Advanced Issues in Computer Architecture

Monday April 26, 10-12 in Elogen.

- Current Trends in Microprocessor Architectures. General Purpose and Embedded Microprocessors. (Petru Eles)

Wednesday April 28, 13-15 in Eliten.

- Current trends in Microprocessor Architectures. Multimedia Extensions and Superscalar Features. (Petru Eles)

Friday May 7, 10-12 in Elogen.

- DSP Applications and Architectures. (Dake Liu, IFM)

Wednesday May 12, 13-15 in Elogen.

- Application Specific Instruction Set Processor Architectures. (Zebo Peng)

Friday May 21, 13-15 in Elogen.

- System-on-Chip. (Petru Eles)

Wednesday June 2, 13-15 in Elogen.

- Distributed Architectures for Automotive Applications. (Paul Pop)

Friday June 11, 10-12 in Elogen.

- Modern Microprocessor Testing and Debugging Facilities. (Zebo Peng)

Friday June 18, 10-12 in Elogen.

- Presentation of Term Papers

Microprocessor Architectures: Past, Present, and Future Trends

1. The Five Generations

2. Some Possible New Approaches

3. The Market

4. PC vs. Workstation

5. Embedded vs. desktop

Microprocessors: Where are we? How it will go on?

☞ Where we are?
Maybe we know (more or less)

☞ How it will go on?
Predictions! Who believes in them?

About Predictions Concerning Computers!

☞ In the 40's/50's *serious people* predicted that a few hundred computers will be the maximum ever needed!

☞ Now we have Billions.

⇩

So, maybe let's go home and wait!

Microprocessor Generations

☞ First microprocessors were explicitly designed for embedded control (1971 Intel 4004, had 2300 transistors).

☞ Later came the general purpose computing area.
- Hobby computers: 1974
- IBM PC: 1981

☞ Today the two markets co-exist.
- The embedded market is much larger (two times, in '97); it goes well beyond the control area and includes multimedia, communication, etc.

Microprocessor Generations (cont'd)

1st generation:
- 1971: Intel 4004 (2300 transistors)
- 1974: Intel 8080 (4500 transistors)
- Instruction processing was strictly sequential. Instructions were fetched, decoded and executed strictly one at a time.

2nd generation:
- 1979: Motorola MC68000 (68000 transistors)
- Primitive pipelining with three stages: fetch, decode, execute; only one instruction is in execution at a certain time.

Microprocessor Generations (cont'd)

3d generation:

- 1984: Motorola MC68020 (240000 transistors)
- Five stage pipeline; increased parallelism

4th generation:

- 1990: MotorolaMC88110, Intel 80960 - these are RISCs (over 1 Million transistors)
- PowerPC604, Pentium
- Superscalar architectures; parallel execution based on multiple pipelines and functional units.

Microprocessor Generations (cont'd)

5th generation:

- 1996: P6, PowerPC 620 (over 5 Million transistors)
- MIPS R10000, AMD K5/K6, UltraSparc (not exactly out of order)
- Superscalars with out of order execution and sophisticated scheduling and renaming strategies.

Some Possible New Approaches (medium term)

- Multiprocessors - *System on Chip*
- VLIW (very large instruction word) - a compiler schedules the instructions statically (instead of dynamic scheduling as with superscalars).

  ⇩

  Compiler is very much architecture dependent ⇒ poor portability.
- Move some system services into hardware
  - Graphic accelerators, . . .
  - MMX like extensions in instruction set
- Programmable logic

The Market

☞ The microprocessor market has two main segments:
1. Desktop microprocessors (CPUs for computers)
2. Embedded systems

☞ The embedded market is much larger!

And then came the RISC!

☞ First PCs were based on microprocessors which were used for embedded applications, or were direct successors of those (Intel, Motorola).

☞ University research tried to develop architectures explicitly for the CPU market ⇒ RISC.
Projects: RISC at Berkeley (1980), MIPS at Stanford (1981).

☞ Several companies started to design/produce RISC-like architectures: Sun (SPARC), MIPS, Motorola (MotorolaMC88110 and family), Intel (i860 family), AMD, etc.

PC vs. Workstation (cont'd)

The desktop market split:

1. PCs low performance, low prise (based on x86 and family).

2. Workstations - high performance, high quality software (based on the “new wave”: RISC architectures).

☞ Size of the markets:
- PC: ~ 20 Million
- Workstations: ~ few hundred thousands

PC vs. Workstation (cont'd)

Why did the PC market not take over the new RISC-based architectures?

- Backward compatibility of software

☞ PCs take over from the Workstation market:

- Prises are lower (large market)
- High-end PCs have performance comparable with low-end workstations.

⇩

The market for (RISC-based) workstations will probably stay in the niche area of high-performance (UNIX-based) machines.

PC vs. Workstation (cont'd)

If we look at microprocessor architectures the cut is by far not as sharp.

☞ Microprocessor architectures have been strongly influenced by RISC concepts (e.g. limited number of instructions and addressing modes, high degree of pipelining)

☞ RISC architectures are also not pure RISCs any more (larger instructions set, smaller number of registers, instruction formats are not fixed)

Microprocessors for Embedded Systems

☞ Embedded applications:
- control systems: they use what usually is called *embedded microprocessors*, or *microcontrollers*.
- DSP applications: they use *DSP processors*.

What makes these processors specific?

Are they so different from desktop processors (general purpose processors - GPPs)?

Microprocessors for Embedded Control

- Traditionally we had a relatively clear cut between *embedded* and *desktop* microprocessors.

  Embedded microprocessors (= microcontrollers): 8 or 16 bit, RAM and ROM instead of caches, a lot of peripherals (ports).
- Mainly as 32 bit embedded processors emerge (handheld, mobile, multimedia applications), the difference is not so sharp any more.

- The difference *is not* any more: register organization, (basic) instruction set, pipelining;
- The difference *is*: power consumption, cost, integrated peripherals, interrupt response time, on-chip RAM and ROM, parallel ports.

Microprocessors for Embedded Control (cont'd)

- Embedded processors have begun to incorporate capabilities traditionally associated with conventional CPUs.

- Main constraints typical to the embedded business:
  - cost
  - power consumption

    three modes: (1) fully operational, (2) standby, (3) clock off;
  - application-imposed (e.g. time, multimedia acceleration)

    functions that are typical for DSP processors are now executed by the embedded microprocessor itself;
  - peripheral integration

    1. basic core + ASIC (system on a chip)
    2. standard microprocessor + companion chip

Microprocessors for Embedded Control (cont'd)

- In the embedded market custom specific software platforms are still current practice.

- There are steps towards standardisation:
  - Windows Compact Edition (CE)
  - Java

  Problem: hard real-time requirements and reliability!

- Will there emerge a “standard”, dominant microprocessor and OS for the embedded market?
  - OSEK in automotive electronics?

DSP Processors

- Some specifics of DSP processors:
  - *multiply-accumulate operations* (GPPs are not even efficient with simple multiplications)
  - *memory architecture*: there are two separate memory spaces, program memory and data memory (Harvard architecture)
  - *explicit programmer controlled on-chip (cache) memory*
  - *zero overhead looping*
  - *fixed-point arithmetic* (rather than floating-point)

DSP Processors (cont'd)

- Problems with execution time predictability
  - dynamic features in the architecture create problems: pipelining (branch prediction), cache memory, out of order execution;

- *With DSP processors execution time prediction is easier*:
  - caches are programmer controlled
  - no branch prediction or other dynamic features;

DSP Processors (cont'd)

- DSPs have very particular asymmetric architectures

⇓

optimization is very important in order to get performance;
often no compilers are available => optimization by hand!

DSP Processors (cont'd)

- GPPs are incorporating more and more DSP features!
- Vendors offer versions of there processors which include DSP hardware (hybrid GP/DSP processors): Intel MMX, AMD K6.

⇓

The same story as with Embedded processors: *the distinction between GPPs and DSP processors is becoming less clear!*

- Some high-end GPPs outperform DSP processors!

DSP Processors (cont'd)

- Why should we use DSP processors?
  - execution time predictability
  - cost
  - power consumption

- If the system already contains a high-end GPP, you don't need to add a DSP processor;
  If you build the system from scratch, it's a good idea to use a DSP processor or a GP/DSP one.

Summary up to this Point

- Typical features of last generation microprocessors:
  - sophisticated superscalar architecture with out of order execution;
  - special features for multimedia.
  - Sophisticated multilevel caches
- One possible evolution: system on chip
- Market forces push the dominance of x86 like architectures and PC systems.
- Overlap between high-end PCs and low-end workstations
- The largest part of the microprocessor market is dominated by embedded applications.
- General purpose processors get more and more features which make them suitable for embedded applications

Multimedia Extensions to General Purpose Processors

1. Applications and Approaches
2. Multimedia Processors
3. Characteristics of Multimedia Applications
4. Instruction Set Extensions
5. Open Questions

Multimedia Applications

- Video playback
- CD-ROM
- Audio
- Image processing
- WWW related applications
- 3D graphics and animation
- Videoconferencing

The Main Approaches

1. Multimedia processors

2. ISA extensions to general purpose processors (GPPs)

Multimedia Processors

☞ Microprocessors targeted towards multimedia applications
- TriMedia (TM-1), from Philips
- Multi-Media Signal Processor (MSP), from Samsung
- Mpact, from Chromatic

☞ Some features:
- Features typical to DSP architectures (see paragraph earlier).
- Hardwired video and audio ports and bitstream codecs.
- Instructions typical to multimedia extensions of GPPs.
- Hardwired motion estimation, ...
- special architectures:
    - very long instruction word (TM-1, Mpact)
    - vector instructions (MSP and Mpact).

Multimedia Extensions to GPPs

☞ Several vendors have extended the instruction set of their GPPs in order to improve performance with multimedia applications:

- MMX for Intel x86 family
- VIS for UltraSparc
- MDMX for MIPS
- MAX-2 for Hewlett-Packard PA-RISC

Basic Assumptions

☞ Characteristics of multimedia applications:

- Small data types (8 or 16 bits).
- Large data set sizes.
- Large amounts of inherent data parallelism.
- Computationally intensive features with highly predictable branches.
- Real-time processing requirements
- Multiple concurrent streams (e.g. video and audio)
- Large I/O bandwidth requirements

Instruction Set Extensions for Multimedia

☞ The basic idea: subword execution

- Use the entire width of a processor data path (32 or 64 bits), even when processing the small native data types used in signal processing (8, 12, or 16 bits).

With word size 64 bits, the adders will be used to implement eight 8 bit additions in parallel

☞ This is practically a kind of SIMD parallelism, at a reduced scale.

Instruction Set Extensions for Multimedia (cont'd)

Packed data types

Packed byte

q7	q6	q5	q4	q3	q2	q1	q0

Packed half word

q3	q2	q1	q0

Packed word

q1	q0

Long word

q0

Instruction Set Extensions for Multimedia (cont'd)

Subword arithmetics

ADD R3 ← R1,R2

a7	a6	a5	a4	a3	a2	a1	a0
+	+	+	+	+	+	+	+
b7	b6	b5	b4	b3	b2	b1	b0
=	=	=	=	=	=	=	=
a7+b7	a6+b6	a5+b5	a4+b4	a3+b3	a2+b2	a1+b1	a0+b0

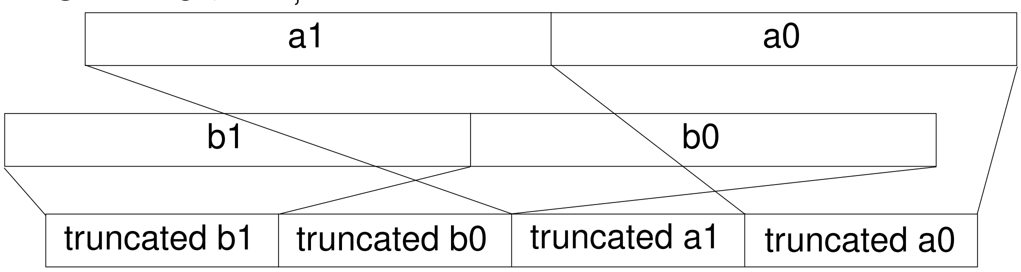
MPYADD R3 ← R1,R2

a7	a6	a5	a4	a3	a2	a1	a0
x-+		x-+		x-+		x-+	
b7	b6	b5	b4	b3	b2	b1	b0
=		=		=		=	
(a6xb6)+(a7xb7)		(a4xb4)+(a5xb5)		(a2xb2)+(a3xb3)		(a0xb0)+(a1xb1)	

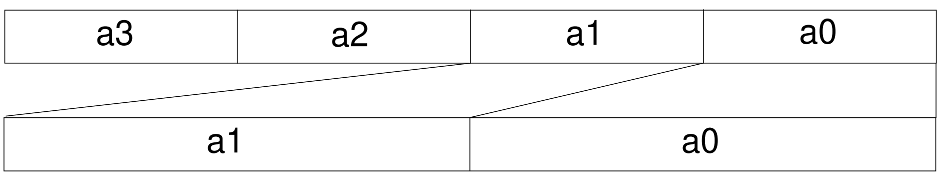
Instruction Set Extensions for Multimedia (cont'd)

Packing and unpacking

PACK.W R3 ← R1,R2



UNPACK R3 ← R1



Instruction Set Extensions for Multimedia (cont'd)

Other extensions (VIS for UltraSparc)

- *Pixel distance instruction*: used predominantly for motion estimation in MPEG video compression and video conferencing protocols.
- *Block load and store*: transfer blocks of data between memory and registers without cache allocation (moves large amounts of data used only once, without displacing useful data from the cache).
- *Partial store instructions*: masked store to memory with byte granularity.

Problems

☞ Compilers

- Like for DSP processors, there are no compilers which can make use in an efficient way of the multimedia extensions. *Really good code has to be handwritten*.
- Adequate languages could facilitate the automatic parallelisation (vectorisation) by compilers.

☞ How do this ad-hoc extensions facilitate the overall performance and cost. Rethinking of the microarchitecture?

☞ Will the ASIC for multimedia processing disappear? Power consumption will influence this aspect?

Summary

- Multimedia applications are becoming an essential part of computer workload.
- Two main approaches for implementation: specific multimedia processors or multimedia extensions to the instruction set of GPPs.
- Instruction set extensions are basically targeted towards subword execution. This introduces a SIMD-like parallelism with operations on short vectors.
- Other instructions are also implemented, which are typical for video applications: pixel distance computation, block load and transfer, etc.
- Some of the main problems are related to availability of software tools, mainly compilers.
- What about the future of ASICS in multimedia?