Extensions to Instruction Sets

Ricardo Manuel Meira Ferrão Luis

Departamento de Informática, Universidade do Minho
4710 - 057 Braga, Portugal
rml@net.sapo.pt

Abstract. This communication analyses extensions to instruction sets in general purpose processors, together with their evolution and significant innovation. It begins with an overview of specific multimedia and digital signal processing requirements and then it focuses on feature detection to properly identify the adequate extension. A brief performance evaluation on two competitive processors closes the communication.

1 Introduction

The purpose of this communication is to analyse extensions to instruction sets, describe the specific multimedia and digital signal processing (DSP) requirements of a processor, detail how to detect these extensions and verify if the processor supports them. To achieve this functionality an IA32 architecture example code will be demonstrated.

To study the instruction sets supported by each processor, an analysis is performed to compare and distinguish each technology in terms of new extensions to instruction sets and architecture evolution, mapping a table that lists the instruction sets supported by each processor.

To clearly identify these differences between architectures, tests are performed with an application example, determining comparable results of two competitive processors.

In this communication, Section 2 gives an overview of multimedia and digital signal processing requirements, Section 3 explains the feature detection of these instructions, Section 4 details some of the extensions supported by IA32 based processors, Section 5 tests and describes the performance of each technology and analyses the results. Section 6 concludes this communication.

2 Multimedia and DSP Requirements

Computer industries needed a way to improve their multimedia capabilities because of the demand for 3D graphics, video and other multimedia applications such as games. Tto achieve these goals instead of the usual loop of similar assembly instructions they could use vector computation, also known as single instruction multiple data (SIMD) for the same set of instructions. For example, a register can hold a vector with numeric values and perform an instruction that adds these values instead of adding registers with values in a loop, having fewer instructions to execute and becoming faster in terms of performance.

Almost simultaneously, computer manufacturers attached new extensions in the instruction set for specific processing operations like a multiply accumulator that is normally used in digital signal computation. With these extensions to the instruction set architecture, computer industries can now outrun lower-end DSP processors [2][3].

Concluding, this technology adds new assembly instructions and data types to the existing architecture to use the data parallelism that is often available in DSP and multimedia applications. Among this, many current general purpose processors are using extensions to the instruction set architecture to enhance the performance of digital signal processing and multimedia applications [4].

3 Feature Detection

In an IA32 Architecture, the CPUID instruction gives programmers the ability to determine the presence of multimedia technology on a processor. This instruction was introduced by Intel in Pentium processor. Since then it became the official method of identification for Pentium class and newer chips. It is also supported by newer 486-class chips made by Intel, AMD, UMC and Cyrix, and newer NexGen CPUs.

In the CPUID instruction an argument is passed in EAX. The argument selects information that should be returned by CPUID. AMD introduced extended CPUID instruction. Extended CPUID is called when EAX content has bit 31 set to 1. The Extended CPUID is supported by various CPU models such as AMD K6 and K6-2, Cyrix III, Transmeta Crusoe and Intel Pentium 4.

Software applications must first test to see if the CPUID instruction is supported.

The presence of the CPUID instruction is indicated by the ID bit 21 in the EFLAGS register. If this bit is writable, the CPUID instruction is supported. The following code sample shows how to test for the presence of the CPUID instruction [6][7].

```
pushfd                  ; save EFLAGS
pop eax                 ; store EFLAGS in EAX
mov ebx, eax            ; save in EBX for later testing
xor eax, 00200000h      ; toggle bit 21
push eax                ; put to stack
popfd                   ; save changed EAX to EFLAGS
pushfd                  ; push EFLAGS to TOS
pop eax                 ; store EFLAGS in EAX
cmp eax, ebx            ; see if bit 21 has changed
jz NO_CPUID             ; if no change, no CPUID
```

Once the software has identified the processors support for CPUID, it must test for extended functions by executing extended function 8000_0000h (EAX=8000_0000h). The EAX register returns the largest extended function input value defined for the CPUID instruction on the processor. If the value is greater than 8000_0000h, extended functions are supported.

The following code sample shows how to test for the presence of extended function 8000_0001h.

```
mov eax, 80000000h      ; query for extended functions
CPUID                   ; get extended function limit
cmp eax, 80000000h      ; is 8000_0001h supported?
jbe NO_EXTENDEDMSR      ; if not, multimedia technology not supported
```

Overview

The extensions to instruction sets are supported by every x86 processor introduced in the market after the Intel Pentium MMX, so it should be fairly safe to assume that the processor that your code is running on has extensions to instruction sets. But checking for these instructions is really simple and avoids embarrassing crashes due to unsupported codes.

Regarding each technology in an IA32 architecture, the ID bit 21 is always used to determine the presence of the CPUID instruction on a processor and after verifying the existence of extensions support, each technology has an extended function limit. Therefore we can test that limit to identify which extensions are supported.

4 Extensions in IA32-based Processors

This section details some of the extensions supported by IA32 based processors.

MMX technology *(Intel Architecture)*

In January 1997, the Pentium MMX Technology was developed. Pentium MMX looks like a classical Pentium but with turbo injection. It has more cache (32 KB, instead of 16 KB) and has new (integer) MMX instructions. These instructions were made to improve the performance and execution of integer operations. In spite of its low performance, it was quite commented in its time.

Multimedia Extension (MMX) is merely a marking name of one of the extensions to instruction sets used in the numerous computer market, the same occurs with 3DNow! technology from AMD.

Both normal Pentium and Pentium MMX are fifth generation processors.

Extended MMX technology *(Intel Architecture)*

The Extended MMX appeared in order to upgrade and extend new instructions on the MMX instruction set. This technology was implemented on Pentium III processors and AMD Athlon and Duron.

SSE technology *(Intel Architecture)*

Pentium III has its principal evolution on the use of new instructions SSE, also known as MMX2, which improved the processor’s performance in games and other applications.

SSE has 52 floating point instructions SIMD, 19 MMX instructions, a total of 71 instructions.

Pentium Pro, Pentium II, Pentium III and Celeron are considered sixth generation processors.

SSE2 technology *(Intel Architecture)*

SSE2 technology contains 144 new instructions when compared with SSE. It is used in the Pentium 4 processor and its newest innovation is the use of 128 bit registers, making it possible to manipulate more data of smaller dimension in each time (for example 16 data of 8 bit each time) and the hyper-pipelined technology that doubles the usual pipeline capacity.

This processor is still a 32 bit processor and still uses the same basic architecture of the 80386 (x86 instructions or IA-32). The name 32 bit processor or 64 bit processor normally refers to the type of software these processors can execute (for example Windows).

3DNow! technology *(AMD Architecture)*

The AMD K6-2 processor was a K6 processor with new 3DNow! instructions, which were the first set of instructions made to improve the performance and execution of 3D graphics. The 3DNow! instructions had such success that Intel resolved to include a similar set of instructions (SSE) in Pentium III processor.

It has 21 floating point instructions SIMD, 19 new MMX instructions and 5 DSP instructions (for soft modems, soft ADSL, MP3, Dolby Digital and Surround sound processing), a total of 45 instructions.

The AMD Athlon processor adds 24 new instructions to the existing 3DNow! and MMX instruction sets. The new enhanced 3DNow! technology implemented in the AMD Athlon processor adds streaming and DSP technologies, which allow faster, more accurate speech recognition, DVD-quality audio and video, and streaming audio and video for a rich Internet experience.

Other processors

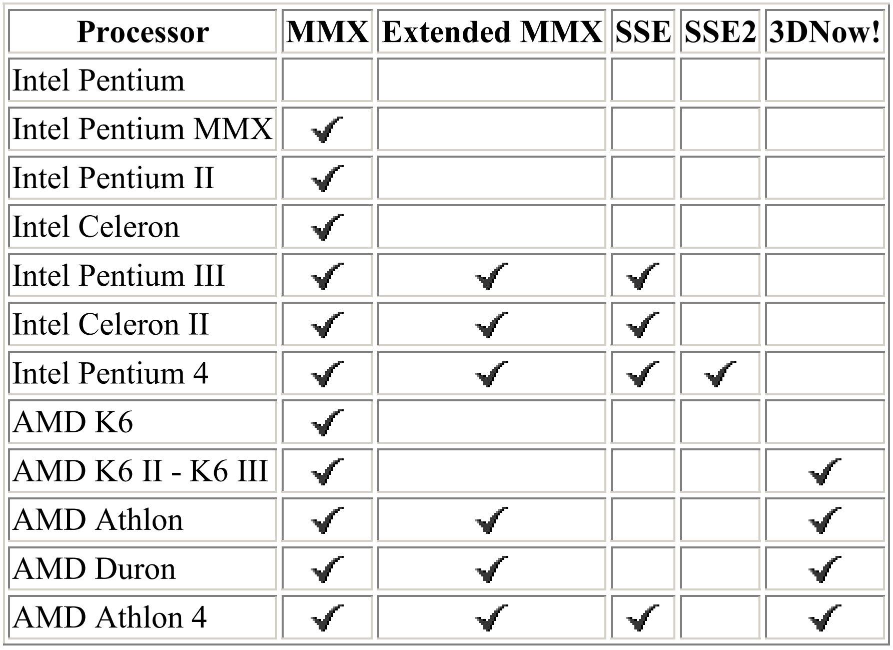
Cyrix processors never had the same impact, success and popularity as the AMD and Intel processors. Recently VIA announced the launch of its first processor named Cyrix III. Cyrix III includes MMX and 3DNow! technology.

The lack of information noticed on the manufacturer's side regarding other processors (like Transmeta) did not allow a comparison between the ones presented.

Extensions to Instruction Set / CPU map

The Extensions to Instruction Sets supported by each processor are given in Table 1.

Table 1. Extensions to Instruction Set / CPU map



Processor	MMX	Extended MMX	SSE	SSE2	3DNow!
Intel Pentium					
Intel Pentium MMX	✓				
Intel Pentium II	✓				
Intel Celeron	✓				
Intel Pentium III	✓	✓	✓		
Intel Celeron II	✓	✓	✓		
Intel Pentium 4	✓	✓	✓	✓	
AMD K6	✓				
AMD K6 II - K6 III	✓				✓
AMD Athlon	✓	✓			✓
AMD Duron	✓	✓			✓
AMD Athlon 4	✓	✓	✓		✓

Since the appearance of the MMX instruction set, all processors were designed to support this technology, but as processors innovate and their developers design new extended instructions, each company goes its own way but always with an eye on its competitors innovations [1].

5 Performance Evaluation

In this chapter, functionality and number of instructions supported by each instruction set are analysed and compared. A detailed analysis will be made on an IA32 instruction set architecture [8].

5.1. Performance

- The original MMX has a total of 57 instructions, all designed to perform integer operations. This technology can manipulate up to 64 bits of data while older processors only manipulated 8 or 16 bits of data. In spite of all the hush, this technology only brought a 10% improvement in multimedia performance.

- 3DNow! has a total of 45 instructions, 21 floating point SIMD instructions, 19 new MMX instructions and 5 new DSP instructions. It also has a CISC/RISC decoder.

- SSE was the technology Intel developed to make face to AMD 3DNow!. SSE has a total of 71 instructions, 52 floating point SIMD instructions, 19 MMX instructions. Comparative to 3DNow! the floating point functionalities are comparable, both technologies support 4 operations per clock. But 3DNow! is simpler to operate. SSE has more instructions because the Intel architecture needs double control of functionalities and 2 floating point instructions per extension: one for SIMD and another for scalable operations.

- SSE2 has a total of 144 instructions. These new instructions do not bring anything new in terms of multimedia performance. The big jump attends to the other Pentium 4 processor innovations such as hyper-pipelined and 128 bit registers.

5.2. Examples and Performance Results

For a better understanding of multimedia technology, it is wise to know how the graphic pipeline works. It consists in 4 parts (Fig. 1 shows the functionality of this technology) [6].

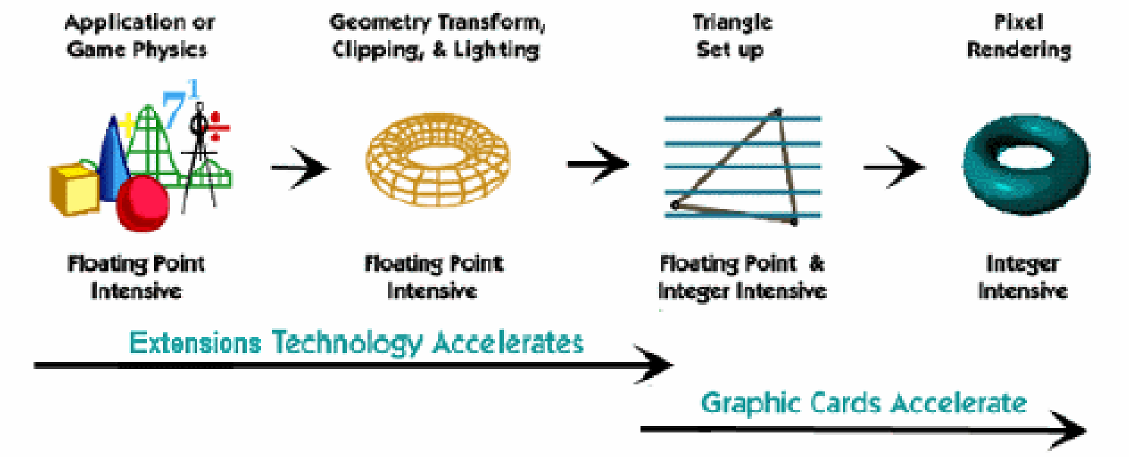

Application or Game Physics
Geometry Transform, Clipping, & Lighting
Triangle Set up
Pixel Rendering
Floating Point Intensive
Floating Point Intensive
Floating Point & Integer Intensive
Integer Intensive
Extensions Technology Accelerates
Graphic Cards Accelerate

Fig. 1. Graphic Pipeline Functionality

Physic – The CPU effectuates floating point calculations to create simulations of the real world and objects in it.

Geometry – Then the CPU transforms math representations of the object in three-dimensional representations, using Floating point 3D geometry.

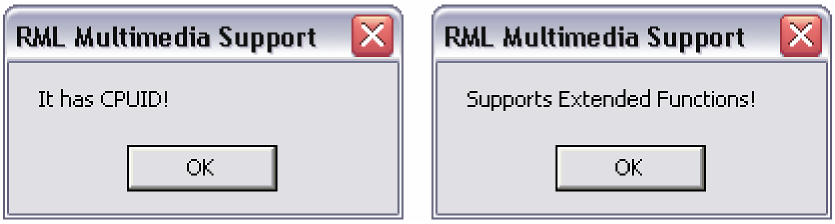
Setup – The CPU starts the process on a 3D vision and the graphic accelerator finishes it.

Rendering – Finally, the graphic accelerator joins almost real textures generated by computer, using calculations per-pixel of colour, shadow and position.

The following example was performed with two computers, an Intel Celeron 800 MHz and an Intel Pentium 4 1.5 GHz. Both have the same graphic card (ATI Radeon), the same monitor and 256 MB RAM.

The Celeron has MMX support. The Pentium 4 has SSE2 support and also recognizes MMX, Extended MMX and SSE instructions.

The feature detection was performed on both computers with the same following results (Fig. 2 shows the detection and results).

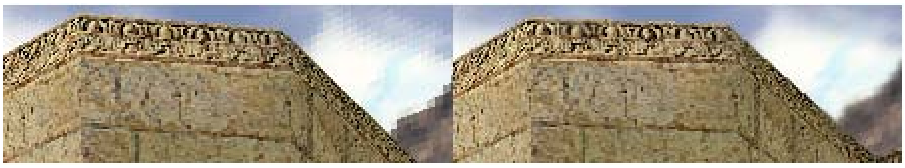


It has CPUID!

Supports Extended Functions!

Fig. 2. Results of the feature detection performed on both computers

This next test is performed on the known game Counter-Strike.



Intel Celeron 800 Intel Pentium 4 1.5

Fig. 3. Graphics generated on both computers

It is obvious that the image (Fig. 3) on the left (Celeron - MMX) has less quality, the image (Fig. 2) on the right (Pentium 4 – SSE2) has better rendering:

- Smoother fluidity and geometries;
- More detailed texture;
- Less pixilated.

This analysis concludes that some extensions are better than other ones and that it can be visible to the human eye and can also be assumed by rendering and motion performance. The latency, number of clock cycles that are required to complete the execution of all of the μops that form an instruction is also an important issue to be considered. In sum, you can have a high-quality graphic card but without the proper instructions to process, in this case the floating point computation instructions, the image generated by your computer can be of low quality.

6 Summary

In conclusion, Multimedia and DSP requirements will keep growing due to the digital communication generation and computer manufactures will certainly keep developing new techniques and computer technologies to satisfy the requirements of market applications. Surely new extensions will be created to assure and optimize these improvements and will be added to the existing instruction sets.

Registers, pipeline and vector computation will certainly keep growing and leave behind the existing 128 bit registers.

Graphic cards and other equipment, as shown in section 5, will keep depending on the execution of the floating point calculations performed by processors to create simulations of the real world and objects in it.

The battle between Intel and AMD in the IA32 architecture will surely keep going on, feeding this novel, without knowing who will succeed or if there will be a third party to run the show. In the mean time Intel seems to lead this market with AMD near behind. By now processors with SSE2 technology or processors with SSE and 3D-Now! technology will keep ahead on performance.

On the other hand, parallel processing may become a big step in the evolution of computers, but this technology will probably take a long time to grow and achieve good results.

References

[1] Vikram Uday Godbole, Performance Characterization of Intel's Internet, Streaming SIMD Extensions, Technical Report, http://www.ece.utexas.edu/projects/ece/lca/sponsors/dell_project1.html, May 2000.

[2] Byeong Kil Lee and Lizy Kurian John, Implications of Programmable General Purpose Processors for Compression/Encryption Applications, http://www.ece.utexas.edu/projects/ece/lca/ps/asap02.pdf, 2002.

[3] Jesus Corbal, Mateo Valero, Exploiting a New Level of DLP in Multimedia Applications, http://palms.ee.princeton.edu/fiskiran/repository/ MICRO/corbal99exploiting.pdf.

[4] Ruby B. Lee, A. Murat Fiskiran and Abdulla Bubshait, Multimedia Instructions in IA-64, http://www.ee.princeton.edu/~rblee/me/bio_ruby2001_2pages.pdf, 2001.

[5] Alex Peleg and Uri Weiser, MMX Technology Extension to the Intel Architecture, IEEE Micro, August 1996.

[6] AMD, 3DNow! Technology Manual, http://www.amd.com/K6/k6docs/pdf/21928.pdf, 2000.

[7] AMD, Extensions to the 3D-Now and MMX Instruction Sets Manual, http://www.amd.com/K6/k6docs/pdf/22466.pdf, 2000.

[8] Ravi Bhargava, Lizy K. John, Bryan L. Evans and Ramesh Radhakrishnan, Evaluating MMX Technology using DSP and Multimedia Applications, 1998.