A new Processor for Power Efficient Computing

José Augusto Fernandes da Silva

Departamento de Informática, Universidade do Minho
4710 Braga, Portugal
jafs@aeiou.pt

Abstract. Mobile computing became a common gadget in spite of its limitation with power, weight and performance. Industries attempted to overcome these limitations and their research lead to power efficient computing. The new processor from Transmeta, Efficeon, works with lower clock frequencies and voltage, offers better performance per clock cycle than previous Crusoe processor, and is highly competitive with the Intel Centrino processor.

1 Introduction

Today, the productivity has become directly linked to its ability to produce power efficient computing technology. The mobile computing devices are an example where the waste of energy must be controlled without losing performance and productivity. These devices, powered solely by batteries, continue to proliferate throughout modern society. Now, people need mobility everywhere. This mobility and the need to avoid energy wastage, lead energy efficiency to the forefront of modern processor design.

The adaptive processing - required for example to improve microprocessor power efficiency - dynamically tunes major processor resources, by implementing part of ISA in software. Dynamic translation is one way to implement this: codes sequences are translated as needed at run time, caching results and optimising code sequences based on dynamic information, and providing the size reduction and power needs of the processor. A processor like this, use only the amount of hardware needed to execute an application and nothing more.

In October 2003, Transmeta Corporation launched the Efficeon, an x86 compatible processor that use this dynamic translation approach. This processor continued a new family of solutions started with another Transmeta processor, the Crusoe. With this solution, companies have now an alternative processor to use in their new mobile products.

Next section we will detail modern processor technologies – such as the ones behind x86, superscalar and VLIW - and the Efficeon processor architecture as a VLIW implementation with his *longrun2* technology and code morphing software. Finally we will make a brief benchmarking analysis between the Efficeon and the Intel Centrino processor, comparing their clock frequency for the same power consumption.

2 The Modern Processor Technologies: x86, Superscalar and VLIW

Following, we will see some modern and important processors that contributed for the evolution of computer architecture.

2.1 The x86 Processor

Starting with the 8088, which powered the first IBM PCs, the x86 architecture has

developed with the popularity of PCs and it is a significant CISC architecture.

Here we consider the basic structure of a simple processor. We see the flow of data through such a simple processor.

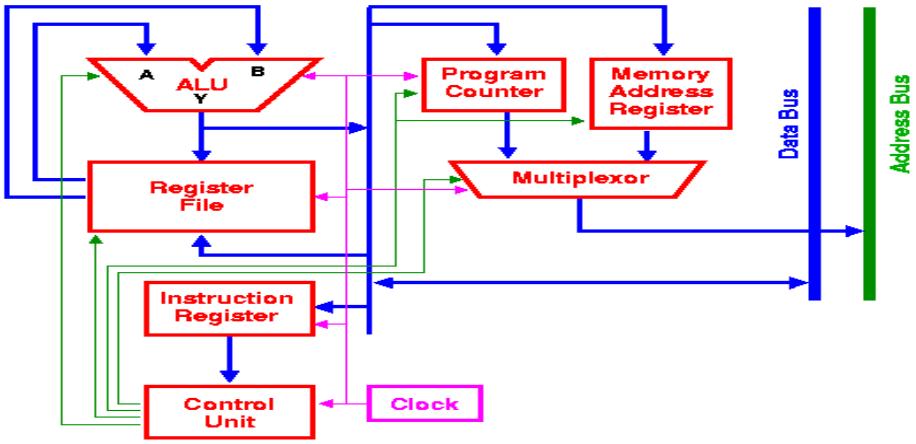


Fig. 1 Simple Processor Structure

2.2 Superscalar Processors

Superscalar processors - the MIPS R10000 for example - have multiple functional units. This enables their instruction issue units to emit multiple instructions into the execution section of the pipeline at the same time.

The instruction issue unit will usually fetch a whole line (4-8 instructions) from the cache at a time and determine whether they can be issued to the various functional units. To do this, it checks whether an instruction's operands are available yet. If the register which contains one of the operands is being written by some instruction that has been despatched into the pipeline, but which has not completed (retired or "graduated") yet, then all instructions reading that register must be stalled (wait in the instruction issue unit) until the writing instruction has completed.

2.3 VLIW Processors

Very Long Instruction Word (VLIW) processors have instruction words with fixed "slots" for instructions that map to the functional units available. This makes the instruction issue unit much simpler, but places a great burden on the compiler to allocate useful work to every slot of every instruction.

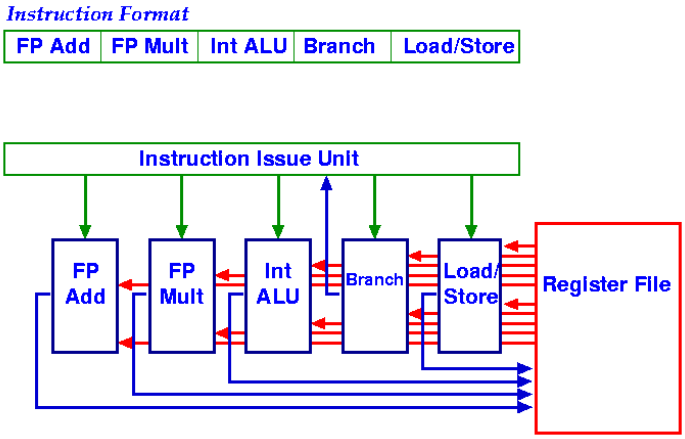


Fig.2 A Possible VLIW processor

This is the architecture used by the new Transmeta processor. It takes a similar structure and evaluates it. The Transmeta approach makes a translation from x86 code to RISC ops and then to VLIW code. The VLIW code results from RISC ops optimization .

For example, from x86 code

```
addl  %eax, (%esp)    # load data from stack, add to eax
addl  %ebx, (%esp)    # load data from stack, add to ebx
movl %esi,  (%ebp)    # load from memory
subl  %ecx,  5        # sub 5 from ecx
```

and after optimizing RISC ops, we have the VLIW code

```
ld    %r30, [%esp];    sub.c %ecx, %ecx, 5
ld    %esi, [%ebp];    add  %eax, %eax, %r30
add   %ebx, %ebx, %r30
```

3 Efficeon Processor Architecture

The Efficeon processor solutions consist of a hardware core engine surrounded by a power management unit (*LongRun2*) and a software layer.

Fig. 3 Efficeon Architecture

LongRun is a proprietary power management technology that enables processors to dynamically change voltage and frequency to provide the performance needed by the application at any moment. Transmeta introduced LongRun power management in the Crusoe processor, and the new Efficeon processor now also incorporates Enhanced LongRun technology. This technology allows the Efficeon™ processor to continuously adjust its operating frequency and voltage to exactly match the requirements of the application workload, which can dramatically extend battery life.

The software layer, called Code Morphing software, dynamically optimize and translate x86 instructions into VLIW instructions that the VLIW engine can process. With the introduction of the code morphing software, the processor engineers can reduce the size and the consumption of this processor. They reduce the numbers of logic transistors and additional hardware needed for the execution and timing of instructions, common in a traditional processor. With fewer transistors we have less heat dissipation and a cooler-running processor.

A long instruction word in an Efficeon processor, a molecule, can be 256 bits long and contain up to eight RISC-like instructions, the atoms. All atoms within a molecule are executed in parallel, and the molecule format directly determines how atoms get routed to

functional units; this greatly simplifies the decode and dispatch hardware. Figure 4 shows a sample 256-bit molecule and the straightforward mapping from atom slots to functional units. Molecules are executed in order, so there is no complex out-of-order hardware. To keep the processor running at full speed, molecules are packed as fully as possible with atoms.

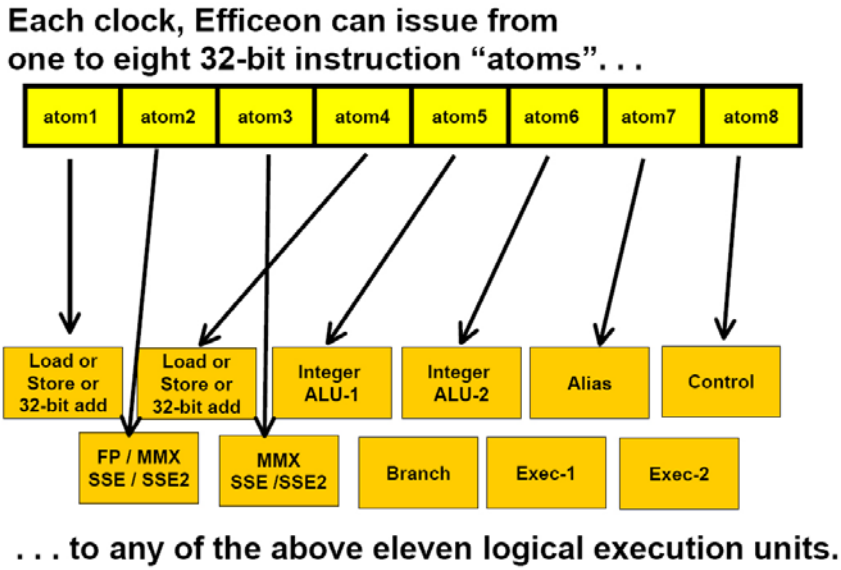


Fig 4. A molecule can contain up eight atoms.

The Efficeon processor is currently produced using advanced 0.13-micron semiconductor technology from TSMC.

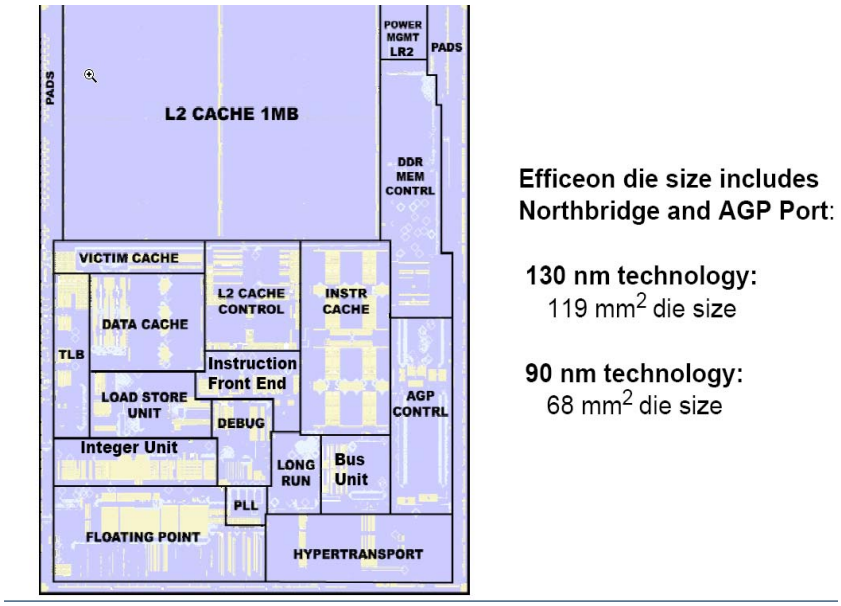
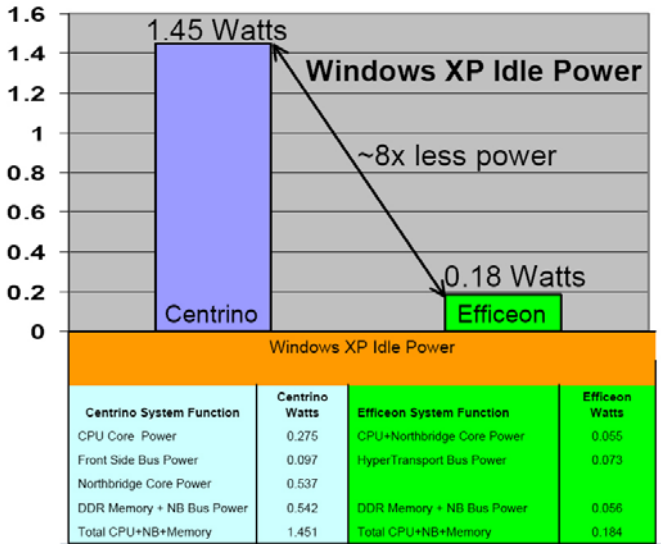


Fig 5. Efficeon Processor Layout

4 Efficeon vs. Centrino

When Efficeon was launched, we already had another efficient mobile processor: Intel Centrino. Although the few information and benchmark, we try to give an idea of their power consumptions.

For many factors, TDP (maximum Thermal Design Power) often limits the maximum MHz. For fanless notebook systems the upper limit tends to be 7 Watts. If you know that different processors achieve different MHz for the same TDP limit, to compare performance, you must compare at the maximum MHz for a given TDP limit. With 7 Watt TDP, Intel Centrino achieves 900MHz and Transmeta Efficeon 1100 MH (next figure).



Centrino System Function	Centrino Watts	Efficeon System Function	Efficeon Watts
CPU Core Power	0.275	CPU+Northbridge Core Power	0.055
Front Side Bus Power	0.097	HyperTransport Bus Power	0.073
Northbridge Core Power	0.537		
DDR Memory + NB Bus Power	0.542	DDR Memory + NB Bus Power	0.056
Total CPU+NB+Memory	1.451	Total CPU+NB+Memory	0.184

Fig 6. Efficeon and Centrino power consumption

5 Conclusions

The Efficeon processor offers a good balance between performance and power. It provides better performance per clock on typical and multimedia application. With the new code morphing software, this solution provides high performance with less power consumption. It is ideal to several mobile devices like tablet PCs and ultra notebooks. It was designed to run with the full suite of x86 compatible operating systems, including Microsoft Windows 9x, Windows Me, Windows NT, Windows 2000, Windows XP and Windows XP Tablet PC Edition. The Efficeon TM8600 processor, for example was also designed to work optimally with Linux operating systems. So, I think this processor or the future family of this processor has a chance to be competitive with Intel or AMD mobile processors.

This processor is already an example of adaptive processing implementation. Here, we have a dynamic power reduction and a better control of leakage power with the LongRun technology. In future, I think these techniques will be improved and become usual. Maybe we will have, not an adaptive processor but an entire adaptive system working with us and doing several tasks on an efficient way. We must think about seriously, since the earth resources are limited and the energy wastage must be controlled.

References

[1] Perry, M., Ditzel, D.: Introduction to the Transmeta Efficeon TM8000 Microprocessor family. http://www.transmeta.com/pdfs/brochures/efficeon_tm8600_processor.pdf (2003)

[2] Miranda,V. :"Crusoe: an Approach for the new Era Computing", ICAA 02 DI-UM http://gec.di.uminho.pt/Discip/MInf/ac0102/ICCA02.htm (2002)

[3] Hennessy, J. ,Patterson, D.,: Computer Architecture, A quantitative approach 3rd Ed. (2003)

[4] Asanavic, K,: Virtual Machines and Dynamic translation, Implementing ISAs in Software http://www.csg.lcs.mit.edu/6.823 (2002)

[5] Morris, J.: Computer Architecture , The Anatomy of Modern Processors http://ciips.ee.uwa.edu.au/~morris/ca406 (1998)

[6] Albonesi, D., Balasubramonian, R, Dropsho,S., Dwarkasdas, S., Friedman,E., Huang,M., Kursun,V., Magklis, G., Semeraro, G., Scott,M., Bose,P., Cook,P., Buyuktosunoglu,A., Shuster, S.: Dynamically tuning processor resources with adaptive processing. IEEE Computer, December 2003