CHAPTER 6

Pentium Pro Microprocessors

Topics - Introduction to Pentium Pro Microprocessor, Internal Structure of the Pentium Pro, The Memory System Multiple Core technology.

Text Book Used – The INTEL Microprocessors; Architecture, Programming and Interfacing By Barry B Brey (8^{th} Edition)

INTRODUCTION TO THE PENTIUM PRO MICROPROCESSOR

Before this or any other microprocessor can be used in a system, the function of each pin must be understood. This section of the chapter details the operation of each pin, along with the external memory system and I/O structures of the Pentium Pro microprocessor.

Figure 18–12 illustrates the pin-out of the Pentium Pro microprocessor, which is packaged in an immense 387-pin PGA (pin grid array). The Pentium Pro is available in two versions: One version contains a 256K level 2 cache; the other contains a 512K level 2 cache.

The most notable difference in the pin-out of the Pentium Pro, when compared to the Pentium, is that there are provisions for a 36-bit address bus, which allows access to 64G bytes of memory. This is meant for future use because no system today contains anywhere near that amount of memory.

As with most recent versions of the Pentium microprocessor, the Pentium Pro requires a single +3.3 V or +2.7 V power supply for operation. The power supply current is a maximum of 9.9 A for the 150 MHz version of the Pentium Pro, which also has a maximum power dissipation of 26.7 W. A good heat sink with considerable airflow is required to keep the Pentium Pro cool. As with the Pentium, the Pentium Pro contains multiple VCC and VSS connections that must all be connected for proper operation. The Pentium Pro contains VCCP pins (primary VCC) that connect to +3.1 V, VCCS (secondary VCC) pins that connect to +3.3 V, and VCC5 (standard VCC) pins that connect to +5.0 V. There are some pins that are labeled N/C (no connection) and must not be connected.

Each Pentium Pro output pin is capable of providing an ample 48.0 mA of current at a logic 0 level. This represents a considerable increase in drive current, compared to the 2.0 mA available on earlier microprocessor output pins. Each input pin represents a small load, requiring only 15 μA of current. Because of the 48.0 mA of drive current available on each output, only an extremely large system requires bus buffers.

Internal Structure of the Pentium Pro

The Pentium Pro is structured differently than earlier microprocessors. Early microprocessors contained an execution unit and a bus interface unit with a small cache buffering the execution unit for the bus interface unit. This structure was modified in later microprocessors, but the modifications were just additional stages within the microprocessors. The Pentium architecture is also a modification, but more significant than earlier microprocessors. Figure 18–13 shows a block diagram of the internal structure of the Pentium Pro microprocessor.

The system buses, which communicate to the memory and I/O, connect to an internal level 2 cache that is often on the main board in most other microprocessor systems. The level 2 cache in the Pentium Pro is either 256K bytes or 512K bytes. The integration of the level 2 cache speeds processing and reduces the number of components in a system.

The bus interface unit (BIU) controls the access to the system buses through the level 2 cache, as it does in most other microprocessors. Again, the difference is that the level 2 cache is integrated. The BIU generates the memory address and control signals, and passes and fetches data or instructions to either a level 1 data cache or a level 1 instruction cache. Each cache is 8K bytes in size at present and may be made larger in future versions of the microprocessor. Earlier versions of the Intel microprocessor contained a unified cache that held both instructions and data. The implementation of separate caches improves performance because data-intensive programs no longer fill the cache with data.

The instruction cache is connected to the instruction fetch and decode unit (IFDU). Although not shown, the IFDU contains three separate instruction decoders that decode three instructions simultaneously. Once decoded, the outputs of the three decoders are passed to the instruction pool, where they remain until the dispatch and execution unit or retire unit obtains them. Also included within the IFDU is a branch prediction logic section that looks ahead in code sequences that contain conditional jump instructions. If a conditional jump is located, the branch prediction logic tries to determine the next instruction in the flow of a program.
Once decoded instructions are passed to the instruction pool, they are held for processing. The instruction pool is a content-addressable memory, but Intel never states its size in the literature. The dispatch and execute unit (DEU) retrieves decoded instructions from the instruction pool when they are complete, and then executes them. The internal structure of the DEU is illustrated in Figure 18–14. Notice that the DEU contains three instruction execution units: two for processing integer instructions and one for floating-point instructions. This means that the Pentium Pro can process two integer instructions and one floating-point instruction simultaneously.
The Pentium also contains three execution units, but the architecture is different because the Pentium does not contain a jump execution unit or address generation units, as does the Pentium Pro. The reservation station (RS) can schedule up to five events for execution and process four simultaneously. Note that there are two station components connected to one of the address generation units that does not appear in the illustration of Figure 18–14.
The last internal structure of the Pentium Pro is the retire unit (RU). The RU checks the instruction pool and removes decoded instructions that have been executed. The RU can remove three decoded instructions per clock pulse.

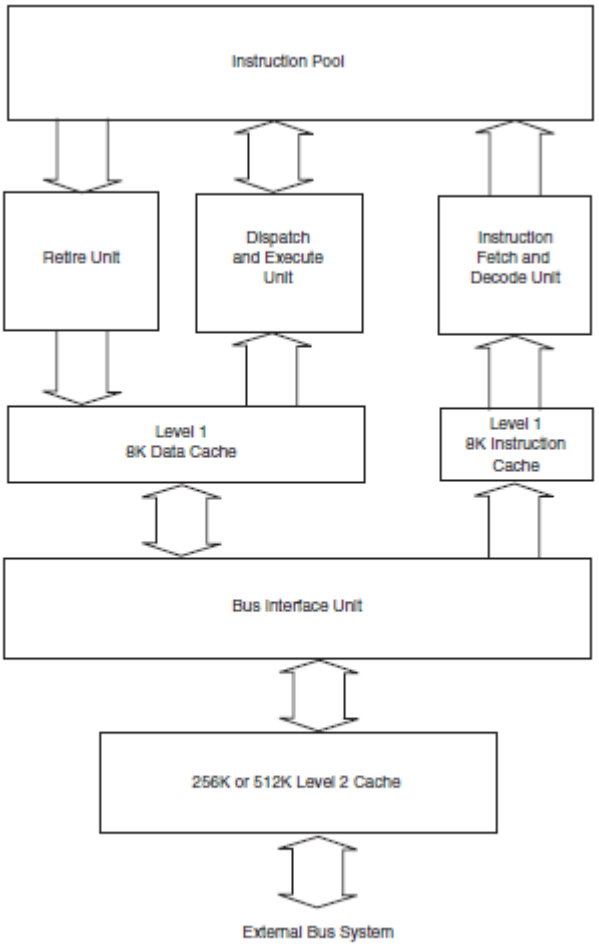

FIGURE 18–13 The internal structure of the Pentium Pro microprocessor.

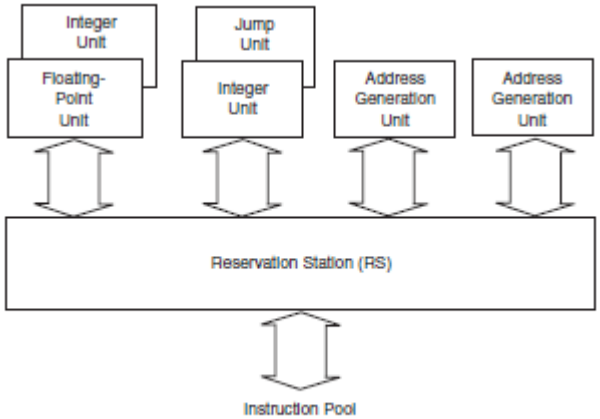

FIGURE 18–14 The Pentium Pro dispatch and execution unit (DEU).

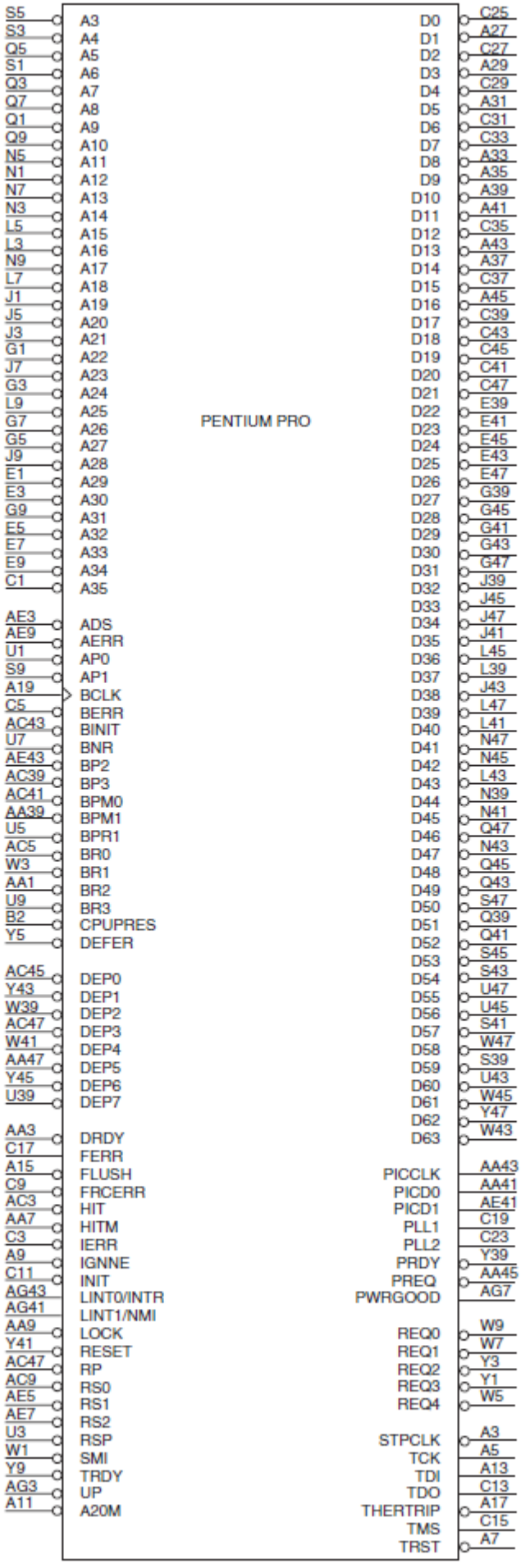

FIGURE 18–12 The pin-out of the Pentium Pro microprocessor.

Pin Connections

The number of pins on the Pentium Pro has increased from the 237 pins on the Pentium to 387 pins on the Pentium Pro. Following is a description of each pin or grouping of pins:

$\overline{\text{A20M}}$ — The **address A20 mask** is an input that is asserted in the real mode to signal the Pentium Pro to perform address wraparound, as in the 8086 microprocessor, for use of the HIMEM.SYS driver.

$\overline{\text{A35}}$–$\overline{\text{A3}}$ — **Address bus** connections address any of the 8G × 64 memory locations found in the Pentium Pro memory system.

$\overline{\text{ADS}}$ — The **address data strobe** becomes active whenever the Pentium Pro has issued a valid memory or I/O address.

$\overline{\text{AP1}}$, $\overline{\text{AP0}}$ — **Address parity** provides even parity for the memory address on all Pentium Pro–initiated memory and I/O transfers. The **$\overline{\text{AP0}}$** output provides parity for address connections A_{23}–A_3, and the **$\overline{\text{AP1}}$** output provides parity for address connections A_{35}–A_{24}.

$\overline{\text{ASZ1}}$, $\overline{\text{ASZ0}}$ — **Address size** inputs are driven to select the size of the memory access. Table 18–6 illustrates the size of the memory access for the binary bit patterns on these two inputs to the Pentium Pro.

BCLK — The **bus clock** input determines the operating frequency of the Pentium Pro microprocessor. For example, if BCLK is 66 MHz, various internal clocking speeds are selected by the logic levels applied to the pins in Table 18–7. A BCLK frequency of 66 MHz runs the system bus at 66 MHz.

$\overline{\text{BERR}}$ — The **bus error** input/output either signals a bus error along or is asserted by an external device to cause a machine check interrupt or a non-maskable interrupt.

TABLE 18–6 Memory size dictated by the ASZ pins.

$\overline{\text{ASZ1}}$	*$\overline{\text{ASZ0}}$*	*Memory Size*
0	0	0–4G
0	1	4G–64G
1	X	Reserved

TABLE 18–7 The BCLK signal and its effect on the Pentium clock speed.

LINT1/NMI	*LINT0/INTR*	*$\overline{\text{IGNNE}}$*	*$\overline{\text{A20M}}$*	*Ratio*	*BCLK = 50 MHz*	*BCLK = 66 MHz*
0	0	0	0	2	100 MHz	133 MHz
0	0	0	1	4	200 MHz	266 MHz
0	0	1	0	3	150 MHz	200 MHz
0	0	1	1	5	250 MHz	333 MHz
0	1	0	0	5/2	125 MHz	166 MHz
0	1	0	1	9/2	225 MHz	300 MHz
0	1	1	0	7/2	175 MHz	233 MHz
0	1	1	1	11/2	275 MHz	366 MHz
1	1	1	1	2	100 MHz	133 MHz

$\overline{\text{BINIT}}$	**Bus initialization** is active on power-up to initialize the bus system.
$\overline{\text{BNR}}$	**Block next request** is used to halt the system in a multiple microprocessor system.
$\overline{\text{BP3}}$, $\overline{\text{BP2}}$	The **breakpoint status** outputs indicate the status of the Pentium Pro breakpoints.
$\overline{\text{BPM1}}$, $\overline{\text{BPM0}}$	The **break point monitor** outputs indicate the status of the breakpoints and programmable counters.
$\overline{\text{BPRI}}$	The **priority agent bus request** is an input that causes the microprocessor to cease bus requests.
$\overline{\text{BR3}}$–$\overline{\text{BR0}}$	The **bus request** inputs allow up to four Pentium Pro microprocessors to coexist on the same bus system.
$\overline{\text{BREQ3}}$–$\overline{\text{BREQ0}}$	**Bus request signals** are used for multiple microprocessors on the same system bus.
$\overline{\text{D63}}$–$\overline{\text{D0}}$	**Data bus** connections transfer byte, word, doubleword, and quadword data between the microprocessor and its memory and I/O system.
$\overline{\text{DBSY}}$	**Data bus busy** is asserted to indicate that the data bus is busy transferring data.
$\overline{\text{DEFER}}$	The **defer** input is asserted during the snoop phase to indicate that the transaction cannot be guaranteed in-order completion.
$\overline{\text{DEN}}$	The **defer enable** signal is driven to the bus on the second phase of a request phase.
$\overline{\text{DEP7}}$–$\overline{\text{DEP0}}$	**Data bus ECC protection** signals provide error-correction codes for correcting a single-bit error and detecting a double-bit error.
$\overline{\text{FERR}}$	The **floating-point error**, comparable to the ERROR line in the 80386, shows that the internal coprocessor has erred.
$\overline{\text{FLUSH}}$	The **flush** cache input causes the cache to flush all write-back lines and invalidate its internal caches. If the $\overline{\text{FLUSH}}$ input is a logic 0 during a reset operation, the Pentium Pro enters its test mode.
$\overline{\text{FRCERR}}$	**Functional redundancy check error** is used if two Pentium Pro microprocessors are configured in a pair.
$\overline{\text{HIT}}$	**Hit** shows that the internal cache contains valid data in the inquire mode.
$\overline{\text{HITM}}$	**Hit modified** shows that the inquire cycle found a modified cache line. This output is used to inhibit other master units from accessing data until the cache line is written to memory.

TABLE 18–8 The $\overline{\text{LEN}}$ bits and data size.

$\overline{\textit{LEN1}}$	$\overline{\textit{LEN0}}$	*Data Transfer Size*
0	0	0–8 bytes
0	1	16 bytes
1	0	32 bytes
1	1	Reserved

$\overline{\text{IERR}}$ — **Internal error** output shows that the Pentium Pro has detected an internal parity error or functional redundancy error.

$\overline{\text{IGNNE}}$ — The **ignore numeric error** input causes the Pentium Pro to ignore a numeric coprocessor error.

INIT — The **initialization** input performs a reset without initializing the caches, write-back buffers, and floating-point registers. This input may not be used to reset the microprocessor in lieu of RESET after power-up.

INTR — The **interrupt request** is used by external circuitry to request an interrupt.

$\overline{\text{LEN1}}$, $\overline{\text{LEN0}}$ — Length signals (bit 0 and 1) indicate the size of the data transfer, as illustrated in Table 18–8.

$\overline{\text{LINT1}}$, $\overline{\text{LINT0}}$ — The **local interrupt** inputs function as NMI and INTR, and also set the clock divider frequency on reset.

$\overline{\text{LOCK}}$ — $\overline{\text{LOCK}}$ becomes a logic 0 whenever an instruction is prefixed with the LOCK: prefix. This is most often used during DMA accesses.

NMI — The non-maskable interrupt requests a non-maskable interrupt, as it did on the earlier versions of the microprocessor.

PICCLK — The **clock** signal input is used for synchronous data transfers.

PICD — The **processor interface serial data** is used to transfer bidirectional serial messages between Pentium Pro microprocessors.

PWRGOOD — **Power good** is an input that is placed at a logic 1 level when the power supply and clock have stabilized.

$\overline{\text{REQ4}}$–$\overline{\text{REQ0}}$ — **Request signals** (bits 0–4) define the type of data-transfer operation, as illustrated in Tables 18–9 and 18–10.

$\overline{\text{RESET}}$ — **Reset** initializes the Pentium Pro, causing it to begin executing software at memory location FFFFFFF0H. The Pentium Pro is reset to the real mode and the leftmost 12 address connections remain logic 1s (FFFH) until a far jump or far call is executed. This allows compatibility with earlier microprocessors.

$\overline{REQ4}$	$\overline{REQ3}$	$\overline{REQ2}$	$\overline{REQ1}$	$\overline{REQ0}$	*Function*
0	0	0	0	0	Deferred reply
0	0	0	0	1	Reserved
0	1	0	0	0	Case 1*
0	1	0	0	1	Case 2*
1	0	0	0	0	I/O read
1	0	0	0	1	I/O write
X	X	0	1	0	Memory read
X	X	0	1	1	Memory write
X	X	1	0	0	Memory code read
X	X	1	1	0	Memory data read
X	X	1	X	1	Memory write

*See Table 18–10 for the second clock pulse for Case 1 and Case 2.

TABLE 18–9 Function of the request inputs on the first clock pulse.

TABLE 18–10 Function of the request inputs for Case 1 and Case 2.

Case	$\overline{REQ4}$	$\overline{REQ3}$	$\overline{REQ2}$	$\overline{REQ1}$	$\overline{REQ0}$	*Function*
1	X	X	X	0	0	Interrupt acknowledge
1	X	X	X	0	1	Special transaction
1	X	X	X	1	X	Reserved
2	X	X	X	0	0	Branch trace message
2	X	X	X	0	1	Reserved
2	X	X	X	1	X	Reserved

$\overline{\text{RP}}$	**Request parity** provides a means of requesting that the Pentium Pro checks parity.
$\overline{\text{RS2}}$–$\overline{\text{RS0}}$	The **response status** inputs cause the Pentium Pro to perform the functions listed in Table 18–11.
$\overline{\text{RSP}}$	The **response parity** input applies a parity error signal from an external parity checker.
$\overline{\text{SMI}}$	The **system management interrupt** input causes the Pentium Pro to enter the system management mode of operation.
$\overline{\text{SMMEM}}$	The **system memory-management mode** signal becomes a logic 0 whenever the Pentium Pro is executing in the system memory-management mode interrupt and address space.
$\overline{\text{SPCLK}}$	The **split lock** signal is placed at a logic 0 level to indicate that the transfer will contain four locked transactions.
$\overline{\text{SPCLK}}$	**Stop clock** causes the Pentium Pro to enter the power-down state when placed at a logic 0 level.
TCK	The **testability clock** input selects the clocking function in accordance with the IEEE 1149.1 Boundary Scan interface.
TDI	The **test data input** is used to test data clocked into the Pentium Pro with the TCK signal.
TDO	The **test data output** is used to gather test data and instructions shifted out of the Pentium with TCK.
TMS	The **test mode select** input controls the operation of the Pentium Pro in test mode.
$\overline{\text{TRDY}}$	The **target ready** input is asserted when the target is ready for a data transfer operation.

$\overline{RS2}$	$\overline{RS1}$	$\overline{RS0}$	*Function*	$\overline{HITM}$	$\overline{DEFER}$
0	0	0	Idle state	X	X
0	0	1	Retry	0	1
0	1	0	Defer	0	1
0	1	1	Reserved	0	1
1	0	0	Hard failure	X	X
1	0	1	Normal, no data	0	0
1	1	0	Implicit write-back	1	X
1	1	1	Normal, with data	0	0

TABLE 18–11 Operation of the response status inputs.

The Memory System

The memory system for the Pentium Pro microprocessor is 4G bytes in size, just as in the 80386DX–Pentium microprocessors, but access to an area between 4G and 64G is made possible by additional address signals A32–A35. The Pentium Pro uses a 64-bit data bus to address memory organized in eight banks that each contain 8G bytes of data. Note that the additional memory is enabled with bit position 5 of CR4 and is accessible only when 2M paging is enabled.

Note also that 2M paging is new to the Pentium Pro to allow memory above 4G to be accessed. More information is presented on Pentium Pro paging later in this chapter. Refer to Figure 18–15 for the organization of the Pentium Pro physical memory system.

The Pentium Pro memory system is divided into eight banks where each bank stores a byte-wide data with a parity bit. Note that most Pentium and Pentium Pro microprocessor-based systems forgo the use of the parity bit. The Pentium Pro, like the 80486 and Pentium, employs internal parity generation and checking logic for the memory system data bus information. The 64-bit-wide memory is important to double-precision floating-point data. Recall that a double precision floating-point number is 64 bits wide. As with

earlier Intel microprocessors, the memory system is numbered in bytes from byte 000000000H to byte FFFFFFFFH. This nine-digit hexadecimal address is employed in a system that addresses 64G of memory.
Memory selection is accomplished with the bank enable signals (–). In the Pentium Pro microprocessor, the bank enable signals are presented on the address bus (A15–A8) during the second clock cycle of a memory or I/O access. These must be extracted from the address bus to access memory banks. The separate memory banks allow the Pentium Pro to access any single byte, word, doubleword, or quadword with one memory transfer cycle. As with earlier memory selection logic, we often generate eight separate write strobes for writing to the memory system. Note that the memory write information is provided on the request lines from the microprocessor during the second clock phase of a memory or I/O access.
A new feature added to the Pentium and Pentium Pro is the capability to check and generate parity for the address bus during certain operations. The pin (Pentium) or pins (Pentium Pro) provide the system with parity information, and the (Pentium) or pins (Pentium Pro) indicate a bad parity check for the address bus. The Pentium Pro takes no action when an address-parity error is detected. The error must be assessed by the system, and the system must take appropriate action (an interrupt) if so desired.
New to the Pentium Pro is a built-in error-correction circuit (ECC) that allows the correction of a one-bit error and the detection of a two-bit error. To accomplish the detection and correction of errors, the memory system must have room for an extra 8-bit number that is stored with each 64-bit number. The extra 8 bits are used to store an error-correction code that allows the Pentium Pro to automatically correct any single-bit error. A 1M × 64 is a 64M SDRAM without ECC, and a 1M × 72 is an SDRAM with EEC support. The ECC code is much more reliable than the old parity scheme, which is rarely used in modern systems. The only drawback of the ECC scheme is the additional cost of SDRAM that is 72 bits wide.

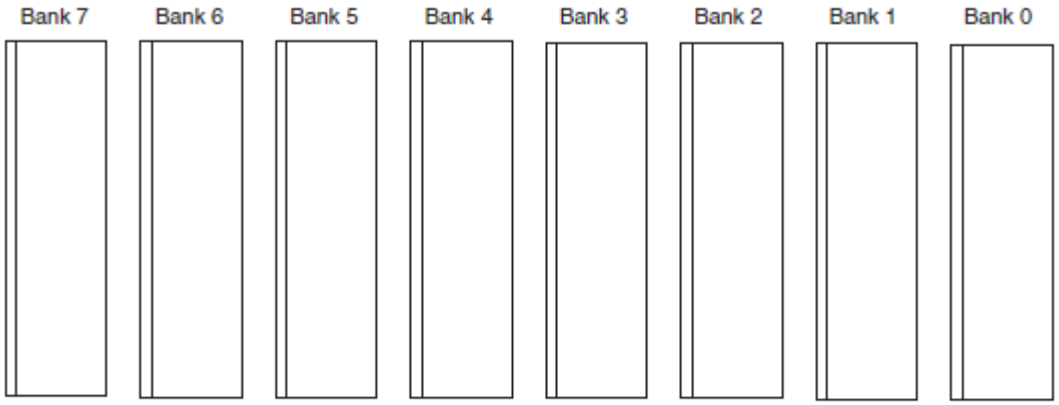

FIGURE 18–15 The eight memory banks in the Pentium Pro system. Note that each bank is 8 bits wide and 8G long if 36-bit addressing is enabled.

Multiple Core Technology

Most new versions of the Pentium 4 and Core2 contain either dual or quad cores. Each core is a separate version of the microprocessor that independently executes a separate task. Three versions are currently available: the Pentium D, which contains two cores; with separate caches; a Core2 Duo version that contains a shared cache, but two cores and a quad core version, which contains four cores. Intel seems to have migrated to a shared cache for multiple core microprocessors. A recent article from Intel stated that in the future the Pentium or whatever it will be called may contain up to 80 cores. The Core2 Duo contains either a 2M or 4M byte cache and operates at frequencies to 3 GHz. It certainly appears that the speed race is over and the clock frequency has stabilized at between 3 and 4GHz. Does this mean that in the future a 5 GHz version will never become available? It is possible, but at this time a much higher clock frequency appears to be impossible, so multiple cores using threaded application seem to be the prospect for some time to come. It appears silicon technology has reached its apex. What this means is that efficient programming will become the avenue for increasing the speed of computer systems.

CPUID

As in earlier versions of the Pentium, the CPUID instruction accesses information that indicates the type of microprocessor as well as the features supported by the microprocessor. In the ever evolving series of microprocessors it is important to be able to access this information so that efficient software can be written to operate on many different versions of the microprocessor. Table 19–7 lists the latest features available to the CPUID instruction. To access these features, EAX is loaded with the input number listed in the table, then the CPUID instruction is executed. The CPUID instruction usually returns information in the EAX, EBX, ECX, and EDX registers in the real or protected mode. As can be gleaned from the table, additional features have been added to the CPUID instruction when compared to previous versions.

TABLE 19–7 Pentium 4 CPUID instruction.

EXA Input Value	*Output Registers*	*Notes*
0	EAX = Maximum input value EBX = "uneG" ECX = "lene" EDX = "letn"	"GenuineIntel" is returned in little endian format
1	EAX = Version information EBX = Feature information ECX = Extended feature information EDX = Feature information	Feature information
2	EAX, EBX, ECX, and EDX	Cache and TLB information
3	ECX and EDX	Serial number in the Pentium III only
4	EAX, EBX, ECX, and EDX	Deterministic cache parameters
5	EAX, EBX, ECX, and EDX	Monitor/Mwait information
80000000H	EAX	Extended function information
80000001H	EAX	Reserved
80000002H, 80000003H, and 80000004H	EAX, EBX, ECX, and EDX	Processor brand string
80000006H	ECX	Cache information

In Chapter 18 software was developed to read and display the data available after the CPUID instruction was invoked with EAX = 1. Here we deal with reading the processor brand string and prepare it for display in a Visual C++ function. The brand string, if supported, contains the frequency that the microprocessor is certified to operate and also the genuine Intel keyword. The BrandString function (see Example 19–3) returns a CString that contains the information stored in the CPUID members 80000002H–80000004H. This software requires a Pentium 4 system for proper operation as tested for in BrandString function. The Convert function reads the contents of EAX, EBX, ECX, and EDX from the register specified as the parameter and converts them to a CString that is returned. The author's system shows that the brand string is
"Intel(R) Pentium(R) 4 CPU 3.06GHz"

EXAMPLE 19–3

```
int getCPU(int EAXvalue)
{
        int temp;
        _asm
        {
                mov eax,EAXvalue
                cpuid
                mov temp1,eax
        }
        return temp;
}
```

```
private: System::String^ BrandString(void)
{
	String^ temp;
	int temp1 = getCpu(0x8000000);
	if ( temp1 >= 0x80000004 ) //if brand string present
	{
		temp += Convert(0x80000002); //read register 80000002H
		temp += Convert(0x80000003); //read register 80000003H
		temp += Convert(0x80000004); //read register 80000004H
	}
	return temp;
}

private: System::String^ Convert(int EAXvalue)
{
	CString temp =“ ”; //must be 16 spaces
	int temp1, temp2, temp3, temp4;
	_asm
	{
		mov eax,EAXvalue
		cpuid
		mov temp1,eax
		mov temp2,ebx
		mov temp3,ecx
		mov temp4,edx
	}
	for ( int a = 0; a <4; a++ )
	{
		temp.SetAt(a, temp1);
		temp.SetAt(a + 4, temp2);
		temp.SetAt(a + 8, temp3);
		temp.SetAt(a + 12, temp4);
		temp1 >>= 8;
		temp2 >>= 8;
		temp3 >>= 8;
		temp4 >> =8;
	}
	return temp;
}
```

The other information available about the system is returned in EAX, EBX, ECX, and EDX after executing CPUID after loading EAX with a 1. The EAX register contains the version information as the model, family, and stepping information, as illustrated in Figure 19–7. The EBX register contains information about the cache, such as the size of the cache line flushed by the CFLUSH instruction in bits 15–8 and the ID assigned the local APIC on reset in bits 31–24. Bits 23–16 indicate how many internal processors are available to hyper-threading (two for the current Pentium 4 microprocessor). Example 19–4 shows a function that identifies the number of processors in a hyper-threaded CPU and returns it as a character string. If more than nine processors are eventually added to the microprocessor, then the software in Example 19–4 would need to be modified.

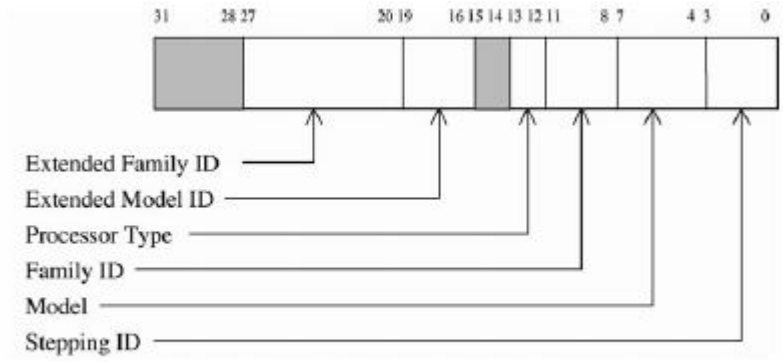

FIGURE 19–7 EAX after a CPUID instruction showing version information.

EXAMPLE 19–4

```
CString CCPUIDDlg::GetProcessorCount(void)
{
        CString temp = “This CPU has ”;
        char temp1;
        _asm
        {
                mov eax,1
                cpuid
                mov temp1,31h
                bt edx,28 ;check for hyper-threading
                jnc GetPro1 ;if no hyper-threading, temp1 = 1
                bswap ebx
                add bh,30h
                mov temp1,bh
                GetPro1:
        }
        return temp + temp1 + “processors.”;
}
```

Feature information for the microprocessor is returned in ECX and EDX as indicated in Figures 19–8 and 19–9. Each bit is a logic 1 if the feature is present. For example, if hyperthreading is needed in an application bit, position 28 is tested in EDX to see if hyper-threading is supported. This appears in Example 19–4 along with reading the number of processors found in a hyper-threaded microprocessor. The BT instruction tests the bit indicated and places it into the carry flag. If the bit under test is a 1, then the resultant carry is one and if the bit under test is a 0, the resultant carry is zero.

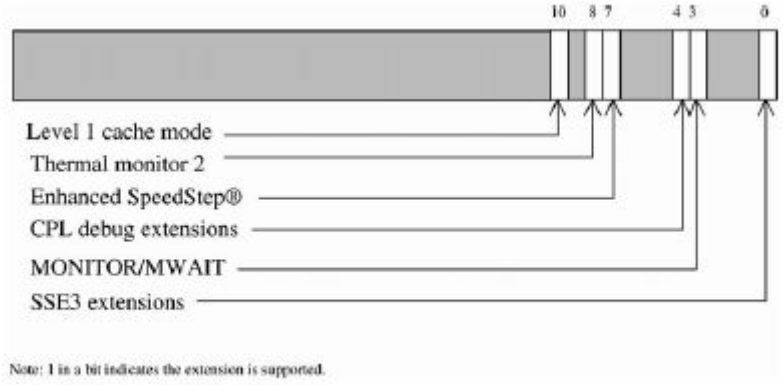

FIGURE 19–8 ECX after a CPUID instruction showing the version extensions.

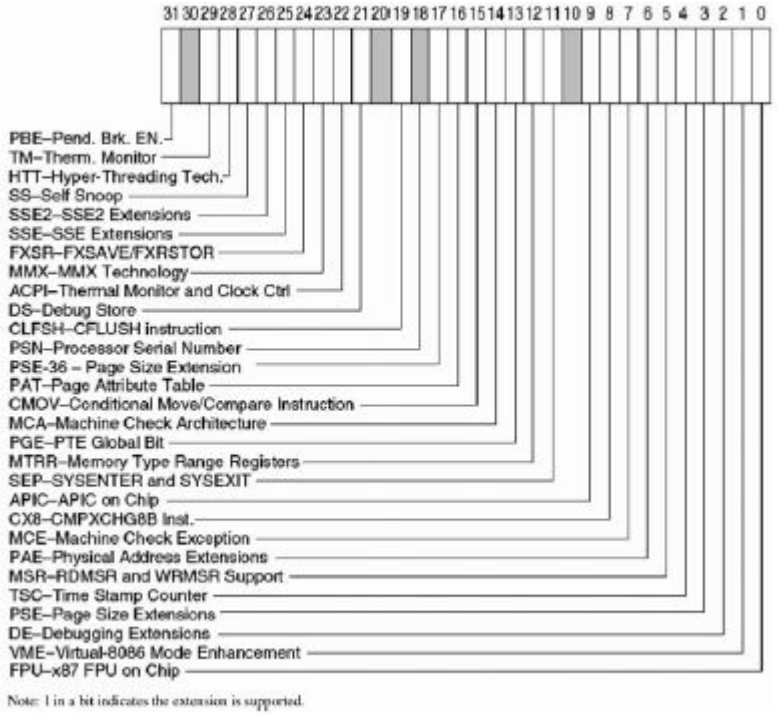

FIGURE 19–9 EDX after a CPUID instruction showing the version extensions.

Model-Specific Registers

As with earlier versions of the Pentium, the Pentium 4 and Core2 also contain model-specific registers that are read with the RDMSR instruction and written with the WRMSR instruction. The Pentium 4 and Core2 each have 1743 model-specific registers numbered from 0H to 6CFH. Intel does not provide information on all of them. The registers not identified are either reserved by Intel or used for some undocumented feature or function.

Both the read and write model-specific register instructions function in the same manner. Register ECX is loaded with the register number to be accessed, and the data are transferred through the EDX:EAX register pair as a 64-bit number where EDX is the most significant 32 bits and EAX is the least significant bits. These registers must be accessed in either the real mode (DOS) or in ring 0 of protected mode. These registers are normally accessed by the operating system and cannot be accessed in normal Visual C++ programming.

Performance-Monitoring Registers

Another feature in the Pentium 4 is a set of performance-monitoring registers (PMR) that, like the model-specific registers, can only be used in real mode or at ring 0 of protected mode. The only register that can be accessed via user software is the time-stamp counter, which is a performance monitoring register. The remaining PMRs are accessed with the RDPMR. This instruction is similar to the RDMSR instruction in that it uses ECX to specify the register number and the result appears in EDX:EAX. There is no write instruction for the PMRs.

64-Bit Extension Technology

Intel has released its 64-bit extension technology for most members of the Intel 32-bit architecture family. The instruction set and architecture is backwards compatible to the 8086, which means that the instructions and register set have remained compatible. (The only things that are not compatible are a few of the legacy instructions and some instructions that deal with AH, BH, CH, and DH.) What is changed is that the register set is stretched to 64 bits in width in place of the current 32-bit-wide registers. Refer to Figure 19–10 for the programming model of the Pentium 4 and Core2 in 64-bit mode.

Notice that the register set now contains sixteen 64-bit-wide general-purpose registers, RAX, RBX, RCX, RDX, RSP, RBP, RDI, RSI, R8–R15. The instruction pointer is also stretched to a width of 64 bits, allowing the microprocessor to address memory using a 64-bit memory address. This allows the microprocessor to address as much memory as the specific implementation of the microprocessor has address pins.

The registers are addressed as 64-bit, 32-bit, 16-bit, or 8-bit registers. An example is R8 (64 bits), R8D (32 bits), R8W (16 bits), and R8L (8 bits). There is no way to address the high byte (as in BH) for a numbered register; only the low byte of a numbered register can be addressed. Legacy addressing such as MOV AH,AL functions correctly, but addressing a legacy high-byte register and a numbered low-byte register is not allowed. In other words, MOV AH,R9L is not allowed, but MOV AL,R9L is allowed. If the MOV AH,R9L instruction is included in a program no error will occur; instead the instruction will be changed to MOV BPL, R9L. AH, BH, CH, and DH are changed to the low-order 8 bits (the L is for low order) of BPL, SPL, DIL, and SIL, respectively. Otherwise the legacy registers can be mixed with the new numbered registers R8–R15 as in MOV R11, RAX, MOV R11D, ECX, or MOV BX, R14W.

Another addition to the architecture is a set of additional SSE registers numbered XMM8–XMM15. These registers are accessed by the SSE, SSE2, or SSE3 instructions. Otherwise, the SSE unit has not been changed. The control and debug registers are expanded to 64 bits in width. A new model-specific register is added to control the extended features at address C0000080H. Figure 19–11 depicts the extended feature control register.

SCE The **system CALL enable** bit is set to enable the SYSCALL and SYSRET instructions in the 64-bit mode.

LME The **mode enable** bit is set to allow the microprocessor to use the 64-bit extended mode.

LMA The **mode active** bit shows that the microprocessor is operating in the 64-bit extended mode.

The protected mode descriptor table registers are expanded in the extended 64-bit mode so that each descriptor table register, GDTR, LDTR, IDTR, and the task register (TR) hold a 64-bit base address instead of a 32-bit base address. The biggest change is that the base address and limits of the segment descriptors are ignored. The system uses a base address of 0000000000000000H for the code segment and the DS, ES, and SS segments are ignored.

Paging is also modified to include a paging unit that supports the translation of a 64-bit linear address into a 52-bit physical address. Intel states that in the first version of this 64-bit Pentium the linear address will be 48 bits and the physical address will be 40 bits. This means that there will be a 40-bit address to support 1T (terra) byte of physical memory translated from a linear address space of 256T bytes. The 52-bit address accesses 4P (peta) bytes of memory and a 64-bit linear address accesses 16E (exa) bytes or memory. The translation is accomplished with additional tables in the paging unit. In place of two tables (a page directory and a page table), the 64-bit extended paging unit uses four levels of page tables.

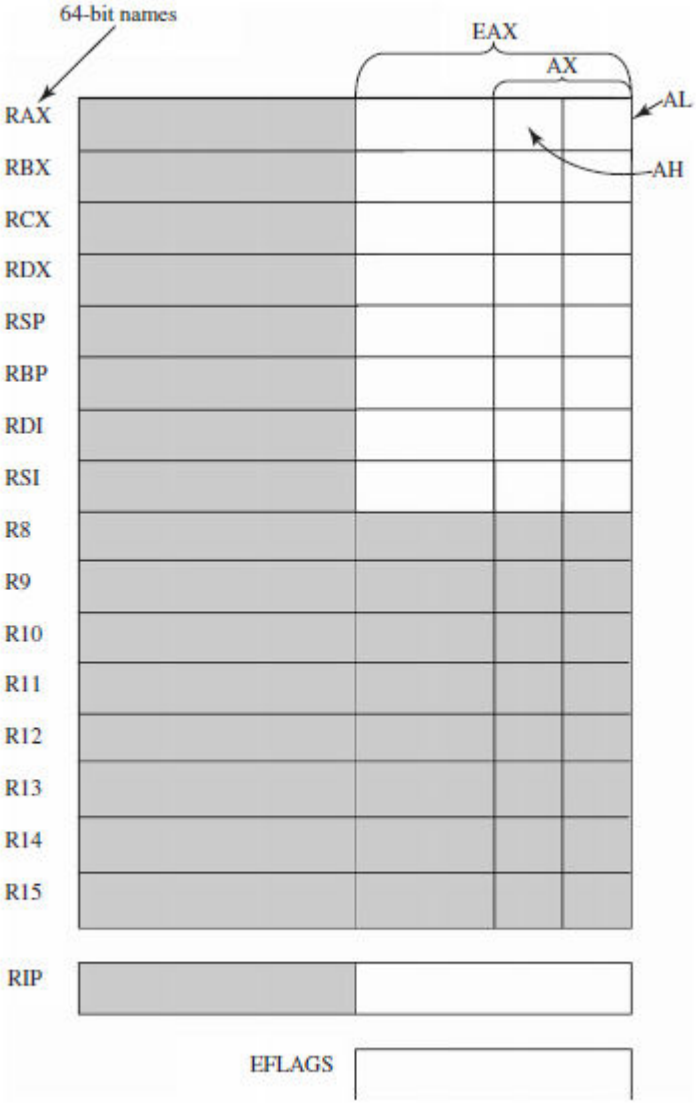

FIGURE 19–10 The integer register set in the 64-bit mode of the Pentium 4. The shaded areas are new to the Pentium 4 operated in the 64-bit mode.

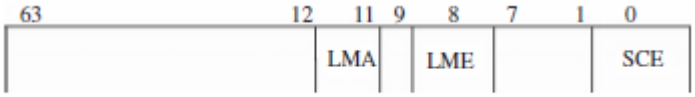

FIGURE 19–11 The contents of the extended feature model-specific register.