IBM Stretch (7030) -- Aggressive Uniprocessor Parallelism

Mark Smotherman

Last major update July 2010; minor addition January 2016

Summary: The IBM Stretch computer from the 1950s contains many high-performance design features that we usually think of as being associated only with current day superscalar microprocessors. Gene Amdahl and John Backus were influential in proposing an instruction "lookahead" approach to start memory fetches early and queue up operations for a fast arithmetic unit. John Cocke and Harwood Kolsky later helped refine the lookahead design by developing a timing simulator used in tradeoff studies. The register set and function unit partitioning as well as the resulting pre-execution of certain instructions in the instruction stream are ideas from Stretch that have influenced high-end processor design within IBM for decades.

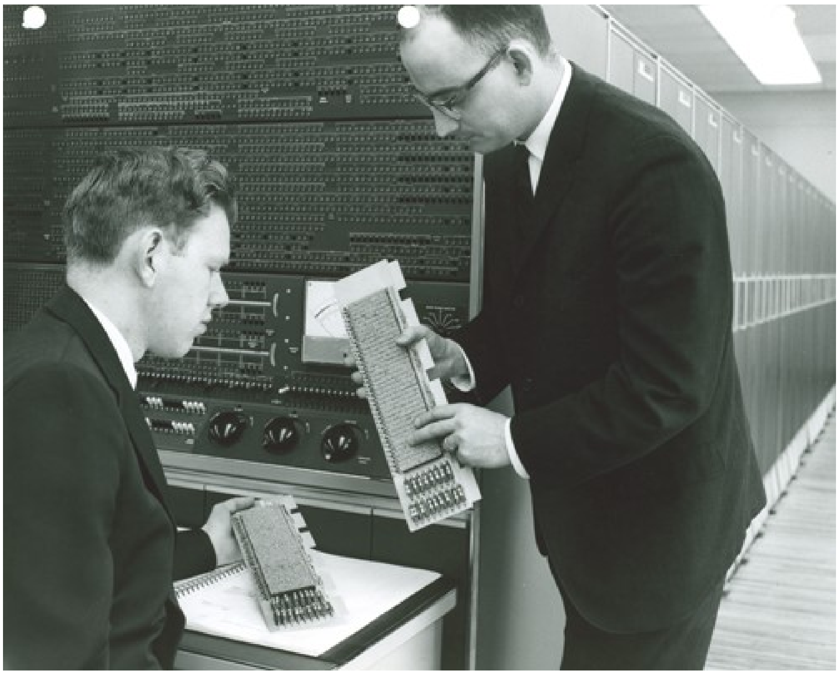

Figure 1. Two Stretch engineers in front of the eighteen frames forming the CPU and the attached maintenance console. (Image courtesy of Computer History Museum)

Introduction

> *STRETCH (the IBM 7030) is the largest, fastest, operating general purpose computer. It combines fixed word length arithmetic for performing floating point operations with the flexibility of variable word length arithmetic in which the words can be composed of "bytes" with from one to eight bits in a byte. The logic includes "look ahead" of instructions, in which as many as four instructions can be processed simultaneously. The arithmetic unit is extremely fast and the main memory has a two micro-second cycle time with multiple memory modules operating simultaneously. It is extremely difficult, however, to estimate accurate timing for individual instructions because of the complexity of "look ahead" plus the variable length of time required to execute VFL (variable field length) instructions. A reasonable estimate, however, is that approximately 1,000,000 instructions per second can be executed.*
>
> *from G.R. Trimble, "STRETCH," Computer Usage Communique, 1963 (pdf)*

The IBM Stretch is an amazing computer designed in the 1950s that included many advanced organization and microarchitecture techniques that would be considered aggressive by even today's standards. These techniques include predecoding, memory operand prefetch, pre-execution of certain instructions (a limited form of out-of-order execution), speculative execution based upon branch prediction, branch misprediction recovery, and precise interrupts. In many ways the Stretch organization of preprocessing the instruction stream to handle branches and memory loads as early as possible is a precursor of later high-end IBM mainframes (e.g., System/360 Model 91, System/370 Model 165, 3033, and 3090) as well as the IBM RS/6000 and PowerPC microprocessors.

The Stretch design had its roots in 1954 from initial studies in "advanced concepts" by Stephen Dunwell and Werner Buchholz, which became known as the "Datatron" memos [Bashe, et al., 1986]. Also in 1954, Nat Rochester, the architect of the 701 and at that time the engineer in charge of IBM's Electronic Data Processing Machines (EDPM), asked Gene Amdahl to design a new high-performance scientific computer in the new transistor technology. Amdahl recalls that Rochester "assured me that it would be my project and that I would get a development contract from either Livermore or Los Alamos" [personal communication, May 2005]. This was subsequent to Amdahl's work on the 704 [Norberg Interview with Amdahl, 1986/1989].

The project started formally after IBM lost an April 1955 bid on a high-performance decimal computer system for the University of California Radiation Laboratory (Livermore Lab). Univac, IBM's competitor and the dominant computer manufacturer at the time, had won the contract to build the 2-megacycle Livermore Automatic Research Computer (LARC) by promising delivery of the requested machine in 29 months. IBM's bid was based on a renegotiation clause for a machine that was four to five times faster than requested, cost $3.5M rather than the requested $2.5M, and proposed delivery in 42 months. In September 1955, IBM proposed a binary computer of "speed at least 100 times greater than that of existing machines" to the Los Alamos Scientific Laboratory and received formal approval of a $4.3M contract in November 1956 for what would become the Stretch computer. Delivery was slated for 1960.

Both Gene Amdahl and Stephen Dunwell were major contributors to the proposed design; but, when Dunwell was chosen at the end of 1955 to head the Stretch project, with Amdahl assigned a lesser role, Amdahl chose to leave the company. Dunwell subsequently recruited Fred Brooks, John Cocke, and Jim Pomerene in the summer of 1956 to join the project, and Harwood Kolsky from LASL joined the team in the summer of 1957. Robert Blosk and Gerrit Blaauw joined IBM in 1953 and 1955, respectively; both joined the Stretch team in 1957.

Blaauw and Brooks investigated key architecture ideas, such as the interrupt system and indexing; their Stretch experience would serve them well later in their careers as they worked on the IBM System/360. Blosk managed the design team for the Stretch indexing and instruction unit, which was essentially a pipelined computer on its own. Cocke and Kolsky constructed a timing simulator that would help the team explore organization options for the lookahead [Cocke and Kolsky, 1959], and Pomerene became the designer and engineering manager for the special-purpose Harvest system that was being built for the NSA. Erich Bloch, later to become Chief Scientist at IBM, was named engineering manager for Stretch in 1958 and led the implementation efforts on prototype units in that year and an engineering model in 1959.

Five test programs were selected for the timing simulations to help determine machine parameters: a hydrodynamics mesh problem, a Monte Carlo neutron-diffusion code, the inner loop of a second neutron diffusion code, a polynominal evaluation routine, and the inner loop of a matrix inversion routine. Several Stretch instructions were defined for scientific computation of this kind, such as branch on count and "cumulative multiply" (i.e., fused multiply and add). For the latter, Stretch expanded the classic 3-register von Neumann datapath by adding a "factor" register so that the innermost loop for matrix multiply requires only four instructions:

```
LOOP:  LOAD FACTOR NORMALIZED, 0(X4)
       MULTIPLY AND ADD NORMALIZED, 0(X5)
       ADD IMMEDIATE TO VALUE, X5, COLUMN LENGTH
       COUNT BRANCH AND REFILL PLUS, X4, LOOP
```

Figure 2. Matrix multiply on Stretch. [adapted from Table B.3, p. 298, Buchholz, 1962].

[Note: cumulative multiply was available in several early machines, including the EDSAC, Ferranti Mark I, ERA 1101, and IBM 650. A fast loop-closing instruction, transfer and increment index (TIX), was available on the IBM 704. The 1956 Ferranti Pegasus provided both cumulative multiply and a fast loop-closing instruction.]

Sidebar: Stretch/7030 Customers

Machine name	Built	Customer	Delivery
X-1	Poughkeepsie	Los Alamos Scientific Lab (LASL)	1961
K-1	Kingston	Livermore Radiation Lab (LRL) [now LLNL]	1961
K-2	Kingston	Atomic Weapons Research Establishment (AWRE), Aldermaston, UK	1962
K-3	Kingston	US Weather Bureau [now NWS]	1962
K-4	Kingston	Naval Weapons Lab (Dahlgren)	1962
K-5	Kingston	MITRE Corporation	1962
K-6	Kingston	Commissariat a l'Energie Atomique (CEA), France	1963
7950 (Harvest)	Poughkeepsie	National Security Agency (NSA)	1962

A ninth Stretch was built and kept by IBM.

The Lookahead Concept

> *control of logic is in parallel:*
> *- works ahead as much as 9 orders looking for indices, etc. ahead of arithmetic.*
> *- on transfers: will go ahead on "main branch"*
> *This multiplexing is automatic - needs no special coding.*
>
> *from Kolsky's 1955 notes*

Even though they are not usually associated with the Stretch project, Gene Amdahl and John Backus worked on the proposal to Livermore in 1955 and defined an instruction lookahead scheme called asynchronous non-sequential (ANS) control [Backus, "Computer System Design and ANS Control Techniques," October 26, 1955; Amdahl, "Logical Equations for ANS Decoder," December 13, 1955]. Cache memory would not be available until a decade later, so the basic lookahead approach was intended to start the slower memory operand fetches early and overlap them with the operation of the fast floating-point arithmetic unit.

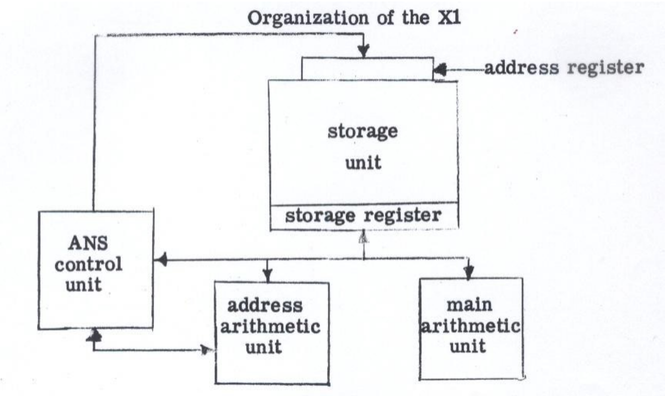

Figure 3. Machine organization in Backus paper. (Image courtesy of Manuscript Division, Library of Congress)

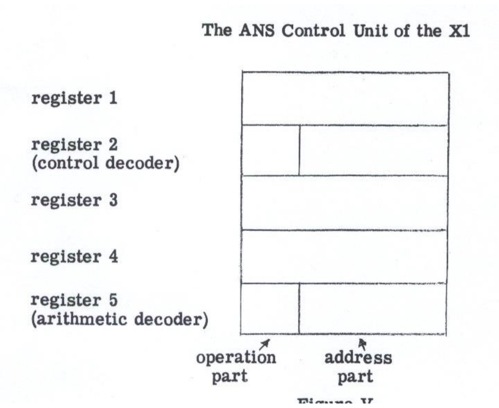

Figure 4. ANS control unit in Backus paper. (Image courtesy of Manuscript Division, Library of Congress)

Amdahl had worked on the initial plans for a high-end scientific computer beginning in November of 1954 and came up with an idea for instruction lookahead. [personal communication, May 2005; see also pp. 71-72 of Norberg Interview with Amdahl, 1986/1989]. Amdahl discussed his original idea for lookahead with John Backus "two or three times". "And John thought what I had proposed initially, he couldn't do a compiler for. So we went ahead and redid it. And we came out with the thing that was the look-ahead structure of the STRETCH." [p. 71, Norberg]. Amdahl recalls that "principally the look-ahead pre-fetched instructions to see branch instructions early enough so that we could get the succeeding instruction and data for each of the two alternative branch paths" [personal communication, May 2005].

In April 1955, Amdahl presented his design to Livermore and estimated it "to be about 40 to 50 times" the performance of the 704. Amdahl also participated in the presentation in May 1955 to Los Alamos. However, a struggle between Amdahl and Dunwell for direction of the design began in the summer of 1955, and Amdahl left IBM in December of 1955 when Dunwell was chosen to lead the Stretch project. (See sections 11.1 and 11.2 of [Bashe, et al., 1986] for more details regarding the start of the Stretch project.)

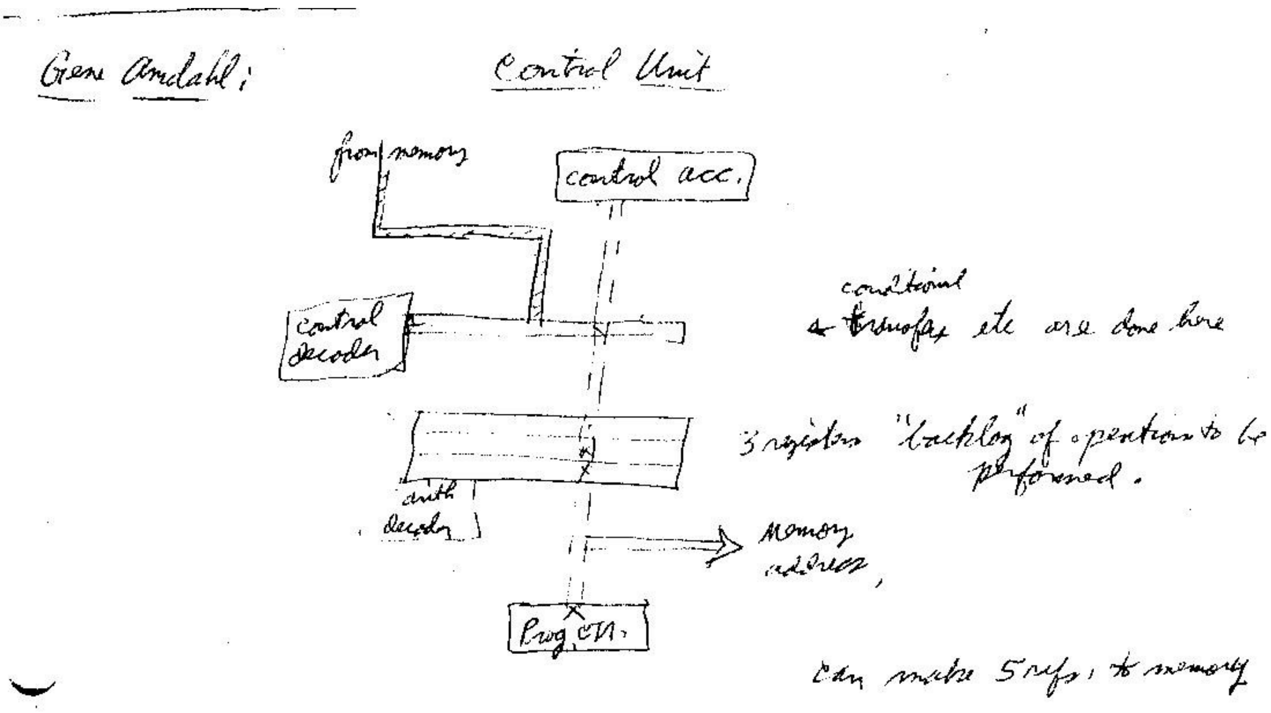
Gene Amdahl:
Control Unit
from memory
control acc.
control decoder
conditional & transfer etc are done here
3 registers "backlog" of operations to be performed.
arith. decoder
Memory address,
Prog. ctr.
Can make 5 refs. to memory

Figure 5. Lookahead control sketch in LASL proposal. [from Harwood Kolsky's notes of the September 20, 1955, presentation to LASL (Image courtesy of Computer History Museum)]

After Amdahl left IBM, John Cocke and Harwood Kolsky helped refine the lookahead concept and used simulations to explore the performance of various design options. Their work resulted in a choice of four levels of lookahead.

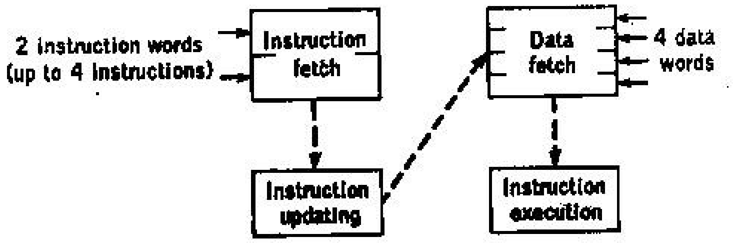

FIG. 14.3. Overlapped operation in 7030.

Figure 6. Overlapped operation. [p. 204, Buchholz]

Cocke and Kolsky's descriptions of the lookahead as implemented in Stretch describe it as both a prefetch and store buffer and also in cache-like terms. [Note that their use of the term "virtual memory" for the lookahead differs from our current-day definition of the term.]

- "The Virtual Memory fetches and receives the data required by the instruction and holds this data until the arithmetic unit is ready for it. The Virtual Memory also performs all store operations. It holds the data generated by the arithmetic unit or indexing arithmetic unit until the memory to which the data must be sent is available. Thus, the virtual memory acts not only as a 'look-ahead' for instructions to be fed to the arithmetic unit, but also acts as a 'look-behind' storage buffer." [Cocke and Kolsky, 1959]
- "The look-ahead unit is a speed-matching device interposed between the arithmetic unit and the memory. ... The time spent by the arithmetic unit waiting for an operand may be greatly reduced by 'looking' several instructions ahead of the one currently being executed. If the memory reference is initiated early enough, the operand will usually be availabe in a buffer register by the time the arithmetic unit is ready for it. Similarly, the arithmetic unit should be allowed to place a just-computed result into a buffer register for storing in memory while it proceeds with the next operation. ... The look-ahead unit may be described as a *virtual memory* for the arithmetic unit. The arithmetic unit communicates only with the look-ahead unit, not directly with the real memory; it receives instructions and operands from the look-ahead and returns its results there. The virtual memory, being small and fast, resembles in some respects the separate fast memory that was originally proposed for Project Stretch. It differs greatly, however, in that it takes care automatically of the housekeeping involved in the data and instruction transfers and thus avoids most of the extra time and all the difficult storage-allocation problems associated with a hierarchy of memories of different sizes and speeds." [pp. 228-229, Buchholz, 1962; emphasis in original; two types of core memory were described in the 1956 proposal - a 2 microsecond "Large Core Memory" and a 0.5 microsecond "Fast Core Memory" - along with "sixteen or more" single-word, 0.2 microsecond, transistorized registers called the "Ultra-Fast Memory"].

The lookahead unit as described by Cocke and Kolsky allowed pre-execution and speculative execution, as detailed in the next section. [Ralph Banhsen and Jules Dirac apparently did the detailed logic design. See their memo dated December of 1957 and patent issued in 1964. In his December 1959 EJCC paper, Bloch credits only Dirac for the lookahead unit.]

[Note: Konrad Zuse used a simple lookahead scheme in the Z4 in the 1940s to swap the order of two instructions and to eliminate repeated operand fetches.]

Sidebar: Backus acknowledgements

John Backus is known for his work in developing Fortran. He served in the Army during World War II and graduated from Columbia in 1950. He was hired by IBM in 1950 to work on programming the SSEC (Selective Sequence Electronic Calculator). His SSEC experience led to work on translators and interpreters, including the widely used Speedcoding. In 1953 he proposed a compiler approach for a user-oriented language for the IBM 704, which became known as Fortran. He received the ACM Turing Award in 1977 in part for this work, and the citation read: "For profound, influential, and lasting contributions to the design of practical high-level programming systems, notably through his work on FORTRAN, and for seminal publication of formal procedures for the specification of programming languages."

John Backus worked on the high-speed machine proposal to Livermore in early 1955, and wrote the "Computer System Design and ANS Control Techniques" paper in October of that year. In the acknowledgements, he wrote:

> The writer wishes to express his gratitude for the opportunity of working with the system design group which prepared the recent computer proposal for the Livermore Radiation Laboratory.
>
> Many of the concepts expressed in this paper were employed in the work done by this group. Among the many people contributing to the Livermore proposal, the writer worked most closely with the following: Dr. G. M. Amdahl, Mr. S. W. Dunwell, Mr. J. E. Griffith and Mr. J. W. Sheldon, consultant. Mr. Sheldon's suggestion to incorporate multiple storage units in the machine and to use uncomplicated instructions gave impetus to the need for finding new control techniques.
>
> Dr. Amdahl was largely responsible for designing a real ANS control unit in terms of the basic structure described in this paper.

Dr. Amdahl, Mr. Dunwell, and Mr. Griffith were working at Poughkeepsie at the time. John Griffith had worked at Livermore before joining IBM. He also worked with Gene Amdahl and Elaine Boehm on ideas for the IBM 709. John Sheldon had headed IBM's Technical Computing Bureau (later known as the Scientific Computing Service) in New York. While there he had recruited John Backus from the SSEC programming team to work on program translators and a floating-point interpretive system for the IBM 701. Sheldon left IBM in 1953 to pursue graduate studies at Columbia; in 1955, he co-founded Computer Usage Company. Bashe, et al., describes him in this way: "With a strong knowledge of both physics and mathematics and a natural aptitude for developing efficient problem-solving procedures, he was influential in establishing a balanced perception of programming at IBM." [Bashe, p. 655, n. 66]

Sidebar: Dunwell and Amdahl

	Stephen W. Dunwell (1913-1994)	**Gene M. Amdahl (1922-)**
employment at IBM	1934-1976	1952-1955, 1960-1970
known within IBM for	product planning	computer organization
projects up until 1955	WWII code-breaking system (US Army), 604, CPC, 650, TPM, 702, Datatron memos (w/ W. Buchholz)	WISC (Wisconsin), 704 (chief architect), 709

Stephen Dunwell started working for IBM Endicott in 1933 as a co-op student in Electrical Engineering at Antioch College of Ohio. He joined IBM as a full-time employee in 1934. In 1938 he transferred to IBM corporate headquarters in New York City and worked in the Future Demands group. He was recruited by the Army Signal Corps in 1942 to be technical director of the machine branch of a new US cryptographic center, attaining the rank of Lt. Colonel by the war's end. Mr. Dunwell returned to IBM headquarters and future product planning in 1946, and moved to Poughkeepsie in 1954 to work under T. Vincent Learson. In the Datamation article on Stretch, Harwood Kolsky says of Dunwell, "His real genius was that he saw where [IBM] should be five or ten years hence and was able to put together a huge project over the endless objections of everybody." In 1961, Mr. Dunwell was demoted to a staff position after Tom Watson, Jr., became angry that he wasn't properly informed about the problems in Stretch performance. In 1966, Mr. Dunwell was named an IBM Fellow, and Mr. Watson made a public apology at the IBM Awards Dinner.

Gene Amdahl taught electronics in the U.S. Navy during World War II and received a BS in Engineering Physics from South Dakota State University in 1948. He next went to the University of Wisconsin and received a PhD in Theoretical Physics in 1952. His dissertation described the WISC (Wisconsin Integrally Synchronized Computer), one of the first pipelined computers. Dr. Amdahl joined IBM in June 1952 and left in December 1955. John Griffith recalls that before Dr. Amdahl left in 1955, Jerry Haddad asked that all of Amdahl's ideas for Stretch be recorded; Griffith and Elaine Boehm, both of whom had worked with Amdahl previously on the design of the 709, coauthored a series of memos with Dr. Amdahl [see Stretch memos 4-5, 10-14, 16 at CHM]. After working at Ramo Wooldridge and at Aeronutronic, he returned to IBM in September 1960. He contributed to Project X, which later resulted in the System/360 Model 91, and then to the IBM System/360 architecture and the data paths for the various System/360 models. Bob Evans, a 33-year veteran of IBM and former division president of four divisions and corporate vice president, described him this way in James Strothman's 1990 article, "The Ancient History of System 360,": "Amdahl, in days before there were truly computer architects, was ... a brilliant architect. I have yet to see his peer. He could visualize what happens internally in a computer during the computational process ... and the flow of things during the solutions of problems."

Instruction Processing in Stretch

> *4.7 Control Decoder*
> *The stream of instructions which is to control the operation of the computer flows into a control decoder. The decoder examines the individual instructions to determine the nature of the action which is called for. It holds an instruction which may be several instructions in advance of the one being executed by the arithmetic system. This permits it to look ahead at the program and determine what preparatory steps must be taken in anticipation of the arithmetic operations to follow. These preparatory actions include, particularly, address modification, references to memory, modification of the contents of index registers and logical transfers in the program.*
>
> *from "Preliminary Description of Proposed Multiplex 10 Megapulse Automatic Computer," February 27, 1956 (pdf)*

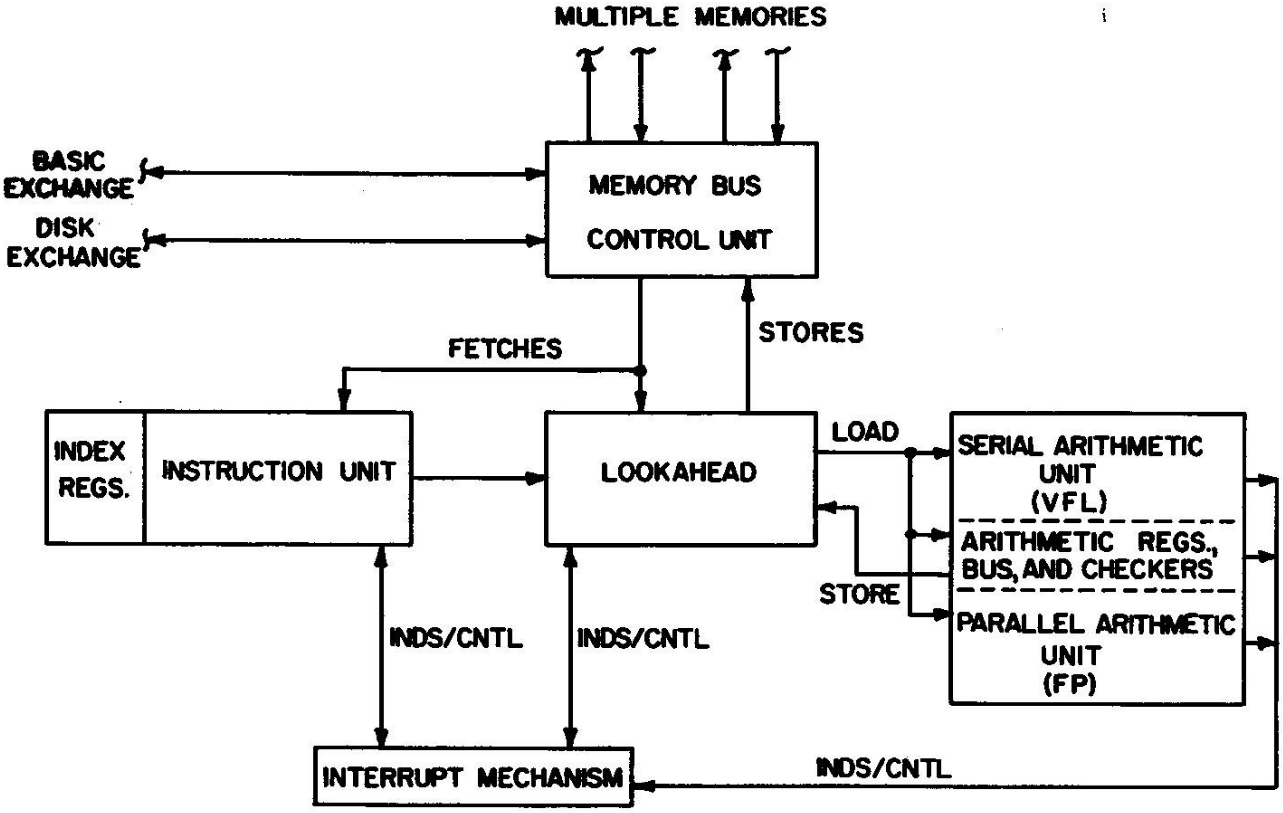

FIGURE 2. STRETCH COMPUTER

Figure 7. Block diagram of Stretch CPU components. [from Blosk, 1960]

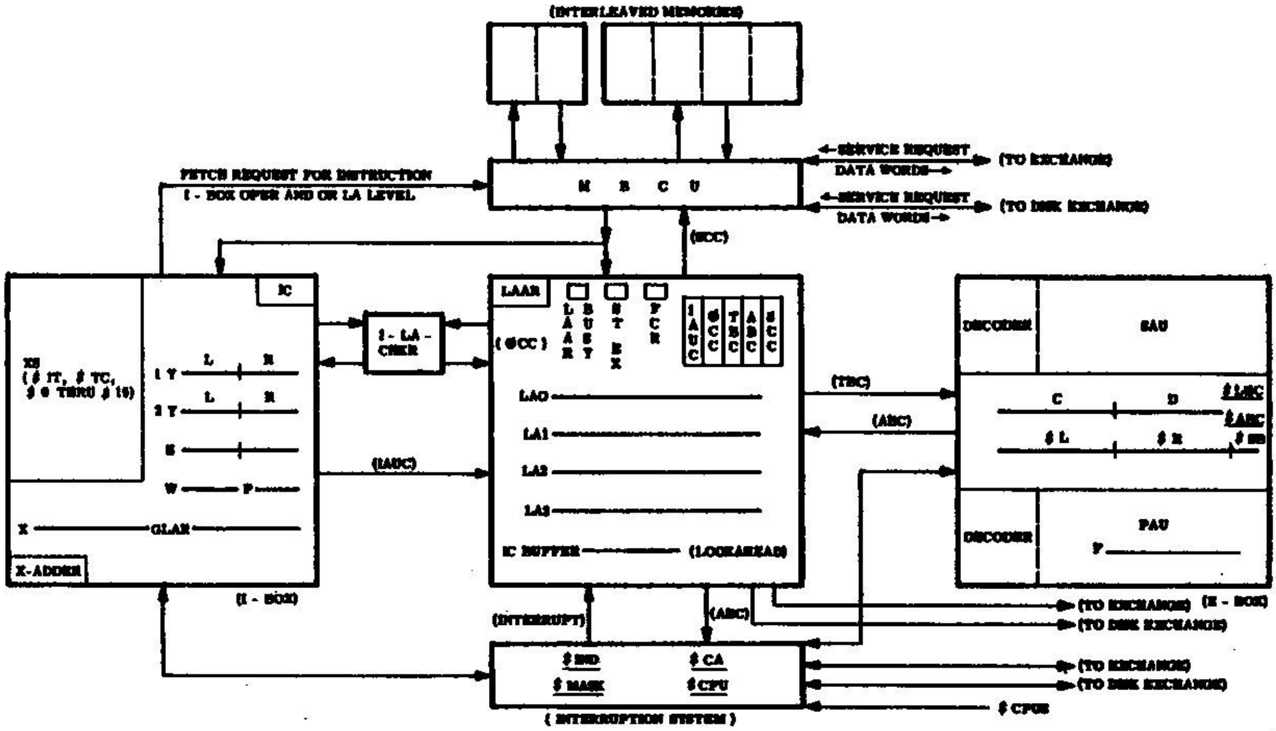

Figure A-1. The 7030 Data Processing Unit

Figure 8. Detailed diagram of Stretch CPU components showing four levels of lookahead (LA0-LA3, in center). [from Performance Characteristics, 1960]

Instructions in Stretch flowed through two processing elements: an indexing and instruction unit that fetched, predecoded, and partially executed the instruction stream, and an arithmetic unit that executed the remainder of the instructions. Stretch also partitioned its registers according to this organization: a set of 16 64-bit index registers was associated with the indexing and instruction unit, and a set of 64-bit accumulators and other registers were associated with the arithmetic unit.

The indexing and instruction unit of Stretch fetched 64-bit memory words into a two-word instruction buffer. Instructions could be either 32 or 64 bits in length, so up to four instructions could be buffered. The indexing and instruction unit directly executed indexing instructions and prepared arithmetic instructions by calculating effective addresses (i.e., adding index register contents to address fields) and starting memory operand fetches. The unit itself was a pipelined computer, and it decoded instructions in parallel with execution [Blosk, 1961]. One interesting feature of the instruction fetch logic was the addition of predecoding bits to all instructions; this was done one word at a time, so two half-word instructions could be predecoded in parallel.

Unconditional branches and conditional branches that depended on the state of the index registers could also be fully executed in the indexing and instruction unit. For the first two production Stretch computers (X-1 and K-1), conditional branches that depended on the state of the arithmetic registers were predicted untaken, and the untaken path was speculatively executed. (Subsequent models predicted taken or untaken based on the currently-available indicator status, with the assumption being that the indicator would be unlikely to change [p. 7, T.C. Chen memo (3), 1961]. Also, static branch prediction had been considered early on, using "guess bits" as part of the branch instructions, but rejected [Cocke and Kolsky, 1959].)

All instructions, either fully or partially executed (i.e., "prepared"), were placed into a novel form of buffering called a "lookahead" unit, which was contemporaneously called a "virtual memory" but which we would view today as a combination of a history buffer and a set of instruction reservation stations. A fully executed indexing instruction would be placed into one of four levels of lookahead along with its instruction address and the previous value of any index register that had been modified. This history of old values provided a way for the lookahead levels to be rolled back and thus restore the contents of the index registers on a mispredicted branch or interrupt. A partially executed arithmetic instruction would also be placed into a lookahead level along with its instruction address, and there it would wait for the completion of its memory operand fetch. Complex instructions were broken into separate parts and thus required multiple lookahead levels. (E.g., some VFL instructions required three levels when an operand crossed memory words [pp. 28-29, Johnson, 1959].)

An arithmetic instruction would be executed by the arithmetic unit whenever its lookahead level became the oldest and its memory operand was available. Arithmetic exceptions and external interrupts were made precise by causing a roll back of the lookahead levels (called "housecleaning mode"), just as in the case of a mispredicted branch.

Stores were also executed whenever their lookahead level became the oldest. For the first two Stretch computers (X-1 and K-1), store forwarding was implemented by checking the memory address to be read of each subsequent load placed in the lookahead levels; if that address matched the memory address to be written by the store (held in the Look-Ahead Address Register, LAAR), the load was cancelled and the store value was directly copied into the buffer reserved for the loaded value. Although allowing for multiple stores within the lookahead was considered in the early phases of the project, the design as implemented provided only one address register (LAAR), and thus only one outstanding store was allowed at a time. Also, because of potential instruction modification, the memory address to be written was compared to each of the instruction addresses in the lookahead levels. (Forwarding was disabled on subsequent models [p. 2, T.C. Chen memo (1), 1961].)

When no stores were present in the lookahead unit, the address register was used to implement a limited type of load forwarding. The register held the address of the most recently loaded operand; if a subsequent load address matched, the load was cancelled and the previously loaded value was copied. Thus, when possible, programs would be ordered to place repeated loads from the same memory word in successive instructions.

Compromises in the Design

> *During the fall of 1957 the Joint IBM-LASL planning group meetings became longer and more feverish. Arguments built up and tempers mounted. Most decisions were reached by the simple compromise of including both proposals in the machine. Only a few voices were raised protesting the complications which all these compromises might impose on the hardware, but they were shrugged off because there was no way of evaluating the cost of anything -- either dollars or performance. When the full horror of the engineering complications began to be felt during early 1959, as the detailed logical design of the boxes was being laid out, it was then "too late" to reconsider the logical structure of the computer ...*
>
> *from Kolsky's 1961 analysis of the project*

As in any project of this magnitude, the Stretch effort was affected by technological and financial challenges, as well as by personnel changes and by marketing and political pressures within the company. While budget reductions and personnel changes are to be expected, a major reorganization of the project was attempted in the spring of 1957 after the NSA requirements for Harvest were documented.

A group called the "Three-in-One" committee (consisting of Brooks, Blaauw, Codd, Griffith, Sweeney, and Wolensky) was formed with the intent of partitioning the Stretch and Harvest designs to obtain a subsetted, commercial machine offering with wider marketing appeal. (Gene Amdahl indicates that this was Dunwell's approach to the scientific computer design for LASL in the 1955 design struggles [Norberg Interview with Amdahl, 1986/1989].) This "basic" machine would consist of the addressing and indexing logic, the variable-field-length logic, and the I/O controls. Stretch would then become the basic computer plus a scientific processor (called "Sigma"), and in like manner Harvest would become the basic computer plus a specialized processor. Transistor count estimates were 65,000 for Basic, 65,000 for Sigma, and 110,000 for Harvest, with little duplication between the designs. The 3-in-1 approach was adopted in June 1957 but dropped by April of 1958. However, Harvest retained a Stretch plus special processor structure.

Bashe, et al., state that the 3-in-1 effort can be viewed in a positive light, as a time when "new ideas could be explored and documented" [p. 443, Bashe, 1986]. However, some of the design requirements frozen during this time and the decisions regarding the operation speeds of the logical and VFL components for the basic machine proved to have lasting damage -- for example, Kolsky attributes the poor performance of arithmetic branching to the after-effects of the partitioning effort [pp. 5-8 and point 4 on p. 12, Kolsky analysis, 1961].

In general, Stretch was an overly complex design, with features added early on without proper cost-benefit analysis. Many such design decisions were made by the joint LASL-IBM planning committee in 1957, but the timing simulation effort of Cocke and Kolsky did not start until late 1957. (Even then Kolsky reported that he felt the results of simulation were "generally disregarded" [point 8 on p. 13, Kolsky analysis, 1961].) Likewise, detailed logic designs were not started until mid-1958; and, when they revealed that some features were more complicated to implement than originally anticipated, it was too late to do a major instruction set redesign.

Stretch went through a series of transistor budget cuts. In April of 1958, the team was told to reduce transistor count by one-fourth, from 210,000 to 162,000. In June of 1958, Erich Bloch proposed a list of features to cut, including reducing the number of lookahead levels, to reduce transistor count from 140,000 to under 115,000. [cover memo; and detailed list, June 1958] In the fall of that year, Harwood Kolsky modified the timing simulator to account for the changes and reported back concerns regarding performance (some of which echo concerns he had noted in the spring of that year). Bloch dismissed the concerns and wrote a counter memo stating that, "the present design in my opinion is an optimum one."

It is a tribute to the Stretch engineers that they were able to implement such a complex machine design. Kolsky wrote in November of 1958, regarding early performance estimates for Stretch and the performance estimates of the revised timing simulator:

> Perhaps the most disturbing part of the comparisons ... are those between SIGMA and the "old STRETCH" as pictured in the hand-drawn timing charts of a year or two ago. What are the reasons for this factor of 2 or 3 reduction in performance? The following seem to be the main causes:
>
> (1) The fundamental transistor circuit speeds are slower by at least a factor of 2 than those originally postulated.
>
> (2) The memories are all slower -- particularly the index registers. Another example, the read-out time of the 2 µsec memory is presently 1.4 µsec instead of 0.8 µsec.
>
> (3) the early arithmetic speed estimates were unrealistic even with the proposed circuit speeds. The "1.8 µsec divide" is particularly hard to explain.
>
> (4) The "old STRETCH" estimates were really based, perhaps unconsciously, on much simpler designs than the present ones. Nothing resembling the intricacies of the interrupt system hardware, the VFL arithmetic, nor the present I-Box interlocks were ever considered in giving the "0.2 µsec indexing time".
>
> The fact that the overall performance has dropped by only a factor of 3 in view of these difficulties is greatly to the credit of the engineers.

Sidebar: Lookahead Timeline

- October 1954 - Datatron No. 0 memo by Dunwell and Buchholz, advocating that IBM "take a giant step" to achieve a "pre-eminent position" in the computer industry [p. 420, Bashe, et al.]
- December 1954 - discussions among Amdahl, Backus, Buchholz, Dunwell, C. Hurd (Director of Electronic Data Processing Machines), and others

- January 1955 - Buchholz writes a memo describing the design of Datatron machine with microprogrammed control and "a small fast memory (equivalent to multiple registers)" [p. 424, Bashe, et al.]; Hurd meets with Edward Teller at Livermore
- 1Q 1955 - discussions among Amdahl, Backus, Dunwell, Griffith, Sheldon, and others
- April 1955 - proposal to Livermore, with request that it be renegotiated on the basis of a 8-10 MHz machine
- August 1955 - first use of the term "Stretch" in reference to the 10-MHz design; R. Palmer and C. Hurd get approval to pursue supercomputer talks with potential customers
- September 1955 - presentation to Los Alamos
- October 1955 - Backus ANS paper
- December 1955 - Amdahl memory reference and ANS memos, Amdahl leaves IBM

- February 1956 - written proposal to Los Alamos
- April 1956 - Los Alamos recommends IBM (three companies submitted written proposals)
- Summer 1956 - John Cocke joins IBM
- September 1956 - first meeting of the Joint IBM-LASL Mathematical Planning Group; Los Alamos members included Harwood Kolsky
- November 1956 - LASL/IBM contract formally approved
- December 1956 - EJCC papers

- May 1957 - Harvest manual, summarizing requirements for the NSA; "3-in-1" design study begins
- June 1957 - "3-in-1" design adopted (Basic, Sigma, Harvest)
- August 1957 - Harwood Kolsky joins IBM
- 4Q 1957 - Cocke-Kolsky simulator developed
- December 1957 - last meeting of the LASL/IBM group - "design complete"; USAF SAC rejects "Basic" as too costly; Bahnsen and Dirac memo on lookahead system for Sigma

- April 1958 - "three-in-one" design dropped; "considerable engineering redesign of Sigma" [p. 443, Bashe, et al.] based on desired reduction in transistor count
- May 1958 - first Stretch operations manual; detailed logical design begins
- June 1958 - Bloch makes recommendations to reduce transistor count
- October 1958 - Kolsky revises simulator to reflect design changes
- November 1958 - Kolsky advises of performance problems; Bloch responds

- January 1959 - assembly of Stretch engineering model begins

- December 1959 - EJCC papers

- 1960 - several engineering improvements for Stretch considered under the name "SuperStretch"
- December 1960 - lookahead patent filed by Bahnsen and Dirac

- February-March 1961 - performance tests run by IBM and Livermore
- April 1961 - X-1 installed at LASL; IBM reduces prices on 7030 and withdraws it as a product
- May 1961 - Watson announcement at WJCC press conference
- 1961 - Stretch improvement program at Kingston; changes in lookahead operation

See also Harwood Kolsky's Stretch timeline

Performance Assessments of Stretch

> *In many of the problems for which it was intended the Stretch system outperforms the 704 by a factor of about 35 to one, while in some arithmetical problems the factor rises to perhaps 50 and in certain logical problems drops to about five. In a few problems which urgently require its large storage, long word-length and built-in checking, Stretch more than meets its design objective of outperforming the 704 by more than a hundred to one. ... Had it not been for the publicity and the competition provided by the 7090 (itself based on Stretch technology), Stretch might well have received unqualified acclaim.*
>
> *from "An Appraisal of the The Stretch System," Adams Associates, May 31, 1961 (pdf)*

The computer sections of the LASL Stretch contained 169,100 transistors, with the lookahead circuits taking 16% of the total [Fig. 14.14, p. 217, Buchholz, 1962]. The core memory sections provided 96K 64-bit words with 2.1 microsecond cycle time. The first 64K words of memory were 4-way interleaved, and the next 32K words were two-way interleaved. While not required, better performance could be obtained by allocating data in the first 64K words and instructions in the next 32K words. (Instruction set addressability allowed 256K words.) The computer sections dissipated 21 KW of power; and, these sections, without the memory banks, measured 30 feet by 6 feet by 5 feet.

Each major processing unit within Stretch had its own clock, and the clock cycle times ranged from 300 to 600 nanoseconds -- up from the initial clock cycle time estimate of 100 nanoseconds. The parallel arithmetic unit performed one floating-point add every 1.5 microseconds or one floating-point multiply every 2.7 microseconds. Up to six instructions could be in flight within the indexing unit, and up to five instructions could be in flight within the lookahead and parallel arithmetic unit. Thus up to eleven instructions could be in some stage of execution within Stretch at any one time.

As the clock cycle change indicates, Stretch did not live up to its initial performance promises; various estimates had ranged from 60 to 100 times the performance of a 704. (Early on, Dunwell had even used the estimate of 200 times a 704.) In 1960, product planners set a price of $13.5M for the product version of Stretch, called the 7030, for Livermore and estimated that its performance would be eight times the performance of a 7090, which was itself eight times the performance of a 704. This estimation was heavily based on arithmetic operation times.

When Stretch became operational in 1961, benchmarks indicated that it was from 0.8 to 10 times the performance of a 7090 [Meager analysis, 1961]. This difference was apparently due to store instruction delays and the misprediction recovery time required for taken arithmetic branches; both cases stalled the fast arithmetic unit. The precise, instantaneous interrupt design was also blamed for poor performance. As compared to the 7090, floating-point arithmetic on the Stretch was ten times faster, stores were the same speed, but taken -- thus mispredicted -- arithmetic branches on Stretch were five times slower. (On Stretch, a taken branch-on-bit takes up to almost 10 times longer than a floating-point multiply and up to 15 times longer than a floating-point add [see Table I, p. 4, of the Ralph Meagher report on Stretch, 1961].) As a result, the high-performance System/360 Model 91 would be designed in the mid-1960s with imprecise interrupts and with branch target path instruction prefetch but no speculative execution. It wasn't until the 3090 design in the mid-1980s that some of the more aggressive Stretch lookahead features reappeared. (See the table below.)

Even though Stretch was the fastest computer in the world (and remained so until 1964), the performance difference caused considerable embarrassment for IBM, and in May of 1961, Tom Watson, Jr., announced a price cut of the 7030s under negotiation to $7.78M and immediate withdrawal of the product from further sales. A Stretch improvement program was instituted in Kingston after the benchmarking results, and several design changes were made, including the branch prediction and store forwarding changes mentioned earlier [Chen memos (1) and (3), 1961], as well as reducing the number of lookahead levels required by the cumulative multiply from two to one [p. 1, T.C. Chen memo (2), 1961]. Also installed on K-2 and subsequent models was a multiplier register, with the result that "The arithmetic time in matrix multiplications is cut in half." [p. 3, Chen memo (2)]

The Legacy of Stretch

> *The 1961 Stretch computer had a phenomenal list of "firsts." But the benefits to IBM extended far beyond that Ferrari of a machine.*
>
> *from Edward Yasaki, "Fastest in its Time," Datamation, 1982 (pdf)*

While Stretch turned out to be slower than expected and was delivered a year later than planned, it provided IBM with enormous advances in transistor design (work on Stretch circuits allowed IBM to deliver the first of the popular 7090 series 13 months after the initial contract in 1958) and computer organization principles (multiprogramming, memory protection, generalized interrupts, the 8-bit byte, and other ideas originated in Stretch and were subsequently used in the System/360).

The Stretch design also influenced instruction sets and processor design within IBM for decades.

- Instruction set
  - The fast loop closer (branch on count) is used in the System/360-370 architecture.
  - The fast loop closer as well as the cumulative multiply are used in the RS/6000 and PowerPC architectures.

- Partitioned function units and lookahead
  - The high-performance IBM mainframes all use two processing elements, I and E, with some form of data prefetch activity in the I element. The 3090 is the closest to the original Stretch design:
    - The I element prefetches data operands and directs them to buffers in the E-element.
    - The I element pre-executes loop-closing and load-address instructions.
    - There is a four-element queue between the I element and E element to place decoded instructions.
    - The E element has the only copies of the floating-point and control registers but both the I and E elements have coherent copies of the general register file.
  - Three partitioned function units with separate register sets appear on the POWER (e.g., RS/6000) and PowerPC architectures: branch unit, integer (fixed-point) unit, and floating-point unit. Most models of the PowerPC provide instruction reservation stations or queues as well as completion buffers for generalized out-of-order execution and speculative execution.

	pre-execution	**inst. prefetch paths**	**branch prediction**	**speculative execution**
Stretch	indexing insts. (including index branches)	one	non-index conditional branches predicted untaken	yes, recovery using lookahead rollback
91		two (instruction stack and "BTB" prefetch buffers)	predicted insts. stall at execution units	no
85/165/168		two	stall in IQ	no
3033		three	stall in IQ	no
3090	LA, BCT, BXH, BXLE	three	BHT (called "DHT")	yes, flush to recover
9000		one	4K-entry BTAC (called "BHT")	yes, insts. tagged with two conditional bits
z900	LA, LD, index and some linking branches	one	8K-entry BTB	yes, flush to recover
z990	LA	five	8K-entry BTB	yes, flush to recover

See a discussion of eager execution, which has some more information about IBM I and E element designs for specific processors. Also see "Was Stretch Superscalar?" for a discussion of whether Stretch could be considered a superscalar design.

Selected Published Stretch References

- Charles Bashe, Lyle Johnson, John Palmer, and Emerson Pugh. IBM's Early Computers. Cambridge, MA: MIT Press, 1986.

- "IBM Stretch," section 13.3 of G.A. Blaauw and F.P. Brooks, Jr., Computer Architecture: Concepts and Evolution. Reading, MA: Addison Wesley, 1997.

- Erich Bloch, "The Engineering Design of the Stretch Computer," Proc. IRE/AIEE/ACM Eastern Joint Computer Conference, Boston, December 1959, pp. 48-58. (scanned pdf version at bitsavers mirror)
- R.T. Blosk, "The Instruction Unit of the Stretch Computer," Proc. IRE/AIEE/ACM Eastern Joint Computer Conference, New York, December 1960, pp. 299-324. [derived from May 1960 TR00.722 report below]
- Frederick P. Brooks, Jr., "Stretch-ing Is Great Exercise - It Gets You in Shape to Win," IEEE Annals of the History of Computing, January 2010, pp. 4-9. (abstract and link)
- Werner Buchholz, editor. Planning A Computer System, McGraw-Hill, 1962. (scanned pdf version at ed-thelen.org, 10.4 MB)
- John Cocke and Harwood Kolsky, "The Virtual Memory in the Stretch Computer," Proc. IRE/AIEE/ACM Eastern Joint Computer Conference, Boston, December 1959, pp. 82-93. (scanned pdf version at bitsavers mirror) [Note that this paper is the basis of Chapter 15 in the Buchholz book but the book updates it with "a simplified description by R.S. Ballance of the actual look-ahead unit as it exists in the Los Alamos system."]
- Dag Spicer, "It's Not Easy Being Green (or "Red"): The IBM Stretch Project," Dr. Dobb's Journal, April 1, 2000. (on-line version)
- Edward Yasaki, "Fastest in its Time," Datamation, January 1982. (scanned pdf version at CHM)

See Eric Smith's site for a more extensive bibliography.

Library of Congress

- John Backus, "Computer System Design and ANS Control Techniques," IBM internal paper, October 26, 1955. Item No. 86, Box No. 2, John W. Backus Papers, Manuscript Division, Library of Congress, Washington, D.C.

On-line Stretch Resources

Computer History Museum

- Stretch timeline
- CHM search page for Kolsky's collection of 900+ Stretch-related documents
- Selected CHM documents
    - Harwood Kolsky, notes from first AEC meeting on Stretch, 1955 (pdf)
    - Gene Amdahl, "Logical Equations for ANS Decoder," 1955 (pdf)
    - "Preliminary Description of Proposed Multiplex 10 Megapulse Automatic Computer," 1956 (pdf)
    - Fred Brooks, et al., report of the 3-in-1 committee (pdf)
    - Ralph Bahnsen and Jules Dirac, "Proposal for a Sigma Lookahead System," 1957 (pdf)
    - Harwood Kolsky, (second) memo on Sigma timing, March 12, 1958 (pdf); report on meeting regarding Kolsky's memo, April 1, 1958 (pdf)
    - Kolsky's notes from Stretch Group Meeting, "New shift in emphasis in project," April 10, 1958
    - Harwood Kolsky, memo on branching on arithmetic results in Sigma, May 19 1958 (pdf)
    - Erich Bloch, proposal to reduce transistor count, June 5, 1958 (pdf); detailed list of simplifications, June 2, 1958 (pdf)
    - Harwood Kolsky, memo on revised simulator, November 7, 1958 (pdf)
    - Harwood Kolsky, Sigma performance compared to other computers, November 20, 1958 (pdf)
    - Bloch response to Kolsky's concerns, November 24, 1958 (pdf)
    - Lyle Johnson, "A Description of Stretch," 1959 (pdf)
    - Harwood Kolsky's analysis of the Stretch project, 1961 (pdf)
    - Adams Associates analysis, 1961 (pdf)
    - Ralph Meagher analysis, 1961 (pdf)
    - T.C. Chen memo (1) on floating-point improvements, Aug. 28, 1961 (pdf)
    - T.C. Chen memo (2) on multiplier register, Sept. 7, 1961 (pdf)
    - T.C. Chen memo (3) on floating-point improvements, Sept. 12, 1961 (pdf)
    - Dag Spicer, Interview with John Griffith, Sept. 28, 2002 (doc)

Stretch manuals and reports at bitsavers (links to textfiles.com mirror)

- 7030 Reference Manual, August 1961 (pdf, 27.4 MB)
- 7030 General Information Manual, 1960 (pdf, 3.8 MB)

- R.T. Blosk, "Design and Performance Goals of the STRETCH Computer Instruction Unit," TR00.722, May 1960 (pdf, 3.5 MB)
- 7030 Performance Characteristics, Vol. 1, March 1961 (pdf, 2.8 MB)

Other links

- Eric Smith, IBM Stretch (aka IBM 7030 Data Processing System)
- Stretch: The Technological Link Between Yesterday and Tomorrow (13 minute video, BYU, 1981)
- IBM Archives page on Stretch
- Harwood Kolsky interview -- memories of John Cocke, May 1990
- Arthur Norberg, Interview with Gene Amdahl, Charles Babbage Institute, 1986/1989.
- "John Cocke: A Retrospective by Friends," 1990, 23 minute video by Mary S. Van Deusen and transcript
- Stretch/Harvest Reunion, September 28-29, 2002
- picture of Dunwell and Bloch standing in front of a real Stretch, discussing a scale model (jpeg)
- See also the IBM ACS timeline which contains some important dates for Stretch activities.

System-Level Stretch Patents (preliminary list)

- US Patent 3,048,332, F.P. Brooks and D.W. Sweeny, Program Interrupt System [179 pp., filed 12/9/57, granted 8/7/62]
- US Patent 3,051,387, J.H. Pomerene and J. Cocke, Asynchronous Adder-Subtractor System
- US Patent 3,058,656, J.H. Pomerene, Asynchronous Add-Subtract System
- US Patent 3,108,256, W. Buchholz and L.E. Kanter, Logical Clearing of Memory Devices
- US Patent 3,108,257, W. Buchholz, Locking and Unlocking of Memory Devices
- US Patent 3,145,296, D.W. Sweeney, Divider Device for Skipping a String of Zeros or Radix-Minus-One Digits
- US Patent 3,156,897, R.J. Bahnsen and J.F. Dirac, Data Processing System with Look Ahead Feature [68 pp., filed 12/1/60, granted 11/10/64]
- US Patent 3,202,971, G.A. Blaauw, Data Processing System Programmed by Instruction and Associated Control Words Including Word Address Modification [609 pp., filed 12/30/1960, granted 8/24/1965]
- US Patent 3,231,862, R.T. Blosk, E.D. Foss, R.E. Merwin, and J.H. Pomerene, Memory Bus Control Unit [113 pp., filed 12/30/60, granted 1/25/66]
- US Patent 3,427,592, R.J. Bahnsen and J. Cocke, Data Processing System [13 pp., improving indexing, continuation of original 12/9/59 filing, granted 2/11/69]

Acknowledgements

This sketch of the organization of Stretch has been revised over several years with the gracious help of the late John Cocke, and Gene Amdahl, Fred Brooks, Norman Hardy, Harwood Kolsky, George Michael, and Stuart Tucker. I very much appreciate their assistance, and I regret if I have misunderstood or misstated anything about the design.

The collection of papers donated by Harwood Kolsky to the Computer History Museum has been an invaluable resource in understanding the history of the Stretch project. The Bashe, Johnson, Palmer, and Pugh book has also been an excellent resource for the history of the Stretch project as well as for other IBM computers.

Postscript -- Some Stretch "War Stories"

Story from Bashe, et al., p. 442

> For step-by-step simulation of unit activity, [Kolsky and Cocke] decided, a step should represent 0.1 microsecond. Given about twenty units to track, clerical simulation was ruled out by the implied volume of tallying, and so their simulation method was programmed for an IBM 704 in November 1957. During the execution of their simulator program on a Stretch kernel, one user option was to print raw results as a table in which the typical column corresponded to a given unit and each row to a step. Each printed line displayed the state of all the units at the beginning of the step, and simulation of a short Stretch kernel could easily yield a listing 50 feet in length. Very soon, the standard procedure became to print summary statistics instead of raw results.

Story from Dag Spicer's interview of Stuart Tucker:

> I remember one absolutely marvelous event when we were probably within a few months of shipping to Los Alamos, and we tracked a bug down on the floor, to the fact that the bus between the arithmetic

> checker and the registers, which were packaged in the VFL unit, had a conflict. We were trying to use it for one of our instructions to move things over to the floating point unit, and totally asynchronously, someone else was trying to use it. There just wasn't any way to synchronize that, and we managed to wire in, find a spare circuit on each of the 128-bit register cards, maybe it was just 64 we needed, and put in a second bus. It was over a thousand wire change as I recall [laughs], which I worked on most of the weekend. I was sitting there watching them wire it and there was something about the pattern of the wiring that said "Oh, I forgot a parity or something". I went back and wrote up another thirty wire change for that - and the whole change came up first time without a bug. But what was interesting is the fact that it was a case where we just hadn't properly dealt with this as asynchronous. There was no connection between when one unit wanted to use that bus and when another unit wanted to use it, and we just couldn't find any way to interlock them. So we had to put in a whole separate bus for it.

Story from Gerard Paul in Wasn't That a Time: Stretch/Harvest Retrospectives (Stretch Reunion book), 2002:

> There was a battery operated phone system so we could talk from one end of the computer to the other without shouting [a distance of 30 feet]. It was in full view, but we didn't use it much - we shouted. One day a visitor thought the battery was used to power the computer.

Story from T.C. Chen in Wasn't That a Time: Stretch/Harvest Retrospectives (Stretch Reunion book), 2002:

> At the first Stretch User's Meeting held at Los Alamos in 1962, which almost became the last, LASL scientists reported on their experience using the machine, and rated it as 2 times the 7090, far from the original aim of 100 times the 704. The room fell deathly silent, and I found myself raising the only dissenting voice, challenging their programming methods, particularly their use of programming tricks, which backfired under the new and unusual architecture.

Story from Dick Holleran, 2010:

> By the way, the way performance was measured in those days was primitive. It consisted of measuring individual arithmentic instructions with a "kernel" of a grouping of several commands, i.e., add, multiply, divide in varying amounts. An anecdote: One of the key Los Alamos software guys wrote a program to measure individual instruction performance. He was astounded (and was extremely vocally upset) when his measurements found out that an "add" or "multiply" would have varying "performance" depending on how the instruction was surrounded by other instructions, particularly branches, and even whether the instruction stream was on half or full word memory boundaries! You could hear him screaming at Eric Bloch who then had called me to come "explain"!

Story from Bob Ramey in Wasn't That a Time: Stretch/Harvest Retrospectives (Stretch Reunion book), 2002:

> My first STRETCH run was an Eigenvalue problem which ran about 10 minutes on the 704. The program was entered on punched cards through the card reader. Pushed "Start" and almost immediately after reading the last card, the "halt" light came on. After a half an hour or so of debugging and poking around on the console, we suddenly realized the program had simply completed successfully! My introduction to 7030 performance.

Assessment of branching in Stretch by Jim Pomerene in "Historical perspectives on computers: components," AFIPS FJCC, 1972, pp. 977-983:

> Ambitious though it was, the two microsecond cycle [time of memory] fell far short of matching transistor speeds. In STRETCH, for example, the logic cycle was 300 nanoseconds, making the memory cycle time seven times greater. In order to offset the speed imbalance the concept of lookahead was introduced. The memory would be kept as busy as possible supplying the next few instructions and operands in anticipation of their use. Unfortunately the critical importance of the branch instruction was not fully recognized. At a branch the program may take one of two paths and if the lookahead had gone down the wrong path considerable unwinding was necessary. The problem proved to be quite fundamental and had a strong effect on high performance machine organization.

[History page] [Mark's homepage] [CPSC homepage] [Clemson Univ. homepage]

mark@cs.clemson.edu