R. E. Meagher
206 South Fair Street
Champaign, Illinois

May 9, 1961

Mr. F. J. Cummiskey
International Business Machines Corporation
112 East Post Road
White Plains, New York

Dear Mr. Cummiskey:

I am here enclosing a report on my evaluation of the Los Alamos STRETCH machine based upon my discussions since April 18, 1961. A summary is also included.

In accordance with your request I will also look and comment on the improvement program associated with STRETCH as soon as it is more definitely formulated in the next few weeks. The enclosed report does not take any part of this proposed improvement program into account.

If this report does not cover the areas in question in sufficient detail, please let me know.

Sincerely,

R. E. Meagher

R. E. Meagher

REM:EB

Enclosure

# SUMMARY

There is no completely satisfactory method of evaluation and basically there cannot be one. However, problems do fall into standard categories and using these and simulation techniques it is worth trying to get better methods of evaluation.

Computers have achieved importance because of: 1) high speed and 2) the concept of a condition transfer or branch instruction. While STRETCH is fast in arithmetic, the conditional transfer is relatively slow.

No single number can be used to compare STRETCH speed to another machine unless the problem type is fairly well specified. STRETCH is a machine whose internal computing speed is from 0.8 to about 8 times that of the 7090 depending upon several factors. With moderately expert programming dominated in favor of add and multiply parallel unit instructions, the machine would average about 6 times the speed of the 7090 (if the word length and memory capacity are sufficient on a 7090). With really expert programming on such problems STRETCH can be 8 times the speed of the 7090. On programs which, because of necessity or inadequate knowledge of STRETCH, are dominated by logical instructions, STRETCH will be about the same speed as a 7090.

The instructions are more general than it is economical to have them and this generality makes the machine slower.

Checking features are valuable for the present reliability of the machine. Based upon initial test results only, STRETCH would not be considered "reliable" without the checking circuits and an effort should be made to see that the reliability without checking is improved.

It is important to note that the 7090, is an extremely good machine. Many of the STRETCH developments, including the biggest factor in circuit speed and the memory, are in STRETCH. By comparison, then, it is more difficult to show STRETCH as being greatly superior on non-ordinary instructions.

STRETCH was an "advanced development" kind of project but had many constraints. The effort to have Harvest, a commercial machine and STRETCH all the same general machine added to the difficulties of constructing STRETCH and is responsible, in part, for the cost and slowness of STRETCH on some instructions. Since the arithmetic speeds were specified, most effort was placed on these. STRETCH supported the development of many now standard IBM products or improvements.

The original plans called for circuits with an average delay of 10 nanoseconds per circuit element. Circuit engineers proposed and later produced circuits with delays of about 20 nanoseconds per element. There was possibly a misunderstanding by the systems planning group that these circuits could be used or that through further work and better transistors 10 nanosecond circuits could be obtained. The circuit speeds did not, in fact, become faster and hence there was a gap between the originally proposed speeds and those which could reasonably be obtained.

The part of STRETCH comprising I-Box, Look-Ahead and the controls connecting these to other parts of the machine is considered very complicated and more complicated than is desired for future machines. The innovations of standard interface junctions in machines is a desirable kind of complication for machines.

Planning groups did not follow through on the machine after its formative stages and no single engineer was assigned to STRETCH early enough.

# METHODS OF EVALUATION AND THE STRETCH SYSTEM

R. E. Meagher
May 9, 1961

## Introduction

The STRETCH system has the fastest operating arithmetic unit. Yet on some otherwise reasonable programs the STRETCH system would be slower than a 7090! How does one evaluate a computer like this and how did it all come about? This report discusses these topics. It consists of two parts. The first part considers methods of evaluation and examines the STRETCH system. The second part discusses the development of the STRETCH system.

## Part I

### Methods of Evaluation

One must understand at the outset that there exists no completely satisfactory method of evaluating a computer. Although the individual steps in computing are precise, the field of computing is broad and general. There is no "standard" problem. There is no general purpose machine. If a definite sequence of steps for a computation is stated, any computer could be examined in the light of these, of course. However, we do not want this because we seek always to solve a problem and more than one sequence of steps will give a solution to the same problem. Perhaps an evaluation could be carried out for a problem or a class of problems -- if we could agree on what the class of problems whould be. Evaluating a computer is somewhat like evaluating a man's life or like evaluating a book. It is impossible unless we have some standards and who is to say what the standards are. I hold no hope, therefore, for a completely satisfactory evaluating method, but it is worth striving for methods which can be used if the problems are defined.

The method of considering the distribution of instructions in the problems in question and then finding how long it would take a computer to execute these instructions is the most widely used method of evaluation. In a "balanced" machine, in the Von Neumann sense, this is certainly a good method. Most of the machines have been fairly similar in the past. For a given problem and with a good understanding of a machine to select the best instructions, it is still a good method, but the evaluation of a given machine may still vary considerably from one problem to the next.

Other Methods of Evaluation

For scientific problems it is possible, I think, to select about 10 representative problems which form the types most frequently used. After a machine is built, a very effective evaluation would consist in observing the operating times on these same problems with due regard for advantageous programming. This is certainly the final kind of proof of a machine's speed.

I am aware that simulation techniques can be carried out before a machine is built to get a measure of its speed. Dr. Harwood Kolsky did this, in part, for STRETCH and became aware that there were some difficulties. However the seriousness of the difficulties was not recognized. Perhaps this is because the numerial times assigned to each box were not correct.

Simulations techniques do have considerable prospects and I would approve of work on them. I would like to compare simulation results with tests on operating machines before applying them to new designs.

An Evaluation of STRETCH

I usually think of most problems, and especially "scientific" problems, as containing fairly short loops of essentially the following instruction types:

load a register from memory

add a number to the number in the register

multiply by another number from memory

conditional transfer or branch

(More complicated mixes including many regularly used at IBM are, I am sure, better than mine.)

Table I gives a list of the instruction operation times for a few of the main instructions in the 7030, the 7090 and the 704 for comparison. These times were obtained by running each of the individual instructions repetitively in STRETCH a very large number of times. The times are "average" (shifts and bit distribution affect these times) and also they represent times when the instruction listed is followed by the same instruction.

TABLE I

Mostly from T. C. Chen

| | 7030 | 7090 | 704 |
|---|---|---|---|
| Multiply | 2. 7 usec. | 28. 6 usec. | 204 usec. |
| Divide | 8 to 18 | 38 | 216 |
| Add | 1. 5 | 11. 0 | 84 |
| Clear and add | 1. 3 | 4. 4 | 24 |
| Store | 4. 8 | 4. 4 | 24 |
| Common branch | 2. 5 | 4. 4 | 24 |
| Branch on zero | 5 to 7 or 15 to 22 | 4. 4 | 24 |

In the case of Stretch these are average numbers except where a range is given.

The very great importance of the automatic computer has been attributed to two things: 1) high speed and 2) the "conditional transfer" or "branch" which depends upon the results of some previous calculation. It should be noted right here that the STRETCH can do arithimetic exceedingly fast, but on the second item, branch, STRETCH is actually slower than a 7090 (except for the simplest branch, namely, count-on-branch). This is a matter of great importance in evaluating STRETCH because the use of branch instructions has become so common and so accepted as a powerful feature of computing machines.

Let us now apply the instruction times to the simple loop I suggested. The results are given in Table II:

TABLE II

| | 7030 | 7090 | 704 |
|---|---|---|---|
| Clear Add | 1.3 usec. | 4.4 usec. | 24 usec. |
| Multiply | 2.7 | 29 | 204 |
| Store | 4.8 | 4.4 | 24 |
| Count and Branch | 2.5 | 4.4 | 24 |
| Total | 11.3 | 42.2 | 276 |
| Speed relative to 704 | 24 | 6.5 | 1 |
| Factor for word length of 60 bits | 36/60 | 1 | 1 |
| Word length adjusted "time" | 6.8 | 42.2 | 276 |
| Adjusted speed relative to 704 | 40 | 6.5 | 1 |

In table II the simple list using a count-on-branch (the shortest branch time in STRETCH) shows the STRETCH to be 24 times as fast as the 704. STRETCH has a word length which is much longer than the 7090 and from a mathematical viewpoint it is proper to say that the amount of computing done is proportional to word length. I have applied this factor as an adjustment. Then the STRETCH is 40 times the speed of the 704 on this simple list of important and frequently used instructions in scientific problems. Word length is important, but depend upon problem size, it is long enough or it isn't long enough for convergence to a correct answer. For "large" problems the word length of the 7090 is not enough and double precision steps must be used which would further show up the advantage of the 7030. Also double-precision, on the 7030, is obtained, except for storage time, in substantially the same time as the single precision 7030 instructions, but probably this is required so seldom that its value may not be important.

Now, the most important thing to note about Table II is that almost any other distribution of instructions than the one I have chosen would make the 7030 take longer in comparison to the 7090 and 704. This is because divisions and branches which depend upon the result in the arithmetic unit are relatively not as fast in STRETCH. Thus a problem with little arithmetic but mostly logical and branching operations (even when the branch is usually not taken) would be slower on STRETCH than on the 7090. Under these fairly extreme conditions, STRETCH would be about 0.8 times as fast as a 7090.

Thus no single number can serve to compare the STRETCH with another system. The comparison is very dependent on the fundamental type of problem (mostly arithmetic or mostly logic) and the programming. With programming which takes into account a knowledge of the machine, and on primarily large problems in arithmetic which can still fit into the 7090 memory, the STRETCH would be on the average about 6 times as fast as a 7090 (8 times in some cases) or about 40 times faster than a 704 (53 times in some cases). At the other extreme, that is, with programming which disfavors STRETCH or on certain problems which use logical orders in the VFL, use branching extensively, and where word length is of no importance, STRETCH would be about 0.8 times as fast as a 7090 and about 5 times as 704.

Other Factors of Importance

There are many other factors which affect the value of a system in some instances.

Word length has already been mentioned and has already been taken into account in the above estimates. From a practical standpoint, only in some big problems is the long word length necessary, but here it is necessary and STRETCH has it.

Instruction repetoire is important and STRETCH has many combination instructions so that the actual number of instructions for a given problem will be somewhat less than on a 7090. This effect, by itself, will not be a large effect, but may amount to a 25% reduction in number of instructions required as compared to a 7090. STRETCH also has an extremely flexible and complete list of instructions. This should in some cases make programming easier, but the resulting program is not necessarily a fast one. For example, the VFL unit contributes to this flexibility, but the VFL instructions are slow. For a machine for "scientific" problems one wonders why the VFL unit is there at all (except for the parts for exponent arithmetic), but it turns out that for checking reasons practically all logical instructions have been designed to use the VFL unit.

Instruction sequence affects the speed of the machine. In many cases groups of instructions can be carried out in a time less than or more than that indicated by the sum of the corresponding times obtained from Table I. This indicates the necessity of understanding the machine to write good programs for it. It is quite possible to incorporate some of these things as rules for a compiler such as FORTRAN. Also, of course, the compiler could be made to favor the fast instructions. This should certainly be done, but it should be remembered that some people do not wish to rely on a compiler and do not use them. STRETCH is an extremely complicated machine and fast programs will depend upon a knowledge of the machine. This is somewhat like minimum access programming on a drum machine. It is an undesirable feature to have to contend with. It places a premium on high quality programming.

Checking features are of importance because in STRETCH almost every machine error is detected and many are corrected. Thus a user can feel fairly certain that a result is not in error as far as the machine is concerned (and may give him a false sense of security into thinking his program is OK if he gets a result). All programs should have mathematical checks, but usually they do not. Problems should not need to be rerun for assurance of machine results but based upon recent tests about half of the problems longer than 45 minutes will show a check error and require rerunning for that reason.

Without the checking features and based upon these first few months of use, the STRETCH would not be considered reliable because an error would occur about every 3/4 hour. A serious effort should be made by IBM to see that the rate of errors, as shown by the checks, is reduced. Restart procedures, now proposed, do solve the problem but will have in my opinion a bad effect on IBM machine reliability reputation.

Memory capacity in STRETCH in both cores and disks is very large indeed. This is obviously an important item to large problems especially from a convenience standpoint. From a basic standpoint the Von Neumann versus Teller attitudes on this have never been settled.

Input-output features on STRETCH, like the memories, are excellent.

Conclusions on Evaluation of STRETCH

STRETCH is a machine whose internal computing speed is from 0.8 to about 8 times that of the 7090. With moderately expert programming dominated in favor of add and multiply parallel arithmetic unit instructions, the machine would usually be about 6 times the speed of a 7090. On programs which, because of necessity or inadequate knowledge of STRETCH operation, are dominated by VFL and branch instructions, STRETCH will be about the same speed as a 7090.

The relatively slow VFL and branch instructions compromise the value of STRETCH. I do not know how to place a numerical value on this. Because branch orders, at least in some quantity, are responsible for the present "automatic" computer, IBM should not underestimate the importance of this. At best these relatively slow orders place a restriction on the programming.

The very flexible and slow instructions in STRETCH are certainly not economical in equipment or relative rental cost.

The checking features in STRETCH appear to be necessary in order to obtain reliable results from the present circuits.

Any evaluation of STRETCH will be colored by the 7090. The easy part of the speed advance of these last few years is in the 7090. It is an extremely good computer. All of the developments in speed which are in the 7090 were started and sponsored for STRETCH. The 7090 has the electrical circuit advancements of STRETCH but with the more conventional organization. It will be hard, then, to show big improvements in speed in STRETCH on every type of instruction. Indeed, it is only through paralleling of operations achieved by a more complicated logic, that we can hope to get a higher speed. When this paralleling idea breaks down, STRETCH cannot be expected to be faster than the 7090. Indeed, if the normal pattern of data flow is upset, we must expect it to take just as long to get STRETCH started up again as it does to get the 7090 going. (It happens that sometimes it can take considerably longer on STRETCH because of the decision to return all registers to their former state before the branch occurred).

In short, the cream of the development of STRETCH is in the 7090 and the costly features, acknowledged to be more of an experiment, are in STRETCH.

# Part II

## The Original Requirements

STRETCH began, obviously, as an advanced development program which was intended to do everything anyone could think about in 1955 and 1956. The enthusiasm of engineers, engineering conjectures, proposals and promotional attitudes all affected the early work. To tie together such a large program and to support it, a large contract appeared to be necessary. A contract was completed with Los Alamos in November 1956 which called for the delivery of an actual machine in 42 months with the following specifications (among others):

Table III

| | |
|---|---|
| Base Frequency | 10 megacycles |
| Add or Subtract Fixed | 0.2 usec. |
| Add or Subtract, Floating | 0.6 usec. |
| Multiply | 1.2 usec. |
| Divide | 1.8 usec. |
| Disk Capacity | 1,000,000 words |
| Word Length | 60 bits |

6-6-3-1 average = 0.65 μs

These instruction times, which also appeared in technical papers, were arrived at by Mr. S. W. Dunwell with some sort of help from Dr. G. Amdahl. They were based upon a 10 megacycle clock pulse rate.

Both men expected or wanted about 10 elementary logical steps to be carried out in a 10 megacycle period of 100 nanoseconds. Hence, the individual logical steps could be allowed on the average 10 nanoseconds each. J. C. Logue wrote about 20 nanosecond circuits which may have been misinterpreted by Amdahl as a double logical step. In any case R. A. Henle, with experimental transistors, with light loads on experimental circuits, did soon produce circuits which had delays of 10 nanoseconds. These circuits were of the "current switching" and "non-saturating" type. They used PNP and NPN transistors in alternate steps, an excellent idea to avoid some potential changes. These circuits are, in general, excellent. They were later "standardized" and are now in the 7090.

The Conflicting Requirements of Harvest and a Commercial Machine

Although the requirements for STRETCH were pinned down by the contract, the pending contract on Harvest and the hopes for a commercial STRETCH soon became involved in the program. In fact, it was thought that all three requirements could be met by the same main program. Thus, the STRETCH was bound by the contract to a high speed main arithmetic unit and checking features and now it had to serve also two rather different needs, logically very different. The decision to make all three under one machine organization was certainly a complicating feature as far as the machine instruction list and control were concerned. I view this now as a mistake.

The Parallel Arithmetic Unit

Clearly the STRETCH contract was binding on arithmetic speeds and memory speeds and capacities. In trying to obtain these speeds E. Block and R. Merwin found that the current switching circuits were not fast enough, so the following steps were taken:

1. Higher circuit speeds were sought by using diode logic and emitter followers helped.
2. More complicated arithmetic logic.
3. Double card packaging.

Even with all of this the arithmetic was not going to be as fast as predicted in the original contract and a revision to the contract was made in July 1959. The revised times together with those actually achieved are given in Table IV.

6-6-31 ave 1.39 µs    2.3 µs

Table IV

| | Revised Contract * July 1959 | | Obtained March 1961 |
|---|---|---|---|
| Add or Subtract | 0.95 usec. | +58% = | 1.5 usec. |
| Load | 0.65 | +100% = | 1.3 |
| Multiply | 1.9 | +42% = | 2.7 |
| Divide | 7.0 | +14% to +157% = | 8 to 18 |
| Large Memory Cycle | 2.0 | +9% = | 2.18 |
| Disk Memory | $4 \times 10^6$ words | | $2 \times 10^6$ words |
| Mean-free-error time | 60 minutes or more | | |

instruction times subject to ± 25% error

* Los Alamos revised contract only. These are not the instruction times in Livermore or other recent proposals.

The group put its greatest effort on the parallel arithmetic unit because this was what was specified and because this was best understood. In spite of this effort the actual times were slower, as noted, by amounts which exceed the ± 25% allowed for in the contract.

The times for branch instructions were not specified in the contract and the desires for generality of instructions for the other two types of machines tended, I think, to let these instructions come about through a complicated interplay between I Box, look-ahead and the arithmetic unit.

It is true that the machine has fallen short of many of its contract specifications. However, it is the logical instructions and look-ahead which make it relatively slow and these were not explicitly stated in contracts.

It may be interesting to refer to the project Stretch Memo #39 which gives 6 papers which were presented at the IBM Engineering Research Conference on June 20, 1956. This series of papers gives the point of view of the Engineers and Planners associated with this system just prior to the contract. In the first paper Mr. Dunwell outlined the general form of the machine and gave some of the anticipated characteristics. He seemed to indicate clearly at that time that it was a development program when he make the following remarks:

> "Project Stretch has as its objective to put together the most advanced computer which can be developed within the next 3 1/2 years, and to have the first model of this machine in operation in the field at the end of that time, which will be January, 1960. Project Stretch is now a part of the Research organization, and will continue to remain in Research for the next year or year and a half. This provides us with a maximum

> opportunity to collect and apply to STRETCH the new ideas which are developing within the research organization. Similarly, it gives us a maximum opportunity to influence further research in the direction of solving the remaining problems which we encounter in laying out the machine."

Mr. Dunwell also gave machine speeds for addition and multiplication, but not division, which agree with the times quoted in the November 1956 Los Alamos Contract.

From this same series of papers it is interesting also to notice the opening remarks made by Mr. J. C. Logue. Mr. Logue had the principal responsibility for developing the dircuits which were later to go into STRETCH, and also the 7090. Quoting from his introduction:

> "The success or failure of this STRETCH program depends in a large measure on the success of the component effort. While this is true of every machine that has ever been built, it is particularly true of STRETCH. Other machines have incorporated one or two new components into their design, they in general, did not base this success on all new components. Many of the components that were used in past machines were considered initially to be standard items, but they turned out to be more complicated than first anticipated. In this respect the STRETCH program is different than past machine efforts. We cannot point to any component that we plan to use and say that its characteristics are known. Why then do we have the optimism, which everyone on the program exhibits to tackle such a task? One reason is that we feel the time is right for such a program. Another, and more important reason is that we feel we have an excellent physical research, component research, and manufacturing effort to back us up. Since the success or failure of STRETCH depends so strongly on research support the feasibility phase of the program has been initiated within the research organization."

I think Mr. Logue pointed out very effectively and clearly that the components and the circuits were by no means in hand. Indeed he indicated that there were prospects for new components but that these were not in hand. His enthusiasm was certainly justified at the time because the industry was beginning to understand magnetic components and had suggestions for faster transistors.

It is a puzzle to understand why IBM, which has traditionally been conservative in its approaches to proposals and commitments should enter into an agreement with Los Alamos for the delivery of a definite machine with fairly definite specifications as far as speed is concerned when the engineers, although enthusiastic, were not at all certain just how they would carry out this work.

The Complexity of the STRETCH System

The STRETCH system is a tremendously complicated system. Where it was possible in the past for many engineers to understand in a matter of a couple of days the flow of data for each instruction in a reasonably detailed way, it is doubtful whether a corresponding level of capability can be achieved for STRETCH in a couple of months by even a very competent engineer. With something as complicated as this, the design problem is very difficult indeed. The design problem was so great that very closely interrelated units in the control and arithmetic unit had to be handled by several different men. None could understand the whole and none could easily be aware of what a compromise in his unit would do to the over-all system.

What happened was that planning groups under Fred Brooks and Werner Buchholz formulated the general pattern and layout. Their job was difficult enough in trying to satisfy the requirements for three systems and to incorporate everyone's wishes for every kind of feature. After this planning stage engineers began to work. Because of the complications, engineers had to work on their own units and did not understand much more than their own units. The system as conceived was too large to build and compromises had to be made. These compromises had to be made without any single person in planning or engineering being aware of how they would affect the machine. Because the construction date was at hand the planning people did not get a chance to check the design and probably no single engineer really understood it until it was being built. I think that both R. Merwin and E. Block should have been called in earlier and the planning groups should have stayed with the project longer. Ideally a second iteration in the design should have taken place.

There is more than one kind of complication. STRETCH, I believe, introduced to IBM the idea of a standard interface between major units especially input-output units. This is a very good idea which I support. It is complicated but it is a simple kind of complication because the several major units do not have to work intimately together. In the control and sequencing steps through I-Box, Look-Ahead and Arithmetic Unit, this is a "complicated" kind of complication because it involves a very tight interplay of information. I view this particular complication with some concern.

In a limited way it is necessary in streaming where you know what is going on. But it is costly and should not, in my opinion, be the pattern regularly followed in "general purpose" machines.

STRETCH Is Too Generalized

It has already been remarked that STRETCH was really a part of a goal for three systems in one. Related to this but separate from it, STRETCH has an extremely generalized set of instructions. In some of these the value of one bit determines even the meaning or use of another bit or bits. These bits take time to sense. The trouble is that it takes time every time that instruction is used. The very general features of the instruction list are slowing down STRETCH. The desire to do everything is costly in time.

Personnel and IBM Organization

STRETCH has been led (and in effect sponsored on many occasions) by Mr. S. W. Dunwell. He is a very capable man in what I might call engineering promotional work. It is a direct result of his work that all of these new speeds could be introduced into the 7090, and 7070 and elsewhere. The STRETCH project was extremely large and needed his leadership. STRETCH is a success in having produced an arithmetic unit which is this fast. Also, it is a success for the innovations for the other machines.

STRETCH did not have competent working engineers on it early enough. Mr. Dunwell did not have constant and expert engineering help in the early period of the program and really did not have proper engineering help until about the early part of 1959.

Many outside of IBM, including those at the University of Illinois, were not able to substantiate the speeds quoted by IBM at that time by about a factor of 2.

Mr. Block is an extremely capable engineer but was not called in early enough. I think that Mr. Fletcher and others under him are very good.

On something as complicated as STRETCH, I think that there has to be one man who almost understands everything, and in particular knows and admits what he does not understand. Mr. Block appears to have grown into that position.

Organizational changes have had their influence. The sectioning off of the circuit work into a "standards" circuit group probably had a detrimental effect on STRETCH and a benefitial effect on the 7090. The 7090 program itself through its temporary priority slowed up the STRETCH program by some amount but the amount is uncertain.

STRETCH really was a developmental program. It was probably undesirable for it to leave "research" as soon as it did because it gave people inside the project and outside of the project the idea that research support was no longer needed. It was, in fact, a highly developmental program, something different from basic research of course, but also not quite a regular machine development program.

It is certainly desirable to have circuits which are "standardized" and to have a functional organization for this for production design work. STRETCH was not a production design but a new development and for those parts of the machine which are truly new there should have been strong capabilities within the STRETCH group.

IBM CONFIDENTIAL

May 31, 1961

Memorandum to: C. R. DeCarlo
B. O. Evans
J. A. Haddad
W. B. McWhirter
E. R. Piore
H. G. Kolsky

Subject: Dr. R. E. Meagher's STRETCH Evaluation

Dr. Meagher has requested that the attached sentence be added to his STRETCH report of May 9 at the end of the first paragraph on Page 12.

Joan Sicignano
Secretary to Mr. F. J. Cummiskey

JS

Attachment

ERRATA:

While lightly loaded experimental circuits had delays as low as 10 nanoseconds, loaded circuits on production boards had delays from 20 to 30 nanoseconds and were, therefore, slower than necessary to meet the original machine system speeds.

June 1, 1961

Memorandum to: C. R. DeCarlo
B. O. Evans
J. A. Haddad
H. G. Kolsky
W. B. McWhirter
E. R. Piore

Subject: Dr. R. E. Meagher's STRETCH Report

Please add the attached Appendix to Dr. Meagher's STRETCH report.

F. J. C.
F. J. Cummiskey

FJC:js

Attachment

Appendix 3
R. E. Meagher
May 26, 1961

# STRETCH RELIABILITY

The acceptance tests for STRETCH carried out at Los Alamos showed the error rate to be very much reduced as compared to the tests carried out in Poughkeepsie. In the main report I had indicated some concern because of the high "residue" error rate in the central processing unit. According to Mr. E. Bloch, there was only one such error in the 40 hours of acceptance tests at Los Alamos. The total number for the CPU was only a few and the total number for the entire system including all input-output units was about 35 for the same test period.

I understand that some of the panel cabling was repaired after shipment and this is probably what reduced the residue error rate. This is a very creditable improvement.

I think that this is a fairly low rate for the central processing unit and a reasonably low rate for the input-output units.

It is my understanding that substantially all of these errors were indicated by the extensive checking system and did not, of course, produce an incorrect result.