Computer Organization And Assembly Report

The Overview of CPU

B94705049　資管三　沈奕超

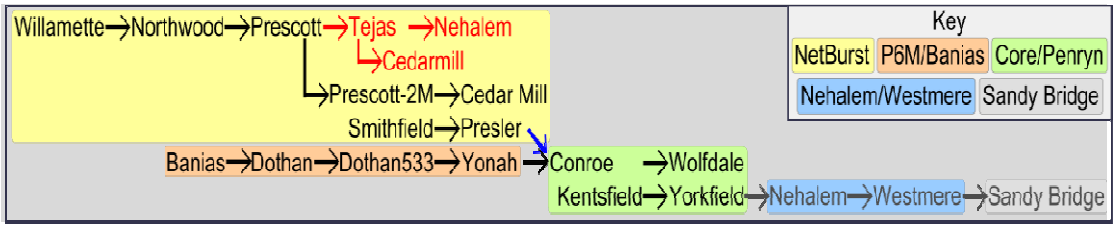
2008/1/22

<引言>:

用電腦用了這麼多年，原本都是使用別人寫好的程式，從開始學習著寫程式之後，對整個電腦系統有著新一層的認識。但是接觸了組合語言之後才開始慢慢讓我對更底層系統有所了解。在現今的電腦架構中，我對 CPU 整體的發展和兩家大廠商的競爭最感到興趣。因此我打算對 **AMD** 和 **INTEL** 在 CPU 近期的發展上先稍微做些重點式的介紹，並且加上兩家廠商針對多媒體的加強各自有哪些嘗試。

<近期發展>:

Intel CPU 架構的發展：



目前及未來的 CPU 列表(focus 在桌上型電腦)

核心架構	處理器名稱	製程	核心數量	FBS	L2 快取
Allendale	Core 2 Duo	65nm	2	1066MT/s	2 Mb
Conroe	Core 2 Duo	65nm	2	1333MT/s	4 Mb
Wolfdale	Core 2 Duo	45nm	2	1333MT/s	6 Mb
Kentsfield	Core 2 Quad	65nm	4	1066MT/s	8 Mb
Yorkfield XE	Core 2 Extreme	45nm	4	1333MT/s	12 Mb

在 CPU 的發展上，不管是 Intel 或者是 AMD 常常都是會提早規劃出許多核心架構，可能都會是未來兩三年才會推出的。但是提早發布規畫也是方便周邊相關的開發廠商可以提早研發對應的東西。而這個所謂的 Microarchitecture 到底到表著甚麼意思呢?

在 Computer Engineering 中，Microarchitecture 代表著電子線路、CPU、還有整個系統中數位訊號傳遞的整體描述以及規劃。Microarchitecture 和 instruction set architecture (ISA)共同組成了整個 Computer architecture。

Microarchitecture 的各個面向：

- chip area/cost
- power consumption
- logic complexity
- ease of connectivity
- manufacturability
- ease of debugging

- testability

大概介紹完了 Microarchitecture 的概念後，讓我們來看看實際上 Intel 和 AMD 所發表的一些 Microarchitecture。

主要的 Intel x86 microarchitectures 發展如下所述：

架構	簡述
NetBurst	**NetBurst**:Microarchitecture used in the Intel Pentium 4, Pentium D, and some Xeon microprocessors.
Pentium M	Pentium M :Updated version of the Pentium III's P6 microarchitecture designed from the ground up for mobile computing
Core/Penryn	**Core**: New microarchitecture used in the Intel Core 2 and Xeon microprocessor families built on a 65nm process **Penryn:** 45nm shrink of Core microarchitecture with new features, such as larger cache, faster FSB and clock speeds and SSE4.1 instructions
Nehalem/Westmere	**Nehalem:** Upcoming microarchitecture expected to be released in 2008, built on a 45nm process. It will be an enhanced version of the Core microarchitecture. **Westmere:** 32nm shrink of the Nehalem microarchitecture with several new features (expected around 2009).
Sandy Bridge	**Sandy Bridge**: Future Intel microarchitecture (expected around 2010) based on 32nm process. Formerly called **Gesher** but renamed in 2007

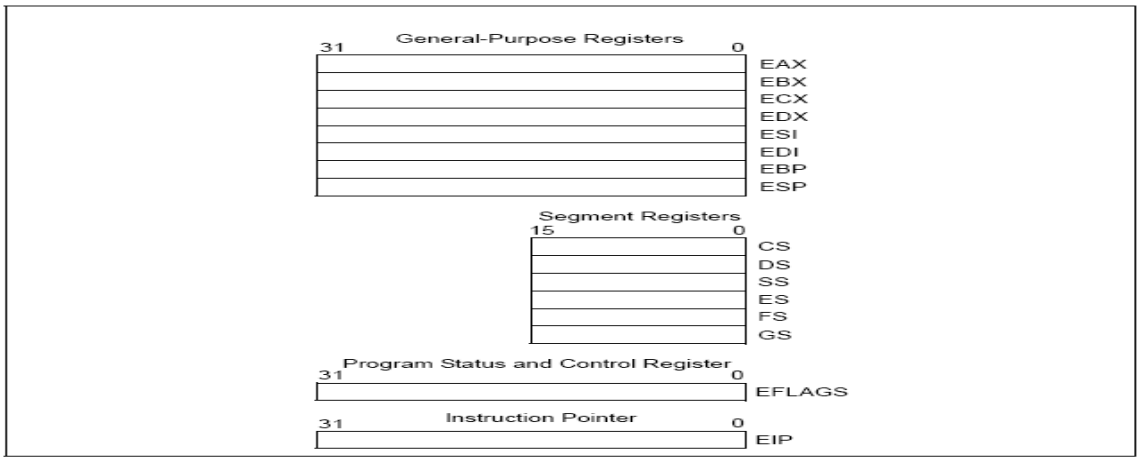
主要的 AMDx86 microarchitectures 發展如下所述：

架構	簡述
K8	**K8:** Also called Hammer, The K8s central processor unit is based on the K7 but was extended to 64-bits, L2 cache sizes up to 1 MB (1128 KB total cache), and SSE2. Later K8 added SSE3. The K8 was the first mainstream Windows compatible 64-bit microprocessor and was released April 22, 2003.
Dual-core K8	**Dual-core K8**:The dual core Athlon 64 X2 was sometimes incorrectly referred to as the K9 by the press after the cancellation of the original K9
K9	**K9**: Canceled microarchitecture rumored to have been an

	8-issue
AMD Family 10h	**AMD Family 10h** :Also incorrectly called K8L in the press, or Barcelona (after the first processor using this microarchitecture) the AMD Family 10h microarchitecture is a new architecture with native quad core, shared level 3 cache, 128-bit floating point units, AMD-V™ Nested Paging virtualization, and HyperTransport 3.0.
Griffin	**Griffin**: Griffin is designed solely for mobile platforms and will be the successor to Turion 64. Griffin will be released with the Puma platform in 2008. Griffin is based on 65nm K8 revision G with power optimization technologies specific to the demands of the mobile market. In face the Griffin power optimizations are beyond those in the AMD Family 10h microarchitecture.
Bulldozer	**Bulldozer**: A microprocessor core after the K10 microarchitecture per AMD M-SPACE modular design mothodology for future microprocessors. Bulldozer will be designed for processors in the 10 W to 100 W category, implementing SSE5 and can be combined with GPU cores (Fusion).

<軟體架構>:

前面對 CPU 的硬體發展架構做過稍微的簡述之後，讓我們來對跟 CPU 有關的軟體方面做一些了解。主要是要探討一些基本的 x86 組合語言的架構以及 Intel 和 AMD 對多媒體 Programming 方面所做的努力。



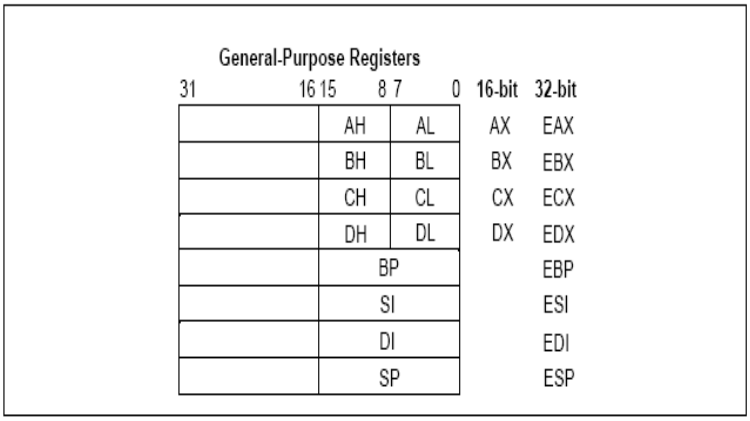
1. **General Purpose Register**

Register Type	Without REX	With REX
Byte Register	**AL, BL, CL, DL, AH, BH,**	**AL, BL, CL, DL, DIL,**

	CH,DH	**SIL, BPL, SPL,R8L - R15L**
Word Register	**AX, BX, CX, DX, DI, SI, BP,SP**	**AX, BX, CX, DX, DI, SI, BP, SP,R8W - R15W**
Doubleword Register	**EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP**	**EAX, EBX, ECX, EDX, EDI, ESI,EBP, ESP, R8D - R15D**
Quadword	**N.A.**	**RAX, RBX, RCX, RDX, RDI, RSI,RBP, RSP, R8 - R15**

這些 Register 有些都有特殊的用途：

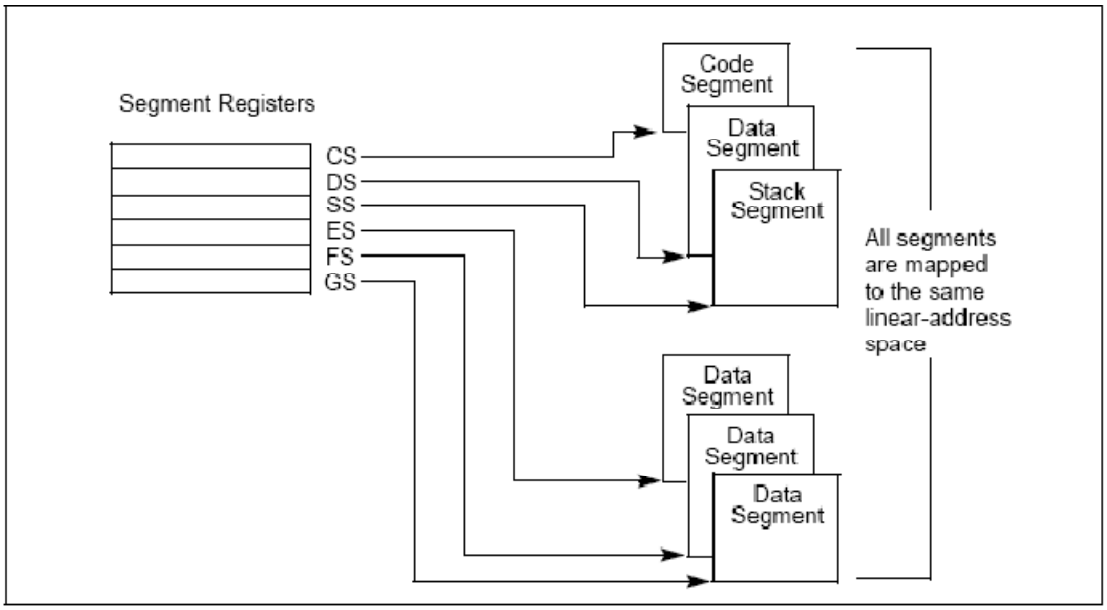
- EAX: Accumulator (automatically used by division and multiplication)
- EBX: Pointer to the data in DS segment
- ECX: Counter for string and loop operation.
- EDX: I/O pointer
- ESI: Pointer to data in the segment pointed to by the DS register; source pointer for string operation.



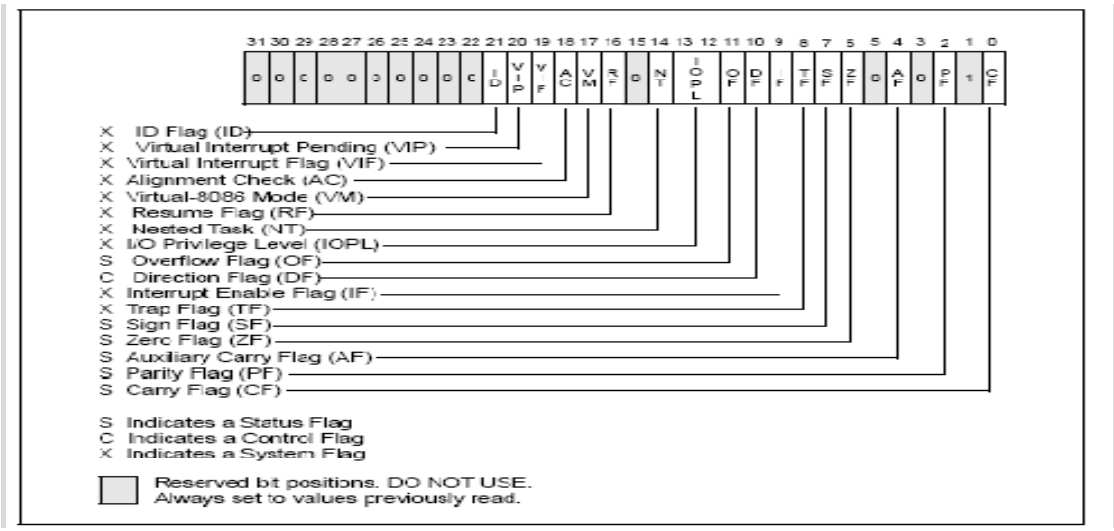
- EDI: Pointer to data in the segment pointed to by the ES register; destination pointer for string operation.
- ESP: Stack pointer (in the SS segment)
- EBP: Pointer to data on the stack(in the SS segment)

2. **Segment Register**

CS(Code Segment),DS,ES,FS,GS(data segment),SS(Stack segment)內含 16-bit 的 segment selector.下圖就是應用了 segment register 的 Segmented Memory Model



3. EFLAGS



4. EIP

這個register會包含下一個要執行的instruction

<多媒體部份>:

在日漸增加的多媒體應用中，處理大量相同性質的資料的速度變得極為重要，因為多媒體資訊如音樂、3D顯示、影片等等，常常都是需要大量的重複性運算。為了這個需求，便有廠商開始開發所謂的SIMD的相關技術。

SIMD(**Single Instruction Multiple Data**)是一種應用在平行處理大量銅質資料的技術。單一指令同時對多資訊是SIMD,相對於多指令同時對多資訊是MIMD。如表格所示。

	單一指令	**多重指令**
單一Data	SISD	MISD
多重Data	SIMD	MIMD

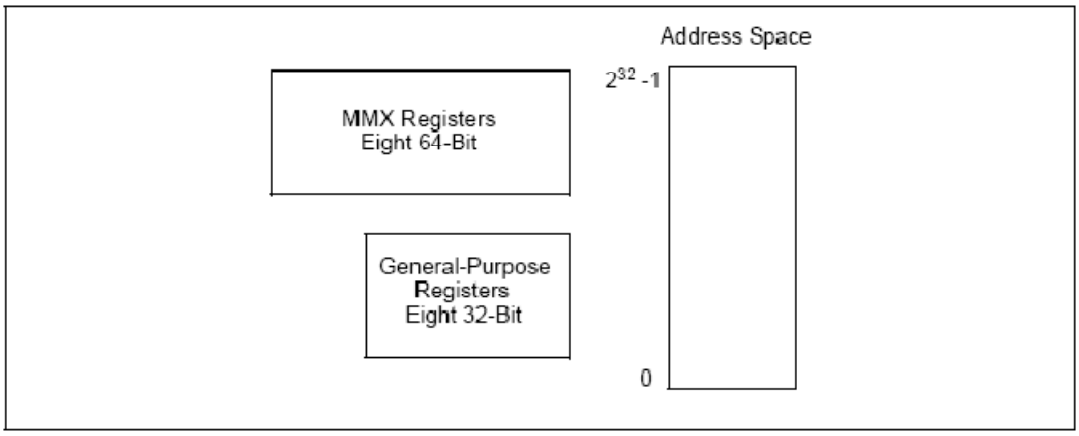
SIMD在x86系統上面應用非常的晚，主要是一開始的SIMD會將系統中的FPU的register占用，會某種程度上降低系統的效能。加上當時大部分的軟體和compiler沒有那麼多這方面的需求，使得一開始要推行x86上面的SIMD指令的廠商其實不是很順利。

一開始Intel推出了MMX，但是AMD也推出了自己的3DNow!，並強調修正了Intel在MMX上面犯的致命的失誤:將FPU的register拿來重複使用在MMX的Register上。不過到最後AMD還是妥協了開始支援Intel後續所推出的SIMD相關指令集:SSE,SSE2,SSE3等等。

其實在別的平台，如Apple的MAC平台上，AltiVec提供了提供了一系列的系統和一些支援了SIMD的compilers，也讓Apple自家的多媒體應用如iTunes和QuickTime，知道後來Apple轉移到Intel的x86系統底下，並且將以往的部份API轉寫爲支援SSE2和SSE3之後的版本。

以下就來稍微介紹比較具代表性的SIMD指令集的演進：

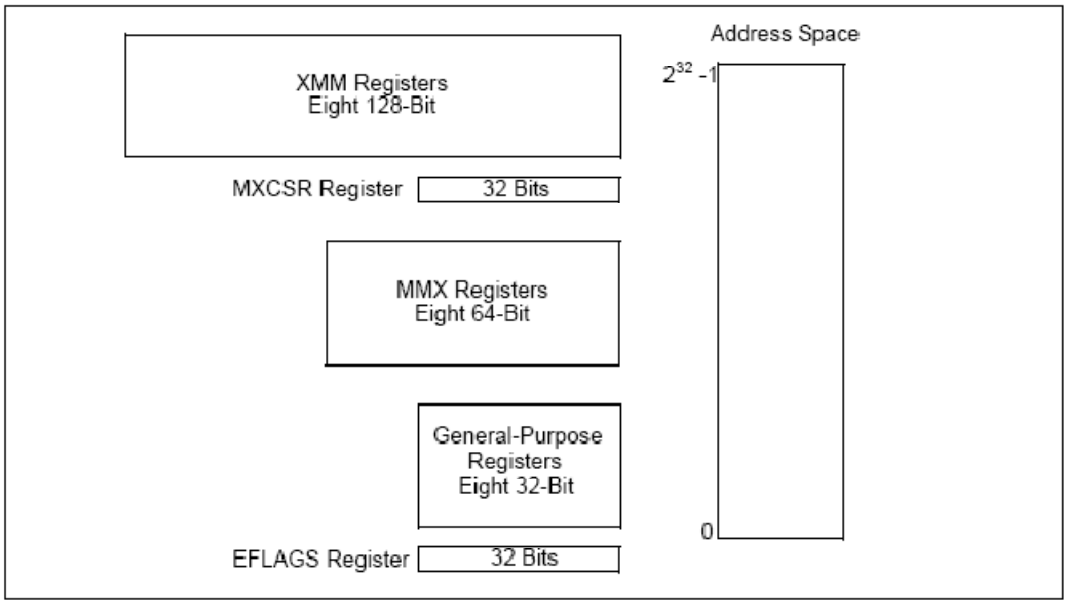
1. **MMX**:



MMX technology 提供了一種簡單且具有彈性的SIMD execution model 用來操作64-bit packed integer data. 但是MMX最大的缺點就如前面所提到的，會使用到x87的暫存器導致系統不能同時執行floating point和MMX的運算。

- 八個新的64-bit data registers,被稱爲MMX registers
- 三種新的資料型態 ：
  - — 64-bit packed byte integers (signed and unsigned)
  - — 64-bit packed word integers (signed and unsigned)
  - — 64-bit packed doubleword integers (signed and unsigned)
- Instructions that support the new data types and to handle MMX state management

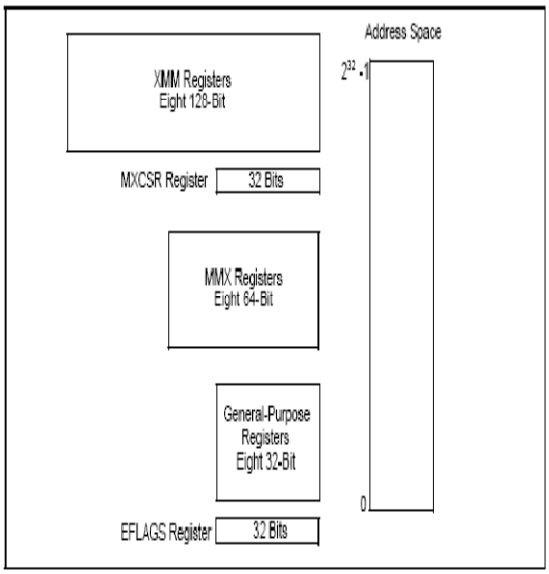
2. SSE:



SSE延伸指令集藉由增加對packed and scalar single-precison floating-point的操作，並且增加了128-bit的暫存器和新型態。

- 增加了八個128-bit的資料暫存器，稱作XMM。
- 沒有和x87共用暫存器
- 一種32-bit的暫存器:MXCSR←提供了在XMM暫存器上control和status bit的暫存器。
- 新的data type: 128-bit packed single-precision floating-point data type.

3. SSE2:



SSE2是繼SSE之後，為了多支援了packed double-precision floating-point與128-bit packed integers而來的。

- 增加了六種新的data type:
  - i. 128-bit packed double-precision floating point
  - ii. 128-bit packed byte integer
  - iii. 128-bit packed word integer
  - iv. 128-bit packed doubleword integer
  - v. 128-bit packed quadword integer
- 為了因應資訊安全上面的顧慮，包括了RSA Encryption和其他加解密方法，提供128-bit packed integers將會大幅加速這方面的程式執行速度。

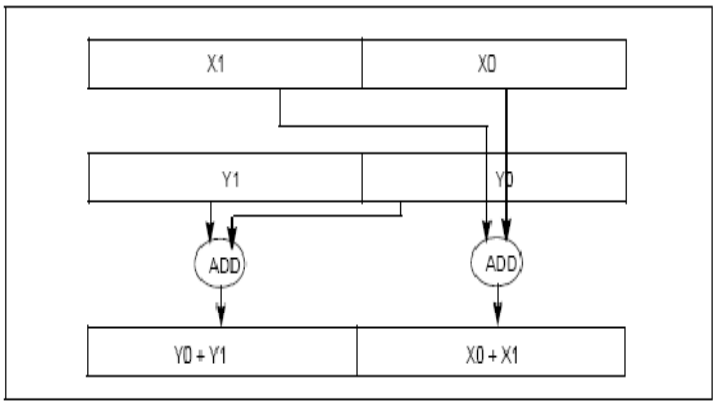
4. SSE3:

SSE3並沒有針對32-bit的系統在對SSE2指令集新增任何data type，而是專門

針對64-bit的架構來設計。

- 在64-bit 模式中，才可以存取新的8個XMM 暫存器(XMM8~XMM15)。
- 以往的SIMD指令大多是做垂直的運算，而SSE3最大的不同就是增加了水平運算的指令。

如圖所示→



5. SSE4:

SSE4將會被下一代的Intel 45nm製程的CPU核心: Penryn 所包含。

SSE4將會出現54個新的指令以支援Intel-64的處理器，並且分成了SSE 4.1和SSE 4.2。SSE 4.1將會在Penryn上就出現，而SSE 4.2則會延遲到Nehalem核心才推出。

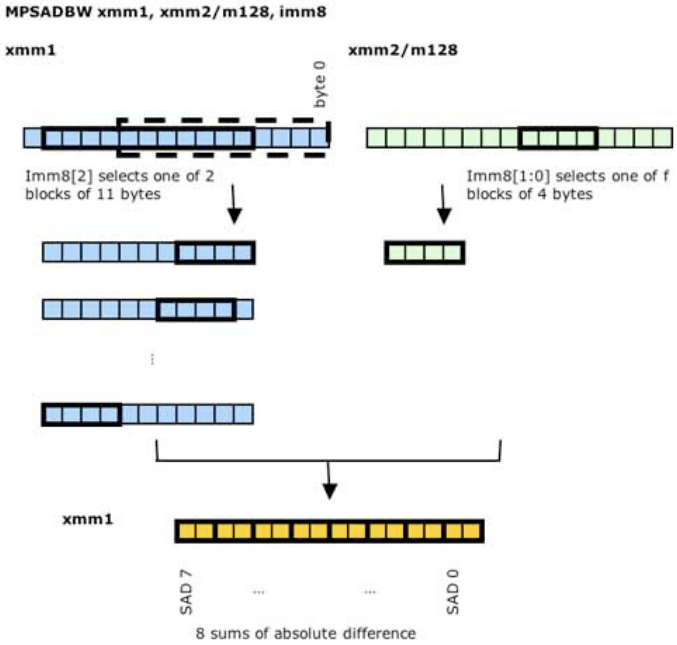
SSE 4.1 並沒有增加任何新的Data types，但是新增加了兩個容許兩個dword相成的instruction，可以增加向量的乘積計算。

		32 bit Integer Operation	
		unsigned x unsigned	signed x signed
Result	**Low 32-bit**	(not available)	PMULLD
	High 32-bit	(not available)	(not available)
	64-bit	PMULUDQ*	PMULDQ

NOTE:

* Available prior to SSE4.1.

實際上SSE 4.1的應用就包括了**Motion Estimation**：利用MPSADBW 和PHMINPOSUW。而SSE 4.1之所以對**Motion Estimation**的應用非常有用，是因為以往在這個operation中，資料都是以一連串blocks的pixels被讀取出來，並且利用在SSE2 就有的PSADBW指令進行加速。這指令一次計算兩對由16 unsigned byte integers組成的資料。但是在SSE 4.1中的MPSADBW指令卻可以一次做到八對。



用SSE 4.1之後和SSE 2的數據比較:

Code Sample	Cycles / Block SAD	Speedup
4x4 Block		
C++	54.84	
SSE2	4.32	1.00
Intel SSE4	2.71	1.59
8x8 Block		
C++	180.55	
SSE2	25.29	1.00
Intel SSE4	6.73	3.83
16x16 Block		
C++	173.01	
SSE2	71.42	1.00
Intel SSE4	26.86	2.66

<Conclusion>:

在CPU相關的議題和領域中，硬體和軟體的進步都同時重要。隨著製程的降低，耗電量也是廠商競爭的目標，加上在開發加速指令集方面，無不對CPU的效能評價有著很大的影響。32-bit的產品也慢慢會退出市場，接著應該會是64-bit以及多核心的天下。

<參考資料>:

- Wikipedia : www.wikipedia.org
- Intel: www.intel.com
- AMD: www.amd.com