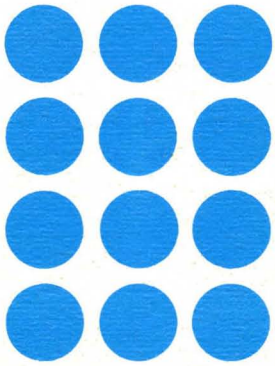
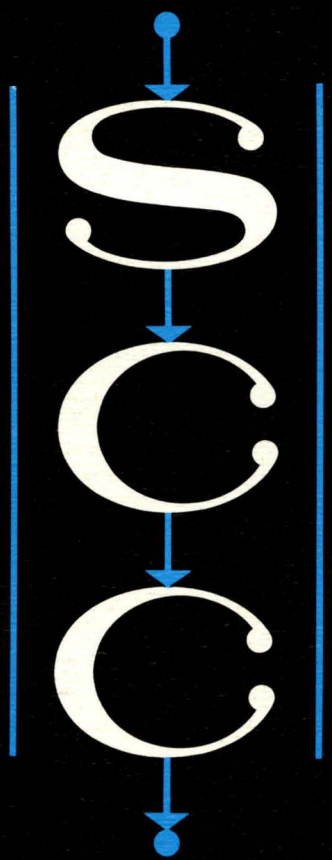
Scientific Control Corporation

Digital Computers
670-2 • 670-5





REFERENCE MANUAL

SCC SERIES 670 DIGITAL COMPUTERS

SCIENTIFIC CONTROL CORPORATION
14008 Distribution Way
Dallas, Texas 75234

Telephone: A/C 214 CHapel 1-2111

TABLE OF CONTENTS

I. INTRODUCTION

Scientific Control Corporation of Dallas, Texas is engaged in the design and manufacture of general purpose digital computers. These systems include the latest techniques of design and programming. The SCC 670 Series of General Purpose Computers are designed to meet the individual's requirements, thereby allowing the user to purchase only as much system as he needs. The computers are solid state, binary, single address with indexing and indirect addressing. A complete package of software is available with the systems.

Installation, warranty, maintenance, documentation, operator and programmer training are offered. Scientific Control Corporation maintains complete support activities by means of of offices located throughout the United States.

II. SCC 670 SERIES COMPUTER, GENERAL DESCRIPTION

The SCC 670 Series consists of general purpose solid state digital computers. The computer may be classified as a binary, single address processor with a high speed magnetic core memory. Internal operation is parallel with powerful indexing and indirect addressing provisions, thus providing for a high rate of computation without sacrificing program flexibility.

A. MEMORY

The memory of the SCC 670 Series Computer is a high speed coincident core memory. The width of the memory is twenty-four data bits per word.

Up to 32,768 words of memory may be attached to the computer.

B. INSTRUCTIONS

The SCC 670 Series contains a powerful instruction set with the full capabilities of performing data movement, arithmetic and logic computation, conditional and unconditional transfers of control, and input/output. The basic add execution requires only two machine cycles. Two registers are available for computation. An index register is also provided. Every address may be indexed. In addition, indirect addressing may be employed, without limit, on any address.

Special techniques are used to enable extremely efficient linkages to subroutines. Instructions are available which will automatically transfer to a subroutine, storing all status and return parameters as well as creating an argument linkage, and which will automatically return to the calling program. The programmer is thus freed from the necessity for creating complex and time-consuming linkage techniques to subroutines. Since a subroutine entered in this manner acts as a functional operator upon a series of arguments, these subroutines are referred to as "programmed operators."

C. INPUT/OUTPUT

The 670 Series Computer contains an input/output buffer which allows buffered transfer of characters at 6, 12, 18 or 24 bits between memory and a selected input/output device. A second buffer is available to allow simultaneous input/output operations through a memory interlace technique. In addition, a twenty-four bit parallel channel is available to allow highly efficient data transfer between the memory and any external device. Two channels of priority interrupt are provided as standard equipment. Additional channels may be added in blocks of eight channels.

The interrupt system may be enabled or disabled under program control. The programmer may thus operate in a "lock-up" mode; i.e., an input/output instruction will wait until the operation is complete before proceeding, or, in an interrupt mode, allow overlap of input/output and computation.

A typical configuration is shown in Figure No. 1.

D. PHYSICAL CHARACTERISTICS

The SCC 670 Series Computer is physically contained in an attractive rack measuring approximately 2' x 4½' x 6'. The basic input/output devices (paper tape reader, paper tape punch, and typewriter) along with the computer control console are housed in a standard size desk with typewriter arm.

Power requirements may be summarized as 115 volts A.C., 60 cps at approximately thirty amps.

The computer will operate over a temperature range of 0° to 60° C. Considerable attention has been paid to the physical design of the processor to insure a high degree of reliability as well as ease of maintenance.

E. SOFTWARE

Scientific Control Corporation offers a comprehensive set of programming aids for use with the 670 Series Computer. Among the set are a symbolic assembly program, a FORTRAN compiler, a utility package and a comprehensive library of subroutines. Diagnostic aids of several varieties are also included. All software is maintained in a program library. Scientific Control Corporation also offers customer programming services by qualified programmers when required.

BLOCK DIAGRAM

SHOWING

INPUT/OUTPUT CONFIGURATION

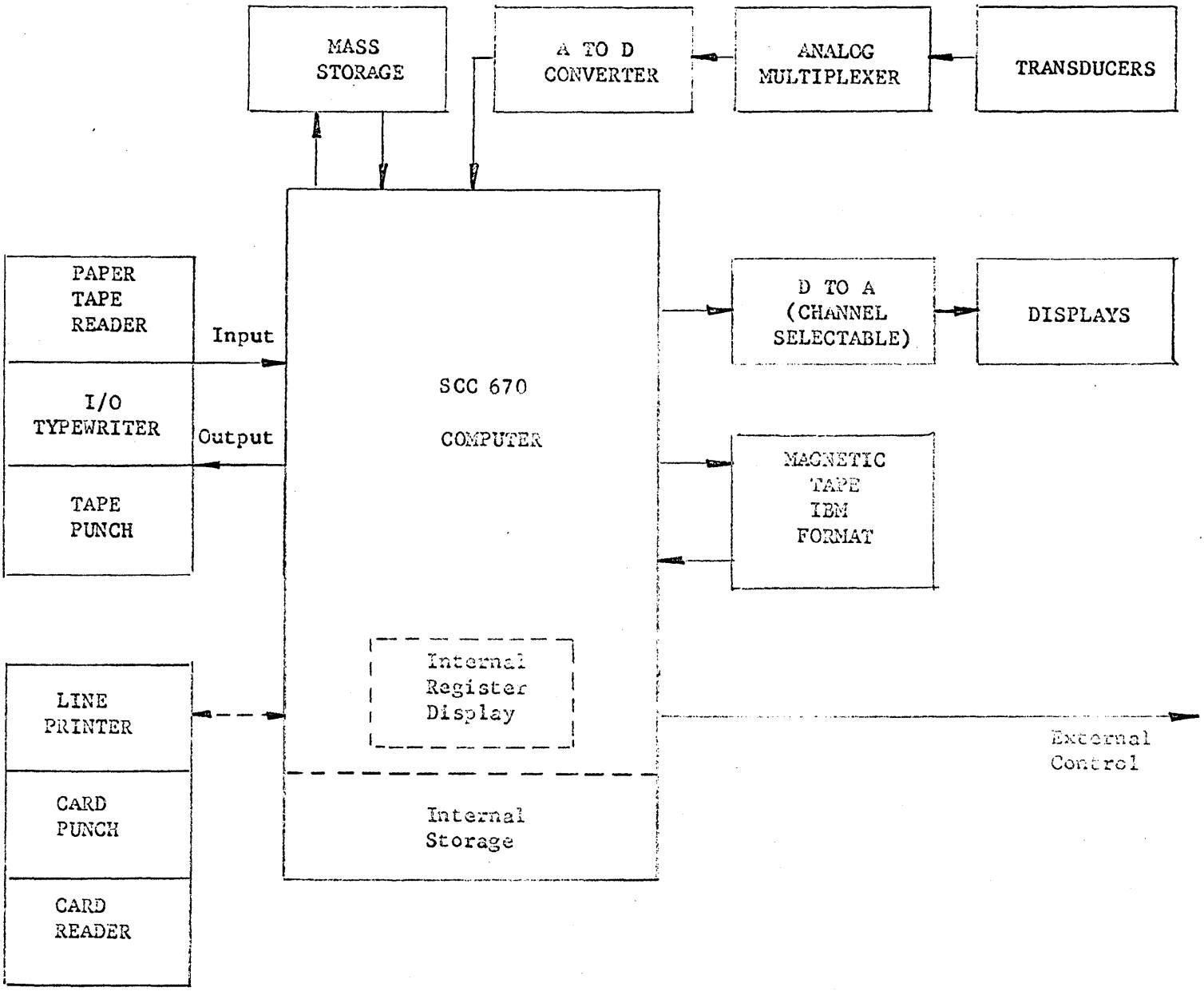
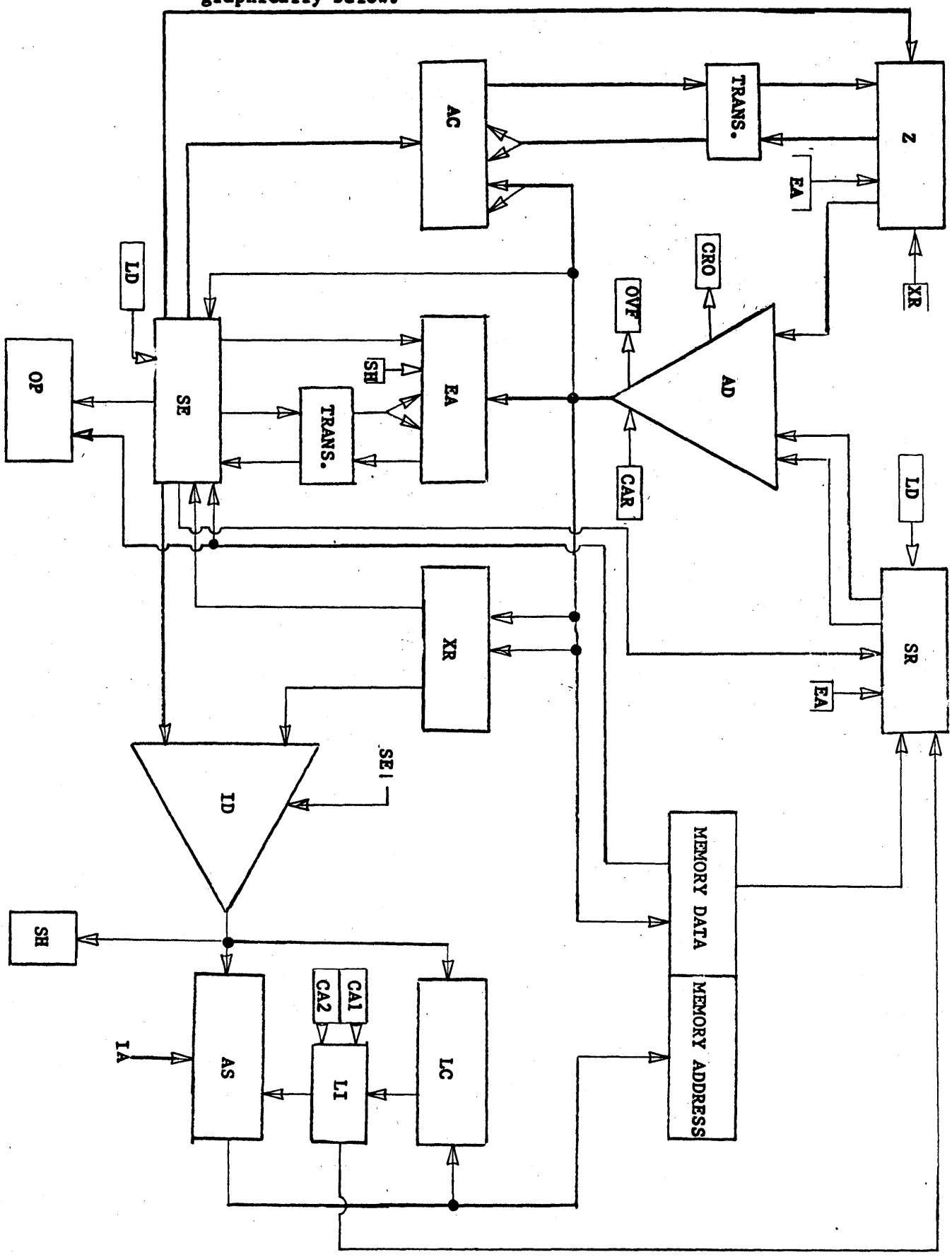


FIGURE 1

III. SCC 670 SERIES COMPUTER DETAILS

The organization of the SCC 670 Series Computer is shown graphically below.



DATA PATHS

The SR Register controls the operation of the computer by analyzing all instructions and furnishing the proper control signals to all of the other registers. The LC Register is referred to as the location counter. It always contains the address of the next instruction to be executed. The M Register is the interface with the memory and holds all data going to or from the memory.

The XR Register is used for indexing and several other special functions. It is a full 24 bit register and the low order bits are used to index other instructions.

The AC Register is referred to as the accumulator and is the main arithmetic register. The EA Register is an auxiliary register which may be treated as an extended accumulator in some operations. In double precision operations, the EA Register will contain the least significant bits of a double precision number.

The B and C Registers are I/O buffer registers which allow overlapped operation of Input/Output and computation. Characters are assembled in these registers for transmission to memory. In addition, a register is available for full word parallel input/output operations.

A. WORD STRUCTURE

Each word in the computer is composed of twenty-four bits, numbered from 0 to 23, from left to right.

Words which represent single precision numeric data use bit 0 as a sign bit. If bit 0 is a "1", the number is negative and is represented in two's complement notation.

Words which represent BCD character data contain four characters per word, with six bits representing each character.

Words which contain machine instructions are in the following format:

I	X	Op-Code	Operand Address
0	1	2 8	9 23

The interpretation placed on the bits is summarized below:

Bit	Meaning
0	Indirect Address Bit: If this bit is a "1", bits 9-23 of the instruction (which may also be indexed) specify the location where the effective operand address of the instruction may be found.
1	Index Bit: If this bit is a "1", the low order 15 bits of the index register are added to bits 9-23 of the instruction to form the effective operand address. If the bit is "0", indexing is not used.
2-8	Op-Code: These bits specify the exact operation to be performed. If bit 2 is a "1", indexing is not used.
9-23	Operand Address: These bits specify the location of the operand, as modified by indexing and indirect addressing considerations mentioned above.

B. PROGRAMMED OPERATORS

Programmed operators are subroutines which simulate machine instructions in that the operation to be performed and the argument for that operation are specified in a single instruction. The automatic linkage and return mechanisms of the computer make programmed operators possible.

Programmed operator entry instructions have a "1" in bit 2 of the instruction. When the computer detects the programmed operator bit, the instruction at the address specified by bits 2-8 of the programmed operator instruction is executed. Execution of a programmed operator entry instruction does not alter the contents of the LC Register, hence a BSL instruction may be placed at the location specified by bits 2-8 of the programmed operator instruction to save the value of the LC and branch to the appropriate subroutine.

Return is accomplished by an instruction which retrieves the stored LC Register value, modifies it by the appropriate amount, and then transfers control to the instruction following the one which caused entry to the programmed operator routine.

C. ITERATIVE INDEXING AND INDIRECT ADDRESSING

In the SCC 670 Series, indexing and indirect addressing are the function of an address and not a function of the instruction.

Iterative indexing - indirect addressing is therefore possible and enhances the programming power of the computer considerably. Indexing takes precedence over indirect addressing, i. e., the address is modified by the index register before indirect addressing takes place. The following is an example of iterative indexing - indirect addressing. The XR Register is assumed to contain the value of 2. An instruction has an operand of 1000, with an indirect address bit and an indexing bit set. Indexing gives an operand address of 1002. Since an indirect address bit is set, location 1002 is assumed to contain the address of the operand. If 1002 contains the address 2000 with indexing and indirect address bits set, then the computer will go to location 2002 to find the operand address. If location 2002 contains the address 500 without indexing or indirect addressing, then the number in location 500 is the actual operand used.

While seemingly complex, this iterative technique offers programming power unavailable in processors where indexing and indirect addressing are associated with the instruction rather than the address.

D. INSTRUCTIONS

This section describes the instruction repertoire of the SCC 670 Series Computer. Instruction description includes mnemonic, computer operation code enclosed in parentheses, instruction name, number of machine cycles required to execute the instruction and machine function performed by the execution. Examples are given, when needed, to clarify the description. In discussing instruction functions the following conventions are used.

(1) The letter "Q" refers to the effective address, i.e., Q refers to the actual address used in the execution of the instruction after all indexing and indirect addressing has been accomplished. In some instructions, Q is not the address of an operand but is itself the operand. When used in this manner, Q is said to be an immediate address.

(2) All numbers, locations, etc. are in octal unless noted otherwise.

(3) Subscription is used to denote bit positions within a register. For example, AC_{9-23} refers to bits 9 through 23 of the AC Register.

(4) A register name or location name enclosed in parentheses denotes the contents of that register or location. For example,(AC) denotes the "contents of the AC" and (Q) refers to the contents of memory at the effective address.

(5) The "N" in the number of cycles column of a shift instruction refers to the number of bit positions shifted.

(6) Each level of indirect addressing adds one cycle time to the number of cycles column of any instruction.

1. Data Transfer Instructions

Mnemonic	Instruction Name	Cycles
LDA (27)	Load AC	2

The contents of memory at the effective address, (Q), replaces the contents of the AC, (AC); the contents of memory at the effective address, (Q), remains unchanged.

LAC (45)	Load AC Complemented	2

The two's complement of the contents of the effective address replaces the contents of the AC. The contents of the effective address remains unchanged by this instruction.

STA (35)	Store AC	2

The contents of the AC, (AC), replaces the contents of memory at the effective address, (Q); the contents of the AC, (AC), remains unchanged.

LDE (75) Load EA 2

The (Q) replaces the (EA); the (Q) remains unchanged.

STE (36) Store EA 2

The (EA) replaces the (Q); the (EA) remains unchanged.

STS (34) Selective Store 3

The "one" bits in the EA are used as a mask to select which bits of the AC are to be stored. Specifically, if $(EA)_i = 1$, the $(AC)_i$ is stored in the $(Q)_i$; if $(EA)_i = 0$ the $(Q)_i$ is unaffected by this instruction.

Example:

	AC	EA	Q
Before Execution	37654227	00007777	62345671
After Execution	37654227	00007777	62344227

LDX (71) Load Index 2

The (Q) replaces the (XR); the (Q) remains unchanged.

LXC (44) Load Index Complement 2

The two's complement of the contents of the effective address replaces the contents of the XR. The contents

of the effective address remains unchanged by this instruction.

STX (37) Store Index 2

The (XR) replaces the (Q); the (XR) remains unchanged.

EAX (77) Effective Address Into Index 1

The $(XR)_{9-23}$ is replaced by Q9-23; the $(XR)_9$ is then copied into $(XR)_{0-8}$. Note that this instruction has a variety of uses, depending upon how indirect addressing and indexing are applied.

Examples:

Instruction		Explanation
EAX	Q	Q is an immediate address and replaces the $(XR)_{9-23}$.
2 EAX	Q	The index register is incremented by Q.

CLA (4600001) Clear AC 1

The (AC) is set to zero.

CLE (4600002) Clear EA 1

The (EA) is set to zero.

CLR (4600003) 1

The (AC) and (EA) are set to zero.

TAE (4600004) Transfer AC to EA 1

The (AC) replaces the (EA). The (AC) remains unchanged by this instruction.

AEC (4600005) Transfer AC to EA, Clear AC 1

The (AC) replaces the (EA). The (AC) is set to zero.

TEA (4600010) Transfer EA to AC 1

The (EA) replaces the (AC). The (EA) remains unchanged by this instruction.

EAC (4600012) Transfer EA to AC, Clear EA 1

The (EA) replaces the (AC), the (EA) is then cleared.

XAE (4600014) Exchange AC and EA 1

The (AC) replaces the (EA); the (EA) replaces the (AC).

TEX (4600020) Transfer EA to XR 1

The (EA) replaces the (XR). The (EA) remains unchanged by this instruction.

TXE (4600040) Transfer XR to EA 1

The (XR) replaces the (EA). The (XR) remains unchanged by this instruction.

XXE (4600060) Exchange XR and EA 1

The (XR) replaces the (EA); the (EA) replaces the (XR).

SXP (4600122) Store Exponent 1

The $(EA)_{15-23}$ is transferred to the $(XR)_{15-23}$. The sign of the exponent, $(XR)_{15}$, is copied into $(XR)_{0-14}$. The $(EA)_{15-23}$ is then cleared.

Example:

	EA	XR
Before Execution	36254025	77777736
After Execution	36254000	00000025

LXP (4600140) Load Exponent 1

The $(XR)_{15-23}$ replaces the $(EA)_{15-23}$. The (XR) and the $(EA)_{0-14}$ are unaffected by this instruction.

Example:

	EA	XR
Before Execution	36254025	77777736
After Execution	36254736	77777736

XXP (4600160) Exchange Exponents 1

The $(XR)_{15-23}$ replaces the $(EA)_{15-23}$; the $(EA)_{15-23}$ replaces $(XR)_{15-23}$. The new sign of the exponent, $(XR)_{15}$, is copied into $(XR)_{0-14}$.

Example:

	EA	XR
Before Execution	36254025	77777736
After Execution	36254736	00000025

TXA (4600200) Transfer XR to AC 1

The (XR) replaces the (AC). The (XR) remains unchanged by this instruction.

TAX (4600400) Transfer AC to XR 1

The (AC) replaces the (XR). The (AC) remains unchanged by this instruction.

XXA (4600600) Exchange XR and AC 1

The (AC) replaces the (XR); the (XR) replaces the (AC).

NGA (4601000) Negate AC 1

The two's complement of the (AC) replaces the (AC).

XAM (62) Exchange AC and Memory 3

The (AC) replaces the (Q); the (Q) replaces the (AC).

2. Arithmetic Instructions

Mnemonic	Instruction Name	Cycles
ADD (55)	Add	2

The contents of the effective address are added to the (AC); their sum replaces the (AC). A carry from bit position 0 of the 24-bit adder sets the carry flip-flop. If both numbers are of the same sign but the sign of the result is different, the overflow flip-flop is set.

Examples:

	AC	Q	OVERFLOW	CAR
Before Execution	36534724	32467345	OFF	0
After Execution	71224271	32467345	ON	0

	AC	Q	OVERFLOW	CAR
Before Execution	47236411	67724415	OFF	0
After Execution	37163026	67724415	ON	1

ADC (57) Add with Carry 2

ADC is the same as ADD, except that the contents of the carry flip-flop is added to the sum.

Example:

	AC	Q	OVERFLOW	CAR
Before Execution	36534724	32467345	OFF	1
After Execution	71224272	32467345	ON	0
Before Execution	47236411	67724415	OFF	1
After Execution	37163026	67724415	ON	1

AAM (63) Add AC to Memory 3

The (AC) is added to the contents of the effective address; the sum replaces the contents of the effective address. As in ADD, overflow is possible and turns on the overflow indicator. The (AC) remains unchanged by this instruction.

SUB (54) Subtract 2

The contents of the effective address is subtracted from the (AC), the difference replaces the (AC). A carry from bit 0 of the 24-bit adder sets the carry flip-flop. If the two numbers have opposite signs, their difference may exceed the capacity of the AC. In this case, the overflow

indicator is set, the difference in the AC is incorrect.

Example:

	AC	Q	OVERFLOW	CAR
Before Execution	47236511	10053463	OFF	0
After Execution	37163026	10053463	ON	1

SBC (56) Subtract with carry 2

The subtract with carry instruction is identical to the subtract instruction if the carry flip-flop is a "one" at the start of the instruction. If the carry flip-flop is zero, the one's complement of the subtrahend is added to the contents of the AC.

Example:

	AC	Q	OVERFLOW	CAR
Before Execution	47236511	10053463	OFF	0
After Execution	37163025	10053463	ON	1

MPY (64) Multiply 14

The (AC) is multiplied by the contents of the contents of the effective address to form a 46 bit product. The sign of the product is in bit 0 of the AC. The most significant

part of the product is in the AC; the least significant part is in the EA. The $(EA)_{23}$ is not significant.

If both numbers have the value 40000000_8, overflow occurs and the computer sets the overflow indicator. In this case, the product is set to zero.

Example:

	AC	EA	Q
Before Execution	00000062	--	00000007
After Execution	00000000	00001274	00000007

DIV (65) Divide 28

The contents of the AC and EA are treated as a double-precision dividend and the contents of the effective address as a 24-bit divisor. The quotient appears in the AC and the remainder in the EA. The sign of the remainder in the EA is the same as the sign of the original numerator.

Division takes place normally if $-1 \leq \frac{(AC, EA)}{(Q)} < 1$, where (Q) denotes the contents of the effective address. If the quotient exceeds these boundaries, overflow occurs and the overflow indicator is turned on. If overflow occurs the result is incorrect.

Example: 537 scaled at binary 46 divided by 7 scaled at binary 23.

	AC	EA	Q
Before Execution	00000000	00001276	0000007
Afeter Execution	00000062	00000001	0000007

MDC (60) Memory Decrement 3

One is subtracted from the contents of the effective address, the difference replaces the contents of the effective address. As in subtract overflow can occur and turns on the overflow indicator. The other registers are not affected by this instruction.

MIN (61) Memory Increment 3

One is added to the contents of the effective address; the sum replaces the contents of the effective address. The other registers are not affected by this instruction. As in ADD overflow can occur and turns on the overflow indicator.

3. Logic Instructions

Mnemonic	Instruction Name	Cycles
AND (14)	AND to the Accumulator	2

Perform a logical "AND" of the contents of the effective address and the contents of the AC; place the result in the AC. The (Q) remains unchanged.

Example:

	AC	Q
Before Execution	43765221	00037777
After Execution	00025221	00037777

ORA (16) OR to the Accumulator 2

Perform a logical "inclusive OR" of the contents of the effective address and the contents of the AC; place the result in the AC. The (Q) remains unchanged.

Example:

	AC	Q
Before Execution	73217654	76320153
After Execution	77337757	76320153

EOR (17) Exclusive OR to the Accumulator 2

Perform a logical "exclusive OR" of the contents of the effective address and the contents of the AC; place the result in the AC. The (Q) is not affected by this instruction.

Examples:

	AC	Q
Before Execution	23417663	23417663
After Execution	00000000	23417663
Before Execution	23417663	77777777
After Execution	54360114	77777777

From the first example, it can be seen that if the AC and Q contain the same bit configurations, the (AC) is set to zero by the instruction. The second example shows how this instruction may be used to complement a number.

4. Branch Instructions

Mnemonic	Instruction Name	Cycles
BRA (01)	Branch	1

Branch to the location specified by the effective address.

BIX (41) Increment Index and Branch 1

Add one to the contents of the XR. If the index register is negative, i.e., $(XR)_8 = 1$, branch to the effective address; if the index register is positive, the computer takes the next sequential instruction.

Example:

			XR BEFORE EXECUTION	XR AFTER EXECUTION	NEXT INSTRUCTION TAKEN FROM
100	BIX	300	00000355	00000356	101
100	BIX	300	00100355	00100356	300

BSL (43) Store Location Counter and Branch 2

Store the LC Register in the location specified by effective address. Store the contents of the overflow indicator in bit 3 of the location specified by the effective address, turn off the overflow indicator, and branch to the effective address plus one.

Example:

LOC	INSTRUCTION			
100	BSL	300		
			OV	LOC 300
Before Execution			1	---
After Execution			0	04000100

The next instruction would be taken from location 301. This instruction, along with the BRT, permits efficient linkages between two programs.

BAN (24) Branch on AC negative 1

The contents of bit 0 of the AC is examined; if the $(AC)_0$ is a one the computer branches to the location specified by the effective address. If the $(AC)_0$ is a zero, the computer takes the next sequential instruction.

BRT (51) Return Branch 2

The low order 15 bits of the effective address replace the contents of the LC register; the location counter is then incremented by one. An "inclusive OR" is performed on bit 3 of the effective address and the overflow indicator; the result is placed in the overflow indicator. This instruction is used as the return linkage in subroutines entered via the BSL instruction.

Example:

LOC	CONTENTS
300	04000100
500	BRT 300

Execution of the instruction at location 500 would cause the overflow indicator to be set (if it was OFF) and the next instruction would be taken from location 101.

BAZ (25) Branch on AC Zero 1

The contents of the AC is examined. If the $(AC)_{0-23} = 0$, the computer branches to the location specified by the effective address. If the $(AC)_{0-23} \neq 0$, the computer takes the next sequential instruction.

BAP (26) Branch on AC Positive 1

The contents of bit 0 of the AC is examined; if the $(AC)_0$ is a zero, the computer branches to the location specified by the effective address. If the $(AC)_0$ is a one, the computer takes the next sequential instruction.

BNZ (15) Branch on AC Non Zero 1

The contents of the AC is examined. If the $(AC)_{0-23} \neq 0$, the computer branches to the location specified by the effective address. If the $(AC)_{0-23} = 0$, the computer takes the next sequential instruction.

5. Skip Instructions

Mnemonic	Instruction Name	Cycles
SNS (40)	Skip if Signal Not Set	1

If the signal indicated by the code in bits 10-23 of the instruction is set ($\neq 0$), the computer takes the next sequential instruction. If the signal is not set ($= 0$), the computer skips the next sequential instruction. Indexing and indirect addressing are not applicable to this instruction.

Example:

		REMARKS
100 SNS	20400	Skip if BP #1 is not set.
101		Control comes to here if BP is set.
102		Control comes to here if BP is not set.

This instruction has the capability of testing 2^{13} different signals. It is used for testing signals both internal and external to the computer. The type of signal to be tested is determined by address bits 10 and 11. These bits are interpreted as follows:

10	11	
0	1	Test the input/output device selected by address bits 18-23.

10	11	
1	0	Test the internal computer condition specified by address bits 12-23.
1	1	Test the external signal selected by address bits 12-23.

(1) Type one, SNS 1XXXX — Skip if STS_1 (See Section VI-E) not set.

(2) Type two, SNS 2XXXX

BIT POSITION	INTERPRETATION
12	**Not Used**
13	Skip if C Buffer ready
14	Skip if B Buffer ready
15	Skip if Breakpoint 1 off
16	Skip if Breakpoint 2 off
17	Skip if Breakpoint 3 off
18	Skip if Breakpoint 4 off
19	Skip if C Buffer error off
20	Skip if B Buffer error off
21	Skip if Interrupt Enabled
22	Skip if Overflow Indicator off and turn off Overflow Indicator

(3) Type three, SNS 3XXXX — Skip if STS_2 (See Section VI-E) is not set.

Bits 12-23 may be assigned as needed to test external signals.

SAG (73) Skip if AC greater 3

The contents of the AC is compared to the contents of the effective address. If the contents of the AC is algebraically greater than the contents of the effective address, the next instruction is skipped. If (AC) is less than or equal to the (Q), the next sequential instruction is taken. The (AC) and (Q) remain unchanged by this instruction.

SMN (53) Skip if Memory Negative 2

The contents of the effective address is examined. If the contents is negative, i.e., $(Q)_0 = 1$, skip the next instruction. If the operand is positive, the next sequential instruction is taken. The operand is unchanged by this instruction.

SAE (70) Skip if AC equals M on EA Mask 3

For each bit position in the EA that is a one, compare the corresponding bits of the AC and the operand. If the accumulator and the operand are equal in these bit positions, skip the next instruction. If the two fail to compare in any of these bit positions, the computer executes the next sequential instruction. The (AC), (EA), and (Q) are unaffected by this instruction.

Example:

If the (AC) = 37652121, the (Q) = 76652121 and (EA) = 77777777, the execution of an SAE would cause all 24

bits of AC and Q to be compared. Since the two do not compare in bits 0 and 5 no skip would occur. However, if (EA) = 00777777, an SAE instruction would compare $(AC)_{6-23}$ with $(Q)_{6-23}$ and the skip would occur.

SAM (72) Skip if AC and M do not compare ones 2

The contents of the AC and the contents of the effective address are compared bit by bit. If a logical "AND" performed on the (Q) and the (AC) would have a zero result, the computer skips the next instruction. If a logical "AND" would produce a one bit in any bit position, the computer takes the next sequential instruction. The (AC) and (Q) are not affected by this instruction.

Examples:

Q	EXPLANATION
00000001	Skip if (AC) is even
77777777	Skip if (AC) = 0
40000000	Skip if (AC) positive
40000001	Skip if (AC) positive and even

SEM (52) Skip if EA and Memory Do Not Compare Ones 2

This instruction is identical to SAM, except that the EA is used instead of the AC.

SAQ (50) Skip if AC Equals Memory 2

The contents of the AC are compared with the contents of the effective address. If the quantities compared are equal, the computer skips the next sequential instruction. If the quantities are unequal, the computer executes the next sequential instruction.

SDE (74)	Difference Exponents and Skip	3

The contents of bits 15-23 of the effective address are subtracted from the $(EA)_{15-23}$; the absolute value of the difference is placed in the $(XR)_{15-23}$ and $(XR)_{0-14}$ is set to zero. If the contents of the least significant 9 bits of the effective address is less than or equal to the corresponding bits of the EA, the computer executes the next sequential instruction. Otherwise the computer skips the next sequential instruction.

Example:

If the (EA) = 36542046 and the (Q) = 25367025, the execution of an SDE instruction would place 00000021 in the XR and the computer would execute the next sequential instruction.

6. Shift Instructions

Mnemonic	Instruction Name	Cycles
RSH (66000XX)	Right Shift	$2 + \frac{N}{8}$

Shift the contents of the AC and the contents of the EA right the number of places specified by the six low-order bits of the effective address. Bits shifted past AC_{23}

enter EA_0. Bits shifted out of the EA are lost. Bit zero of the AC is not shifted but is copied into vacated bit positions.

Example:

RSH 00006

	AC	EA
Before Execution	36722154	67261251
After Execution	00367221	54672612
Before Execution	66722154	62261251
After Execution	77667221	54622612

RCY (66200XX) Right Cycle $2 + \frac{N}{8}$

Rotate the contents of the AC and the contents of the EA right the number of places specified by the six low-order bits of the effective address. Bits shifted past AC_{23} enter EA_0; bits shifted past EA_{23} enter AC_0. No information is lost during the execution of this instruction.

Example:

RCY 00006

	AC	EA
Before Execution	66722154	67261251
After Execution	51667221	54672612

LSH (67000XX) Left Shift $2 + \frac{N}{8}$

Shift the contents of the AC and the contents of the EA left the number of places specified by the six low-order bits of the effective address. Bits shifted past EA_0

enter AC_{23}; bits shifted past AC_0 are lost. A shift which causes the sign of the AC to change turns on the overflow indicator.

Example:

LSH 00006

	AC	EA	OV
Before Execution	00365422	32765472	OFF
After Execution	36542232	76547200	OFF
Before Execution	01665422	32765472	OFF
After Execution	66542232	76547200	ON

LCY (67200XX) Left Cycle $2 + \frac{N}{8}$

Rotate the contents of the AC and the contents of the EA left the number of places specified by the six low-order bits of the effective address. Bits shifted past EA_0 enter AC_{23}; bits shifted past AC_0 enter EA_{23}. No information is lost during the execution of this instruction. The overflow indicator is unaffected by the instruction.

Example:

LCY 00006

	AC	EA
Before Execution	37552761	77777777
After Execution	55276177	77777737

NDX (67100XX) Normalize and Decrement Index $2 + \frac{N}{8}$

Shift the contents of the AC and the contents of the EA left until bit one of the AC is opposite to the sign of the AC, i.e., (AC) $\geq$ 1/2, or until the number of positions

specified by the six low-order bits of the effective address have been shifted. The contents of the index register is decremented by one for each bit position shifted. Bits shifted past EA_0 enter AC_{23}.

Examples:

NDX 00014

	AC	EA	XR
Before Execution	77327622	72553216	00000000
After Execution	55371135	26550700	77777773
Before Execution	00001276	57362314	00000000
After Execution	12765736	23140000	77777764

In the first example, the number in the AC was normalized after five shifts; in the second example, the count was exhausted before the number was normalized. The latter example emphasizes the fact that the programmer must give an initial count large enough to insure that normalization will take place.

7. Miscellaneous Instructions

Mnemonic	Instruction Name	Cycles
HLT (00)	Halt	1

The computer halts awaiting manual intervention from the console. If the computer is in the run mode, the HLT instruction steps the location counter before halting, hence the LC displays the location of the halt plus one.

NOP	(20)	No Operation	1

No operation is performed.

XEC	(23)	Execute	1 + time of instruction executed

Execute the instruction whose location is determined by the effective address. This instruction does not alter the location counter unless the instruction it executes changes the location counter. If a skip instruction is executed, the skip occurs relative to the XEC instruction.

Example:

```
LOC Instruction
100 XEC 200
 ⋮
200 ADD 300
```

Execution of the XEC instruction at location 100 would cause the following sequence of events:

(1) The ADD instruction at location 200 is brought to the SR; the LC is unchanged.

(2) The instruction "ADD 300" is executed.

(3) The LC is incremented by one and the next instruction is taken from location 101.

8. Input/Output Instructions

Mnemonic		Instruction Name	Cycles
TMB	(12)	Transfer Memory to B	2+Wait

The contents of the effective address is transferred to the B buffer. If the B buffer is not ready, the

instruction causes the computer to "lock up" or wait until the B buffer is ready to accept new information.

TMC (10) Transfer Memory to C 2+Wait

The contents of the effective address is transferred to the C buffer. If the C buffer is not ready, this instruction causes the computer to "lock up" or wait until the C buffer is ready to accept new information.

WTP (13) Write Parallel 2+Wait

The contents of the effective address is brought to the SR register and held, awaiting transfer to an external device. The computer locks up until the external device signals it is ready to accept new information. This instruction allows up to 24 bits to be transmitted in parallel to an external device.

TBM (32) Transfer B to Memory 2+Wait

The contents of the B buffer is transferred to the memory location specified by the effective address. The B buffer is cleared and is ready for reloading by the external device. If the B buffer is not loaded, this instruction causes the computer to "lock up" or wait until the B buffer signals that it is loaded.

TCM (30) Transfer C to Memory 2+Wait

The contents of the C buffer is transferred to the memory location specified by the effective address.

The C buffer is cleared and is ready for reloading by the external device. If the C buffer is not loaded, this instruction causes the computer to "lock up" or wait until the C buffer signals that it is loaded.

RDP (33) Read Parallel 2+Wait

Twenty-four parallel bits are input into the contents of the memory location specified by the effective address. The computer locks up until the external device signals that it has up to 24 bits of information ready to be entered into the computer.

ACT (02) Activate 1

This instruction performs four major functions:

Controlling the I/O buffers
Controlling I/O units
Setting internal computer conditions
Controlling special external devices

Address bits 10 and 11 of the instruction determine the function to be performed. These bits are interpreted as follows:

Bit 10	Bit 11	Function
0	0	Buffer control
0	1	I/O unit control
1	0	Set internal computer conditions
1	1	Special external device control

(1) Input/output buffer control - ACT 0XXXX

The primary purpose of this instruction is to connect input/output units to the B or C buffer. Address bits 12-23 of the instruction are interpreted as follows:

Bit Position	Interpretation
9	1 = Enable interlace
12	1 = Reverse 0 = Forward
13	1 = Accept or generate tape without leader 0 = Accept or generate tape with leader
14	1 = Binary (odd parity) 0 = BCD (even parity)
15-16	No. characters per buffer load: 00 = 1 character 01 = 2 characters 10 = 3 characters 11 = 4 characters
17	0 = B buffer 1 = C buffer
18-23	Unit address code: 00 = Disconnect 01 = Typewriter input 04 = Paper Tape input 41 = Typewriter output 44 = Paper tape punch output

Examples:

Instruction	Explanation
ACT 01004	Start paper tape reader 1, connect the B buffer to the paper tape reader, set B buffer to accept binary input data, and assemble 1 character per word.
ACT 01644	Start paper tape punch 1, connect the B buffer to the paper tape punch, set B buffer to output binary data, generate approximately one (1) inch of leader, output 4 characters per word.
ACT 01641	Select typewriter 1, connect the B buffer to the typewriter, set B buffer to output mode, output 4 characters per word.

(2) Input/Output Unit Control - ACT 1XXXX

This type activate instruction controls the various input/output units which can be connected to the computer. The interpretation of address bits 12 - 16 is dependent upon the type of device being used.

Bit Position	Interpretation
9	1 = Enable interlace
12	1 = Control Function No. 1 (When bits 18 - 23 are zeros bit 12 terminates outputs.)
13-16	1 = Control functions 2 through 5, respectively
17	0 = B buffer control 1 = C buffer control
18-23	Unit address code

(3) Setting Internal Computer Conditions - ACT 2XXXX

The address portion of the instruction selects the internal computer condition to be set. The address bits are interpreted as follows:

Bit Position	Interpretation
12-20	Not used
21	Disable Interrupt System
22	Enable Interrupt System
23	Turn off overflow indicator

(4) Special External Device Control - ACT 3XXXX

The address bits of this instruction may be assigned as required to control a special external device. Indexing and indirect addressing are not applicable to this instruction.

E. INPUT/OUTPUT OPERATION

In general, I/O operations are initiated by giving an ACT, or activate command, to the specific device involved. Data is then transferred between memory and the register involved by the appropriate command until a gap or end of record signal is received or transmitted to the device. The status of the buffers involved in the transfer may be determined by using appropriate SNS (Skip if Signal Not Set) commands.

Automatic parity is provided on all input/output operations. In addition, the user may specify whether an I/O operation is to use 6, 12, 18 or 24 bits per word, thus permitting considerable flexibility.

The SCC 670 series computers contain priority interrupt channels which may be enabled or disabled under program control. If the interrupt system is enabled, then the completion of any input/output operation will generate a priority interrupt. An interrupt on a channel will cause

a transfer of control to a fixed location associated with that channel. The LC Register is not disturbed by the transfer to the fixed location, therefore, a subroutine may be initiated which can later return to the point of interrupt.

F. INPUT/OUTPUT BUFFER OPERATION

The B buffer is the primary input/output register on the SCC 670 series computers. The C buffer is an optional input/output register identical to the B buffer. Any comments pertaining to the B buffer below may be applied equally well to the C buffer.

Input

When operating in the input mode, the B buffer mechanism accepts a 6-bit character plus a character parity bit in parallel from an external device. The parity bit is checked, but is not stored in the buffer; a parity error turns on the ERROR flip-flop on the computer console. The B buffer transmits words to memory in one, two, three or four characters per word modes. The number of characters per word is selected by the activate instruction.

After the number of characters per word has been selected, characters are assembled in the buffer until the number of characters in the buffer equals the count set by the activate instruction. At this time the B buffer signals the computer

that it is ready to transmit a "word" of information to memory. If interrupts are enabled, the ready signal is in the form of an interrupt to location 31. If interrupts are inhibited, the computer will "lock up" on a TBM instruction until the buffer loaded signal is received from the buffer.

When the input device is paper tape or magnetic tape, an end-of-record gap causes the buffer to disconnect from the input device. If the number of characters in the record was not a multiple of the number of characters per word, zeros are forced into the least significant bits of the buffer before the computer is signalled that the buffer is loaded. If the number of characters in the record was multiple of the number of characters per word, the B buffer will contain the last "word" of the record if a TBM instruction has not been given for this word or zeros if a TBM has transferred the word to memory. In either situation, a TBM instruction may be given after disconnect to insure that the last word was received.

Output

Output through the B buffer is quite similar to input. When operating in the output mode, the B buffer mechanism transmits a six bit character plus a character parity bit in parallel to an external device. The parity bit is generated by the B buffer

mechanism. The B buffer receives words from memory in one, two, three or four characters per word modes. Characters are assembled in the B buffer from left to right; for example, when operating in the one character per word mode, the 6 high order bits of the memory word referenced are brought to the buffer.

When the contents of the buffer have been transmitted to an external device, the B buffer signals the computer that it is ready for reloading from memory. If interrupts are enabled, the ready signal is in the form of an interrupt to location 31. If interrupts are inhibited, the computer will "lock up" on a TMB instruction until the ready signal is received.

Input/Output Termination

In addition to the end-of-record disconnect, inputs may be terminated by the buffer disconnect instruction, ACT 00000. The disconnect instruction causes the external device to be disconnected immediately. An interrupt is generated by an end-of-record gap; the disconnect instruction does not generate an interrupt.

Outputs may be terminated with either the disconnect instruction, ACT 00000, or the terminate output instruction, ACT 14000. The disconnect instruction causes the external device to be disconnected immediately, regardless of whether or not the last group of characters in the buffer has been transmitted to the

external device. The terminate output instruction allows the last group of characters in the B buffer to be transmitted to the external device before disconnecting. The terminate output instruction generates an interrupt after the disconnect takes place. In the case of magnetic tape, an end-of-record gap is generated before the unit is disconnected. As with input, the disconnect instruction does not cause an interrupt.

If interrupts are disabled, completion of input/output operations may be detected using the instruction to test if the B buffer is ready (SNS 21000). The status of the input/output error flip-flop may be tested using the instruction SNS 20010.

Information Loss

Incorrect use of the input/output buffer may result in loss of information during both input and output. When inputting information from an external device such as magnetic tape, the programmer has the responsibility of clearing the B buffer with a TMB instruction between successive buffer loads by the external device. If the buffer is full at the time the next character arrives from the input device, that character is lost and the input/output error light is turned on.

An analogous situation occurs during output. If the B buffer has not been loaded with a word from memory via the TMB instruction at the time the output tape requests information from the buffer, a blank space will be generated on the tape.

Buffer Interlace System

The buffer interlace system, which is optional equipment, includes the B buffer and a 30-bit interlace register. The interlace system relieves the programmer of the chores of address modification and counting during an input/output operation. The interlace register is divided into two parts: a 15-bit counter and a 15-bit address. The counter allows blocks of up to 32,767 words to be transmitted between memory and an external device. The 15-bit address allows any location in memory to be referenced.

The interlace system is enabled with the activate instruction. The enabling activate instruction may be followed by another activate instruction which sets the six high order count bits of the interlace register. The activate instruction(s) above are immediately followed by a write parallel instruction, WTP, which sets the low order 24 bits of the interlace register. The WTP instruction loads the interlace register with the word in memory specifed by the effective address. The memory word loaded into the interlace register contains the 9 low order count bits in bits 0 - 8 and the interlace register address bits in bits 9 - 23.

After the interlace system has been enabled and the external device started, characters are assembled in the B buffer as

described above. However, when the B buffer is loaded, interrupt does not occur; instead, the computer locks up until the contents of the B buffer are stored in the memory location specified by the interlace register address. The address portion of the interlace register is incremented by one and the count portion is decremented by one.

The operation of the interlace system is independent of the other computer operations and occurs simultaneously with these operations. Termination of an input/output operation disables the buffer interlace system. Output is terminated when the count in the interlace register reaches zero. Input is terminated if the count in the interlace register reaches zero or if an end-of-record gap is encountered.

Example:

Paper Tape Input

The following program reads one record from a binary (odd parity) paper tape. The record is read into memory starting at location 5000.

Location	Instruction	Comments
100	SNS 21000	Is B buffer in use?
101	BRA 00100	Yes, wait until through
102	ACT 20004	Disable interrupts
103	LDX 00113	Load XR with starting address
104	ACT 01604	Fire up paper tape reader
105	2 TBM 00000	Store B buffer in memory
106	SNS 21000	End of record (disconnect)?
107	BIX 00105	No, increment index, go to read next word
110	SNS 20010	Yes. Input/output error?
111	BRA ERROR	Yes, branch to error routine
112	BRA CONTINUE	No, continue rest of program
113	**001 05000**	Starting location for inputs

Explanation

The instructions at location 100 - 101 form a wait loop. As long as the B buffer is being used for some other input/output

operation, the computer will stay in the loop. When any pending input/output operations are complete, the B buffer becomes ready and the computer drops out of the wait loop. The activate instruction at location 102 disables priority interrupts. At location 103, the index register is loaded with the first word address (5000) of the block of memory into which the record is to be read. Bit 8 of the starting address constant stored in location 113 is set so that the constant is considered negative by the index register. The activate instruction at location 104 starts the reader and sets the B buffer to accept four characters per word. The computer locks up on the TBM instruction at location 105 until the B buffer is loaded from the paper tape reader. At this time the TBM stores the word assembled in the next word of the input block. Note that this instruction is indexed, hence the first word is stored in location 5000. At location 106 the SNS instruction tests for an end-of-record gap (disconnect). If the end-of-record gap has not been encountered, the computer executes the BIX instruction at location 107. Here, the contents of the index register are incremented by one; the branch to location 105 always takes place since the XR was loaded with a one in bit 8. When the end-of-record gap is encountered, the buffer disconnects. The SNS instruction at location 106 will now cause the computer to skip to location 110. At location 110,

the input/output error indicator is tested. If the error indicator was on, the program branches to an error routine; if the error indicator was off, the computer takes its next instruction from location 112 which contains a branch to the rest of the program.

G. PARALLEL INPUT/OUTPUT

The SCC 670 series computers have the capability of transmitting or receiving up to 24 bits of parallel data from an external device. The computer communicates with the external device through the read parallel, RDP, and write parallel, WTP, instructions. For output, the memory word to be transmitted is brought to the SR register and held, awaiting transfer to the external device. Input data does not require the SR register and is transferred directly to memory.

The computer "locks up" on parallel operations until the external device presents a signal to the computer indicating that the external device is ready for the transfer. If a particular application requires compute - I/O overlap, the ready signal from the external device may be used to generate an interrupt to the computer.

The "lock up" mode of the parallel operation is useful when a high information transfer rate is required. For example, an indexed RDP or WTP instruction followed by a BIX instruction

may be used to achieve a transfer rate of 166,666 words per second on the SCC 670-2.

H. PRIORITY INTERRUPTS

Each SCC 670 series computer has two (2) interrupt channels as standard equipment. These two interrupt channels are associated with the B buffer. Up to 62 additional priority interrupt channels may be added as optional equipment. Each interrupt channel is numbered. The number assigned to a channel is determined by the location to which the computer branches when interrupted by the given channel. For example, an interrupt on channel 30 causes the computer to branch to location 30. The interrupt channels are assigned as follows:

Channel Number	Function
30	C buffer end-of-word (optional)
31	B buffer end-of-word
32	C buffer end-of-record (optional)
33	B buffer end-of-record

There are two flip-flops associated with each interrupt channel. These flip-flops indicate the status of the corresponding interrupt channel.

RFF	PFF	Status
0	0	Interrupt inactive
1	0	Interrupt requested but not being processed (Waiting)
1	1	Interrupt requested and being processed (Active)
0	1	Invalid bit configuration

Interrupts are honored at the end of the instruction being executed when the interrupt request was received. The waiting status may exist if interrupts are disabled, if the computer is processing a higher priority interrupt, or if the computer is executing a long-duration instruction when the interrupt request is received.

When an interrupt causes the computer to execute the location specified by the channel number, the contents of the location counter is not altered. A BSL instruction may then be used to save the address of the interrupted program.

When an interrupt for a given channel is being processed, a higher priority channel may interrupt the processing subroutine. If this situation occurs, the interrupted channel remains in the active status. After the higher priority interrupt has been processed, return is made to the subroutine processing the original interrupt. If an interrupt on a given

channel is being processed, any new interrupt requests to that channel will not be honored, i.e., no "trap" will occur.

Interrupts are cleared and the interrupt channel flip-flops are reset by an indirectly addressed unconditional branch instruction.

The lower number channels have higher priority, hence given two channels, M and N; M>N implies priority N > priority M.

Example:

Paper Tape Input

The following program inputs one record from paper tape. The B buffer interrupt channels are used to synchronize word inputs and to detect the end-of-record gap. The program assumes that interrupts are enabled.

Location	Instruction	Comments
31	BSL 00107	End-of-word interrupt entry
33	BSL 00113	End-of-record interrupt entry
100	_____	Initialization subprogram
101	LDE 00117	Load EA with mask
102	STS 00110	Store the first word address of input block
103	SNS 21000	Wait on any pending
104	BRA 00103	I/O operation

Location	Instruction	Comments
105	ACT 01604	Start paper tape reader
106	BRT 00100	Return to calling program
107	_____	
110	TBM **	Store next input word
111	MIN 00110	Increment address
112	BRA I 00107	Clear interrupt, return to interrupted program
113	_____	
114	SNS 20010	I/O error?
115	BRA ERROR	Yes, branch to error routine
116	BRA I 00113	Clear interrupt, return to interrupted program
117	000 77777	Mask

Explanation

The section of the program in locations 100 - 106 is the initialization section. This section is entered via a BSL instruction in the calling program. The desired starting address of the input record is assumed to be in the AC when the section is entered. Initialization consists of storing the starting address, $(AC)_{9-23}$, in location 110 and starting the paper tape reader. Return is then made to the calling program.

The section of the program in location 107 - 112 is the end-of-word interrupt subprogram. This subprogram stores the contents of the B buffer in memory and increments the address of location 110. Location 110 is now set to read a word into the next available location of the input block. The indirectly addressed BRA instruction clears the interrupt and returns to the interrupted program.

The subprogram in locations 113 - 116 is entered when an end-of-record gap is encountered. This subprogram tests for an input error. Detection of an error causes the computer to branch to an error routine. If no errors are detected, the indirectly addressed BRA instruction clears the interrupt and branches to the interrupted program.

I. CONTROL CONSOLE

The SCC 670 series computers are equipped with a control console to permit manual control of the operations of the 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.

1. Console Switches

Switch	Function
POWER	Controls the application of power to the computer.
START	This switch initializes the computer for processing. Depressing the switch clears the LC register, resets the B buffer logic and clears the associated I/O indicators, and resets the overflow indicator.
LOAD	This switch initiates the computer for loading a bootstrap or self-loading program. Lowering and raising this switch causes the following functions to be performed: (1) One instruction is read from the paper tape reader into location 2. (2) The index register is set to a minus 7. (3) The instruction at location 2 is executed.

The following is an example of a bootstrap program which may be loaded and executed by lowering and raising the LOAD switch:

Location	Instruction	Explanation
0002	2 TBM 00012	Read next instruction from tape
0003	BIX 00002	
0004	LDX 00011	Load index with starting address of program
0005	2 TBM 00000	
0006	SNS 21000	Test for EOR gap

Switch	Function

LOAD (continued)

Location	Instruction	Explanation
0007	BIX 00005	
0010	_____	Instruction executed after EOR gap is detected.
0011	_____	Starting address with a "one" in bit position 8.

The standard procedure for loading a bootstrap program is given below:

(1) Set the RUN switch to run

(2) Depress the START switch

(3) Lower and raise the LOAD switch

HOLD	Prevents the LC register from being incremented, allowing manual insertion of instructions.
RUN	Setting this switch puts the computer in the RUN mode.
STEP	Each time this momentary switch is depressed, the computer executes a single instruction.
INTERRUPT ENABLE	If this switch is set to ENABLE, the interrupt system is enabled, regardless of the program control.
BREAKPOINT	These four (4) switches do not control the processor, but may be interrogated by a program by using the SNS command.
REGISTER	This switch selects either the AC, EA, SR or XR register to be displayed or entered.

Switch	Function
CLEAR	This switch clears the register selected by the REGISTER switch to zeros.
REGISTER SET	This is a group of 24 switches located below the register display lights. Depressing any one of the switches places a one bit in the corresponding position of the selected register.

2. Console Displays

Display	Function
PROGRAM LOCATION	Displays the 15-bit LC register.
REGISTER	Displays the 24-bit register selected by the REGISTER switch.
OVERFLOW	This indicator is lit if an overflow has occurred.
HALT	This indicator is lit if the computer is in idle status.
ERROR	This indicator is lit whenever an I/O error occurs.
INTERRUPT ENABLED	This indicator is lit whenever the interrupt system is enabled.
INPUT/OUTPUT	These six (6) indicators show the I/O unit selected by the B buffer.

IV. INTERFACE CONNECTIONS

A. LOGIC LEVELS

All logic voltages from and to the computer are as follows:

Output

Logic "One"	+ 8 ± 2 VDC
Logic "Zero"	0 V to + .6 VDC

Input

Logic "One	+ 7.0 V to + 20V
Logic "Zero"	+ 1.0 V to - 1.0 V

B. INPUT/OUTPUT CONNECTORS

Connection into and out of the computer is made through three functional types of connectors. These are as follows:

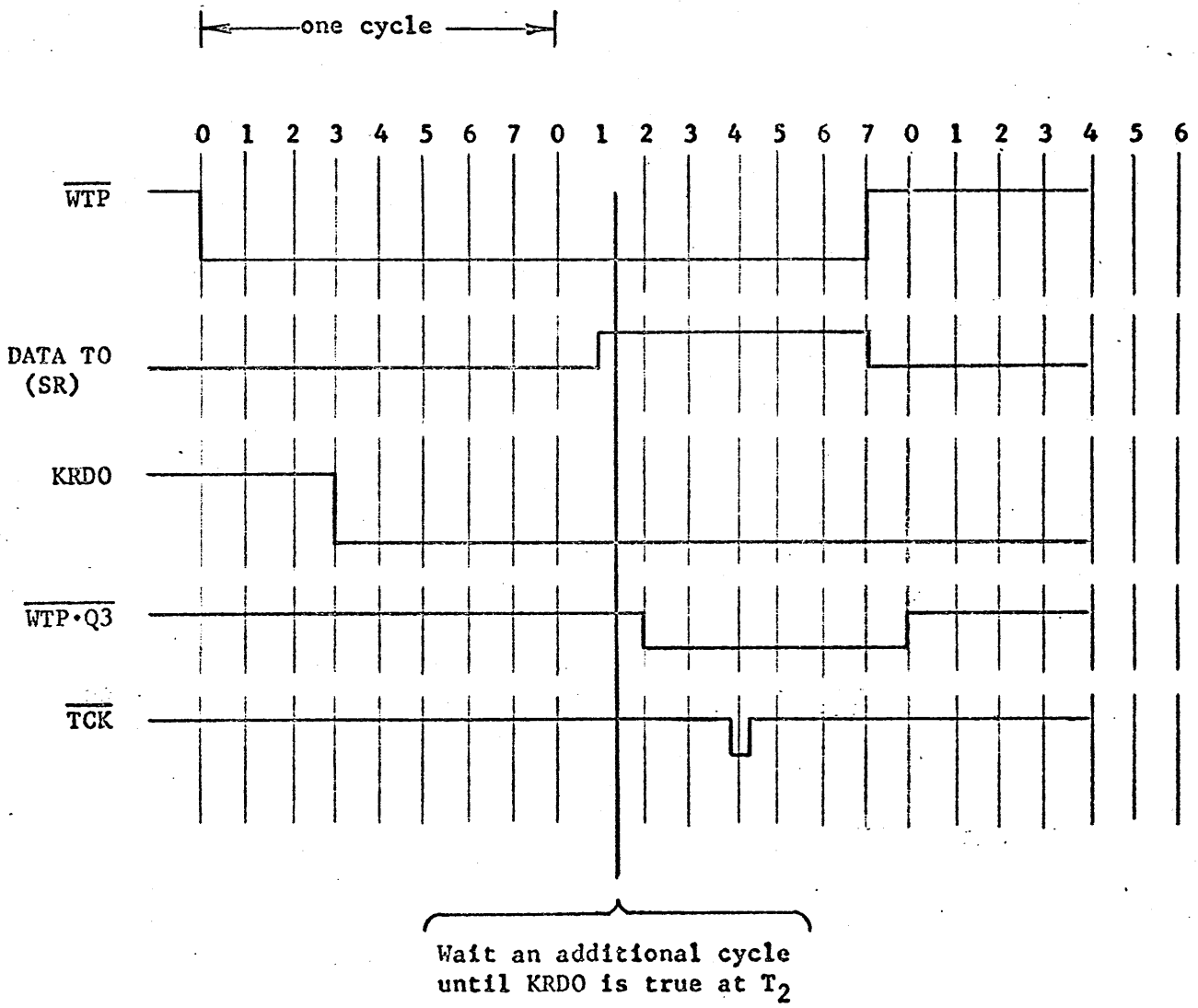
Function	No. of Connectors
1. Parallel output (WTP)	One
2. Parallel input (RDP)	One
3. Buffered input/output	Four

These connectors are panel mounted at the rear-bottom of the rack assembly.

C. PARALLEL OUTPUT

A single computer instruction allows for a 24 bit word to be transferred in parallel to an external device. This instruction locks up the computer until the external device signals that it is ready for the transfer. The output

PARALLEL OUTPUT TIMING



lines are designated as PO_o through PO_{23} with PO_o the most significant bit. These lines are changing and may be read only at transfer time. The Ready signal from the external device is designated $\overline{KRDO}$ (inverse logic) and must be supplied by the external device to release the computer.

Pin	Function	Remarks
EE	$\overline{TPO}$	A strobe signal from the computer. This signal indicates data on the PO_{0-23} may be read.
FF	$\overline{RTP \cdot Q_3}$	A signal from the computer which indicated the $\overline{TPO}$ will follow.
HH	$\overline{WTP}$	A signal from the computer which goes to 0 V and remains through - out the $\overline{WTP}$ instruction.
JJ	$\overline{KRDO}$	A ready indication generated by the external device which allows the transfer to be completed. The computer will maintain a wait state just prior to the transfer until release by the external device pulling the $\overline{KRDO}$ line to ground. If left unconnected, the $\overline{KRDO}$ line will go to + 8 V

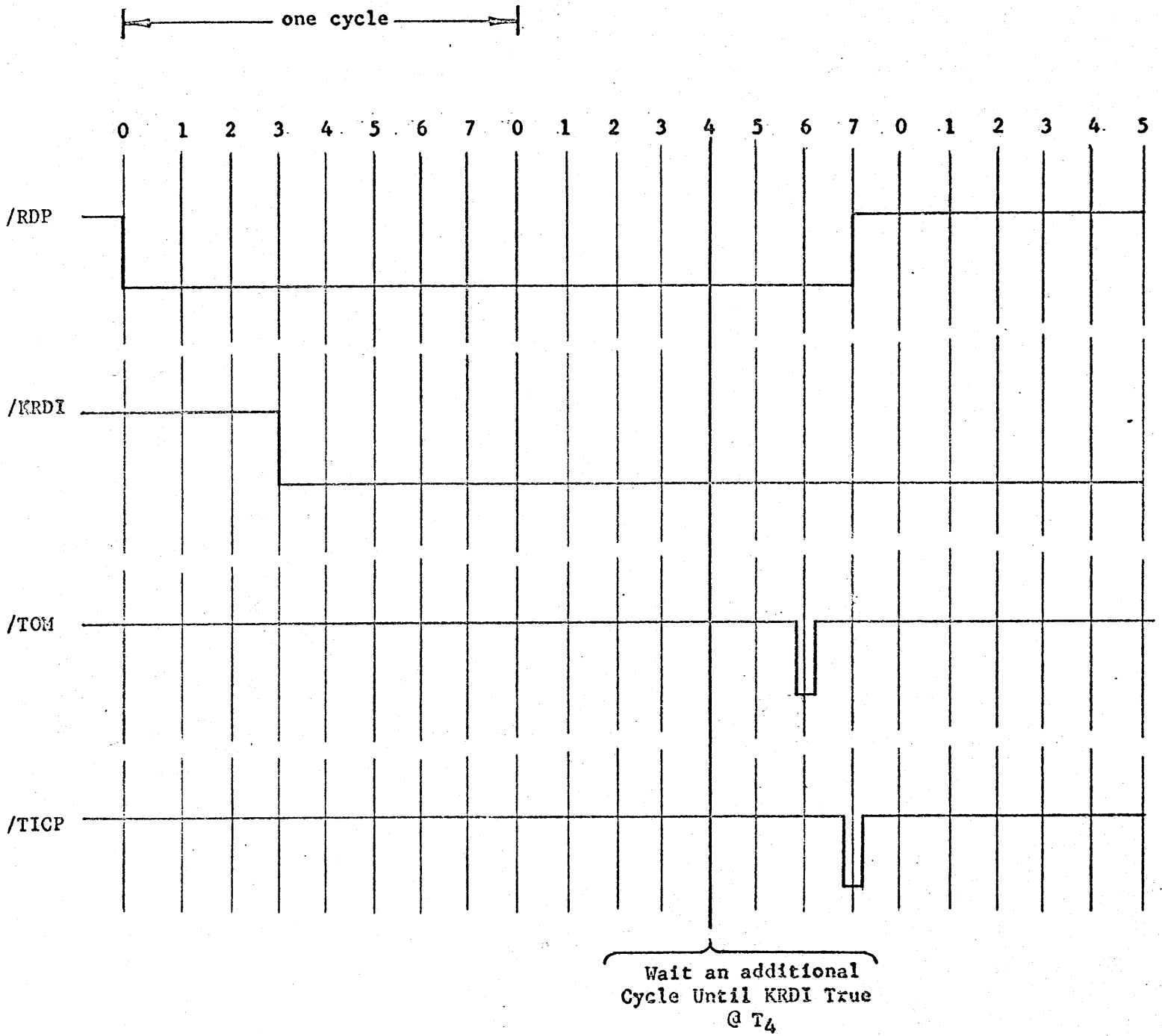
Pin	Function	Remarks
		and the computer will lock up on a WTP instruction until released by bringing the $\overline{\text{KRDO}}$ line to 0 V or by pushing the start push-button.
A	PO_0	Twenty four data line which may be read by the external device when $\overline{\text{TPO}}$ signal from the computer is at 0 V. At other times, data on these lines is changing and meaningless.
B	PO_1	
C	PO_2	
.	.	
.	.	
BB	PO_3	

D. PARALLEL INPUT

A single computer instruction allows for a 24 bit word to be transferred in parallel from an external device to the computer. This instruction locks up the computer until a signal is received from the external device that indicates the transfer is ready. The word is stored at a memory location dsignated by the effective address.

The 24 data lines are designated PI_0 thru PI_{23} with PI_0 the most significant bit. A signal designated $\overline{\text{RDP}}$ indicates the computer is in a state for parallel input transfer. A signal $\overline{\text{KRDI}}$ generated by the external device indicates data to be transferred is present. $\overline{\text{TCP}}$ indicates transfer is complete.

PARALLEL INPUT TIMING



Pin	Function	Remarks
A	PI_0	Input line for the entry of a parallel data. These lines will assume a logic one state unless held to zero by the external device. Data on the line will be read in only after the external device indicates data is ready.
B	PI_1	
C	PI_2	
D	PI_3	
.	.	
.	.	
BB	PI_{23}	
EE	$\overline{TCP}$	Transfer completed signal, indicates the computer has read the data lines.
FF	$\overline{RDP}$	A signal which goes to zero volt throughout a parallel input instruction.
HH	$\overline{KRDI}$	A ready signal generated by the external device indicates the data on the PI lines may be read into the computer. If left unconnected, this signal will assume +8V or the not ready state. The computer will therefore hang up on a parallel input instruction until the $\overline{KRDI}$ line is brought to zero or the start push-button has been pushed.

Pin	Function	Remarks
JJ	$\overline{\text{TIM}}$	Transfer input to the M register. A signal which is used by the computer to strobe data into the M register.

In addition to the above function, the parallel input connector is used for interrupt signal inputs.

Pin	Function	Remarks
KK	IN_1	A true signal will interrupt the computer to a specific address at the completion of its present instruction, provided an interrupt of a higher order is not present.
LL	IN_2	
MM	IN_3	
NN	IN_4	

E. BUFFERED INPUT/OUTPUT

External devices communicate with the computer through the input/output buffer. Data is received by the buffer serial-by-character. Each character contains six data bits plus a parity bit.

Six lines are used for device selection. The lines are designated and decoded for selection of an external device.

Input/Output Transfer Function

Pin	Function	Remarks
A thru F	$\overline{ZW_1}$ thru $\overline{ZW_6}$	Input data line for one six-bit character. $\overline{ZW_6}$ is least significant

Pin	Function	Remarks
		bit of the character. If unconnected, the line will assume +8 volts and input data will be all zeros.
H	$\overline{ZW_7}$	Input parity bit associated with the data character. The computer can check for even or odd parity.
J	$\overline{GA}$	Input/output clock supplied by external device. Indicates the transfer of a character to or from the computer.
D	$\overline{NP}$	Indicates to the computer that no parity will be supplied with the input character. If at 0 V, the computer will not check parity.
K	TWCH	A computer generated signal, indicating it has read a data character.
L	BRDY	Buffer-ready signal from the computer.
M thru T	WU_1 thru WU_6	Peripheral unit address. These signals are decoded by external devices to determine which device has been selected.

Pin	Function	Remarks
EE thru LL	W_0 thru W_5	Output data character. W_5 is the least significant bit.
CC	START	A signal which goes true when the start push-button is activated. May be used to reset external devices.
DD	DISC	A signal which goes true when the computer disconnects from external devices. All WU lines go to false state.
Y	$\overline{\text{LCH}}$	Last character signal used by the computer to set the disconnect flipflop on the last character of a record.
MM	W_p	Output parity bit. Either even or odd parity may be generated by the computer.

Activate Functions

During execution of ACT, instruction control pulses as well as the effective address are sent out through the input/output connector. These signals may be decoded for use in controlling external devices.

Pin	Function	Remarks
c	BUC	Buffer Control - a signal derived from ACT 0XXXX, indicating a buffer control instruction.
d	EXC	External Control - a pulse derived from an ACT 3XXXX signal for controlling external devices. The EXC signal is gated externally with the address line for control purposes.
j	I/OC	Input/output Control - a pulse derived from an ACT 1XXXX signal for controlling input/output devices. The I/OC signal is gated externally with the address lines for unit control.
h	ST	A computer reset signal by the START switch. It is used to reset external devices at the start of operation.
n thru AA	SR_{12} thru SR_{23}	The effective address lines. These signals are decoded and gated with the activation signal for control purposes.

Skip Functions

A SNS instruction will cause the computer to skip if certain conditions depending on the address portion of the instruction are not true. Two external inputs may be tested. Address bits 10 and 11 determine which lines are to be tested. All address bits may be gated with many external signals to generate the signal to be tested, thereby providing provisions for testing 2^{13} external signals.

Pin	Function	Remarks
a	STS_2	Skip test signal #2. The computer will skip an SNS 3XXXX instruction if the signal on this line is at +8V. If left unconnected, this line will assume + 8 V. The address portion on the instruction may be gated externally with many signals to generate the test input.
b	STS_1	Skip test signal #1. The computer will skip an SNS 1XXXX instruction if the signal on this line is at +8V. If left unconnected, this line will assume + 8V. The address portion on the instruction may be gated externally with many signals to generate the test input.
n thru AA	SK_{12} thru SK_{23}	Address line to be gated with external signal generating skip test signals.

APPENDIX

APPENDIX A

SIX BIT CHARACTER CODES

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

APPENDIX B

SCIENTIFIC CONTROL CORPORATION 670-2 COMPUTER

INSTRUCTION LIST

1. Data Transfer Instructions

Mnemonic	Op Code	Instruction Name	Cycles
LDA	27	Load AC	2
LAC	45	Load AC Complemented	2
STA	35	Store AC	2
LDE	75	Load EA	2
STE	36	Store EA	2
STS	34	Selective Store	3
LDX	71	Load Index	2
LXC	44	Load Index Complement	2
STX	37	Store Index	2
EAX	77	Effective Address Into Index	1
CLA	4600001	Clear AC	1
CLE	4600002	Clear EA	1
CLR	4600003		1
TAE	4600004	Transfer AC to EA	1
AEC	4600005	Transfer AC to EA, Clear AC	1
TEA	4600010	Transfer EA to AC	1
EAC	4600012	Transfer EA to AC, Clear EA	1
XAE	4600014	Exchange AC and EA	1
TEX	4600020	Transfer EA to XR	1

Data Transfer Instructions (contd.)

Mnemonic	Op Code	Instruction Name	Cycles
TXE	4600040	Transfer XR to EA	1
XXE	4600060	Exchange XR and EA	1
SXP	4600122	Store Exponent	1
LXP	4600140	Load Exponent	1
XXP	4600160	Exchange Exponents	1
TXA	4600200	Transfer XR to AC	1
TAX	4600400	Transfer AC to XR	1
XXA	4600600	Exchange XR and AC	1
NGA	4601000	Negate AC	1
XAM	62	Exchange AC and Memory	3

2. Arithmetic Instructions

ADD	55	Add	2
ADC	57	Add with Carry	2
AAM	63	Add AC to Memory	3
SUB	54	Subtract	2
SBC	56	Subtract with Carry	2
MPY	64	Multiply	14
DIV	65	Divide	28
MDC	61	Memory Decrement	3
MIN	61	Memory Increment	3

3. Logic Instructions

AND	14	AND to the Accumulator	2
ORA	16	OR to the Accumulator	2

Logic Instructions (contd.)

Mnemonic	Op Code	Instruction Name	Cycles
EOR	17	Exclusive OR to the Accumulator	2

4. Branch Instructions

BRA	01	Branch	1
BIX	41	Increment Index and Branch	1
BSL	43	Store Location Counter and Branch	2
BAN	24	Branch on AC Negative	1
BRT	51	Return Branch	2
BAZ	25	Branch on AC Zero	1
BAP	26	Branch on AC Positive	1
BNZ	15	Branch on AC Non Zero	1

5. Skip Instructions

SNS	40	Skip if Signal Not Set	1
SAG	73	Skip if AC Greater	3
SMN	53	Skip if Memory Negative	2
SAE	70	Skip if AC equals M on EA Mask	3
SAM	72	Skip if AC and M do not Compare Ones	2
SEM	52	Skip if EA and Memory do not Compare Ones	2
SAQ	50	Skip if AC Equals Memory	2
SDE	74	Difference Exponents and Skip	3

6. Shift Instructions

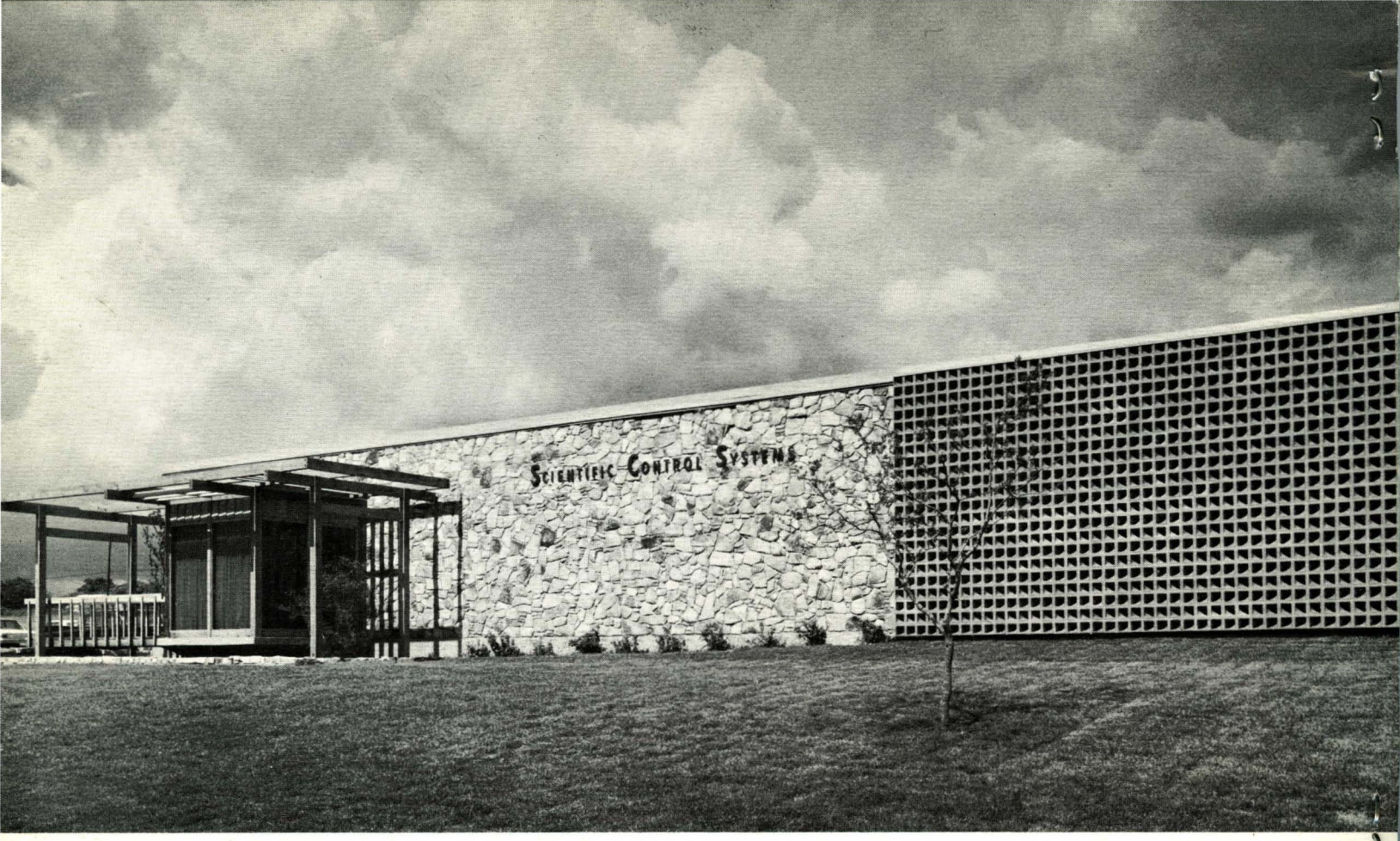
Mnemonic	Op Code	Instruction Name	Cycles
RSH	66000XX	Right Shift	$2 + \frac{N}{8}$
RCY	66200XX	Right Cycle	$2 + \frac{N}{8}$
LSH	67000XX	Left Shift	$2 + \frac{N}{8}$
LCY	67200XX	Left Cycle	$2 + \frac{N}{8}$
NDX	67100XX	Normalize and Decrement Index	$2 + \frac{N}{8}$

7. Miscellaneous Instructions

HLT	00	Halt	1
NOP	20	No Operation	1
XEC	23	Execute	1 + time of instruction executed

8. Input/Output Instructions

TMB	12	Transfer Memory to B	2 + Wait
TMC	10	Transfer Memory to C	2 + Wait
WTP	13	Write Parallel	2 + Wait
TBM	32	Transfer B to Memory	2 + Wait
TCM	30	Transfer C to Memory	2 + Wait



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 local sales office or the Marketing Department in Dallas.

Arlington, Massachusetts
30 Park Avenue
617 — 648-2922
(Boston)

Skokie, Illinois
125 Old Orchard Arcade
312 — 675-6700
(Chicago)

Houston, Texas
7800 Westglen Drive
713 — 782-9851

Seattle, Washington
1806 South Bush Place
206 — 324-7911

Midland Park, New Jersey
36 Central Avenue
201 — 652-6750
(New York)

Crofton, Maryland
Village Green
301 — 647-6431
(Baltimore)

Orlando, Florida
2319 E. South Street
305 — 841-3556

Pasadena, California
180 East California Blvd.
213 — 681-2651
(Los Angeles)

Other SCC products include: telemetry systems and airborne signal conditioning equipment such as amplifiers, demodulators and converters.

Scientific Control Corporation

14008 Distribution Way • Dallas, Texas 75234 • 214 — 241-2111

Printed in U.S.A.