RETROSPECTIVE:

Decoupled Access/Execute Architectures

James E. Smith

Department of Electrical and Computer Engineering
University of Wisconsin-Madison
jes@ece.wisc.edu

In early 1981, I was still at Control Data in the Twin Cities. The Cyber 180/990 project was close to the prototype stage, so no new performance features could be added. It was apparent there wasn't much left for me to do. Also, there were a number of interesting problems that had come up during my year and a half at CDC, and time hadn't allowed pursuing them as much as I had wanted — returning to the University of Wisconsin would provide the opportunity. I decided to resume my academic career, this time in computer architecture, and in late May I headed back to Madison.

The Cyber 180/990 had issued instructions in order, at most one per cycle. And, within CDC at the time, these were treated as fundamental constraints. I had some rather vague notions of how to overcome these barriers — but left CDC more with goals in mind than any specific solutions.

Because my prior experience had been with numerical problems, back at Wisconsin I hit upon a way of achieving multiple issue and dynamic scheduling with two instruction streams and queues. One instruction stream was for addressing and one for computation. Each stream would issue in order — maintaining simplicity. I remember being pretty excited about the novelty of the concept — but was a little deflated a few weeks later when I read about the CSPI array processors in the Sept. '81 issue of Computer Magazine [1]. These weren't general purpose computers, but used basically the same access/execution decoupling. After the ISCA paper appeared, I also became aware of the SMA work that Andy Pleszkun had done for his Ph.D. with Ed Davidson at Illinois [2]. And, about a year later at a workshop in New Orleans appeared yet another machine with similar concepts [3]. It was a proposed machine called FOM (FORTRAN Oriented Machine) from IBM — a place where superscalar concepts had been kicked around for a long time.

The benchmarking in the paper was pretty miserable by today's standards. I used compilations for the Cray-1 as a guide. The actual simulations were done by hand, and average speedups were calculated as, *ahem*, the arithmetic mean.

The simplicity of in-order instruction issue had been drilled into me at CDC — in retrospect, too much. It probably prevented me from looking at more flexible superscalar machines early on. It is my observation that a common mistake of architects has been (and continues to be) overestimating the complexity of dispatch/issue logic.

I still think the idea of two separate instruction streams connected with branch queues was neat. And having two PCs helped with the precise interrupt problem. But later in a study published at a small conference, Tom Kaminski and I looked at a scheme that combined instruction streams in the binary and had a hardware "splitter" that divided the stream after it was fetched [4]. This was the form that showed up later in the ZS-1 [5] (another, longer story). With this modification, decoupled machines were similar in appearance to the first IBM RS/6000s. A major difference is that the decoupled machines use architectural queues for renaming memory operands. This had the advantage of renaming the values that were most important — load values, and the queue discipline made management of the physical locations very straightforward.

Following this paper, the research got a big boost when Shlomo Weiss came along. Building on tools Nick Pang had developed, Shlomo did substantial performance studies (and he and I realized in the process that harmonic mean speedups should be used). These more detailed results appeared in the IEEE Transactions on Computers a few of years later [6]. Along the way, Honesty Young also added significantly to the Cray-1 simulation tool set which benefited this research. After I left the University in 1983 to work on the ZS-1, research on decoupled architectures at Wisconsin continued with the PIPE project [7], headed by Jim Goodman, Andy Pleszkun, and Randy Katz. The PIPE project produced a number of significant papers on decoupled architectures — including one that appeared in the 12th ISCA [8].

References

[1] E. U. Cohler and J. E. Storer, "Functionally Parallel Architecture for Array Processors," *Computer*, vol. 14, no. 9, pp. 28-36, Sept. 1981.

[2] A. R. Pleszkun, *A Structured Memory Access Architecture*, Computer Systems Group Report CSG-10, Coordinated Science Lab., Univ. of Illinois, Urbana, IL, Oct. 1982.

[3] W. C. Brantley and J. Weiss, "Organization and Architecture Trade-offs in FOM," *IEEE Workshop on Computer Systems Organization*, New Orleans, pp. 139-143, March 1983.

[4] J. E. Smith and T. J. Kaminski, "Varieties of Decoupled Access/Execute Computer Architectures," *20th Allerton Conference on Communication, Control, and Computing*, Monticello, IL, pp. 577-586, Oct. 1982.

[5] J. E. Smith, G. E. Dermer, B. D. Vanderwarn, S. D. Klinger, C. M. Rozewski, D. L. Fowler, K. R. Scidmore, and J. P. Laudon, "The ZS-1 Central Processor," *Second Int. Conf. on Arch. Support for Programming Languages and Operating Systems*, pp. 199-204, Oct. 1987.

[6] J. E. Smith, S. Weiss, N. Pang, "A Simulation Study of Decoupled Architecture Computers," *IEEE Transactions on Computers*, Vol. C-35, pp. 692-702, Aug. 1986.

[7] J. E. Smith, A. R. Pleszkun, R. H. Katz, and J. R. Goodman, "PIPE: A High Performance VLSI Architecture", *IEEE Workshop on Computer Systems Organization*, New Orleans, pp. 131-138, March 1983.

[8] J. R. Goodman, J. T. Hsieh, K. Liou, A. R. Pleszkun, P. B. Schechter, and H. C. Young, "PIPE: A VLSI Decoupled Architecture," *12th Int. Symp. on Computer Architecture*, pp. 20-27, June 1985.