LYNN CONWAY
PROFESSOR OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE
THE UNIVERSITY OF MICHIGAN
COLLEGE OF ENGINEERING

170 ATL BUILDING
ANN ARBOR, MICHIGAN 48109-2110
313 763-5509 FAX 313 763-1260
INTERNET: conway@engin.umich.edu

2 January 1999

Dr. Mark Smotherman
Department of Computer Science
Box 341906
Clemson University
Clemson, SC 29634-1906

Dear Dr. Smotherman:

When I came upon your web site identifying the IBM-ACS machine as "the First Superscalar" computer, many past events came rushing back into my mind. I had been at ACS, first at Yorktown Heights, then in Sunnyvale and then up on Sand Hill Road, during the period when the exciting architectural work was being done there.

There were publications and talks, by Herb Schorr in the early 70's and later by John Cocke and others, that hinted at the scope of the ACS innovations. But these early retrospectives lacked detail about the system's architecture and lacked a context in which to embed the ideas so as to fully convey their significance. Many computer architects sensed that amazing things had happened at ACS, but few could be sure quite what, or why it even mattered.

As modern VLSI superscalars emerged into widespread application, and details of their architectures were described, I became aware that important early ACS innovations had transferred directly into those machines. Even the early ACS name for one of those innovations, dynamic instruction scheduling, is now used by superscalar architects, and is described as such in modern computer architecture textbooks.

More than thirty years after the original work, modern superscalars now at last provide a context for understanding and appreciating the value of the early ACS innovations. For some time now, I've hoped that someone from the ACS team might step forward and point towards the sources of those concepts. However, no one has come forward.

When I read the ACS retrospective on your web site, I began thinking about why such claims haven't been made before. The sudden elimination of the project, followed by exits and transfers of the architecture team members, must have meant that few, if any, original ACS documents were saved by anyone. Thus the machine seemed to have just "vanished", and there was little material evidence on which to base any retrospectives.

It vanished almost everywhere, that is, except in a notebook, documents and computer listings that I compiled and kept stored away all these years.

Hopefully, the materials that I have saved can be used to reconstruct many details of ACS machine architecture, and more fully document the accomplishments of the ACS team. I'm interested in helping with such an effort, and in helping contact other ACS alums who might have original artifacts and personal knowledge of events there.

The years I spent at IBM-ACS were among the most intellectually exciting of my life. It was an incredible opportunity for me to be able to work with John Cocke, Herb Schorr, Fran Allen, Ed Sussenguth, Don Rozenberg and all the others upon just finishing my graduate work at Columbia. Reflections on my experiences at ACS, and the documents relating to my work there, may help you and others reconstruct the overall story.

When I joined ACS, the team was based at IBM Research in Yorktown Heights N.Y., and the effort went by the code name "Project Y". I joined in a support role to build the register-transfer timing simulator for the emerging supercomputer. In that role, I had ongoing access to almost all the team's architectural discussions and debates.

During the early phases of the project, I became fascinated with John Cocke's "open questions" about computer architecture. By an amazing stroke of luck, I hit upon a pretty good general solution to one of those questions, namely the problem of multiple issuance. The team was very democratic and open to suggestions and proposals from any member, at any level. They listened to my ideas, and then acted on them.

We initially called the resulting invention "dynamic instruction scheduling". It went on to play an important role in the overall system architecture of the ACS main processing module (MPM). Fortunately, among my documents are those describing this invention, and showing how it was exploited in the ACS-MPM. These documents are identified in an annotated list attached to this letter.

Included in the attached list are my reference notebook, the source code and a detailed user's manual for the MPM timing simulator. During 1967, the timing simulator became the de facto formal description of much of the machine's architecture. Therefore, these materials can be used to reconstruct many details of ACS machine architecture. It's even conceivable that a running timing simulator could be reconstructed someday, based on these materials.

Given the significance and impact of superscalar computers, I really do feel the need to set the story straight, namely that the ACS machine, a long forgotten "orphan", was never really dead. ACS lives on after all, as the original source of many fundamental innovations that have since passed on into modern machines.

I commend you on your efforts to reconstruct events at ACS and to document details of ACS machine architecture. The independent, detailed context that you have already established, together with my materials, should at least confirm the origins of generalized dynamic instruction scheduling. That invention is one of the coolest ideas I've hit upon. It would mean a very great deal to me for its origins in my ACS work to be acknowledged.

I'm not sure how to best proceed from here, but I do suggest that initially we try to acquire more materials, contact more ACS alums, work on a project timeline, etc., before releasing further preliminary conclusions. Also, by putting more ACS materials on a web site, we could perhaps clarify that a lot of materials do still exist, and thereby interest others in participating in reconstruction efforts.

Many of the events surrounding ACS were shaped by internal IBM politics that I and most of my colleagues were unaware of at the time. The sudden demise of the project completely stunned us. I never understood why the decision had been made that ACS must be 360 compatible. However, it was clear right away that the 360 decision meant that the ACS architectural innovations were going to be shelved.

You can imagine what the project's demise meant to those who had done the creative work there. Sure, John Cocke went on to become famous among the cognoscenti in computing. Indeed, four members of the early ACS architecture team, including John Cocke, Fran Allen, Ed Sussenguth and myself, were later elected to the National Academy of Engineering for a variety of other contributions. But imagine how much it would have meant to John and the rest of us if the ACS designs at least had been saved, and approved for later publication. Instead, almost all that wonderful work was discarded, as if it had never existed.

Since I'm not sure what sensitivities remain regarding theories about the project's cancellation, I'd like to proceed carefully when gathering information on the overall story. It is certainly important to try to contact ACS team members named in the various documents in advance of any public uses of those documents. Efforts should also be made to involve as many ACS alums as possible, so that a wider set of perspectives can be gained and a more thorough history compiled.

I really enjoyed talking with you recently about ACS. I look forward to interacting with you further on this interesting project.

Sincerely,

Lynn Conway

Lynn Conway

Professor of EECS, Emerita
University of Michigan, Ann Arbor, MI

Attachment: Annotated list of reference materials regarding the ACS-1 machine