Massively parallel, but massive challenges

09/02/2010 Email to a friend

Massively parallel computing systems are coming, but what issues need to be solved?

Embedded designers will only be able to exploit the potential of multicore processor architectures if the applications software takes advantage of parallelism. This was the starting point for the Massively Parallel Computing seminar held during IP/ESC09 in December 2009.

Massively parallel computing architectures and parallel programming are not new, even in the embedded world – remember the Transputer? So why are they an issue now?

Session chair Huy Nam Nguyen commented: "The performance potential of single chip multicore devices, combined with their low power and small size, has created an expectation. In the embedded world, until now, engineers could cope with developing limited multiprocessor systems using manual design and implementation techniques." But now, Nguyen inferred, architectural and methodology changes are needed.

Dr Frédéric Pétrot said: "We expect to see 128 processor cores being implemented by 2020. There will be an evolution in architecture, with a shift to homogenous designs based on a single software stack and built in redundancy will become increasingly necessary. Tomorrow's SoCs will be composed of multiple, possibly highly parallel, processors (MPSoCs)."

Addressing the homogenous vs heterogeneous debate, Joseph van Vlijmen brought a commercially oriented voice to the debate. "It's well known that STMicroelectronics' mantra on this subject is 'largely homogenous, slightly heterogeneous MPSoCs'."

Nguyen agreed, but expanded the case for a continuing demand for heterogeneous architecture, with different types of cores and clusters of cores, depending on function. "Amdahl's Law still holds in some parts of some applications that are inherently sequential. Handling those functions using smaller cores may not be efficient, so there will always be a demand for high performance processors."

Dr Pétrot believes yields of heterogeneous multicore designs will be too low. "However, heterogeneous MPSoC architectures can provide higher performance and flexibility with less power consumption and lower cost than homogeneous devices," he admitted. "But, as processor instruction sets for general heterogeneous MPSoCs are not identical, task migration between two heterogeneous processors is not generally possible." His plan is to build a MPSoC platform in which all heterogeneous processors are based on the same core instruction set, which can then be better exploited by the operating system.

Confirming that Dr Pétrot is on the right track, van Vlijmen added: "The use of a common core instruction set, together with specialised extra cores, seems to be the preferred hardware oriented approach."
And the next challenge? van Vlijmen believes one of the most pressing challenges is how to handle tasks that can be dealt with in parallel, across multiple processor cores. "Where is the data?" he asked. "Transforming programs from single threaded to multithreaded requires additional functionality."

Dr Pétrot retorted: "Mastering huge parallelism at the task level is the next SoC challenge. Design methodologies must change their focus to the selection, specialisation and usage of processors, either programmable or dedicated, as basic components, rather than logic modules."

Prof Alain Greiner is more concerned with the challenges of massively parallel computing architectures targeting teraflops performance: systems with up to 4096 processor cores, each typically a 32bit risc core such as the Sparc V8 or MIPS32. His work, part of the Catrene project TSAR, is focused on the implementation of a coherent, scalable shared memory.
When a program is split across processors, one challenge is to ensure code is executed in the correct order. Similarly, if two or more threads access shared data, there is the potential of sequencing errors, race conditions and deadlocks. Beyond a couple of threads, it becomes increasingly difficult for the programmer to visualise the system and the ensuing sequencing complexities.

What are the possible solutions? "Our critical achievement so far is the development of a

Related News

ARM links with G&D to secure mobile phone payments
Cadence releases latest version of EDI system
Intel raises bar for mission critical computing
Semiconductor industry set to boom
Analogue signal processor cuts power consumption in heart monitors

Related Technology

Differentiate or die
Powering performance
Pushing packet performance
Chips with everything – reloaded
Showcasing talent - Hard times for start ups, but investors are still interested in opportunities

Related Products

Module uses Freescale i.MX51 with 800MHz ARM Cortex-A8
18 pin microcontrollers feature enhanced midrange core
Low power motherboard
MSP430 MCUs with embedded full speed usb
Compatible devices deliver scalable performance and features

Related Events

Embedded World Exhibition and Conference 2010

Downloads

P20-21.pdf

distributed hybrid cache coherence protocol," Prof Greiner stated. Essentially, it is a multicast policy to a certain threshold, then a broadcast policy. The project architecture consists of clusters of cores, supporting a non uniform memory access (NUMA) shared address space – shared logically, but distributed physically – and a two tier (within and between clusters) interconnect.

"The challenge has been to ensure scalability; that the cache coherence protocol supports up to 4096 processors, implementing different policies for data and instruction caches. Plus, it must support legacy code," Prof Greiner summarised.

"Interconnect is also key and, for this, we implemented a network on chip (NoC) architecture. With NoC, snooping is not possible, so a directory based approach was essential." Although snooping protocols can be faster, they are not scalable. "The NoC technology was critical as it allowed us to adopt a write-back policy to ensure memory coherence with the scalable bandwidth we needed," Prof Greiner explained.

According to Prof Greiner, the NoC and distributed memory architecture avoids several potential bottlenecks in multicore system design. "Although access to external memory can still cause a problem, this could be solved by using 3d chip stacking, which could cope with some of the bandwidth limitation issues."

Nguyen sees 3d 'through silicon via' technology as the solution to efficient interconnects for network processors and network memory. "We will still need arbitration schemes, but the delays are less. However, scheduling becomes more critical and more complicated," he warned.

What about efficiency of access? Dr Pétrot concurred that logically shared, physically distributed memory meets the need for efficient data access: low latency, high bandwidth, low energy. "The issue here is placement. Preplacement of shared data is no good anymore: too far, and the access cost in terms of time and power increases; too close, and there are potential hotspot, contention or congestion problems."

Prof Greiner took up the OS issue. "The big issues are task scheduling and memory management. These are not normally interactive functions, but will need to be for NUMA applications. In fact, we will need a coplacement function of task transaction and memory access and this has to be an OS function."

Van Vlijmen: "Parallelism is here. The silicon is available, but the right tools are not." ACE is concerned that its customers lack a coherent set of tools, such as profilers, visualisers and, particularly, schedulers, to help the move towards the successful exploitation of parallelism in multicore technology.

ACE is also a partner in TSAR and Van Vlijmen has been monitoring progress in the cache coherence and task migration programmes closely. "These two different aspects are fermenting nicely," he said. "They are prevalidating a number of practical approaches, some of which will be instantly usable. They are definitely on the right track, especially in answering the questions about the scalability of cache coherence memory architectures. These projects form a very important part of our embedded system ecosystem."

Nguyen summarised "Cache coherence is the solution for a large class of problems. In the near future, cache coherent architectures using shared memory – which could be physically distributed – will become dominant. They will make life much easier for general software applications development and programming tasks."

The experts:

Prof Alain Greiner
Head of the SoC department at Université Pierre & Marie Curie's LIP6 Laboratory.

Dr Frédéric Pétrot
Assistant professor in computer science at Université Pierre et Marie Curie.

Huy Nam Nguyen
Head of R&D modelling and hardware verification at Bull.

Joseph van Vlijmen
Director, ACE Compilers.

Author
Louise Joselyn

Download Articles

P20-21.pdf

Supporting Information
http://www.ace.nl/compiler/
http://www.bull.com/index.php
http://www.lip6.fr/index.php?LANG=en
http://www.upmc.fr/