

**ARCHIVE OF DOCUMENTS AND REFERENCE MATERIALS  
REGARDING THE IBM ACS-1 MACHINE**

**Lynn Conway\***  
February 16, 1999

This volume contains documents and reference materials that I have compiled regarding the IBM Advanced Computing Systems ACS-1 supercomputer. These are copies of original documents dating back to the ACS project itself. Taken together, they may be sufficient to disclose many of the system architectural innovations of the ACS architecture team.

The front-matter for the archive contains a brief, but important overview, of each document, including some details regarding the document's context within the ACS project. Also included is my initial letter to Dr. Mark Smotherman of Clemson University regarding the possibilities of reconstruction of many details of the ACS-1 machine.

<b>CONTENTS:</b>	<b>PAGE:</b>
i. Overviews of Archive Papers and Documents.	i. 1
ii. Letter to Dr. Smotherman, January 2, 1999.	ii. 1
1. "Dynamic Instruction Scheduling", February 23, 1966.	001
2. "ACS Simulation Technique", March 15, 1966.	022
3. "Dual Arithmetic on ACS-1", May 1, 1967.	051
4. "Architecturally Critical Paths in the MPM", May 12, 1967.	054
5. "MPM Timing Simulation", August 25, 1967.	059
6. MPM Architecture and Simulator Reference Notebook, as of August 1967.	093
7. Timing Simulator Source Code Listings, as of August 1967.	211
8. "ACS Logic Design Conventions: A Guide for the Novice", November 29, 1967.	328
9. "A Proposed ACS Logic Simulation System". October 31, 1967.	347
10. "The Computer Design Process: A Proposed Plan for ACS", August 6, 1968.	367

\* My name was legally changed to Lynn Conway on January 30, 1969. Since I am widely known under my new name, we've chosen to use it on my earlier papers in this archive.

## 1. "Dynamic Instruction Scheduling", February 23, 1966:

L. Conway, B. Randell, D. Rozenberg, D. Senzig

The background on this paper is as follows. Sometime in late '65, I suddenly visualized a solution to the general multi-issuance and conflict-resolution problem. I quickly compiled block diagrams and notes to capture the ideas, and during the next few days I presented these ideas in staff meetings in the architecture group. There was a rapid, very positive reaction. I was tasked to document the ideas in more detail, to incorporate one of the branching schemes then under study, and to turn the scheme into an architectural "proposal".

Since I was quite junior and had little experience with coordinating and writing ACS proposals, I worked with a number of ACS staff members, including Don Rozenberg, Brian Randell, Don Senzig and others to produce the resulting paper. There was a sense that these weren't just ordinary ideas, and we worked hard to frame the concepts in a tutorial form, so that they would be clear to team members. Brian Randell in particular came up with some wonderful articulations about the DIS schemes, in his inimitable British manner. We hoped to be able to publish the ideas openly later on.

But things then moved fast, and within a year the ideas in the paper had become the basis for, and were implemented within, a fully revised ACS-MPM architecture.

Although the original dynamic instruction scheduling ideas were mine alone, the paper was a team effort. As inventor, I was the lead author, and was followed by Brian Randell, Don Rozenberg and Don Senzig. I think Ed Sussenguth and Herb Schorr gave useful feedback too; had the paper gone on to publication they might have been included as co-authors.

The dynamic instruction scheduling paper is labeled "[DRAFT]". I believe that by late February '66, we saw this paper as a work in progress towards formal publication. The ideas were already, in parallel, being evaluated for use in the actual machine. Thus in this draft I think we stepped back from revealing thinking on exactly how the ideas might be applied in the machine, as, for example, by using dual instruction windows.

But by then we also needed a tutorial on the ideas for those outside the architecture group, such as the logic designers, to use as a reference. Thus this "draft" version of 2-23-67 was released within ACS. After that date, no further work was done on the paper. It was completely overtaken by the escalating events surrounding adoption of this scheme for use in the ACS machine. Thus the invention itself then became quite "secret".

Interestingly, the name "dynamic instruction scheduling" never really entered into the team's "lingo". Instead, the relevant structures were usually just called "instruction queues", or "instruction buffers", or "contender stacks" for short, as is seen in all the later documents. It's possible that many ACS vets won't recall the specific title of the paper. Could that perhaps explain why no one from the team has ever come forward and mentioned this work?

On the other hand, it is very likely that copies of this paper surreptitiously passed into circulation outside IBM during the late 60's and early 70's, providing a path for transfer of this knowledge, and its name, into computer architecture circles outside of IBM.

## 2. "ACS Simulation Technique", Mar. 15, 1966: D. Rozenberg, L. Conway, R. Riekert

This paper documents the methods used to build the ACS MPM register-transfer level simulator. This paper may prove valuable by helping later analysts better understand and interpret the source code and the output results of the "MPM Timing Simulator".

The simulator was built in FORTRAN IV. Thus it is relatively easy to "read the code" that defines the workings of each module and functional unit. The simulation methods were also aimed at being fast enough to support long runs involving many, many variations of the machine architectural parameters.

The simulator was initially used to take quick looks at architectural variants, watch code passing through them, and figure out why things got blocked or didn't work as expected. Later it was used to gather data on the performance of many serious MPM variants running lots of real code, and then to "balance and tune" the emerging ACS-1 machine.

Notice the use of a "memory queue" function as the tutorial example in this paper. I believe that by this time in '66, we were already doing basic simulator implementations and evaluations of various "instruction queuing" structures and controls, as part of our explorations of dynamic instruction scheduling methods. I think we may have just simplified and then "reused" some of that code to create the example in this paper.

Don Rozenberg was lead author, I was second and Bob Riekert was third. Bob had done important work on the simulation methods at Yorktown, but didn't go west with ACS.

## 3. "Dual Arithmetic on ACS-1", May 1, 1967: T. C. Chen

This paper is an internal proposal from Tien Chi (T. C.) Chen to Jack Bertram regarding methods for implementing dual floating point arithmetic in ACS-1. It contains interesting references to dual arithmetic on the ILLIAC IV machine.

I include this paper as a good example of an ACS "proposal", though I do not recall right now the details of how this particular one turned out.

Note that the data-path register-transfer-level details of the arithmetic-functional units were an independent architectural dimension of the project that had to meet logic design/machine-cycle constraints on the one hand, and bussing/pipelining/issuance-control/architectural constraints on the other.

Thus only the timings of the ACS-1's arithmetic units, and not those units' internal functional details, were modeled in the timing simulator. (An "unroller" processed assembly code input instructions to produce the input instruction stream to the timing simulator). This was in contrast to the OP fetch, Bussing, OP interlocking and issuance, SKIP, Branch and Exit functioning, etc., which were fully modeled in the timing simulator.