Weifan Lin

Csc342 Section G

04/01/2015

Homework

Title:

Report on CPUID

Objective:

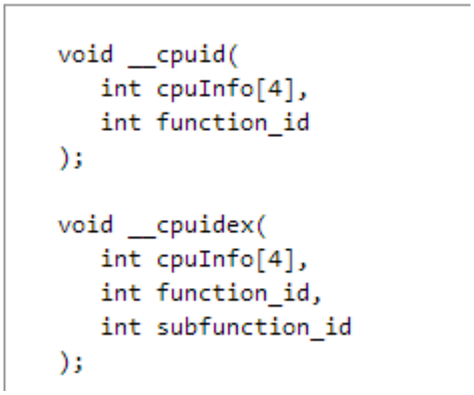
The goal of this assignment was to help us to find the field of the process support for my laptop. It was required to allow software to discover details of the processor that my laptop is currently using and which SSE that it belongs and features that it will support. In addition, I will add other information such as CPU information Type at different location of the CPU.

Specifications and Functionality:

Cupid is an instruction that is available on x86 and x64 architecture. This instruction queries the processor for information about supported features and CPU type. Intrinsic stores the supported features and CPU information returned by the **cupid** instruction in cpuInfo, which is an array of four 32-bit integers that is filled with the values of EAX, EBX, ECX, and EDX register. The information returned has a different meaning depending on the value passed as the function_id parameter. The information returned with various values The information returned with various values of function_id is processor-dependent.

The __cpuid intrinsic clears the ECX register before calling the cpuid instruction. The __cpuidex intrinsic sets the value of the ECX register to subfunction_id before it generates the cpuid instruction. This enables you to gather additional information about the processor.

An example of the code is listed below



```
void __cpuid(
   int cpuInfo[4],
   int function_id
);

void __cpuidex(
   int cpuInfo[4],
   int function_id,
   int subfunction_id
);
```

Input

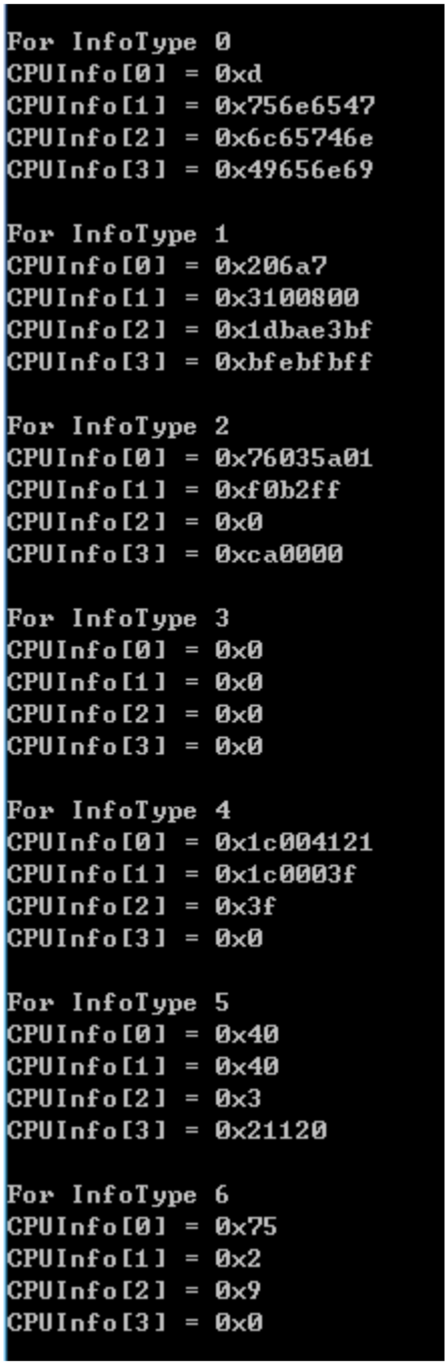
Function_id: This is a code that specifies the information to retrieve, passed in EAX.

Subfunction_id: An additional code that specifies information to retrieve, passed in ECX.

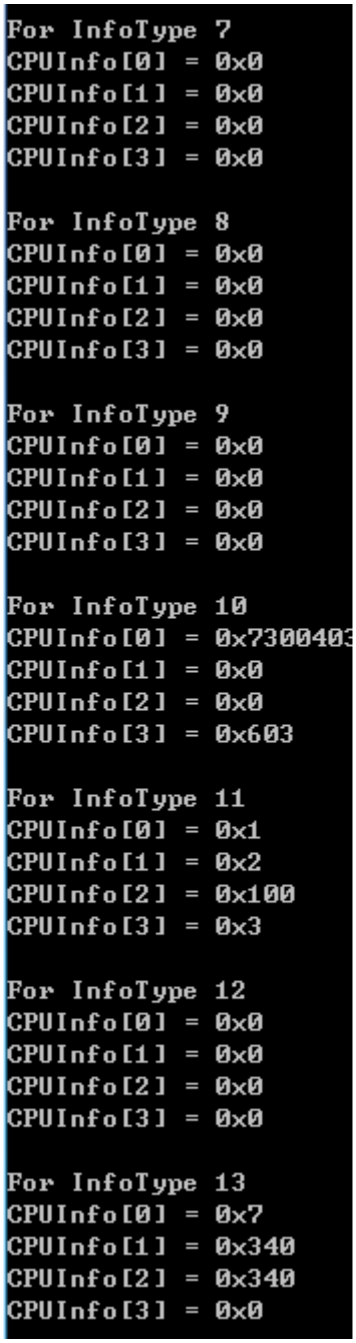
Output:

CpuInfo: An array of four integers that contains the information returned in EAX, EBX, ECX, and EDX about supported features of the CPU.

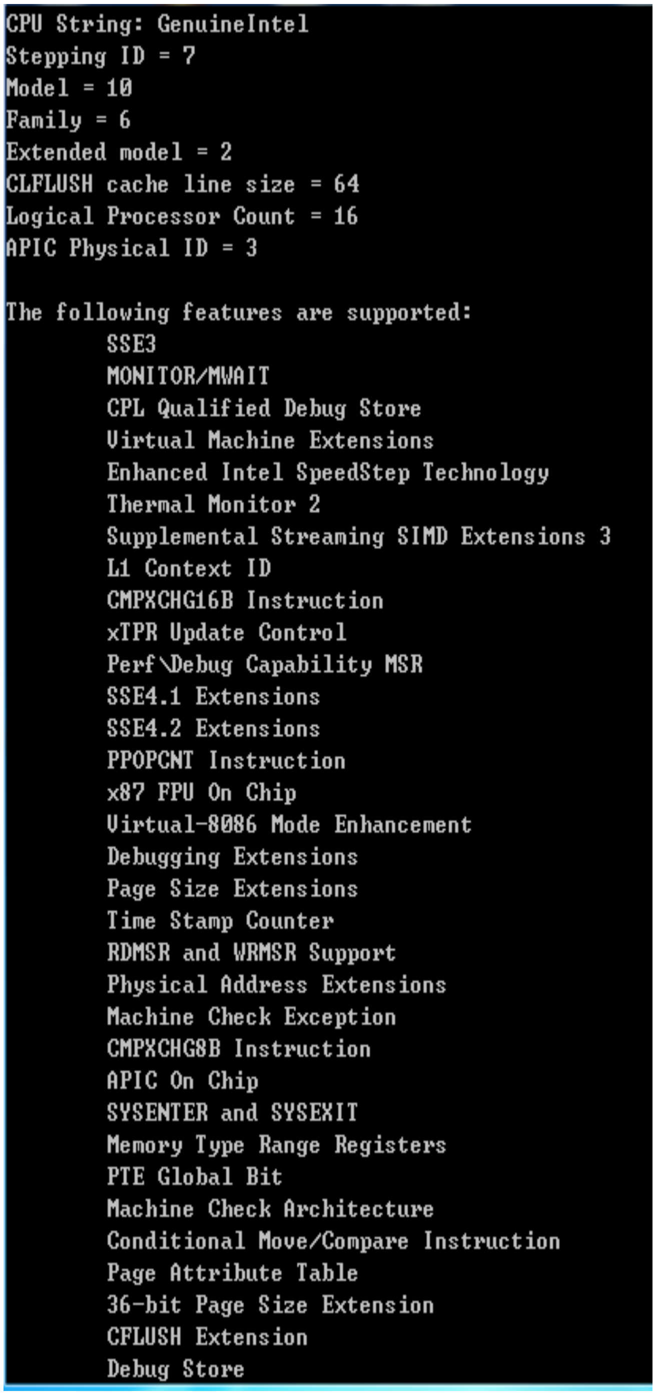
Design:



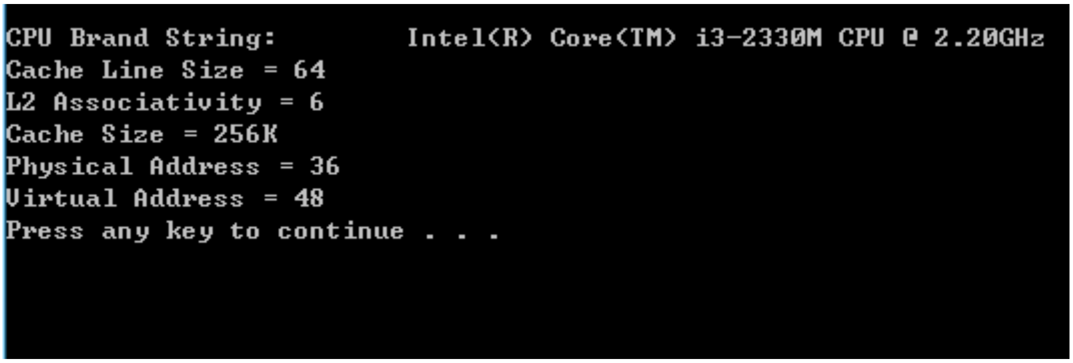
```
For InfoType 0
CPUInfo[0] = 0xd
CPUInfo[1] = 0x756e6547
CPUInfo[2] = 0x6c65746e
CPUInfo[3] = 0x49656e69

For InfoType 1
CPUInfo[0] = 0x206a7
CPUInfo[1] = 0x3100800
CPUInfo[2] = 0x1dbae3bf
CPUInfo[3] = 0xbfebfbff

For InfoType 2
CPUInfo[0] = 0x76035a01
CPUInfo[1] = 0xf0b2ff
CPUInfo[2] = 0x0
CPUInfo[3] = 0xca0000

For InfoType 3
CPUInfo[0] = 0x0
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x0

For InfoType 4
CPUInfo[0] = 0x1c004121
CPUInfo[1] = 0x1c0003f
CPUInfo[2] = 0x3f
CPUInfo[3] = 0x0

For InfoType 5
CPUInfo[0] = 0x40
CPUInfo[1] = 0x40
CPUInfo[2] = 0x3
CPUInfo[3] = 0x21120

For InfoType 6
CPUInfo[0] = 0x75
CPUInfo[1] = 0x2
CPUInfo[2] = 0x9
CPUInfo[3] = 0x0
```



```
For InfoType 7
CPUInfo[0] = 0x0
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x0

For InfoType 8
CPUInfo[0] = 0x0
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x0

For InfoType 9
CPUInfo[0] = 0x0
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x0

For InfoType 10
CPUInfo[0] = 0x7300403
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x603

For InfoType 11
CPUInfo[0] = 0x1
CPUInfo[1] = 0x2
CPUInfo[2] = 0x100
CPUInfo[3] = 0x3

For InfoType 12
CPUInfo[0] = 0x0
CPUInfo[1] = 0x0
CPUInfo[2] = 0x0
CPUInfo[3] = 0x0

For InfoType 13
CPUInfo[0] = 0x7
CPUInfo[1] = 0x340
CPUInfo[2] = 0x340
CPUInfo[3] = 0x0
```



```
CPU String: GenuineIntel
Stepping ID = 7
Model = 10
Family = 6
Extended model = 2
CLFLUSH cache line size = 64
Logical Processor Count = 16
APIC Physical ID = 3

The following features are supported:
        SSE3
        MONITOR/MWAIT
        CPL Qualified Debug Store
        Virtual Machine Extensions
        Enhanced Intel SpeedStep Technology
        Thermal Monitor 2
        Supplemental Streaming SIMD Extensions 3
        L1 Context ID
        CMPXCHG16B Instruction
        xTPR Update Control
        Perf\Debug Capability MSR
        SSE4.1 Extensions
        SSE4.2 Extensions
        PPOPCNT Instruction
        x87 FPU On Chip
        Virtual-8086 Mode Enhancement
        Debugging Extensions
        Page Size Extensions
        Time Stamp Counter
        RDMSR and WRMSR Support
        Physical Address Extensions
        Machine Check Exception
        CMPXCHG8B Instruction
        APIC On Chip
        SYSENTER and SYSEXIT
        Memory Type Range Registers
        PTE Global Bit
        Machine Check Architecture
        Conditional Move/Compare Instruction
        Page Attribute Table
        36-bit Page Size Extension
        CFLUSH Extension
        Debug Store
```



```
CPU Brand String:        Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz
Cache Line Size = 64
L2 Associativity = 6
Cache Size = 256K
Physical Address = 36
Virtual Address = 48
Press any key to continue . . .
```

The above pictures illustrate that the function_id argument is 0, cpuInfo[0] returns the highest available non-extended function_id supported by the processor. The processor manufacturer is encoded in cpuInfo[1], cpuInfo[2], and cpuInfo[3]. I've given 13 infoType to demonstrate each supported function_id for my processor. Sometime if you encounter function_id value from 0x80000000, this is because your processor supports Extended Function CPUID information. Next you see a long list of set extension showing a possible result for my processor through **_cpuid and _cpuidex** intrinsics. Then the last picture shows the processor of my laptop; including cache line size, L2 Associativity , cache size, physical address , and virtual address.

Conclusion

This was a very important assignment to run the code using intrinsic format in order to explore the CPU information for my processor. The result of the program demonstrated a great length of information such as the feature it will support, Hex-address, and mostly CPU related information. Overall of the assignment was to teach us that each laptop is compatible with different features and speed of our CPU.