SDS
SCIENTIFIC DATA SYSTEMS

INFORMATION BULLETIN

26 November 1963

SDS 930

The SDS 930 is the third in the 900 Series of SDS computers. It is program-compatible with the SDS 910 and SDS 920 but is approximately four times faster than the SDS 920. All SDS 900 Series software and all 900 Series peripheral devices can be used with the SDS 930. Essentially, the SDS 930 is identical to the SDS 920 except for:

1) Speed
2) Input/Output
3) One additional instruction
4) Additional memory addressing

SPEED

Since the SDS 930 is approximately four times as fast as the SDS 920, programs that are not input/output limited take only one quarter of the comparable SDS 920 time. The following are specific execution times including both memory access and indexing:

Cycle time	1.925 microseconds
Add	3.85 microseconds
Multiply	7.7 microseconds
Divide	19.25 microseconds
Shift (24 places)	5.775 microseconds

Floating Point Operations
(24-bit mantissa plus 9-bit exponent)

Add	81 microseconds
Multiply	59 microseconds
Divide	81 microseconds

(39-bit mantissa plus 9-bit exponent)

Add	91 microseconds
Multiply	152 microseconds
Divide	162 microseconds

INPUT/OUTPUT

The word parallel (PIN/POT), single bit (EOM/SKS), and priority interrupt I/O systems in the 930 are identical to those of the 920. An additional feature has been added to all 900 Series machines which allows interrupts to be armed and disarmed under program control. The primary change that has been made in the SDS 930 is to provide for up to eight I/O channels in place of the two (W and Y) which are available on the 920. Of these eight, four may be Time Multiplexed Channels and four may be Direct Access Channels.

Time Multiplexed Channels

The Time Multiplexed Channels operate like a W or Y buffer, and one of these (called the W Channel) is standard equipment with the SDS 930. This standard channel is always a 6-bit character device -- like the W buffer -- but additional channels may be extended up to 24-bits. When the predesignated number of characters is assembled, a signal is generated at the register. This signal can be used in any one of three ways, depending on the requirements of the programmer:

1) It can cause the execution of an input instruction on which the computer is locked. This execution will cause the contents of the register to be stored in memory.

2) It can activate an interrupt that transfers control to an input instruction.

3) It can cause the suspension of execution of a series of instructions and, under the control of an interlace control unit, permit the storage of the contents of the register in the designated memory location. This is done without the execution of an instruction, since the interlace control unit designates the address for storage of the input word and keeps track of the number of words stored. The interlace unit is integral to the Time Multiplexed Communication Channel. Output operates in a similar manner.

The three additional Time Multiplexed Channels are designated as follows:

Y: Identical to W except that it can have an extended character register.

C and D: Identical to Y except that they can be operated only in the interlace mode.

The Time Multiplexed Channels, like the W buffer, require two cycles (approximately 4 microseconds) for each word of I/O.

Direct Access Channels

The Direct Access Channels operate exactly like the C or D channels with the following exceptions:

1) Under programmer control, I/O may be either 6 bits or 24 bits.

2) The Input/Output operation requires only a single cycle. Further, if two banks of memory are available, I/O and computation may be overlapped. That is, if the I/O operation uses a memory bank that is different from that used by the computation, both processes can proceed simultaneously so that I/O effectively does not slow down computation.

For certain special purposes, it is appropriate to construct the I/O control external to the computer as an integral part of a peripheral system. To meet this requirement, a Direct Memory Access Connection is available for I/O. Although this Connection is an optional feature with any SDS 930, it is provided at no extra charge on computers purchased with a Direct Access Channel.

An example of the use of the Direct Memory Access Connection is a system that requires information to be handled which includes both data and the memory address of the data. The following information is supplied to the Direct Memory Access Connection:

1) Memory Address

2) Data

3) Synchronization and control signals including, for example, whether the operation is input or output. Note that if two or more banks of memory are available, the I/O and computation may be overlapped.

ONE ADDITIONAL INSTRUCTION

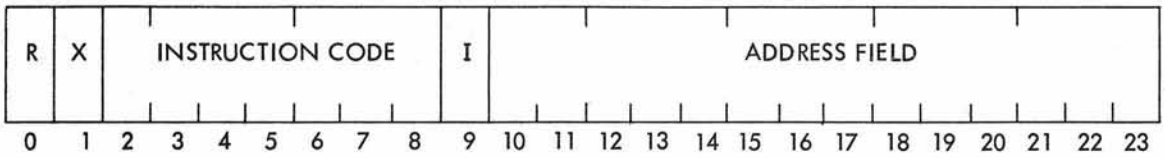
An additional instruction has been added (see the instruction list below) to permit additional EOM instructions which do not violate the basic 900 Series software system.

ADDITIONAL MEMORY ADDRESSING

Bank switching is available so that up to 32,000 words of core storage can be employed. Both manual and programmable memory lockout features are available as options.

Descriptive brochures and reference manual for the SDS 930 will be available in January.

Instruction Format:



Bit Positions	Function
0	Relative Address Bit - A "1" in this position causes the location of the instruction to be added to the address at loading. This bit position is not used in the logic of the computer.
1	Index Register Bit - A "1" in this position causes the contents of positions 10 - 23 of the Index Register (X Register) to be added to the address portion of the instruction prior to execution.
2 - 8	Instruction Code - The contents of bit position 2 (Programmed Operator Bit) determine the method of interpretation of the contents of the remaining 6 positions. If position 2 contains a "0", the contents of positions 3 through 8 are decoded as a normal instruction. If position 2 contains a "1", the instruction code is used to determine a subroutine entrance address.
9	Indirect Address Bit - A "1" in this position causes the computer to interpret the contents of positions 10 - 23 of the instruction (possibly modified by indexing) as the memory location where the effective address of the instruction may be found. A "0" causes the contents of positions 10 - 23 (possibly modified by indexing) to be interpreted as the effective address of the instruction.
10 - 23	Address - The contents of these positions normally determine the memory address referred to by the Instruction Code.

SDS 930 INSTRUCTION LIST

Designation	Instr. Code	Name	Function	Timing
LOAD-STORE:				
LDA	76	LOAD A	(M)→A	2
STA	35	STORE A	(A)→M	3
LDB	75	LOAD B	(M)→B	2
STB	36	STORE B	(B)→M	3
LDX	71	LOAD INDEX	(M)→X	2
STX	37	STORE INDEX	(X)→M	3
XMA	62	EXCHANGE M AND A	(A)↔(M)	3
ARITHMETIC:				
ADD	55	ADD M TO A	(A) + (M)→A	2
ADC	57	ADD WITH CARRY	(A) + (M) + Carry→A	2
ADM	63	ADD A TO M	(A) + (M)→M	3
MIN	61	MEMORY INCREMENT	(M) + 1→M	3
SUB	54	SUBTRACT M FROM A	(A) - (M)→A	2
SUC	56	SUBTRACT WITH CARRY	(A) - (M) - Carry→A	2
MUL	64	MULTIPLY	(A) × (M)→A, B	4
DIV	65	DIVIDE	(A, B) ÷ (M)→A, R→B	10
BRANCH-SKIP:				
BRU	01	BRANCH UNCONDITIONALLY	M→P	1
BRX	41	INCREMENT INDEX AND BRANCH	(X) + 1→X If X Neg., M→P If X Pos., P + 1→P	1, 2
BRM	43	MARK PLACE AND BRANCH	(P)→M; M + 1→P	2
BRR	51	RETURN BRANCH	(M) + 1→P	2
SKS	40	SKIP IF SIGNAL NOT SET	If Signal = 1, P + 1→P If Signal = 0, P + 2→P	2, 3

Designation	Instr. Code	Name	Function	Timing
BRANCH-SKIP:				
SKE	50	SKIP IF A EQUALS M	If (A) $\neq$ (M), P + 1 → P If (A) = (M), P + 2 → P	2, 3
SKG	73	SKIP IF A GREATER THAN M	If (A) $\leq$ (M), P + 1 → P If (A) > (M), P + 2 → P	2, 3

SKR	60	REDUCE M, SKIP IF NEGATIVE	(M) - 1 → M
			If (M) Pos., P + 1 → P
			If (M) Neg., P + 2 → P
				3

SKM	70	SKIP IF A = M ON B MASK	If (B)(A) $\neq$ (B)(M), P + 1 → P If (B)(A) = (B)(M), P + 2 → P	2, 3
SKN	53	SKIP IF M NEGATIVE	If (M) $\geq$ 0, P + 1 → P If (M) < 0, P + 2 → P	2, 3
SKA	72	SKIP IF M AND A DO NOT COMPARE ONES	If (A) (M) $\neq$ 0, P + 1 → P If (A) (M) = 0, P + 2 → P	2, 3
SKB	52	SKIP IF M AND B DO NOT COMPARE ONES	If (B)(M) $\neq$ 0, P + 1 → P If (B)(M) = 0, P + 2 → P	2, 3
SKD	74	DIFFERENCE EXPONENTS AND SKIP	$\lvert (B_{15-23}) - (M_{15-23}) \rvert \rightarrow X_{15-23}$ If Difference is Pos., P + 1 → P If Difference is Neg., P + 2 → P	2, 3
LOGICAL:				
ETR	14	EXTRACT	(A) and (M) → A	2
MRG	16	MERGE	(A) or (M) → A	2
EOR	17	EXCLUSIVE OR	(M)$(\bar{A})$ or $(\bar{M})$(A) → A	2
REGISTER CHANGE:				
CLA	4600001	CLEAR A	"0" → A	1
CLB	4600002	CLEAR B	"0" → B	1
CLR	4600003	CLEAR AB	"0" → A, B	1
CAB	4600004	COPY A INTO B	(A) → B	1
CBA	4600010	COPY B INTO A	(B) → A	1
XAB	4600014	EXCHANGE A AND B	(A) $\leftrightarrow$ (B)	1
BAC	4600012	COPY B INTO A, CLEAR B	(B) → A, "0" → B	1
ABC	4600005	COPY A INTO B, CLEAR A	(A) → B, "0" → A	1
CXA	4600200	COPY INDEX INTO A	(X) → A	1
CAX	4600400	COPY A INTO INDEX	(A) → X	1
XXA	4600600	EXCHANGE INDEX AND A	(X) $\leftrightarrow$ (A)	1
CBX	4600020	COPY B INTO INDEX	(B) → X	1

Designation	Instr. Code	Name	Function	Timing
REGISTER CHANGE:				
CXB	4600040	COPY INDEX INTO B	$(X) \rightarrow B$	1
XXB	4600060	EXCHANGE INDEX AND B	$(X) \longleftrightarrow (B)$	1
STE	4600122	STORE EXPONENT	$(B_{15-23}) \rightarrow X_{15-23}$ $0 \rightarrow B_{15-23} X_{15} \rightarrow X_{0-14}$	1
LDE	4600140	LOAD EXPONENT	$(X_{15-23}) \rightarrow B_{15-23}$	1
XEE	4600160	EXCHANGE EXPONENTS	$(B_{15-23}) \longleftrightarrow (X_{15-23})$	1
CNA	4601000	COPY NEGATIVE INTO A	$-(A) \rightarrow A$	1
SHIFT:				
RSH	6600XXX	RIGHT SHIFT AB	AB Shift Right N Places	
RCY	6620XXX	RIGHT CYCLE AB	AB Cycled Right N Places	
LSH	6700XXX	LEFT SHIFT AB	AB Shift Left N Places	
LCY	6720XXX	LEFT CYCLE AB	AB Cycled Left N Places	
NOD	6710XXX	NORMALIZE AND DECREMENT X	AB Left and $X - 1 \rightarrow X$ Until $A_0 \neq A_1$, or N Shifts	

Shift timing is as follows:

Cycles	Shifts Performed
2	Right 0 - 3 Left 0 - 6
3	Right 4 - 14 Left 7 - 28
4	Right 15 - 25 Left 29 - 48
5	Right 26 - 36
6	Right 37 - 47
7	Right 48

CONTROL:

HLT	00	HALT	Halts Computation	1
NOP	20	NO OPERATION	----	1
EXU	23	EXECUTE	Instruction M is performed, P unchanged	1
EAX	77	COPY EFFECTIVE ADDRESS INTO INDEX	Effective Address → X	2

INPUT/OUTPUT:

MIW	12	M INTO W BUFFER WHEN READY	(M) → W	2 + wait
WIM	32	W BUFFER INTO M WHEN READY	(W) → M	3 + wait
MIY	10	M INTO Y BUFFER WHEN READY	(M) → Y	2 + wait
YIM	30	Y BUFFER INTO M WHEN READY	(Y) → M	3 + wait
POT	13	PARALLEL OUTPUT	(M) → Unit M in Parallel	3 + wait
PIN	33	PARALLEL INPUT	(Unit M) → M in Parallel	4 + wait
EOM	02	ENERGIZE OUTPUT M	2 μsec Pulse to Point(s) Addressed	1
EOD	06	ENERGIZE OUTPUT TO DIRECT ACCESS CHANNEL	2 μsec Pulse for functions of Direct Access Channel	1