Appended Modifications of SODA.

This addenda includes improvements that have been made in SODA since the publication of the Instruction Manual.

Present Versions of SODA:

Version I.

The first version of SODA, for which the instruction manual exists includes the facility for handling programmes of up to 768 symbolic addresses. It is worthy to note that a symbolic address must contain either a punch in the Y, X, or 0, when double punched forms alphabetic characters. However (for the purposes of tabulating a list of instructions before a translation,) some tabulators are not equipped to handle this double punched alphabetic convention, therefore by giving a symbolic address containing an overpunched + or - sign, or a pre-punched sign, the requirement for one alphabetic character, is satisfied, and in this manner the location of this instruction, the location of the operand, and the location of the next instruction may be tabulated. For example, a +6800 might be given as a location for an instruction. Care should be taken so that signs will always be punched, otherwise, a numeric absolute address (0-95), (100-106), might be mistakenly given, if a pre or over-punched sign is not used.

Version II. of SODA includes provisions for a maximum of 512 symbolic addresses. This version of SODA permits errors to be fixed by re-punching a card instead of altering the punching through console manipulation. By placing an "M" or middle type restore control card in the machine through initial input key, the erroneous card can then be re-translated. The erroneous card would have been the last card in the reader hopper. The other type of control card which allows another programme to be translated after one translation has finished is called a "B" or beginning type restore control. At least the first SODA instruction of the programme must be translated correctly in order for the "M" type restore control to operate properly.

Version III includes facilities for a maximum of 512 symbolic addresses. It will accept both the B and M type restore controls. This version, however, has numeric operation numbers, corresponding to the SODA operations. A list of these numbers is given on the following pages, corresponding to the various SODA operations. The purpose of this, again, was to make is possible to list, or tabulate a programme prior to its translation. For example, Clear and Add to the Short, is punched 001 for Version III, and is punched CAS in versions I, and II. A II zeros may be omitted in card punching, except when punching alphabetic characters S through Z.

Restore Control for the Machine or Translated programme.

Sometimes when checking out a translated programme it is desired to run in the programme again, together with the initial pack. This procedure may be eliminated by using an RCM type restore control card. This type of restore control may be effected by running in the RCM card (on initial input key), followed by the last 3 cards of the translated programme being checked. This procedure, also eliminates the need for running in the initial pack when checking out more than one translated programme. In this case, 2 cards (corresponding to the load drum routing ZP 14) follow the restore control, RCM, and the translated programme which is read to drum follows the 2 cards of the load drum.

Errors in a translated programme.

Thus far, no errors which are attributable to the translator have been found. Most errors occur either through incorrectly checked punching, or through careless programming. There are two types of errors which can possibly make the machine go dead, or lead to erroneous results.

One error occurs when using the index registers, and the index register contains a negative number, if this negative number is used in an instruction which is selecting an element. A negative number which is selecting an element may cause the programme to go dead. Since the last element of an array is reached by an index register containing a zero, then this negative number is meaningless.

The other error occurs when a symbolic name for the location of the next instruction does not exist in the rest of the programme, as a name for the location of this instruction. Also, unless a forward jump indication or non-jump operation is given (Col. 26) to a location which has not been previously defined, then erroneous results may result.

Existing Deuce Subroutines which have been modified for SODA:

Read Floating Decimal, RFD. This routine reads in a floating number in standard floating decimal and converts the number to standard floating binary, and places the result in the Floating Accumulator. The layout for the number, $a \times 10^b$, is punched with the sign of b (an integer) in column 1 followed by the magnitude of b in columns 2 to 9. Column 11 contains the sign of a, and the magnitude of a, is punched in columns 12 to 20. The range of a, is as follows:

$$1 \leqslant |a| < 10$$

Punch Floating Decimal, PFD. This subroutine punches out the contents of the Floating Accumulator which is in standard floating binary, in standard floating decimal in the layout like that given for Read Floating Decimal.

Read Decimal Data, RDD. This programme converts the 32 columns of a card which is punched in any layout to binary integers (0 binary places) and places the decimal integers (in their binary form) in Working Store One, or Numeric Addresses 00 up to 31. The parameters of the subroutine give the layout of the cards being read in. The criteria for the selection and punching of these parameters is given as follows:

Card 2 Row 3, if P16 is zero, overpunched signs must be used, if P16 is a one the pre-punched signs must be used.

Card 3 Row 0, this codeword contains a one in every column that is to be read.

Card 7 Row 5. This parameter contains a one in the position corresponding to the last column of a number which is being converted.

All columns must be punched if they are so indicated by the second codeword. The 1st 2 cards of the subroutine are un-numbered, and therefore the card numbers apply only to cards numbered 1 to 12.

Sine and Cosine, SIN, COS. These routines are Deuce subroutines T01/2. When either SIN or COS order is given, the sine or the cosine of $(\Pi/2)\ x$ is found where x is the number in the Short Accumulator and given to 30 binary places. The result is in the Short Accumulator.

PHD - PuncH decimal Data.

This routine corresponds to Deuce subroutine P14. All numbers punched must be less than 10 to "p" binary places. The numbers are punched from binary words located in 00, --31.

The following parameters must be placed in the subroutine:

1. A 1 is punched in Card 2, r 5, row 8 of the 1st digit of the 1st word to be punched. (Cols. 21 - 52).
2. A 1 is punched in card 5, row 4, in the position corresponding to the last digit of each number punched, 1's also indicate a blank col of a card.
3. Card 1, row 8 is a roundoff constant (to 32 binary places) for all numbers punched.
4. Each binary word is punched from "p" binary places. $(2p - 3) \times P_{17}$ is punched in Card 1, Row 3.

Appended List of SODA Instructions for Numerical Version of SODA.

The number in parenthesis is the identification for the trace. The underlined number is the associated number of machine instructions needed for use of a loop control card. The number to the left of the abbreviation (used in Alphabetic Version of SODA) is the operation number.

A. Instructions Pertaining to the Short Accumulator.

(00)1 CAS - Clear and Add to the Short (1) 1.
2 CSS - Clear and Subtract from the Short (2) 2.
3 ADS - ADd to Short (3) 1.
4 SBS - SuBtract from the Short (4) 1.
5 STS - STore the Short(5) 1.
6 SSL - Shift the Short to the Left (6) 2.
7 SSR - Shift the Short to the Right (7) 3.
8 LAS - Logical And with the Short (8) 3.
9 LOS - Logical Or with the Short (9) 6.
(0)10 LNS - Logical Non-equivalence with the Short (10) 3.

B. Instructions Pertaining to the Lower Accumulator.

(0)11 CAL - Clear and Add to the Lower (11) 2.
12 CSL - Clear and Subtract from the Lower (12) 3.
13 ADL - ADd to the Lower (13) 2.
14 SBL - SuBtract from the Lower (14) 2.
15 STL - STore the Lower (15) 2.
16 SLL - Shift the Lower to the Left (16) 3.
17 SLR - Shift the Lower to the Right (17) 3.
18 LAL - Logical And with the Lower (18) 3.
19 LOL - Logical Or with the Lower (19) 6.
20 LNL - Logical Non-equivalence with the Lower (20) 3.

C. Instructions Pertaining to the Upper Accumulator.

21 CAU - Clear and Add to the Upper (21) 2.
22 CSU - Clear and Subtract from the Upper (22) 3.
23 ADU - ADd to the Upper (23) 2.
24 SBU - SuBtract from the Upper (24) 2.
25 STU - STore the Upper (25) 2.
26 SUL - Shift the Upper to the Left (26) 3.
27 SUR - Shift the Upper to the Right (27) 3.
28 LAU - Logical And with Upper (28) 3.
29 LOU - Logical Or with the Upper (29) 6.
30 LNU - Logical Non-equivalence with the Upper (30) 3.

31 MPY - MultiPlY (31) 3.
32 DIV - DIVide (32) 3.

D. Instructions Pertaining to the Double Accumulator.

41 CAD - Clear and Add to the Double (33) 1.
42 CSD - Clear and Subtract from the Double (34) 2.
43 ADD - ADd to the Double (35) 1.
44 SBD - SuBtract from the Double (36) 1.
45 STD - STore the Double (37) 1.
46 SDL - Shift the Double to the Left (38) 2.
47 SDR - Shift the Double to the Right (39) 2.
48 ASD - Add a Single length word to the Double (69) 2.
49 SSD - Subtract a Single length word from the Double (70) 2.

E. Instructions Pertaining to Floating Point Arithmetic.

51 CAF - Clear and Add a Floating point number (40) 1.
52 CSF - Clear and Subtract a Floating point number (41) 4.
53 FAD - Floating point ADd (43) 4.
54 FSB - Floating point SuBtract (44) 4.
55 STF - STore a Floating point number (42) 1.
56 PRF - PRepare a Floating point number (76) 3.
57 FMP - Floating point MultiPly (45) 4.
58 FDV - Floating point DiVide (46) 4.

E. (Cont'd)

61 FSR - Floating point Square Root (47) 4.
62. FLG - Floating point LoGarithm (48) 2.
63 FEX - Floating point EXponential (49) 2.
64 FSN - Floating point SiNe (50) 2.
65 FCS - Floating point CoSine (51) 2.
66 FAT - Floating point Arc-Tangent (52) 2.

F. Instructions Pertaining to the Index Registers.

71 LXP - Load an indeX register Positive (53) 2.
72 LXN - Load an indeX register Negative (54) 5.
73 LXL - Load an indeX register positive Less one (71) 5.
74 ADX - ADd to an indeX register (55) 5.
75 SBX - SuBtract from an indeX register (56) 5.
76 STX - STore an indeX register (57) 2.

G. Decision or Jump Instructions.

101 JSZ - Jump if the Short is Zero 3.
+101 JSP - Jump if the Short is Positive 3.
111 JLZ - Jump if the Lower is Zero 3.
+111 JLP - Jump if the Lower is Positive 3.
121 JUZ - Jump if the Upper is Zero 3.
+121 JUP - Jump if the Upper is Positive 3.
141 JDZ - Jump if the Double is Zero 3.
+141 JDP - Jump if the Double is Positive 3.
-171 JLX - Jump on IndeX 9.
-173 JXH - Jump on indeX High or equal 9.
-172 JXE - Jump on indeX Equal 6.

H. Instructions involving the Working Stores.

81 RWO - Read into Working storage One (65) 5.
82 RWT - Read into Working storage Two (66) 5.
83 WWO - Write from Working storage One (67) 8.
84 WWT - Write from Working storage Two (68) 8.
85 CWO - Clear Working storage One (72) 2.
86 CWT - Clear Working storage Two (73) 2.

I. Instructions Pertaining to Input and Output.

91 RDO - ReaD into One from the reader 2.
92 RDT - ReaD into Two from the reader 2.
93 RDC - ReaD a Card 2.
94 RDA - ReaD Array 3.
95 PHO - PuncH from working storage One 2.
96 PHT - PuncH from working storage Two 2.
97 PHC - PuncH a Card 2.
98 PHA - PuncH Array 3.
99 RIL - Read from the Input Lights (79) 1.
100 WOL - Write into the Output Lights (77) 1.

J. Miscellaneous Instructions.

180 STZ - STore a Zero (60) 1.
181 STO - STore a One (61) 1.
182 STA - STore an Address unit (62) 1.
183 STH - STore a High position bit (63) 1.
184 STM - STore a Minus one (64) 1.

190 ACA - ACtivate Alarm (58) 1.
191 SPA - StoP Alarm (59) 1.
192 COL - Clear the Output Lights (78) 1.
193 ENS - ENter Subroutine 6.
194 LVS - LeaVe Subroutine 2.

198 HPR - Halt and PRoceed 2.
199 STP - SToP 1.

SODA PROGRAMMING SHEET

CARD NOS. ______

PROB. ______

CARD NO.	CONTROL	LOCATION THIS INSTR.	OPERATION	LOCATION OF OPERAND	LOCATION NEXT INSTR.	IND. REG.	DECREMENT	SPEC JMP	COMMENTS
col	1	2 3 4 5 6	7 8 9	10 11 12 13 14	15 16 17 18 19	20	21 22 23 24 25	26	