

## Re: Interview with Alan Kay

**Source:** <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2005-02/1461.html>

---

**From:** Dave Roberts ([ldave\\_at\\_droberts.com](mailto:ldave_at_droberts.com))

**Date:** 02/22/05

Date: 22 Feb 2005 13:51:10 -0800

Tim May <[timcmay@removethis.got.net](mailto:timcmay@removethis.got.net)> writes:

> *This is not quite so, from what I saw (knowing some of the*  
> *IA64/VLIW/EPIC designers). Firstly, it was expected that the first*  
> *IA64s would hit the market around 1998 and that transaction processing*  
> *would be a dominant market (and hence x86 binaries less of an issue*  
> *than with desktops, as the main competition was non-x86 machines*  
> *anyway). This was similar to the 432 experience, and Intel and Siemens*  
> *even had a joint venture in the 1980s to develop a highly capable,*  
> *fault-tolerant business machine ("BiiN," which wags dubbed "Billions*  
> *invested in Nothing," though I think this is too harsh, for various*  
> *reasons).*  
>  
> *(A lot of groups were coming out with novel chip and system*  
> *architectures in the 1980s. Essentially they \_all\_ failed or were*  
> *absorbed into other companies and used with fairly conventional chip*  
> *architectures.)*  
>  
> *Secondly, by the time the first IA64s hit the market, Xeon performance*  
> *had increased enormously (recall that this was during that "sprint"*  
> *from 400 MHz machines (PCs, Macs, Suns) to 1 GHz, then 2 GHz, then 3*  
> *GHz and beyond, depending on exact architecture, e.g., pipelines).*  
> *Thirdly, the initial IA64 was disappointing. The second and third*  
> *iterations (McKinley, Madison, etc.) were better, and have scored well*  
> *on a bunch of benchmarks, winning some large installations (such as the*  
> *"Columbia" supercomputer, currently either #2 or #3 in ranking).*

Yes, I used to work in the lab at HP that had a hand in Madison. \*Very\* talented people work there. The product also reflects their architectural philosophy: \*big\*, \*fast\* caches.

> *Fourthly, software just hasn't changed much, at least not for most*  
> *users. Little demand for 64 bits. I suppose this is the same as your*  
> *"legacy x86 code" point.*

Yes, that's exactly it. Anybody who says that Itantic's competition was simply non-x86 architectures misses the point. That's very poor market strategy. In the same way that x86 competed against those other

architectures, it has to compete versus EPIC-based designs, too.

In any case, my comments weren't meant to be anti-Intel or anti-EPIC/Itanic (though I do continue to call it Itanic ;-). Indeed, EPIC is a very interesting architecture and is successful technically on some axes (great floating point). The technical guys who worked on it are to be commended. The problem is the marketing folks to agreed to junking backward compatibility. The fact that it hasn't been a commercial success is, IMO, directly attributable to the fact that it doesn't run x86 code as fast as Xeon and AMD processors. The fact that AMD64, on the other hand, has been a (relative, at this point) commercial success is directly attributable to the fact that it does run standard 32-bit x86 binaries as fast or faster than other 32-bit only x86 processors.

> *In any case, Intel \_was\_ pursuing 64-bit extensions to the x86, as the  
> current product announcements show (these are multi-year efforts, and  
> the work was started a long time before AMD announced parts).*

That's all well and good. I wasn't suggesting that Intel wasn't pursuing such extensions, only that such efforts were not brought to market and EPIC was. That was Andy Glew's reason for leaving Intel and going to AMD, no? (Though the recent news is that he's gone back to Intel, so perhaps they are putting more effort in this direction.)

> *Eventually there \_will\_ be a major shift to a new ISA. Intel has  
> expected this for a long time...and has, in my opinion as a long-time  
> stockholder, generally done the right thing in supporting the x86 while  
> also attempting to introduce other architectures.*

I guess we disagree here. I agree with you that ISAs are not static and will shift over time, but given the dominant code base is x86 for the average IT buyer, any changes will have to evolve from here to there. Note that each market segment may have different preferences. If you're taking the mainframe market forward, you better make sure you take all the legacy IBM code into account. Etc. In that sense, another architecture (something like IBM's Cell, perhaps) might be able to flourish in a particular new niche (like Playstations), but it's unlikely such an architecture can will ever penetrate the segments dominated by x86 without providing a very graceful transition strategy.

>> *In any case, I submit that the market would actually not reject such  
>> things as enhanced Lisp support in a processor, as long as that  
>> processor was capable of running standard 16-bit, 32-bit, and (now)  
>> 64-bit x86 binaries in addition to the extended capabilities.*  
>  
> *A fascinating topic. I tend to disagree, for reasons it would take me  
> too much time and space to get into. Sure, if supporting "Language X"  
> cost nothing more (to add to the chip), it would not "hurt" sales. But  
> would Dell or HP or anyone else bother to build a box that used the X*

- > *features in \_any\_ significant way? I doubt it.*
- >
- > *And, in fact, I can't see many ways that adding Language X features*
- > *could be reasonably done without affecting the die size, the design and*
- > *debugging costs, the yields, etc. Perhaps if we had some specific*
- > *proposals made?*

Of course. I'm assuming that basic features could be added without too much of an end-user cost hit. Indeed, AMD did it with AMD64. Without knowing details, it's hard to understand whether a particular extension would make it or not. Basically, I'm just arguing along Alan Kaye's line: I think such improvements could be introduced into products. The new innovations have to be cost-neutral, however.

- > *(For example, adding tag bits. One of the 960 processors had a tag bit*
- > *(I think just one, but maybe more). As I recall, this was the 960XA.*
- > *Designed to support some capability-based features, along the lines of*
- > *IBM's System/38. Related to some Ada work. Boeing was enthusiastic*
- > *about having the XA, and designed an avionics system for the 7J7 jumbo.*
- > *Which got cancelled. Last I checked, no systems were using the tag*
- > *bit(s).)*

This argument as a couple of problems. First, you're arguing a negative: the failure of one processor that had tag bits doesn't imply anything about tag bits. The processor also had all manner of other differences and its failure doesn't necessarily imply anything about those other innovations. Secondly, you're sort of proving my point: without x86 compatibility and neutral cost structure for those innovations, almost all processors trying to address the x86 market space will sink (and yes, I realize that the 960 wasn't trying to compete with the x86 space).

- > *In fact, going out on a speculative limb here, it could be argued that*
- > *"listening to requests from language fans" was precisely what led to*
- > *the 432 situation and the recent IPA/EPIC/VLIW efforts.*

Hmmm... I'm not sure I see that. Which language fans were arguing for EPIC or VLIW? Some processor architecture people might have been, but I don't see the translation to HLL fans.

- > *Could either OO/tag bits or VLIW features be added at little or no cost*
- > *to the x86 ISA? I doubt it.*

I bet you could. AMD added a whole 64-bit architecture there, with new registers, etc., etc., and still kept the cost structure under control. A few more instructions for handling tagged data types in the ALU would likely not add a any cost whatsoever. Obviously, VLIW would (changing the whole pipeline, etc.). Heck, with the way that x86s are architected these days, you could probably handle some of the tagged data type instructions with microcode such that they were built out of slightly larger instructions. Not as efficient as hard-coded

primitives, but probably better than separate primitive instructions being fed into the fetch unit.

> *It sure looks to me that the architects look at the usual benchmarks, for integer and FP performance, for transactions per second, for C++ and Java code, and then tweak architectures with more cache, more registers, etc. Exotic things to improve "Language X" performance, where Language X has a small market share, don't get done.*

Oh, agreed. I wasn't arguing that Lisp *will* be implemented in such processors. In fact, I said quite the opposite. The fact is, however, that many such other functionalities *will be* (again, crypto, graphics, etc.).

>>> *(Something that may change this is the "slowing down" of Moore's observation about doubling rates. For good physics reasons, clock speeds are not doubling at the rates seen in the past. This may push architectures in different directions.)*

>>

>> *I agree that this should help, but I think the bigger pressure is not so much physics as economics. The fact is, Intel has been trying to figure out how to use all those transistors it has for something that people will actually buy for quite sometime.*

>

> *I think it's much more about physics. By physics I mean heat production (exceeding 100 W on some of the mid-3 GHz processors, requiring very, very large heat sinks and costly cooling systems, as seen even in Apple's dual-2.5 GHz PPC machines, using water-cooling) and leakage/channel length/dopant variation problems.*

While all the problems you cite are very real, I think they are totally orthogonal to this discussion. Again, AMD and even Intel itself have given us existence proofs that such specialized functionality can be integrated into x86 CPUs for minimal added cost and that such functionality will be adopted and used if it provides value for people. I would cite MMX, AMD64, AltiVec in the case of Motorola/Freescale with the PPC architecture, etc.

--

Dave Roberts  
dave at findinglisp dot com