The design of a technology platform for custom VLIW embedded processors

Paolo Faraboschi
Hewlett-Packard Laboratories Cambridge, 1 Main St, 10th floor, Cambridge, MA 02142, USA

1. Introduction

The center of gravity in computing is shifting down once again, largely as a result of improving circuit density, which lowers the cost and size of computers, and enables them to be deployed in entirely new ways. Change on this scale has happened several times before, as we look back on the move from mainframes to minicomputers and then to desktop systems. Every generational change has driven an orders of magnitude increase in the range of applications and number of systems sold. We have already started the next generational move, which puts an entire system on a single chip, together with many other peripheral functions. This will make general-purpose machines even more affordable, but the really exciting thing will be the infusion of computers into a wide range of products that you do not think of as computers.

The activities at HP Labs Cambridge (http://www.hpl.hp.com/cambridge) are centered on technologies at the border of CPU architectures, compilers and tools. We currently have projects on: dynamic optimization in a practical environment, embedded CPU cores customized for their intended use, retargetable ILP compiler backends in a product environment.

This paper explores the design aspects of Lx, a Technology Platform for Custom VLIW Embedded Processors.

1.1. The embedded processors arena

Many competitors recently announced products (or soon-to-be products) in the embedded processor space. Here we list some of the relevant players:
- Texas Instruments C6xxx. Clustered VLIW, general purpose ISA, unprotected pipelines (no precise interrupts), fully predicated architecture, 32 regs, 8-issue.
- Motorola/Lucent StarCore. “Natural” VLIW extension of traditional DSPs, address/data regs, special addressing modes, loop counters, non-standard data widths, 16 regs, 4-issue.
- Philips Trimedia. VLIW core + media processing hardware, optimized for video, FP, rich ISA, 128 reg, 5-issue.
- Analog Devices TigerSharc. “Static Superscalar” (VLIW + scoreboard), DSP features (XY memories, etc.) optimized for telecom applications.
- Equator MAP1000. High-end VLIW + DSP functionality, optimized for video.
- STMicroelectronics ST100. DSP-oriented decoupled architecture (access-execute).

In addition, a few companies are starting to move in the configurable processor space, among which:
- Tensilica Xtensa. Simple single issue RISC soft core. Customizable, but not scalable. Uses and “a-la-carte” approach with some support for custom instructions. Users can choose from functional units, memory, and peripherals or user-defined instructions with an RTL-like language.
1.2. Why custom-fit VLIW embedded processors?

We believe that custom-fit VLIW processors represent a technology that strikes the right balance between flexibility and speed. The following figure highlights the differences between four competing technologies in the embedded world, and it shows how uniquely positioned is the Custom-VLIW processor technology (see Fig. 1).

Many approaches advocate custom-fit processors as a “push-button” solution to generate a processor from an application. We believe the hardest task is building the infrastructure that makes customization practical. In particular, managing the complex database of machine-specific data is an effort that is often largely underestimated. We think we have done good progress, but still far from “push-button” automation.

2. Lx architecture overview

Lx is a technology platform for customizable System-on-Chip (SoC) cores jointly designed by Hewlett-Packard and STMicroelectronics. First parts are expected for 2nd half 2000 with a base architecture and standard peripherals. In parallel we are investigating for product-oriented specializations that will follow soon after.

For Lx, we have built an “architecture framework” that covers the shared properties of the target domains. This includes toolchain and hardware that were designed within some rigid constraints so that it would be easy to specialize it for a specific application domain. These constraints include a common basic ISA, a common run-time architecture, and a common decoding scheme and pipeline structure.

Both hardware and toolchain for Lx specializations are generated from tables. For the hardware, this step still involves significant manual work, but still far less than a custom approach. For example, we chose to implement Lx as a “hard core” because of cost and performance reasons. A “soft core” Lx would be easier to automate, although it would probably leave too much speed and area on the table.

The target domains for Lx can be defined as integer computation-intensive media-processing applications. These can be characterized as “DSP-style” computational kernels with a significant “control” component. They are usually programmed in a high-level language (C, C++) and involve large code bases (∼100K–1M lines of code).

Examples of domains that share these common properties include digital still-imaging (printers, scanners, digital cameras, compression), video processing (video encoders, decoders, set-top boxes), networking,
cryptography (network processor, routers, switches, firewalls), audio processing (compression, encoding, manipulation), and so on.

Lx is a Clustered VLIW Architecture. Clusters are composed of a mix of Register Banks, Constant Generators and Functional Units. Different clusters may have different unit/register mix, but they all run in lockstep off a Single PC, a unified I-cache and the same pipeline structure. Inter-cluster Communication happens through explicit register-to-register move operations (send/receive), is compiler-controlled and invisible to the programmer. Fig. 2 shows the logical organization of Lx clusters.

A single Lx cluster is a 4-Issue VLIW engine, composed of 4 ALUs, 2 16×32 MULs, 1 Load/Store, and 1 Branch Unit. The ISA is a simple integer RISC instruction set, and minimal predication is supported with select instructions. The Memory Unit allows 1 D-cache access (2cycles latency, 1 cycle throughput), through Base+Offset addressing. Additional features include software prefetch, speculative execution of load operations. Registers are organized in two banks: 64×32b (General Purpose) and 8×1b (branch, predicates, carry). The I-cache is compressed on fetch (horizontal no-ops are removed in the decoding phase), and can be optionally compressed on refill. Fig. 3 shows the single Lx cluster.

2.1. Lx performance analysis

The following figure shows Lx performance relative to a baseline machine. Our baseline numbers (speedup = 1) are for a Pentium-II at 333 MHz, and we also show performance of a typical high-end 32-bit embedded processor, the StrongArm SA-110 at 275 MHz. The numbers for Lx are for single-cluster implementations at 200, 300 and 400 MHz.

From an analysis of the graph we can see that specializing a technology platform pays off: we get 4x–8x performance gains vs. a general-purpose architecture for the target benchmarks (starting from C-level code with similar cost and technology).

In addition, scaling speed vs. power pays off fairly uniformly and gains are almost linear in the considered frequency range.

Outside the application domain we still get good performance, which is very important to de-risk the new technology. Not surprisingly though, performance for general-purpose applications scales poorly.

3. Customization risks and rewards

Given the advantages of domain-specific customization, one could be tempted to push the customization barrier and design CPU processors (or cores) that are
customized for a given application. Although sometimes the rewards can be big, it is important to also point out the risks:

- **Excessive customization impacts Time-To-Market.**

  Often the development flow of embedded products has concurrent Hardware/Software engineering as an integral part of the production process. Sometimes, the software application itself may not even exist in time for hardware customization. Asking to freeze the application too early kills the num-
ber one advantage of using a processor: the fact that the software application can change till the last minute. Obvious mitigating factors are diminishing hardware NRE costs, deriving from improved CAD tools, synthesis, and so on. Much work however remains to be done.

- **Over-Specialization.** First of all, a too narrow specialization may be harmful to other parts of the program. More important, when customizing for an application, there is a heavy risk to specialize for a specific algorithmic implementation more than the application itself. In real life, it can be hard to distinguish between an algorithms and specific implementations (or poor implementations). Sometimes a small implementation change can completely shift the computational balance in an algorithm, thus exposing the risks of over-specialization.

4. Conclusions

Embedded markets are important and are becoming the “new center of gravity” of computation.

In this paper we have shown that embedded designs require a new perspective on core architecture design, where cost becomes a dominant factor. We have also pointed out how VLIW is becoming the predominant embedded/DSP technology and Custom-VLIW strikes the right balance of performance and flexibility.

In addition, we presented the HP/ST Lx custom-VLIW “technology platform” and described how it can be effectively customized to a domain.

Finally, we warned that heavy customization could be impractical and/or too risky and what are the major pitfalls to avoid. More information on this subject can be found in Refs. [1–17].

References


