upspacing of 1 to 7 lines, as well as page control (upspacing to line positions designated on the format loop), may be accomplished.

Printer Controls

The printer controls for SCC line printers consists of five switches and indicators.



PRINTER CONTROL
SWITCHES AND INDICATORS

Power/On

The computer must be turned on for this switch to be activated. Pressing POWER/ON lights the indicator, turns on the motors and hammer driver power supply.

End of Paper

If paper is in the lower set of form tractors, the END OF PAPER indicator is off. Detection of the end of paper in the lower set of form tractors turns on the END OF PAPER indicator.

Ready

When the printer is initially turned on, this indicator is turned on if:

1. paper is loaded in the line printer,
2. the hammer power supply is on.

This indicator automatically goes off when the above conditions are not realized. The printer is ready for operation when READY is turned on. READY is reset to preclude computer intervention while changing paper or ribbon. If the printer is ready, the printer may be placed in the non-ready state by depressing the READY switch.

Skip Feed

Pressing SKIP FEED causes the printer to continually skip feed paper independently of the vertical format tape while the switch is depressed.

Top of Form

Pressing TOP OF FORM causes the printer to position paper according to format channel 1. This indicator is lighted only when the format tape is positioned at channel 1, that is, top-of-form on a standard tape loop.

PAPER TAPE FORMAT LOOP

A paper tape format loop, placed in the printer, allows upspacing to proceed to prespecified vertical positions on the print page. The format loop is an eight-channel paper tape. Putting a punch in the specified channel at the desired vertical spacing selects the channel upspace. Channel 1 is the top of form channel, channel 7 is the bottom of form channel, and channel 0 is the single upspace channel. When printing with no format loop inserted in the printer, single upspacing occurs regardless of the channel specified.

LINE PRINTER INSTRUCTIONS

PRN Print 0 02 01660

The line printer is connected to the B buffer. The buffer is initialized to output four characters per word.

PRDY Printer Ready 0 40 10060

The Printer Ready instruction is a skip instruction. If the line printer is not connected to a buffer and the line printer is ready (ready indicator on), the Printer Ready instruction will skip. If the line printer is connected to a buffer and a printer fault did occur during the previous line, the Printer Ready instruction will skip. This test for a printer fault is not valid after a character is sent to the printer. The printer fault indicator is turned off after receipt of the first character, thus, the fault indicator must be tested after the printer is connected to the buffer and before the first character is sent to the printer.

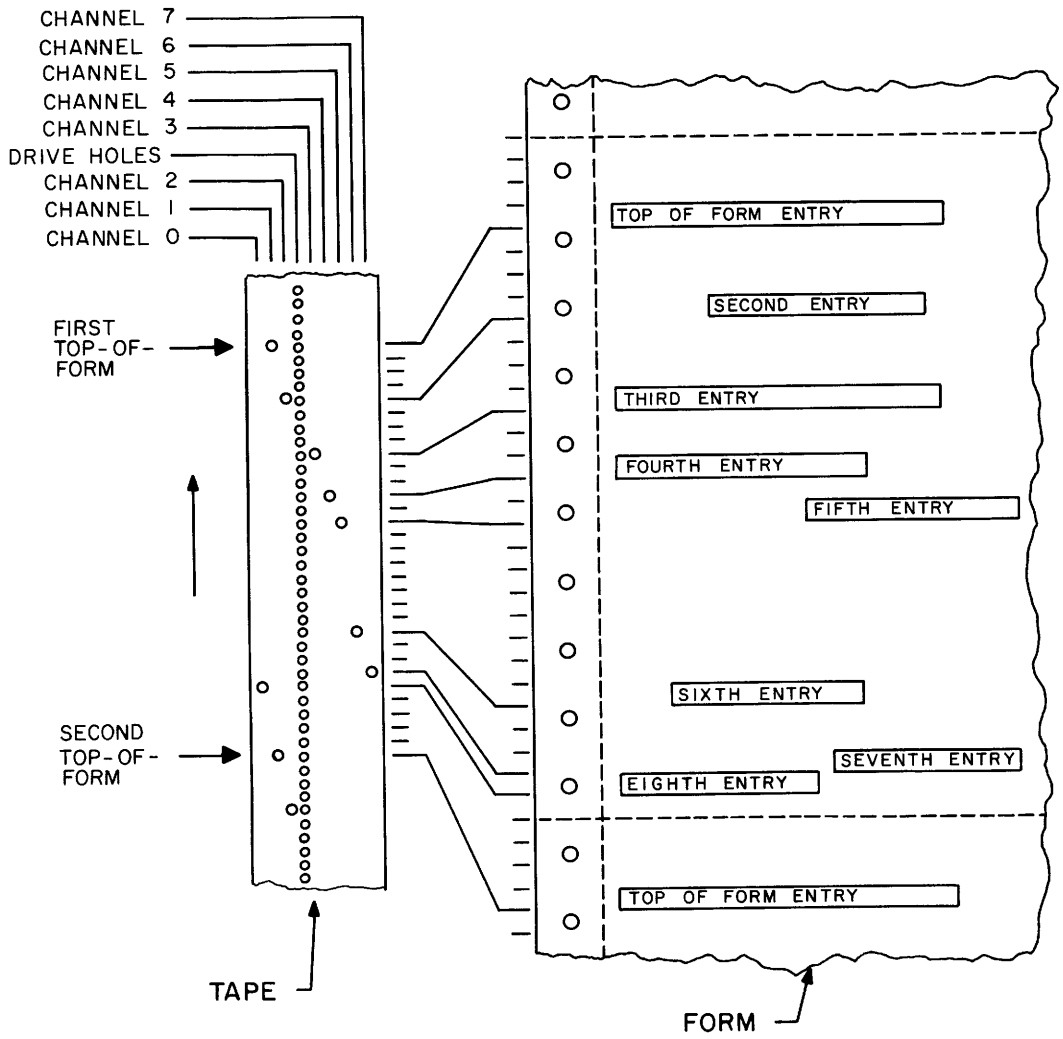
The first character sent to the printer following the PRN instruction is the carriage control character. The carriage control character is followed by 0 thru 132 data characters. Execution of the Terminate Output (TOP) instruction disconnects the printer from the buffer and prints the line. If more than 132 data characters are sent to the printer, the additional characters do not enter the print buffer. Thus, execution of a Terminate Output (TOP) instruction results in printing of the first 132 data characters.

CARRIAGE CONTROL CHARACTER

The carriage control character is the first character sent to the printer following the PRN instruction. The function designated by the carriage control character is performed before the line is printed. The following table specifies the function and corresponding code for carriage control.

CARRIAGE CONTROL

Code	Character	Function
00	0	Skip to Channel 0
01	1	Skip to Channel 1
02	2	Skip to Channel 2
03	3	Skip to Channel 3
04	4	Skip to Channel 4
05	5	Skip to Channel 5
06	6	Skip to Channel 6
07	7	Skip to Channel 7
10	8	Do not upspace
11	9	Upspace 1 line
12	space	Upspace 2 lines
13	=	Upspace 3 lines
14	'	Upspace 4 lines
15	:	Upspace 5 lines
16	>	Upspace 6 lines
17	√	Upspace 7 lines



NOTE:

PAPER LINE SPACING IS SIX LINES PER INCH. EACH HOLE SPACE ON TAPE CORRESPONDS TO ONE PAPER LINE SPACE.

TAPE-LOOP PREPARATION

EXAMPLE: Print a line

This routine upspaces the printer one line and prints the following:

A BIT IS TWELVE AND A HALF CENTS.

Location	Instruction	Address	Comments
PRINT	PZE	**	Reserve the entry location
	BRD		Test buffer for ready
	BRA	*−1	Busy, test again
	PRDY		Test printer for ready
	BRA	*−1	No ready, test again
	PRN		Connect printer to buffer, 4 characters/word
	PRDY		Test for printer fault from previous line
	BRA	PRNERR	Error, go to print error routine
	LDX	NEGTEN	Put negative of word count in XR
	TMB	LINE+10,X	Output a word
	BIX	*−1	Increment XR by 1, go to output a word if XR not zero
	TOP		
	BRD		Test buffer for ready (Disconnect)
	BRA	*−1	Busy, go test again
	BRT	PRINT	Return to caller
** CONSTANTS			
NEGTEN	DEC	−10	
LINE	BCI	1, 9	Carriage control and 3 blanks
	BCI	4,A BIT IS TWELVE	
	BCI	5, AND A HALF CENTS	

EXAMPLE: Print routine

This routine checks printer for ready. If not ready after two seconds, a message is typed on the typewriter. The routine continues after printer is made ready. The routine prints 54 lines per page (single spaced) with 6 blank lines at the top and bottom of each page. The routine requires one parameter – the start address of the line to be printed. The call sequence is two lines of code. The first is a BSL to PRINT. The second is a PZE specifying the address of the line to be printed in the address field. The end of a line is denoted by a word containing all 1's (77777777_8). The first time PRINT is called the paper is positioned to the top of form by a skip to channel 1. The routine saves and restores the registers.

```
** MAIN PROGRAM
*** PRINT LINE 1

START       BSL     PRINT
            PZE     LINE 1
            BSL     PRINT
            PZE     LINE 2
            LDX     =-500
            BSL     PRINT
            PZE     LINE 3
            BIX     *-2
            HLT

** PRINT SUBROUTINE

PRINT       PZE     **          Reserve entry location

*** SAVE REGISTERS

            STA     AC
            STE     EA
            STX     XR

*** UPDATE PRINT TO POINT TO LOCATION OF BSL+1

            MIN     PRINT

*** FETCH PARAMETER (START ADDR. OF LINE)

            EAX*    PRINT

*** STORE PARAMETER

            STX     PARAM
```

Location	Instruction	Address	Comments
*** WAIT TILL BUFFER READY			
BUFRDY	BRD		
	BRA	*−1	
*** WAIT TILL PRINTER READY			
	LDX	=−'100000	
	PRDY		
	BRA	*+2	
	BRA	READY	
	RCY	48	This instruction requires 14 cycles
	LCY	48	This instruction requires 14 cycles
	BIX	*−5	If XR_8 is zero, two seconds elapsed.
	WTY		Connect typewriter for
	LDX	=−10	output message to
	TMB	HELPME+10,X	operator. Send
	BIX	*−1	message, go check
	TOP		buffer ready and printer
	BRA	BUFRDY	ready again.
*** POSITION TO TOP OF FORM IF 1ST CALL			
READY	SMN	FIRST	FIRST is assembled containing
	BRA	IS1ST	zero. So if still zero, is 1st call.
	BRA	NOT1ST	
IS1ST	PRN		Connect printer and send carriage
	TMB	TOF	control and 3 data characters
	MDC	FIRST	Set switch to indicate that initial forms positioning is accomplished.
	TOP		Disconnect when buffer
	BRA	BUFRDY	empty, go to buffer test.

Location	Instruction	Address	Comments
NOT1ST	PRN		
*** OUTPUT MESSAGE TO OPERATOR IF ERROR OCCURRED *** DURING PREVIOUS LINE			
	PRDY		Test for error, go
	BRA	PRNERR	output message if error
	BRA	TOFCHK	If good, go to TOF check.
PRNERR	WTY		Connect typewriter.
	LDX	=–9	Tell operator of error
	TMB	ERRMSG+9,X	
	BIX	*–1	
	TOP		Disconnect typewriter,
	BRD		Wait for buffer to
	BRA	*–1	empty
	PRN		Reconnect printer
*** POSITION TO TOP OF FORM IF 54 LINES PRINTED			
TOFCHK	LDA	LINNUM	Get line count
	LDE	ONES	Put mask into EA
	SAE	=54	Test for 54
	BRA	UPSPAC	
	TMB	TOF	If 54, send top of form
	TOP		Disconnect printer when buffer empties.
	CLR		Reset line counter
	STA	LINNUM	to zero.
	BRA	BUFRDY	Go to buffer ready test
UPSPAC	TMB	SPACE	
*** OUTPUT LINE TO PRINTER			
	LDA	PARAM	Get start address
	ORA	='00100000	Set bit 8 and put

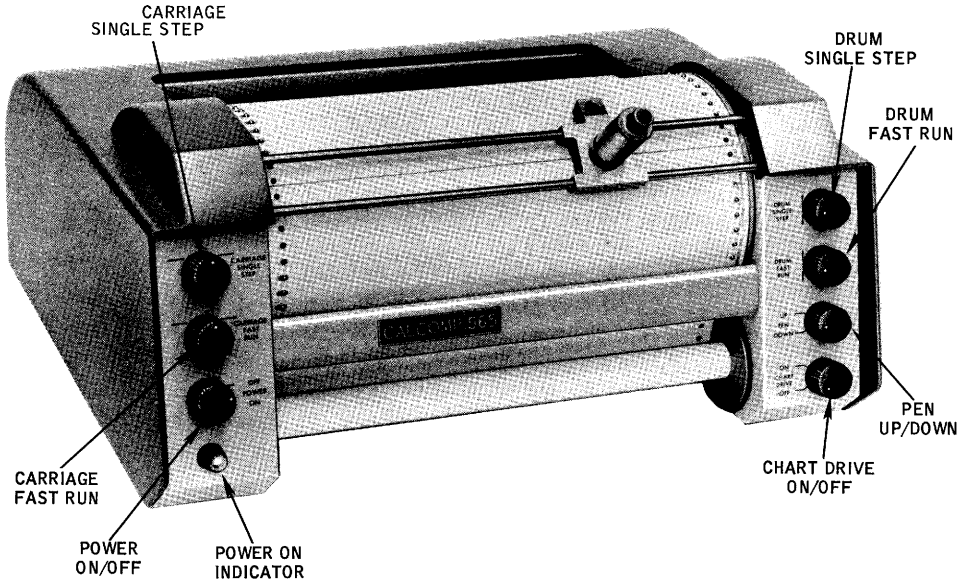
Location	Instruction	Address	Comments
	STA	PARAM	into PARAM
	LDX	PARAM	and XR
	LDE	ONES	Put mask into EA
	LDA	0,X	Get word
	SAE	='77777777	Test for ender
	BRA	OUTPUT	Go to output word.
	BRA	PUTOUT	Go to print line
OUTPUT	TMB	0,X	Output word,
	BIX	*–5	increment XR and go again
*** PRINT THAT LINE			
PUTOUT	TOP		The printer will disconnect after buffer empties and the the line will be printed.
*** UPDATE LINE COUNTER			
	MIN	LINNUM	
*** RESTORE REGISTERS			
	LDA	AC	
	LDE	EA	
	LDX	XR	
*** RETURN TO CALLING PROGRAM			
	BRT	PRINT	
** CONSTANTS, ETC.			
AC	OCT	0	
EA	OCT	0	
XR	OCT	0	
PARAM	OCT	0	
FIRST	OCT	0	
ONES	OCT	77777777	
HELPME	OCT	52525252	Carriage returns

Location	Instruction	Address	Comments
	BCI	4,ATTENTION	
	BCI	5,MAKE PRINTER READY.	
TOF	OCT	01121212	The first character is a skip to channel 1. The other three are spaces.
ERRMSG	OCT	52525252	
	BCI	4,ATTENTION.	
	BCI	4,PRINTER ERROR.	
SPACE	OCT	11121212	The first character is an upspace one line. The other three are not used.
LINNUM	OCT	0	
LINE1	BCI	3,I HAVE 132 P	
	BCI	6,RINT POSITIONS AND 64 CH	
	BCI	3,ARACTERS.	
	OCT	77777777	
LINE2	BCI	4,I CAN PRINT 1000	
	BCI	5, LINES PER MINUTE.	
	OCT	77777777	
LINE3	BCI	4,I AM A SCC LINE	
	BCI	2,PRINTER .	
	OCT	77777777	
	END	START	

PLOTTER

The digital incremental plotter provides high speed, high resolution plotting of digital computer outputs. The plotter is a high speed two-axis plotter designed for plotting one variable against another. The actual plot is produced by the movement of a pen over the surface of the chart paper. The Y-axis plot is produced by lateral movement of the pen carriage and the X-axis plot by rotary motion of the chart drum. The pen can be lifted or lowered to the plotting surface.

The plotter employs a bi-directional rotary step motor on both the X and Y-axis drives. Each step causes the drum or pen carriage to move .01 or .005 inch in either a positive or a negative direction. The motors are capable of operating at a rate of up to 300 steps per second. A roll paper feed and takeup mechanism is provided which accepts chart paper rolls 12 inches wide by 120 feet long. The feed and takeup mechanism is bi-directional. The paper is driven by sprocket teeth on the drum which engage the sprocket holes on both edges of the paper, thus maintaining accuract registration between the recording pen and the paper.



Plotter Instructions

PLOT	Activate Plotter	0 02 01662

The incremental plotter is connected to the B buffer. The buffer is initialized for four characters per word.

PLRDY	Plotter Ready	0 40 10062

The Plotter Ready instruction is a skip instruction. If the plotter is ready, the Plotter Ready instruction will skip.

The plotter is connected to a buffer with a PLOT instruction. The plotter receives characters and decodes these characters as plotter commands. Plotter operations must be terminated with a Terminate Output (TOP) instruction. The plotter does not generate error signals, but if a parity error or character rate error is detected by the buffer, the buffer error indicator is set and the ERROR indicator on the control panel is turned on. If the plotter is connected to a buffer, the buffer ready test will not skip. The plotter commands and execution times are listed in the following table.

PLOTTER COMMANDS

Bit	Function	Execution Time
0	1 increment in +X direction	3.3 millisec
1	1 increment in -X direction	3.3 millisec
2	1 increment in +Y direction	3.3 millisec
3	1 increment in -Y direction	3.3 millisec
4	Pin Up	100 millisec
5	Pin Down	100 millisec

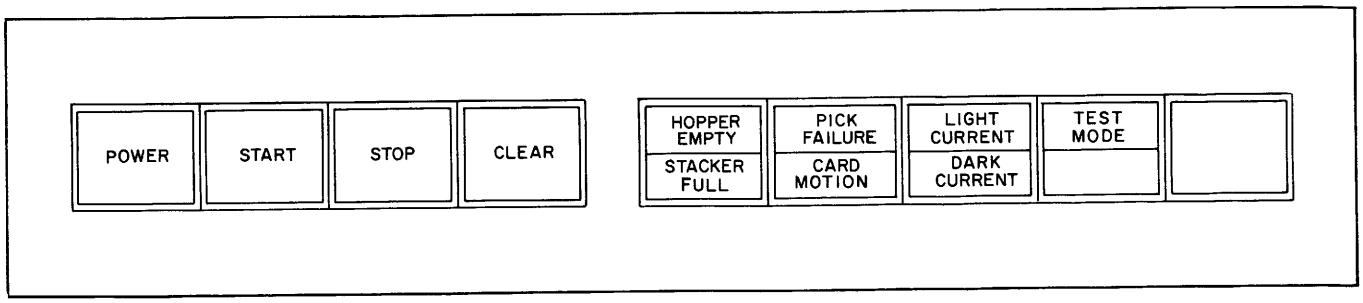
CARD READER

The SCC card reader reads standard 80 column punched cards at a maximum rate of 1100 cards per minute in column-binary format. Binary format consists of two 6-bit characters per column. The top 6 rows (rows 12-3) of column 1 form the first

character (with the most significant bit in row 12), the bottom 6 rows (rows 4-9) form the next character (with the most significant bit in row 4). When reading in the four character per word mode, the first character in column 1 enters bit positions 0-5 of the first computer word, the second character of column 1 enters positions 6-11, and so on. Thus, a single card may represent up to 160 characters (40 words at 4 characters/word) in binary format.

OPERATING PROCEDURES

The input hopper holds a maximum of 1000 cards and is loaded with the 9 edge toward the panel and column 1 toward the read station (operator's left). The output stacker holds a maximum 1000 cards. Cards in the output stacker are in the same sequence and orientation in which they were placed in the input hopper. The reader control panel consists of four switch/indicators and four indicators.



READER CONTROL PANEL

Switch/Indicators

POWER

Depressing the POWER switch turns the card reader on. The POWER switch is illuminated when the card reader is on.

START

Depressing START makes the reader ready for read operation if all error indicators are off. The START switch is illuminated if the above conditions are satisfied.

STOP

The STOP switch places the reader in a not ready condition. If the reader becomes not ready, the STOP switch is illuminated.

CLEAR

The CLEAR switch resets any error indicator which has been set, if the condition which caused the error has been corrected. This indicator is illuminated if any error indicator is set.

Indicators

HOPPER EMPTY

This indicator is turned on when there are no cards in the hopper. The indicator remains on until cards are placed in the hopper and the CLEAR switch is depressed.

STACKER FULL

This indicator is turned on when the stacker becomes full (approximately 1000 cards). The indicator is turned off by removing cards from the stacker and depressing the CLEAR switch.

PICK FAILURE

The PICK FAILURE indicator is turned on when a card fails to reach the read station within a specified time after the picker is energized. Pick failures are usually due to damaged or out of tolerance cards. Inspect the leading edge of the bottom card in the hopper if a pick failure occurs.

CARD MOTION

If the CARD MOTION indicator is turned on, a card failed to complete passage through the read station within a set time. If a card jam has occurred, place the unit in a POWER OFF condition before removing damaged cards from the card track.

LIGHT CURRENT

If a read station phototransistor fails to see light after a card is picked and before the leading edge reaches the read station, the LIGHT CURRENT indicator is turned on. A defective light source or foreign matter in the read station can cause this condition.

DARK CURRENT

If a read station phototransistor is not darkened by the leading edge of a card before column 1 is sensed, the DARK CURRENT indicator is turned on. A torn leading edge or read station malfunction can cause this condition.

TEST MODE

The TEST MODE indicator is turned on if the maintenance TEST/NORMAL switch is in the TEST position.

Card Reader Instructions

RCB Read Card Binary ACT '3606

The card reader feeds a card from the hopper to the read station to be read in binary format. Each column is transmitted as two 6-bit characters with the first character corresponding to rows 12, 11, 0, 1, 2, 3 and the second character corresponding to rows 4, 5, 6, 7, 8, 9.

CRO Card Reader Offset ACT '12006

The card presently in the read station will be offset by 1/4" when placed into the output stacker.

CRDY Card Reader Ready SNS'10006

If the card reader is ready, a skip will occur. If the card reader is not ready a skip will not occur. The card reader is ready if:

1. Power ON
2. START is illuminated
3. Card is not being fed thru read station
4. No trouble indicator
5. Hopper is not empty
6. Stacker is not full
7. Error indicators are reset

SNCRE Skip if No Card Reader Error SNS'11006

If no card reader error condition exists, a skip will occur. If an error condition exists, a skip will not occur. A card reader error exists if:

1. Hopper Empty ON
2. Stacker Full ON
3. Light Current Error ON
4. Dark Current Error ON
5. Pick Failure ON
6. Card Motion Error

SRHNE Skip if Read Hopper Not Empty SNS '14006

If the card read hopper is not empty, a skip will occur. If the card read hopper is empty, a skip will not occur.

MAGNETIC TAPE

The SCC magnetic tape units are seven level, IBM-compatible. Tape densities are 200, 556 and 800 bits per inch. Read, write, scan and erase operations may be performed in a forward direction. Scan and erase operations may also be performed in the reverse direction. Tape speed in the forward and reverse direction is 75 inches per second.

Magnetic Tape Test Instructions

The magnetic tape unit tests instructions are coded for the B buffer, with n being the number (0-7) of the magnetic tape unit.

TRDY 0,n Skip if Tape Ready SNS '1041n

If tape unit n is ready, a skip will occur. If tape unit n is not ready, a skip will not occur. Conditions which place the tape unit in a not ready status are:

1. No tape unit is set to the number being tested
2. The selected unit is busy
3. The selected unit is off-line.

SNR 0,n Skip if No Ring SNS'1401n

If the file-protect ring is not inserted, a skip will occur. If the file-protect ring is inserted, a skip will not occur. No skip will occur if no tape unit is set to n. The file-protect ring must be inserted for write operations.

SLP 0,n Skip if at Load Point SNS '1201n

If tape unit n is positioned at the load point, a skip will occur. If tape unit n is not positioned at the load point a skip will not occur. If no tape unit is set to n, a skip will not occur.

SET 0,n Skip if at End of Tape SNS '1101n

If tape unit n is positioned past the end-of-tape marker, a skip will occur. If tape unit n is not positioned past the end-of-tape marker, a skip will not occur. If no tape unit is set to n, a skip will not occur.

SDEN5 0,n Skip if Density 556 SNS '1021n

If tape unit n is set for 556 bits per inch, a skip will occur. If tape unit n is not set for 556 bits per inch, a skip will not occur. If no tape unit is set to n, a skip will not occur.

SDEN8 0,n Skip if Density 800 SNS '1061n

If tape unit n is set for 800 bits per inch, a skip will occur. If tape unit n is not set for 800 bits per inch, a skip will not occur. If no tape unit is set to n, a skip will not occur.

SNG Skip if No Gap SNS'10010

If the tape unit presently connected to the tape controller is not positioned over a gap, a skip will occur. If the tape unit presently connected to the tape controller is positioned over gap, a skip will not occur. If no tape unit is presently connected to the tape controller, a skip will not occur.

Magnetic Tape Control Instructions

The magnetic tape control instructions that follow are coded for the B buffer, with n being the number (0-7) of the magnetic tape unit.

REW,0,n Rewind ACT '1401n

Tape unit n is started in a rewind. After completion of the rewind, the tape is positioned at the load point.

RUN 0,n Rewind and Unload ACT '1201n

Tape unit n is started in a rewind. Once started, the tape continues in rewind until the tape is unloaded.

CRS Convert Read to Scan ACT '10400

The tape unit currently on the buffer is instructed to convert from the read mode of operation to the scan mode of operation.

READ OPERATIONS

Tape may be read in even or odd parity. Even parity is referred to as the BCD mode and odd parity is referred to as the binary mode. Tape movement is initiated by a read command and continues until gap is encountered. When the tape stops, the tape unit disconnects from the buffer. The single character per word mode should not be used for tape read operations due to the physical characteristics of magnetic tape recording. Consequently, the longitudinal check character will always be read into memory with the end-of-file character. In a four character per word read operation, the end-of-file character will be read into memory as 17170000_8.

RTB 0,n Read Tape Binary ACT '361n

Tape unit n is initiated. Tape unit n is started in a binary (odd parity) read mode.

RTD Read Tape (Decimal) BCD ACT '261n

Tape unit n is started in a BCD (even parity) read mode.

SCAN OPERATIONS

During scan operations, the data is transferred from the tape controller to the buffer as in read operations. However, the buffer does not consider a word complete until gap is encountered. When gap is encountered, the buffer contains the last characters of the record. This "last" word can be read into memory with a TBM instruction. When scanning in reverse, this word consists of the first four characters of the record. These characters will be in reverse order. For example, if the first four characters of the record were HOPE and the record was scanned in reverse, these would appear as EPOH in the buffer. This word can be stored in memory with the TBM instruction. All scan operations must be in the two-, three-, or four-character per word mode.

SFB 0,n Scan Forward Binary ACT '363n

Tape unit n is started forward in a binary scan mode.

SFD 0,n Scan Forward Decimal (BCD) ACT '263n

Tape unit n is started forward in a BCD scan mode.

SRB 0,n Scan Reverse Binary ACT '763n

Tape unit n is started in reverse in a binary scan mode.

SRD 0,n Scan Reverse Decimal ACT '663n

Tape unit n is started in reverse in a BCD scan mode.

WRITE OPERATIONS

The file-protect ring must be inserted in the tape unit to write on a tape. A tape write instruction will start the tape in motion and the tape will remain in motion until the terminal signal from the buffer is received. The tape unit will then stop and disconnect from the buffer. An end-of-file record may be written using a one character per word, BCD write instruction. The word to be written must contain the tape mark character (17_8) in the leftmost character. The write instructions that follow are coded for the B buffer, with n being the number (0-7) of the magnetic tape unit.

WTB 0,n Write Tape Binary ACT '365n

Tape unit n is started in a binary write mode.

WTD 0,n Write Tape Decimal ACT '265n

Tape unit n is started in a BCD write mode.

ERASE OPERATIONS

Erase operations are similar to write operations. A tape erase operation will start the tape in motion and the tape will remain in motion until the termination signal from the buffer is

received. The erase operation will write blank tape while the tape is in motion. Thus, the length of tape erased is variable. A gap of 3.75 inches blank tape may be accomplished with an erase of 417 four-character words at 556 bpi or 600 words at 800 bpi. The erase instructions to follow are coded for magnetic tape unit n (0-7) on the B buffer, using the 4 characters per word mode.

ETF 0,n	Erase Tape Forward	ACT '367n

Tape unit n is started forward in an erase mode.

ETR 0,n	Erase Tape Reverse	ACT '767n

Tape unit n is started in reverse in an erase mode.

SECTION V

SCC 660 CONTROL CONSOLE

SCC 660 CONTROL CONSOLE

The SCC 660 is equipped with a control console to permit manual control of the central processor. The control console has been carefully designed to permit maximum communication between operator and computer. The console contains switches which allow the operator to communicate with the computer and displays which indicate the computer status to the operator.

CONSOLE SWITCHES

Control Panel

When this key operated switch is in the "ON" position all switches operate normally. However, when this switch is in the "OFF" position those switches which might cause the program to be disturbed are disabled.

Memory Protect

The memory protect switch turns the protection subsystem on or off. When this switch is ON any attempt to write into protected memory will be inhibited and an interrupt request to the internal condition interrupt channel will be generated. When this switch is in the OFF position, the protection subsystem is disabled.

Breakpoint

These four switches do not directly control the processor, but may be interrogated by a program via the breakpoint test instructions. The breakpoint switches are operative regardless of the control panel switch position.

Start

The start switch initializes the computer for processing. Depressing this switch performs the following functions:

(a) The HALT mode is immediately entered and the HALT indicator is turned ON.

(b) The I/O ERROR, OVERFLOW and CARRY OUT indicators are all turned off and all interrupts are disarmed. In addition, if the INTERRUPT ENABLED switch is OFF the interrupt system is disabled and the INTERRUPT ENABLED indicator is turned OFF.

(c) All device controllers are reset to the inactive state and all I/O status indicators are turned off.

(d) The AC, EA, XR and C Registers are cleared and the C register is selected for display.

(e) The program location counter is cleared.

This switch is inoperative when the CONTROL PANEL switch is in the OFF position.

Clear

The CLEAR switch resets to zero the register currently being displayed by the REGISTER DISPLAY indicator.

The CLEAR switch is inoperative when the CONTROL PANEL switch is in the OFF position.

Register

These four switches select the AC, EA, C and XR Registers, respectively, to be displayed or entered. If the CONTROL PANEL switch is in the OFF position, these registers may be displayed but not altered from the console.

Bootstrap

The BOOTSTRAP switch initiates a bootstrap or self loading program. Depressing this switch causes the following functions to be performed.

(a) The paper tape reader on the B buffer is selected for input in the four characters per word mode and one instruction is read from the paper tape reader into memory location 2.

(b) The index register is set to a minus 7.

(c) The instruction at location 2 is executed.

The following is an example of a bootstrap loader program which will load itself into memory locations 00002 thru 00011 and then load a block of instructions starting at the memory address specified by the seventh word on the tape.

Location	Instruction	Address	Comments
2	TBM	10,X	Read next instruction from tape
3	BIX	2	
4	LDX	9	Load XR with starting address of program
5	TBM	0,X	
6	BRD		Test for end-of-word gap
7	BIX	5	
8			Instruction executed after gap is detected
9			Starting address with a "one" in bit position 8

Store

The STORE switch provides a convenient means of manually entering instructions or data into memory. Depressing the STORE switch causes the contents of the C register to be stored into the memory location being displayed by the location counter and then increments the location counter.

I/O Display

The I/O Display switch has two positions labeled B and C. When this switch is in the "B" position the unit address of the device attached to the B buffer is displayed by the INPUT OUTPUT UNIT display. When the I/O display switch is in the "C" position the unit address of the device attached to the C buffer is displayed by the INPUT OUTPUT UNIT display.

Interrupt

When the INTERRUPT switch is set to ENABLE the interrupt system is enabled. The INTERRUPT switch overrides the disable interrupt (DIS) instruction.

Run

With this switch in the RUN position, instruction execution proceeds automatically. When this switch is taken out of the RUN position, the computer enters an "idle" state. The C register contains the next instruction to be executed. The address of the next instruction to be executed is presented in the program location display.

Test

The TEST switch will illuminate all console indicators. Any lamps which remain off when the TEST switch is depressed should be replaced.

Step

If the computer is in the idle state, depressing the STEP switch causes the computer to execute the current contents of the C register, load the C register with the next instruction in sequence, and automatically return to an idle state.

Power

This key-operated switch turns the computer power system on or off.

DISPLAYS

Register Display

These 24 indicators display the contents of the register selected by the REGISTER switches. With the computer in the idle state, pressing the CLEAR button clears the selected register to all zeros. Depressing the pushbuttons beneath the register display indicators will set the corresponding bits of the register selected by the REGISTER switches.

Program Location

The current contents of the LC (Location Counter) register is displayed in these 15 binary indicators. When the computer is in the idle state, this display contains the memory address of the next instruction to be executed.

Input Output Unit

The current contents of the unit address register of the buffer specified by the I/O DISPLAY switch is displayed.

Control

Execution of a Halt (HLT) instruction or manually placing the computer in the idle state will illuminate the HALT indicator. The HALT indicator is turned off and program execution resumed by placing the computer in the idle state and then in the RUN state.

The ERROR indicator displays the current status error flip-flop for the buffer selected by the I/O DISPLAY switch. When the ERROR indicator is on, an error occurred during the previous input/output operation involving the specified buffer.

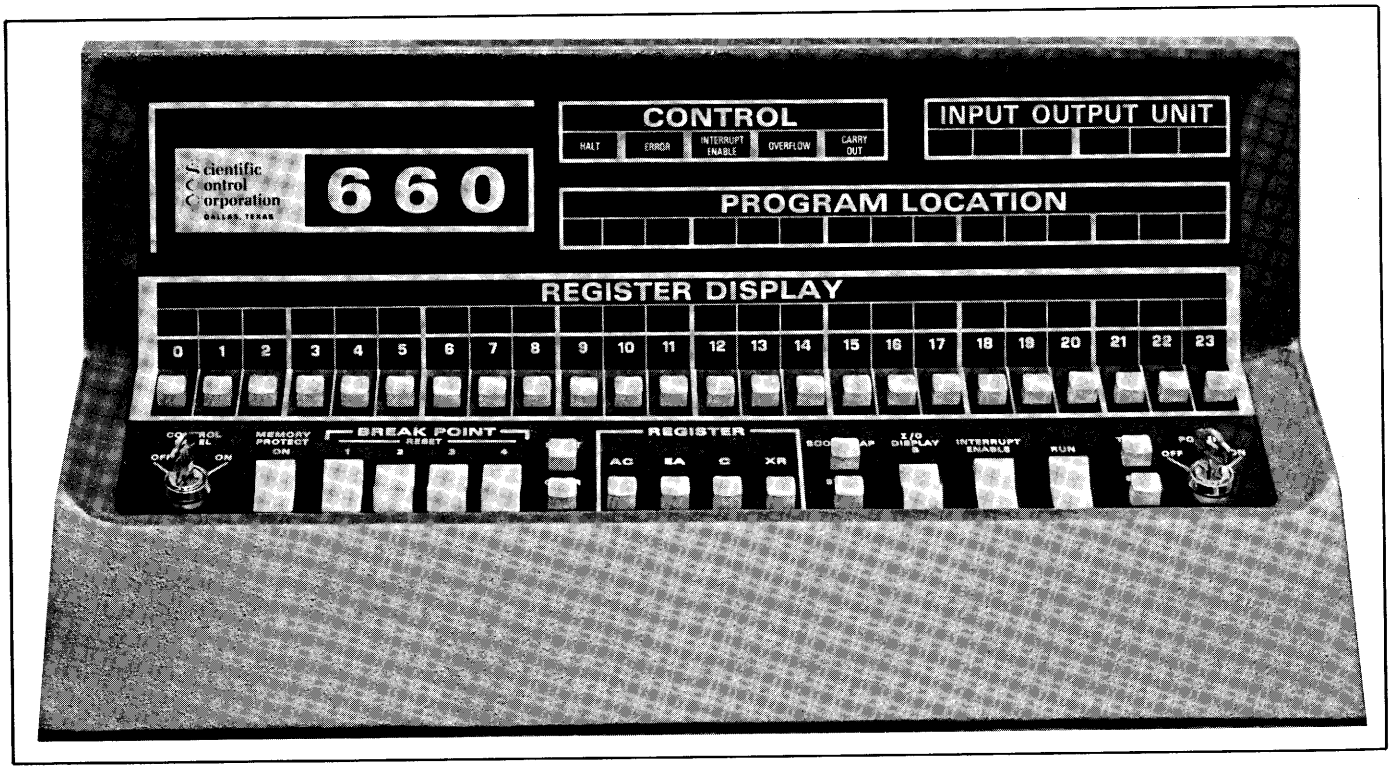
The INTERRUPT ENABLE indicator is on if the interrupt system is presently enabled and off if the interrupt system is disabled.

The OVERFLOW indicator is illuminated if the central processor overflow indicator is set.

The CARRY OUT indicator is illuminated if the central processor carry out indicator is set.

SECTION V

SCC 660 CONTROL CONSOLE



APPENDIX A

NUMBER SYSTEMS

The decimal number system is in common use because early man used his ten fingers for counting whereas increased use of the binary and octal number systems is due to the advent of the binary computer. The similarity of the fundamental principles of the binary, octal and decimal number systems is such that a review of the decimal number system is beneficial to an understanding of the binary and octal number systems.

DECIMAL NUMBER SYSTEM

The decimal number system is so named because it uses ten distinct symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) for representing numbers. (The word "decimal" comes from the Latin word for ten – "decem." Other familiar words using the same root are: decade – ten sided figure, December – the tenth month of the Roman Calendar.)

In the decimal number system the position in which a numeral appears determines its "value." For example, the "two" in 203 represents two hundreds and the "two" in 162 represents "two" ones. Thus, 162 is one hundred, 6 tens and 2 ones; 203 is two hundreds, 0 tens and 3 ones. 48,715 can be written as:

$4 \times$ ten thousand $+ 8 \times$ one thousand $+ 7 \times$ one hundred $+ 1 \times$ ten $+ 5 \times$ one

or $4 \times 10000 + 8 \times 1000 + 7 \times 100 + 1 \times 10 + 5 \times 1$

or $4 \times 10^4 + 8 \times 10^3 + 7 \times 10^2 + 1 \times 10^1 + 5 \times 10^0$.

(10^0 is 1. Any number other than zero raised to the zero power is one. For example, $6^0 = 1$, $2^0 = 1$, $8^0 = 1$, $10^0 = 1$.)

A decimal fraction such as .237 can be written as:

$2 \times$ one tenth $+ 3 \times$ one hundredth $+ 7 \times$ one thousandth

or $2 \times \frac{1}{10} + 3 \times \frac{1}{100} + 7 \times \frac{1}{1000}$

or $2 \times \frac{1}{10^1} + 3 \times \frac{1}{10^2} + 7 \times \frac{1}{10^3}$

or $2 \times 10^{-1} + 3 \times 10^{-2} + 7 \times 10^{-3}$

Thus 435.62 can be written as:

$$4 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 2 \times 10^{-2}.$$

To summarize, the two main characteristics of the decimal number system are

1) Ten symbols

   0, 1, 2, 3, 4, 5, 6, 7, 8, 9

2) Power of Ten

   $10^6, 10^9, 10^0, 10^{-3}, 10^{-1}, \ldots$

OCTAL NUMBER SYSTEM

The octal number system is so named because it uses eight distinct symbols (0, 1, 2, 3, 4, 5, 6, 7) for representing numbers, and the positional value of a numeral is a power of eight (8^0, 8^1, 8^2, etc.). Since the octal and the decimal number system use the same symbols (0, 1, 2, 3, 4, 5, 6, 7), a subscript is used to denote the intended number system. For example, 47_{10} denotes the decimal number system and 57_8 denotes the octal number system.

The decimal number system uses ten symbols and powers of ten to represent numbers; the octal number system uses eight symbols and powers of eight, thus:

$$475_8 = 4 \times 8^2 + 7 \times 8^1 + 5 \times 8^0.$$

Thus: 5347221_8 can be written as:

$$5 \times 8^3 + 3 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 2 \times 8^{-2} + 1 \times 8^{-3}.$$

which is $5 \times 256_{10} + 3 \times 64_{10} + 4 \times 8_{10} + 7 \times 1 + 2 \times \frac{1}{8_{10}} + 2 \times \frac{1}{64_{10}} + 1 \times \frac{1}{256_{10}}$.

To summarize, the two main characteristics of the octal number system are:

1) Eight symbols

   0, 1, 2, 3, 4, 5, 6, 7

2) Powers of Eight

2) $8^7, 8^2, 8^0, 8^{-3}, 8^{-1}, \ldots$

The usefulness of the octal number system is as a shorthand for binary numbers, since three binary digits may be represented by one octal digit.

BINARY NUMBER SYSTEM

The binary number system is so named because it uses two distinct symbols (0, 1) for representing numbers, and the positional value of a numeral is a power of two (2^0, 2^1, 2^2, 2^3, etc.). Since the binary symbols (0, 1) are used by the octal and the decimal number system, a subscript is used to denote the intended number system. For example, 1001_{10} denotes the decimal number system, 1001_8 denotes the octal number system, and 1001_2 denotes the binary number system.

The decimal number system uses ten symbols and powers of ten. Thus, 1001_{10} is:

$$1 \times 10^3 + \times 10^2 + 0 \times 10^1 + 1 \times 10^0$$

or $1 \times 1000_{10} + 0 \times 100_{10} + 0 \times 10_{10} + 1 \times 1$

or $1000_{10} + 0 + 0 + 1$

or 1001_{10}.

The octal number system uses eight symbols and powers of eight. Thus, 1001_8 is:

$$1 \times 8^3 + 0 \times 8^2 + 0 \times 8^1 + 1 \times 8^0$$

or $1 \times 256_{10} + 0 \times 64_{10} + 0 \times 8_{10} + 1 \times 1$

or $256_{10} + 0 + 0 + 1$

or 257_{10}.

The binary number system uses two symbols and powers of two. Thus, 1001_2 is:

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

or $1 \times 8_{10} + 0 \times 4_{10} + 0 \times 2_{10} + 1 \times 1$

or $8_{10} + 0 + 0 + 1$

or 9_{10}.

Thus, 1101.101 can be written as:

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + \frac{1}{2^1} + \frac{0}{2^2} + \frac{1}{2^3}$$

or $$8_{10} + 4_{10} + 0 + 1 + \frac{1}{2_{10}} + \frac{0}{4_{10}} + \frac{1}{8_{10}}$$

or $$8_{10} + 4_{10} + 0 + 1 + .5_{10} + 0 + .125_{10}$$

or $$13.625_{10}$$

To summarize, the two main characteristics of the binary number system are:

1) Two symbols

   0, 1

2) Powers of two

   $2^7, 2^5, 2^0, 2^{-3}, 2^{-6}, \ldots$

APPENDIX B
LOGICAL OPERATIONS

Logical Operations

All logical operations are performed bit by corresponding bit between two words; one word is in the AC accumulator and the other is in the effective location. The result of the logical operation is placed in the AC accumulator. The following paragraphs discuss three logical operations: AND, inclusive OR, and exclusive OR.

When two numbers are combined by an AND, they are matched bit-for-bit. If the same position in each word contains a 1, the result is a 1. If in one word the position is 0 and in the other word it is a 1, the result is a 0. If the same position in both words is a zero, the result is a 0. The following is an example of a logical AND operation:

```
101 101 011 011
101 001 001 101
101 001 001 001     Resulting AND
```

The following examples illustrate two cases of special interest.

```
111 111 111 111
101 001 001 101
101 001 001 101     Resulting AND

000 000 000 000
101 001 001 101
000 000 000 000     Resulting AND
```

The first example illustrates that any number "anded" with 1's produces a result identical to the number, or 1's AND number = number. The second example illustrates that any number "anded" with 0's produces a result of zero, or 0's AND number = 0.

The following example illustrates an application of these rules.

```
111 111 000 000     Mask
101 001 001 101     Data
101 001 000 000     Resulting AND
```

The six leftmost bits of the result are identical to the six leftmost bits of the data word and the six rightmost bits of the result are zeros. The first of the two words is referred to as a "mask". The six rightmost bits are said to be "masked out" and the procedure is referred to as "masking".

An OR operation (sometimes called "inclusive OR") also matches two numbers bit-for-bit. The difference, however, when compared with an AND, is (1) if the same position in either word contains a 1, the result is a 1; (2) if the same position in both words is a 1, the result is again a 1; (3) only if the same position in both words is a 0, is the resulting position a 0. For example:

```
011 010 110 101
001 100 100 100
011 110 110 101     Resulting inclusive OR
```

The following examples illustrate two cases of special interest.

```
111 111 111 111
001 101 100 110
111 111 111 111     Resulting inclusive OR

000 000 000 000
001 101 100 110
001 101 100 110     Resulting inclusive OR
```

The first example illustrates that any number "ored" with 1's produces a result of 1's, or 1's OR number = 1's. The second example illustrates that any number "ored" with 0's produces a result identical to the number, or 0's OR number = number.

The following example illustrates an application of these rules.

```
101 100 000 000     Data
000 000 011 001     Data
101 100 011 001     Resulting inclusive OR
```

The six leftmost bits of the result are identical to the six leftmost bits of the first data word and the six rightmost bits of the result are identical to the six rightmost bits of the second data word.

The EOR (called exclusive OR) is another of the logical operations. In this operation, only those positions which do not match result in a 1. If the same position in each word is a zero or if the same position in each word is a 1, the result is a zero. If the same position in one word is a 1 and in the other a 0, then the result is a 1. For example:

```
101 101 100 101
001 011 001 101
100 110 101 000     Resulting exclusive OR
```

The following examples illustrate three cases of special interest:

```
111 111 111 111
001 011 001 101
110 100 110 010     Resulting exclusive OR
```

```
000 000 000 000
001 011 001 101
---------------
001 011 001 101     Resulting exclusive OR


001 011 001 101
001 011 001 101
---------------
000 000 000 000     Resulting exclusive OR
```

The first example illustrates that any number "exclusive ored" with 1's produces the one's complement of the number, or 1's EOR number = one's complement of number. (The one's complement of 0 is 1. The one's complement of 1 is 0.) The second example illustrates that any number "exclusive ored" with 0's produces a result identical to the number, or 0's EOR number = number. The third example illustrates that any number "exclusive ored" with itself produces a result of zero.

APPENDIX C

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 171 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
291 484 976 710 656	48	0.000 000 000 000 003 552 513 678 800 500 929 355 621 337 890 625

0.001	1.00069	33874	62581
0.002	1.00138	72557	11335
0.003	1.00208	16050	79633
0.004	1.00277	64359	01078
0.005	1.00347	17485	09503
0.006	1.00416	75432	38973
0.007	1.00486	38204	23785
0.008	1.00556	05803	98468
0.009	1.00625	78234	97782
0.01	1.00695	55500	56719
0.02	1.01395	94797	90029
0.03	1.02101	21257	07193
0.04	1.02811	38266	56067
0.05	1.03526	49238	41377
0.06	1.04246	57608	41121
0.07	1.04971	66836	23067
0.08	1.05701	80405	61380
0.09	1.06437	01824	53360
0.1	1.07177	34625	36293
0.2	1.14869	83549	97035
0.3	1.23114	44133	44916
0.4	1.31950	79107	72894
0.5	1.41421	35623	73095
0.6	1.51571	65665	10398
0.7	1.62450	47927	12471
0.8	1.74110	11265	92248
0.9	1.86606	59830	73615

APPENDIX D

POWERS OF TEN EXPRESSED IN OCTAL

10^n Expressed in Octal	n	10^{-n} Expressed in Octal
1	0	1.000 000 000 000 000 000 00
12	1	0.063 146 314 631 463 146 31
144	2	0.005 075 341 217 270 243 66
1 750	3	0.000 406 111 564 570 651 77
23 420	4	0.000 032 155 613 530 704 15
303 240	5	0.000 002 476 132 610 706 64
3 641 100	6	0.000 000 206 157 364 055 37
46 113 200	7	0.000 000 015 327 745 152 75
575 360 400	8	0.000 000 001 257 143 561 06
7 346 545 000	9	0.000 000 000 104 560 276 41
112 402 762 000	10	0.000 000 000 006 676 337 66
1 351 035 564 000	11	0.000 000 000 000 537 657 77
16 432 451 210 000	12	0.000 000 000 000 043 136 32
221 411 634 520 000	13	0.000 000 000 000 003 411 35
2 657 142 036 440 000	14	0.000 000 000 000 000 264 11
34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01

APPENDIX E

MATHEMATICAL CONSTANTS

Constant	Decimal	Octal
$\sqrt{10}$	3.16227 76601 68379	3.123 054 072 67
$\ln 10$	2.30258 40929 94046	2.232 730 673 55
$\sqrt{2}$	1.41421 35623 73095	1.324 047 463 20
$\ln 2$	0.69314 71805 59945	0.542 710 277 60
$\log_{10} 2$	0.30102 99956 63981	0.232 101 152 04
π	3.14159 26535 89793	3.110 375 524 21
$1/\pi$	0.31830 98861 83790	0.242 763 015 56
$\sqrt{\pi}$	1.77245 38509 05516	1.613 376 110 67
$\ln \pi$	1.14472 98858 49400	1.112 064 044 35
e	2.71828 18284 59045	2.557 605 213 05
$1/e$	0.36787 94411 71442	0.274 265 306 61
$\sqrt{e}$	1.64872 12707 00128	1.514 112 307 04
$\log_{10} e$	0.43429 44819 03252	0.336 267 542 51
$\log_2 e$	1.44269 50408 88963	1.342 521 662 45
γ	0.57721 56649 01533	0.447 421 477 07
$\ln \gamma$	−0.54953 93129 81645	−0.431 272 336 02

APPENDIX F

OCTAL—DECIMAL TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

4000 (Octal)	2048 (Decimal)
to	to
4777 (Octal)	2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 (Octal)	2560 (Decimal)
to	to
5777 (Octal)	3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000	3072
to	to
6777	3583
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000	3584
to	to
7777	4095
(Octal)	(Decimal)

APPENDIX G

OCTAL—DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

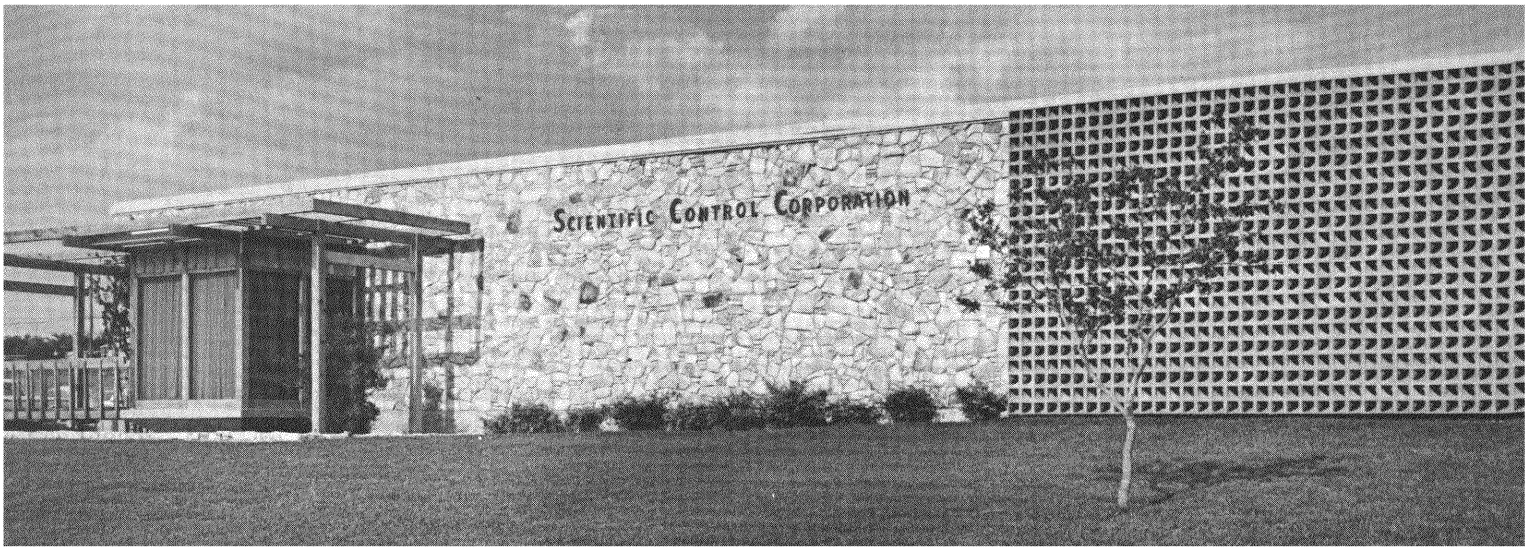
APPENDIX H
BAUDOT CODE

OCTAL	BINARY	CHARACTER	
		LETTERS	FIGURES
00	00 000	Blank (▽)	Blank (▽)
01	00 001	T	5
02	00 010	Car. Ret. (<)	Car. Ret. (<)
03	00 011	O	9
04	00 100	Space (▭)	Space (▭)
05	00 101	H	
06	00 110	N	,
07	00 111	M	.
10	01 000	Line Feed (≡)	Line Feed (≡)
11	01 001	L	)
12	01 010	R	4
13	01 011	G	&
14	01 100	I	8
15	01 101	P	0
16	01 110	C	:
17	01 001	V	;
20	10 000	E	3
21	10 001	Z	"
22	10 010	D	$
23	10 011	B	?
24	10 100	S	Bell
25	10 101	Y	6
26	10 110	F	!
27	10 111	X	/
30	11 000	A	-
31	11 001	W	2
32	11 010	J	'
33	11 011	Figs (∧)	Figs (∧)
34	11 100	U	7
35	11 101	Q	1
36	11 110	K	(
37	11 111	Ltrs (∨)	(∨)

APPENDIX I

CHARACTER CODES

CODE	TYPEWRITER	LINE PRINTER	CODE	TYPEWRITER	LINE PRINTER
ØØ	Ø	0	4Ø	–	–
Ø1	1	1	41	J	J
Ø2	2	2	42	K	K
Ø3	3	3	43	L	L
Ø4	4	4	44	M	M
Ø5	5	5	45	N	N
Ø6	6	6	46	O	O
Ø7	7	7	47	P	P
1Ø	8	8	5Ø	Q	Q
11	9	9	51	R	R
12	SPACE	SPACE	52	CARRIAGE RETURN	"
13	=	=	53	$	$
14	'	'	54	×	*
15	:	:	55	]	]
16	>	>	56	;	;
17	√	√	57	Δ	#
2Ø	+	+	6Ø	ↄ	&
21	A	A	61	/	/
22	B	B	62	S	S
23	C	C	63	T	T
24	D	D	64	U	U
25	E	E	65	V	V
26	F	F	66	W	W
27	G	G	67	X	X
3Ø	H	H	7Ø	Y	Y
31	I	I	71	Z	Z
32	BACKSPACE	←	72	TAB	↑
33	.	.	73	,	,
34	)	)	74	(	(
35	[	[	75	¤	@
36	<	<	76	\	\
37	‡	%	77	##	?



SCC maintains complete support activities for its users. Installation and maintenance services are available through SCC offices strategically located throughout the United States. For pre-procurement demonstration of hardware and programs in Dallas, contact nearest regional office or the marketing department in Dallas.

EASTERN REGION
College Park, Maryland
(Washington, D.C.)
301/779-2510

SOUTHEASTERN REGION
Huntsville, Alabama
205/881-8805

WESTERN REGION
Los Angeles, California

SOUTHWESTERN REGION
Dallas, Texas
214/241-2111

MIDWESTERN REGION
Dallas, Texas
214/241-2111

CORPORATE MARKETING
Dallas, Texas
214/241-2111

Scientific Control Corporation

P.O. Box 34529 • Dallas, Texas 75234 • 214 — 241-2111 • TWX 910-860-5509

SCC 06-0168