The Intel MMX technology

M. Sonza Reorda

Politecnico di Torino
Dipartimento di Automatica e Informatica

Introduction

The Pentium MMX was introduced in 1997 and could run at 166, 200, and 233 MHz.

The new Pentium (also named P55C) has several major differences w.r.t. the previous version:

- **MMX technology**
- **L1 caches of 16 Kb each (4-way instead of 2-way)**
- **Additional pipeline stage.**

When the same clock rate is used, the Pentium MMX is about 10-15% faster than the previous Pentium.

SIMD paradigm

The MMX technology derives from the SIMD computation paradigm.

In the SIMD (Single Instruction Multiple Data) paradigm, each instruction executed by the processor operates on several data in parallel, executing the same operation on all of them.

This paradigm can be very effective when certain applications are considered (e.g., DSP and multimedia).

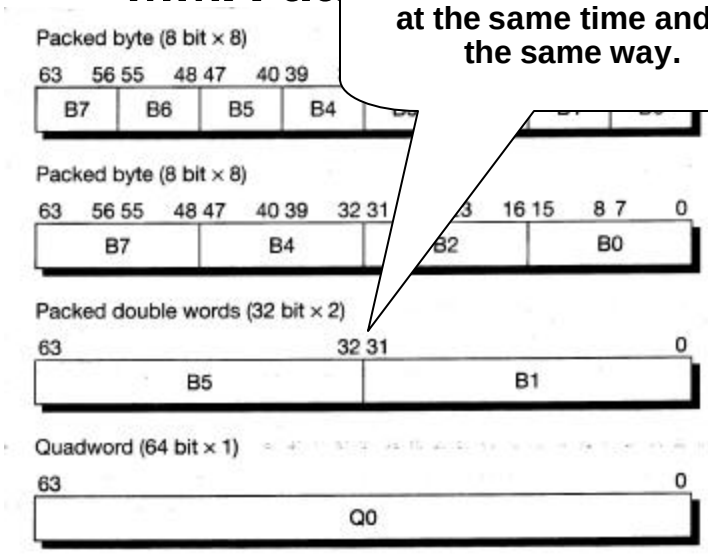
Several SIMD architectures have been devised and sold starting from the '60s (e.g., Connection Machine by Thinking Machine Co.).

MMX technology

It is based on

- **MMX data types, that can be stored in the MMX registers**
- **MMX instructions.**

MMX da



Example

Let consider an application which performs color manipulation on images.

If the color of each pixel is stored in a byte, and the application has to operate on all the image pixels in the same way, we can pack 8 pixels in one MMX register and resort to MMX instructions; each of them will operate on 8 pixels at once.

The theoretical speed-up that can be obtained in this way is 8.

MMX registers

The 8 64-bit MMX registers are physically overlapped with the 32 80-bit FP registers.

MMX instructions work on these registers, whose FP content must previously be saved if relevant.

MMX instructions

There are 57 MMX instructions divided in 3 groups:

- **Those used to control the MMX status**
- **Those for type conversion**
- **The processing instructions.**

Comments

MMX technology requires new programs to be written. These programs can provide a speed-up of up to 5 times with respect to corresponding "traditional" programs when special applications are considered.

The SW solution they implement is still clearly slower than the HW solution implemented by HW accelerators.