Poughkeepsie, New York
December 15, 1955

PROJECT STRETCH
FILE MEMO # 11

COMPANY CONFIDENTIAL

SUBJECT: Partial Word Data System
By: G.M. Amdahl, E.M. Boehm, J.E. Griffith

Editing

Not all data fields correspond in size to the size of a Stretch word. In many applications it is desirable to put more than one field in the 60 bit word. A logical extension of "packing" multiple fields in one word is starting a field in the unused portion of one word and finishing the field in the next word. Provision is made for addressing fields whose size does not match a Stretch word. These fields may be moved from one storage location to another and may be used in arithmetic operations. This report will describe the method of addressing and the execution of arithmetic operations on these fields. The reader is referred to the report on Editing for a description of data movement and other logical operations.

A field is addressed in an instruction by defining four values; (1) the size of a character S, (2) the number of characters in the field N, (3) the location of the starting character L, and (4) the address of the word in which the field starts A.

Opn.	(S)	N	L	Tag	A

A minimum of 3 character sizes is provided. A data field may be in binary (S=1), decimal (S=4), or binary coded decimal (S=6) for arithmetic operations. The size of the field is defined by N. If N is left blank, the field is assumed to be 60 bits. Otherwise, the number of bits used by the field is N multiplied by the size of the character (S). L indicates the starting position of a field. Numbering of characters commences in the high order position of the word with zero and proceeds to the right. The memory location in which the field starts is given by A. The starting bit can be obtained by multiplying S by L. If the product of this multiplication is greater than 60, the search for the starting position of the field is extended to the next location in memory. Successive words are shifted left until the product of S L equals the number of bits shifted left.

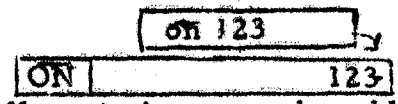
	L	N	A	
Add (S=4)	33	33	1000	The starting position in memory of the field referred to in this instruction is bit 12 in location 1002.

A field used in arithmetic operations will be 60 bits or less. If an instruction designates a field larger than 60 bits, the low order 60 bits of the field are used. Arithmetic operations will be performed on two classes of data -- signed fields and unsigned fields. A signed field will contain the sign, status of overflow, and or alphanumeric information. An unsigned field will be assumed positive, assumed not to have overflow, and contain alphanumeric information.

Signed fields are stored in memory with the sign and overflow information adjacent to the remainder of the field. When data is in binary, 2 bits are used for this information. With decimal data, 4 bits are required, and with alphanumeric data 6 bits are used. When assigned field is transmitted between memory and the arithmetic or control section of Stretch the sign and overlow indicator must be counted in N.

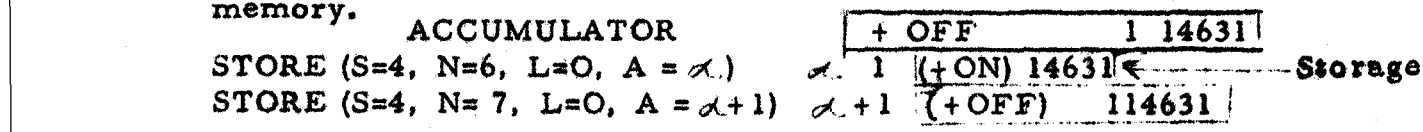
	Sign	Overflow		S	N
Binary	+	OFF	101101	1	8
Decimal	+	OFF	10100011	4	3
Alphanumeric	+	OFF	001010001000	6	3

While data is addressed by its sign or high order position, data enters the low order end of the arithmetic or control register.



Thus, fields of different sizes may be added or subtracted without the necessity of loading the largest field first or shifting data in the arithmetic section to expand the field size. An unsigned field is addressed by the high order position and is loaded from storage into the low order end of the arithmetic or control register.

Overflow in partial word data is defined as a non-zero bit in the arithmetic register to the left of the field stored. When a signed field is stored into memory, the overflow indicator will be turned on if a non-zero bit is found in the portion of the accumulator not written into memory.



The presence or absence of overflow in the accumulator may be determined by zero test (also non-zero test) starting at a specified position in the accumulator.

ACCUMULATOR _ - 1465320

Transfer zero (S=4, N= 6)	Proceed to next instruction
Transfer non-zero (S=4, N=)	Transfer)
Transfer zero (S=4, N=8)	Transfer
Transfer non-zero (S=4, N=9)	Proceed to next instruction

When a signed field is written into memory, the sign and overflow indicator are inserted in the high order portion of the field addressed.

Instructions are provided to easily translate numeric information from a decimal representation (4 bits to a character) to an alphanumeric form (6 bits to a character) and from an alphanumeric representation to a decimal form. These instructions are termed - (1) Expand, and (2) Contract.

When the expand instruction is executed, the contents of the accumulator are shifted left and 2 zero bits are inserted between each 4 bit decimal digit to form a 6 bit character. The 6 bit characters may be held in the arithmetic register, and if the 4 bit character field is larger than 10 digits, the high order characters (11-15) are lost. With an expand instruction 0101 00111000 becomes 000101 000011001000. When the contract instruction is executed, the accumulator is shifted right to form a 4 bit representation for each 6 bit character. No test is made for numeric information, and if alphanumeric information is held in the accumulator the zone portion of the character is eliminated. 011001 100011 000001 000101 with a contract instruction becomes 1001001100010101. Zeros are placed in the high order 10 positions of the arithmetic register.

Additional instructions are provided for more complex data transformations. These instructions are described in the memorandum titled, "The Editing System". The report on editing will also describe compare instructions for fields which do not occupy an integral number of Stretch words, and which do not necessarily have the same starting locations within a word.

G.M. Amdahl
E.M. Boehm
J.E. Griffith