PRODUCT PLANNING & MARKET ANALYSIS DIVISION

IBM

MEETING OR CONTACT REPORT

70-6076-0

Project: Stretch Mathematical Planning Meeting - No. 2	Date of Report: 9/26/56
Purpose of Meeting or Contact: To investigate the mathematical planning effort for Stretch.	Date of Meeting or Contact: 9/19-21/56
	Reported by: D. W. Sweeney
	Dept.: Product Planning
Place of Meeting or Contact: ☐ WHQ ☐ Phone Other: Los Alamos, New Mexico	Follow-up Date:

List Personnel Participating
Give Report (including next action)
Indicate Distribution

Group Present:

IBM	Los Alamos
F. Beckman	R. Bivens
R. Bemer	B. Carlson
F. Brooks	I. Cherry
J. Cocke	R. Frank
S. Dunwell	M. Goldstein
J. Griffith	T. Jordan
F. Johnston	H. Kolsky
B. Sarahan	R. Lazarus
D. Sweeney	M. Storm
	E. Voorhees
	D. Woods
	W. Worlton

On September 19, 1956, the group met at 9:00 a.m. at Los Alamos. Mr. S. Dunwell opened the meeting by saying that IBM was going to present some of the machine requirements, but that they were preliminary and although not definite would be a basis for further thought. He said that the word size being considered is 64 bits and that instructions will probably be indefinite in size (i.e. they will be made up of bits and pieces and will be as long as necessary to describe the operation to be performed). He felt that the main memory bus traffic problem would probably be one of the most critical aspects of the machine and that indexing by use of a special memory to contain the index registers might alleviate this problem. He mentioned the possibility of 0.2 μsec. core memory which would contain more words than the currently proposed sixteen 0.2 μsec. transistor registers.

He also stated that a number of operations might be performed directly in memory to prevent main memory but traffic. Some of these might be counting in memory, indexing, and index modification such as $j + 1 \longrightarrow j$ and $j + 1$ compared to J in a standard memory cycle.

He presented a possible method of coding for multiple level indexing, and table storage. The example shown was the summation, $\sum \alpha_{j,j'} A_{i-\frac{1}{2},j'}$ in which the programmer would write:

Reset Add α

subscript j

subscript j'

Multiply A

subscript $i-\frac{1}{2}$

subscript j'

In this method the tables of values of α and A would be arranged so that the table dimensions would be powers of 2. This allows the computation of the address of $\alpha_{j,j'}$ to be accomplished by shifting and adding rather than multiplication. Both Mr. E. Voorhees and Mr. R. Frank pointed out that this system has been used on the 704 but was definitely restricted particularly for large indices.

Mr. R. Lazarus pointed out that this method was good to obtain a random entry in a table, but that the majority of problems required a regular sequence of points to be investigated and this only required addition of indices for either rectangular or triangular data arrays.

Mr. S. Dunwell then mentioned that it would be useful to examine the coding system he proposed using combinations of full and half word instructions and data.

Mr. R. Lazarus then presented the thinking of the Los Alamos group about floating point operations.

1. Exponent should have both sign and magnitude. The ability to multiply by $2^{\pm n}$ should be a simple, logical exponent operation.

2. Exponent underflow should be handled by two modes of operation: (a) Error indication for special conditions, and (b) Set the value to zero with a possible indication of such action.

3. Exponent overflow should give an error indication except X/0 should give a separate error indication.

4. The handling of zero fractions should imply the mathematical operations with zero. (i.e. $X \pm 0 = X$, $X \cdot 0 = 0$, and $0/X = 0$). He noted that zero detection facilities implied a time saving.

5. Normalization should take minimum time with no lost information bits. The cost of implied unnormalized operations might be great for such operations as compare accumulator with storage, division, or pre-normalization of factors during multiplication or addition.

6. Double precision was investigated, and it was felt that if more flexible floating point operations were available programmed double precision operations would be about as fast as the same automatic operations. The time ratios presented were 6:4:1 for programmed double precision: automatic double precision: single precision.

He pointed out that there were certain considerations to using a number base larger than 2 for the exponent, but that the Los Alamos people in general did not favor it.

A general discussion followed about normalization and the treatment of unnormalized numbers. This seems to be the area of least definition of any floating point system. The question of address size was raised and Los Alamos seemed to feel that 16 bits was sufficient with indirect addressing for all over 32,000 words.

Mr. F. Beckman then presented some of the problems related to matrix work and the solution of simultaneous linear equations. The following are some of the figures presented from various fields to show the problem magnitude:

Econometrics	675th order system
Surveying	2400th order system
Aircraft Design	200th order system

(Lockheed would like to do up to 7000th order)

Statistics	50 x 1000 regression
Factor Analysis	100th order characteristic roots and vector solutions
Partial Differential Equations by implict methods	100th order system

He pointed out that most large matrix systems have many zero elements and that about the cube of the order is the number of operations required for solution. Zero detection may speed up the arithmetic.

Referring to some of the work done by Von Neuman and Goldstine, he estimated the word length requirements to be about 45 bits for the mantissa. He then discussed some of the problems of the numerical techniques involved in the calculation of characteristic roots and vectors and matrix inversion requiring "positioning for size" or iterative methods.

This ended the first day of the meeting. The next day Mr. R. Bivens presented a large Monte Carlo problem which was used to determine the events occurring when high energy particles enter an atomic nucleus. He stated that the problem was done in three dimensions with relativistic effects taken into account. The problem included conversion from laboratory to center-of-mass coordinates and vice-versa requiring about 15 square roots and the calculation of many cosines for each collision. The time for the square roots accounted for about 1/3 of the problem running time.

Storage was definitely a problem as there were about 3500 instructions in a 1024 word machine. He estimated that 10,000 words would have been adequate for the task. The problem results were stored on 40 tapes with about 16,000 words each, representing 640,000 events. Four analysis routines have been written and used on these results so far. He pointed out that about 12 random numbers were used per collision. A discussion followed about the generation of random numbers.

The flow diagram and description of the problem were very general, and a better problem statement and flow diagram were requested by IBM.

A small group then left the meeting to program and investigate the timing for full and half word instructions and data.

The rest of the group then heard a presentation by Mr. R. Bivens and Mr. R. Bemer of the problems and trends in automatic coding. Mr. R. Bivens discussed some of the work of the Los Alamos group. Mr. R. Bemer presented a brief history of the accomplishments of IBM and discussed the aims of programming research and the formulation of COMTRAN. He also discussed briefly the requirements of supervisory programs.

The group then split into three committees to consider data word format, instruction word format, and arithmetic instructions.

Late that afternoon the group met to discuss the first committee's results. Mr. S. Dunwell discussed the results of the programming of the inner loop of the problem presented by Mr. R. Lazarus at the previous meeting.

This problem was timed out on a chart for several different cases. In general, it was revealed that half word airthmetic (vs. full word arithmetic) saved .6μ sec. out of 3.8μ sec. assuming HW Mpy = 0.6μ sec. and FW Mpy = 1.2μ sec. Full word instructions were as fast as half word instructions and no difference in time was noted as a result of storing two instructions per word. However, it was evident that the problem was not very typical, and therefore, could not draw general conclusions from the timing chart of this one example, which was only seven instructions in length.

The next day (September 21st) the group meet to discuss the results from the other committee's. Mr. D. Woods presented the findings about instruction word format. They felt the operations should be in several classes with modifiers (4 groups of 4 bits each). The address should be tagged to indicate an immediate, direct, or indirect address [X, L(X), L(L(X)) or $L^n(X)$.] The main instruction came out to about 48 bits. The remaining bits could be used in other ways as specified below:

Operation	16 ± 4 bits
Address	16 bits
Address Modifier	2 bits
Index	10 bits
Return Address	4 bits
	48 ± 4 bits

Bit Address	6 bits
Field Length	6 bits
Character Length	3 bits
Second Index	8 bits
Break Point Indicator	Not Specified

The return address specifies one of the fast transistor registers for such operations as A · B + C → C.

The committee felt that these partitions of the instruction word would be sufficient but that further study was necessary. A discussion followed concerning the number of bits in the operation part for ease of decoding versus information content.

Mr. D. Sweeney then reported on arithmetic operations. The committee felt that the arithmetic operations could be classified to MS op A → → A (or S) where M meant sign manipulation of four kinds (use sign, invert sign, set plus, and set minus), A means Accumulator contents, S means storage contents and op means add, subtract, multiply, or divide. This set would probably be sufficient if an interchange indication could be given to have operations of the type MA op S → A (or S) performed. This is useful for sign manipulation as well as the operation S/A. (Current machines only have the operation A/S). The committee recommended that in every case that it made sense that there be reversible operations (i. e. if there is a Store Address order, there should be a corresponding Reset Add Address order). An automatic square root command was requested. An extensive set of commands to manipulate exponents and mantissas separately was recommended.

The committee further requested a study of the feasibility of a modified two address system in which A in the above equations would be any one of the 16 transistor registers.

Mr. R. Lazarus then reported the findings of the committee on data word format. The committee had restricted its work to floating point and felt that the format should be 42 bits for mantissa with sign, 9 bits for the exponent with sign, and at least 5 additional bits to be used for a zero indicator and boundary condition indicators. The 41 bit mantissa was felt to be a sufficient data length. The 8 bit exponent was chosen so that an early indication of exponent overflow or underflow would be given in cases of instability.

The group then discussed its future planning activity. Mr. R. Lazarus asked that whenever any particular machine functions or specifications are fixed because of other considerations that Los Alamos be notified so that they would not have to spend time investigating problems to which the solu-

tions are already fixed. The group agreed on a list of committee assignments. Committee members were to be assigned during the week of September 24-28. These small groups would keep in contact mainly by phone. The next meeting of the entire group was not set.

The committee assignments are as follows:

Performance
Instruction Word Format
Data Word Format
Floating Point
Arithmetic Operations
Indexing and Addressing
Automatic Programming
Input/Output

Mr. S. Dunwell then spoke of the organization of IBM Research, Product Development, and Stretch.