L - 92
1 OF 2

6.3 A SHORT GUIDE TO CODING
(Using the Whirlwind I code of October 1949)

COMPUTER PROGRAMS

Program. A program is a sequence of actions by which a computer handles a problem. The process of determining the sequence of actions is known as programming.

Flow diagrams. A flow diagram is a series of statements of what the computer has to do at various stages in a program. Lines of flow indicate how the computer passes from one stage of the program to another.

Coded program. Programs and flow diagrams are largely independent of computer characteristics, but instructions for a computer must be expressed in terms of a code. A set of instructions that will enable a computer to execute a program is called a coded program, and the process of preparing a coded program is known as coding.

Orders and operations. Individual coded instructions are known as orders and call for specific operations such as multiply, add, shift, etc.

The computer code. The computer code described here is that of Whirlwind I, an experimental computer using binary digits, single-address order code, parallel operation, and electrostatic storage. It is expected that computers of this type will ultimately achieve an average speed of 50,000 operations per second.

COMPUTER COMPONENTS

Registers and words. A register has 16 digit positions each able to store a one or a zero. A word is a set of 16 digits that may be stored in a register. A word can represent an order or a number.

Arithmetic element. Arithmetic operations take place in the arithmetic element, whose main components are three flip-flop registers, the A-register, the accumulator, and the B-register (AR, AC, BR). The 16 digit positions of AR starting from the left are denoted by AR 0, AR 1, . . ., AR 15. Similarly for AC, BR. Words enter AC through AR; BR is an extension of AC.

Storage. The term "register" by itself refers to the main electrostatic storage, which consists of 2^{11} or 2048 registers, each of which is identified by an address. These addresses are 11-digit binary numbers from 0 to 2047. The computer identifies a register by its address.

Input-output. All information entering or leaving the computer is temporarily stored in the input-output register (IOR). The computer regulates the flow of information between the internal storage and IOR, and also calls for any necessary manipulation of external units. The descriptive names of the input-output orders were chosen for photographic film reader-recorder units, but the orders are applicable to other types of external equipment.

Control element. The control element controls the sequence of computer operations and their execution. Instructions are obtained from storage in the form of individual orders, each of which is represented by a single word.

Inter-connections. The four main elements (storage, control, arithmetic, and input-output) are connected by a parallel communications system, known as the bus.

REPRESENTATION OF ORDERS

Operation section. When a word is used to represent an order the first (left-hand) 5 digits, or operation section, specify a particular operation in accordance with the order code.

Address section. The remaining 11 digits, or address section, are interpreted as a number with the binary point at the right-hand end. In the majority of orders this number is the address of the register whose contents will be used in the operation. In orders sl, sr, the number specifies the extent of a shift; in rf, rb, the number specifies an external unit; in ri, rs, the address section is not used.

Example. The order ca x has the effect of clearing AC (making all the digits zero) and then putting into AC the word that is in the register whose address is x. If q is a quantity in some register, the order needed to put q in AC is not ca q but ca x, where x is the address of the register that con-

REPRESENTATION OF NUMBERS

Single-word representations. When a word is used to represent a number the first digit indicates the sign and the remaining 15 are numerical digits. For a positive number the sign digit is zero, and the 15 numerical digits with a binary point at their left specify the magnitude of the number. The negative -y of a positive number y is represented by complementing all the digits, including the sign digit, that would represent y. (The complement is formed by replacing every zero by a one and every one by a zero.) In this way a word can represent any multiple of 2^{-15} from $2^{-15} - 1$ to $1 - 2^{-15}$. Neither +1 nor -1 can be represented by a single word. Zero has two representations, either 16 zeros or 16 ones, which are called +0 and -0 respectively.

Overflow — increase of range and accuracy. With single-word representation the range is limited to numbers between $2^{-15} - 1$ and $1 - 2^{-15}$. Programs must be so planned that arithmetic operations will not cause an overflow beyond this range. The range may be extended by using a scale factor, which must be separately stored. Accuracy can be increased by using two words to represent a 30-digit number.

COMPUTER PROCEDURE

Sequence of operations. After the execution of an order the program counter in the control element holds the address of the register from which the next order is to be taken. Control calls for this order and carries out the specified operation. If the order is not sp or cp(-) the address in the program counter then increases by one so that the next order is taken from the next consecutive register. The sp and cp(-) orders permit a change in this sequential procedure.

Transfers. A transfer of a digit from one digit position to another affects only the latter digit position, whose previous content is lost.

Negative zero. The subtraction of equal numbers produces a negative zero in AC, except when AC contains +0, and -0 is subtracted from it.

Manipulation of orders. Words representing orders may be handled in the arithmetic element as numbers.

Procedure in the arithmetic element. The execution of an addition includes the process of adding in carries; this process treats all 16 digits as if they were numerical digits, a carry from AC 0 being added into AC 15. A subtraction is executed by adding the complement. Multiplication, division, shifting and round-off are all executed with positive numbers, complementing being performed before and after the process when necessary. For round-off the digit in BR 0 is added into AC 15.

NOTATION FOR CODING

Addresses. A coded program requires certain registers to be used for specified purposes. The addresses of these registers must be chosen before the program can be put into a computer, but for study purposes this final choice is unnecessary, and the addresses can be indicated by a system of symbols or index numbers.

Writing a coded program. Registers from which control obtains orders may be called action registers, and should be listed separately from registers containing other information, which may be called data registers. A coded program is written out in two columns; the first contains the index number of each action or data register, and the second column indicates the word that is initially stored in that register. In many cases part or all of a word may be immaterial because the contents of the register in question will be changed during the course of the program. This state of affairs is indicated by two dashes, for example, ca --.

The abbreviations RC, CR. Abbreviations used in referring to the register that contains a certain word or to the word in a certain register are

RC . . . = (Address of) Register Containing . . .
CR . . . = Contents of Register (whose address is) . . .

The symbol ri x. When an address forms part of an order it is represented by the last 11 digits of a word whose first 5 digits specify an operation. An address x that is not part of an order is represented by the last 11 digits of a word whose first 5 digits are zero, which is equivalent to specifying the operation ri. Thus the word for an unattached address x

L-92
2 of 2

AC = Accumulator   AR = A-Register   BR = B-Register

x is the address of a storage register; n is a positive integer; k designates an external unit

Order	Operation			Function
	Name	Code		
		Decimal	Binary	
ri --	read initially	0	00000	Take words from external unit until internal storage is full.
rs --	remote unit stop	1	00001	Stop external unit.
rf k	run forward	2	00010	Prepare to use external unit k in forward direction.
rb k	run backward	3	00011	Prepare to use external unit k in backward direction.
rd x	read	4	00100	Transfer to register x a word supplied by external unit.
rc x	record	5	00101	Arrange for transfer of contents of register x to external unit.
ts x	transfer to storage	8	01000	Transfer contents of AC to register x.
td x	transfer digits	9	01001	Transfer last 11 digits from AC to last 11 digit positions of register x.
ta x	transfer address	10	01010	Transfer last 11 digits from AR to last 11 digit positions of register x.
cp(-)x	conditional program	14	01110	If number in AC is negative, proceed as in sp; if number is positive disregard the cp(-) order, but clear the AR.
sp x	subprogram	15	01111	Take next order from register x. If the sp order was at address y, store y + 1 in last 11 digit positions of AR.
ca x	clear and add	16	10000	Clear AC and BR, then put contents of register x into AC. If necessary, add in carry from previous sa addition.
cs x	clear and subtract	17	10001	Clear AC and BR, then put complement of contents of register x into AC. If necessary, add in carry from previous sa addition.
ad x	add	18	10010	Add contents of register x to contents of AC, storing result in AC.
su x	subtract	19	10011	Subtract contents of register x from contents of AC, storing result in AC.
cm x	clear and add magnitude	20	10100	Clear AC and BR, then put positive magnitude of contents of register x into AC. If necessary add in carry from previous sa addition.
sa x	special add	21	10101	Add contents of register x to contents of AC, storing result in AC and retaining any overflow for next ca, cs, or cm order. Only orders 1 through 15 may be used between the sa order and ca, cs, or cm orders for which the sa is a preparation.
ao x	add one	22	10110	Add the number 1×2^{-15} to the contents of register x. Store result in AC and in register x.
mr x	multiply and round off	24	11000	Multiply contents of register x by contents of AC; round off result to 15 numerical digits and store in AC. Clear BR.
mh x	multiply and hold	25	11001	Multiply contents of register x by contents of AC and retain the full product in AC and the first 15 digit positions of BR, the last digit position of BR being cleared.
dv x	divide	26	11010	Divide contents of AC by contents of register x, leaving 16 numerical digits of the quotient in BR and ±0 in AC according to sign of the quotient. (The order sl 15 following the dv order will round off the quotient to 15 numerical digits and store it in AC.)
sl n	shift left	27	11011	Multiply the number represented by the contents of AC and BR by 2^n. Round off the result to 15 numerical digits and store it in AC. Disregard overflow caused by the multiplication, but not that caused by round-off. Clear BR.
sr n	shift right	28	11100	Multiply the number represented by the contents of AC and BR by 2^{-n}. Round off the result to 15 numerical digits and store it in AC. Clear BR.
sf x	scale factor	29	11101	Multiply the number represented by the contents of AC and BR by 2 sufficiently often to make the positive magnitude of the product equal to or greater than 1/2. Leave the final product in AC and BR. Store the number of multiplications as last 11 digits of register x, the first 5 digits being undisturbed.

NOTES ON THE ORDER CODE

Effect of operations. The functions of the various orders are described above. It is to be assumed that AR, AC, BR, and the register whose address is x are undisturbed unless the contrary is stated.

AR. AR is primarily a buffer register for passing words into AC. After orders ca x, cs x, ad x, su x, sa x, and ao x it contains the number originally contained in register x. After orders cm x, mr x, mh x, and dv x it contains the magnitude of the contents of x. The effect of sp x and cp(-)x is stated above. No other order changes the contents of AR.

BR. A number stored in BR always appears as a positive magnitude, the sign of the number being assumed to be that indicated by the sign digit in AC. This convention has no effect on the logical result of the operations involving BR except that when BR contains a number that will be used later it is necessary to retain the appropriate sign digit.

Alarms. If the result of an arithmetic operation exceeds the register capacity (i.e., if overflow occurs), a suitable alarm is given except as mentioned in connection with orders sa x and sl n.

Shift orders. A multiplication overflow in sl is lost without giving an alarm, but an overflow from round-off gives an alarm. Orders sr 0 and sl 0 only cause round-off, an alarm being given if an overflow occurs. The integer n is treated modulo 32, i.e., sl 32 = sl 0, sl 33 = sl 1, etc.

Scale factors. If all the digits in BR are zero and AC contains ±0, the order sf x leaves AC and BR undisturbed and stores the number 33 in the last 11 digit positions of register x.

Division. Let u and v be the numbers in AC and register x when the order dv x is used. If $|u| < |v|$ the correct quotient is obtained and no overflow can arise. If $|u| > |v|$ overflow occurs and gives an alarm. If $u = v \neq 0$ the dv order leaves 16 ones in BR and round-off in a subsequent sl 15 would cause overflow and give an alarm. If $u = v = 0$ a zero quotient is obtained.