Министерство образования и науки Украины

Харковский национальный университет радиоэлектроники

Кафедра ЭВМ

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ 10

по предмету

«Архитектура компьютеров»

на тему:

«ИДЕНТИФИКАЦИЯ ПРОЦЕССОРОВ CЕМЕЙСТВА X86-64»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Приняли: |
| ст.гр. КИУКИ-18-4 |  | Голубничий Д.Ю. |
| Кравченко Н.С. |  | Дзюбенко В.Ф. |

Харьков 2020

1 Цель работы

Изучение способов идентификации микропроцессоров, совместимых с архитектурой семейства Intel x86-64; получение практических навыков по разработке низкоуровневых программ, определяющих тип установленного центрального процессора.

2 Постановка задачи

С целью получения практических навыков по разработке и выполнению программ, определяющих тип CPU, необходимо самостоятельно составить программу в соответствии с заданием:

1. Используя регистр флагов, необходимо убедиться в наличии 32-разрядного процессора в системе.
2. Убедиться в поддержке команды CPUID (посредством бита ID регистра EFLAGS) и определить максимальное значение параметра ее вызова.
3. Получить строку идентификации производителя процессора и сохранить ее в памяти.
4. Получить сигнатуру процессора и определить его модель, семейство и т.п. Выполнить анализ дополнительной информации о процессоре.
5. Получить флаги свойств. Составить список поддерживаемых процессором свойств.

3 Теоретическая часть

Одним из простых способов идентификации является контроль за изменением разрядов в регистре флагов (E)FLAGS. При этом используются следующие биты указанного регистра: ID, VIP и VIF, которые присутствуют в регистре флагов процессоров, начиная с процессора Pentium (P5); AC (i486); VM, RF (i386); NT, поле IOPL (i286).

Для идентификации 16-битных процессоров анализируют значение разрядов 12-15 регистра флагов после попыток его модификации.

Для обращения к регистрам FLAGS и EFLAGS используются команды PUSHF/POPF и PUSHFD/POPFD, соответственно.

BIOS может получить информацию о типе 32-разрядного процессора в виде сигнатуры идентификации, которая формируется в регистре EDX непосредственно после сигнала сброса процессора. Следует обратить внимание на некоторые отличия сигнатуры современных микропроцессоров от i386.

Поле "Extended family" (биты 20-27) используется совместно с "Family code" (биты 8-11) для обозначения принадлежности процессора семействам i386, i486, Pentium, Pentium Pro или Pentium 4. Семейство P6 включает все процессоры, основанные на микроархитектуре Pentium Pro, имеет значение поля "Extended family" равное 00h и значение поля "Family code" равное 06h. Семейство Pentium 4 включает все процессоры, основанные на микроархитектуре Intel NetBurst, имеет значение поля "Extended family" равное 00h, значение поля "Family code" равное 0Fh.

Поле "Extended model" (биты 16-19) используется совместно с полем "Model number" (биты 4-7) для обозначения модели процессора в рамках семейства. Поле "Stepping ID" (биты 0-3) обозначает модификацию (ревизию) конкретной модели. Поле "Type" (биты 12 и 13) может обозначать, что процессор является оригинальным OEM-процессором (00), OverDrive-процессором (01) или процессором, который может быть использован в двухпроцессорных системах (10).

Возможность получения сигнатуры процессора в любое время без необходимости сброса присутствует в процессорах, поддерживающих команду CPUID (см. 3.2.3). Такая поддержка была реализована, начиная с поздних версий процессоров i486.

Команда CPUID предоставляет программному обеспечению подробную информацию о процессоре, на котором она выполнена. Ее функция определяется содержимым регистра EAX в момент вызова. Все множество поддерживаемых функций разделено на два подмножества: функции первого и второго подмножеств возвращают, соответственно, базовую и расширенную информацию о процессоре.

4 Экспериментально-практическая часть

1. Определяю процессор 8086/8088 по регистру флагов, используя ассемблерную вставку:

bool is\_32x;

\_asm {

mov is\_32x, 0 ; сброс результата

pushf ; Сохранение исходное состояние регистра флагов

pushf

pop ax

xor ah, 11110000b ; Замена значений старших 4 битов

push ax

popf

pushf

pop bx

popf ; Восстановление исходное состояние регистра флагов

xor ah, bh ; AH = 0 → 808x - 80286, иначе 80386 +

mov is\_32x, ah ; сохранение результата

}

return is\_32x;

Если регистр AH = 0 (биты в регистре флагов не поменялись) - 808x-80286, иначе 80386 +.

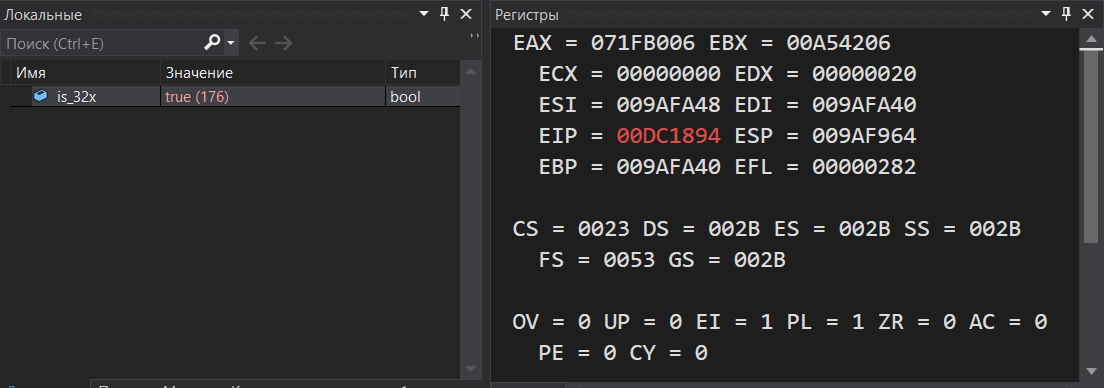


Рисунок 1 – Результат определения процессора по регистру флагов

1. Убедился в поддержке команды CPUID (посредством бита ID регистра EFLAGS)

bool is\_cpuid\_command\_supported;

\_asm {

mov is\_cpuid\_command\_supported, 0 ; сбрасываем результат

pushfd ; Скопировать состояние регистра флагов

pushfd ; Скопировать регистр флагов...

pop eax

or eax, 200000h ; установить 21 - й бит(флаг ID) в 1

push eax ; Скопировать регистр EAX в регистр флагов

popfd

pushf ; Скопировать регистр флагов в регистр EBX

pop bx

popfd ; Восстановить исходное состояние регистра флагов

xor eax, ebx ; EAH = 0 → CPUID не подерживается, иначе - поддерживается

jz exit ; если флаг ID = 0 → CPUID не поддерживается, можно выходить

mov is\_cpuid\_command\_supported, 1 ; иначе - поддержка присутствует

exit : }

Если EAH = 0 (биты в регистре флагов не поменялись) - CPUID не подерживается, иначе – поддерживается.

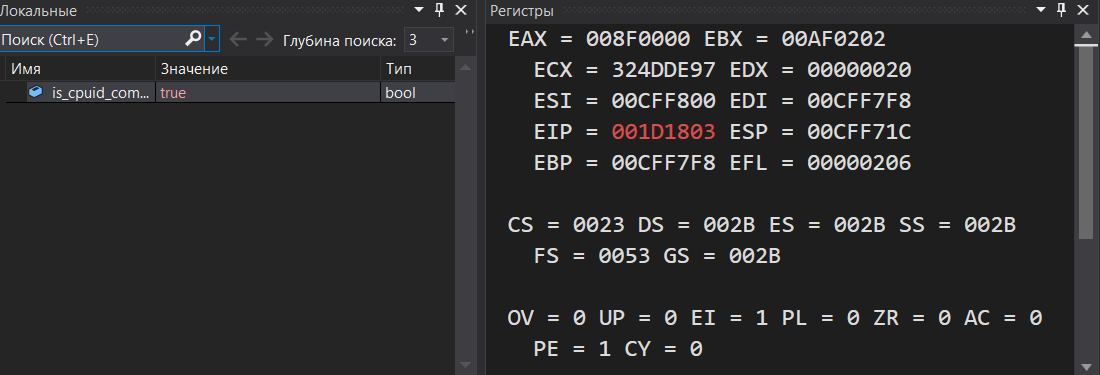
****

Рисунок 2 – Результат определения поддержки команды CPUID

Определил максимальное значение параметра вызова команды CPUID с помощью следующей ассемблерной вставки:

int max\_cpuid\_value;

\_asm {

mov EAX, 00h ; установка входного параметра

cpuid ; получение выходных параметров

mov max\_cpuid\_value, eax ; сохранение максимального входного параметра

}

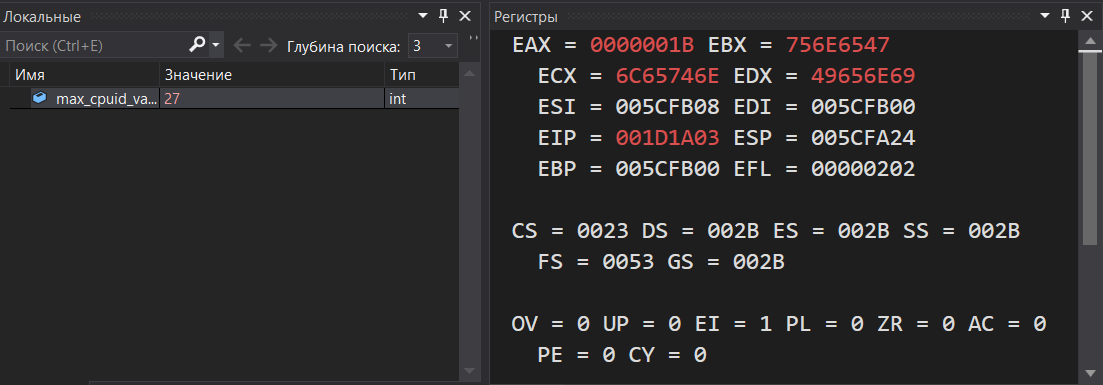
****

Рисунок 3 – Результат определения максимального значения параметра вызова команды CPUID

1. Получаю строку идентификации производителя процессора и сохраняю ее в памяти используя следующую вставку:

\_asm {

mov EAX, 00h ; установка входного параметра

cpuid ; получение выходных параметров

mov edi, \_pVendorString

mov[edi], EBX ; копируем первые 4 символа

mov[edi + 4], EDX ; копируем следующие 4 символа

mov[edi + 8], ECX ; копируем последние 4 символа

}

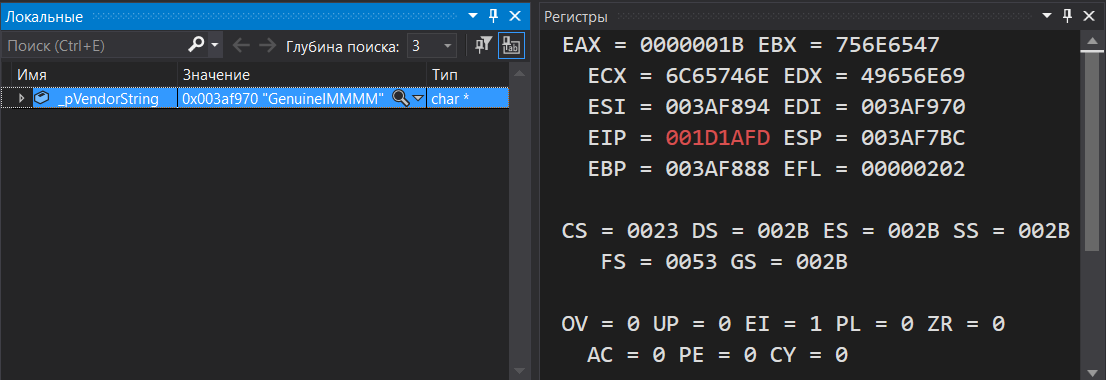
****

Рисунок 4 – Результат определения производителя процессора

При EAX=0 CPUID возвращает в регистрах EBX, ECX и EDX идентификатор производителя процессора (“Vendor ID”) в виде 12 символов ASCII.

|  |  |  |
| --- | --- | --- |
| ASCII-строка | HEX-значение EBX:EDX:ECX | Производитель |
| GenuineIntel | 759E6547:49656E69:6C65746E | Intel |

Таблица 1 - Идентификация производителя процессора

1. Получаю сигнатуру процессора и определяю его модель и семейство.

\_asm

{

mov eax, 01h

push eax

cpuid

}

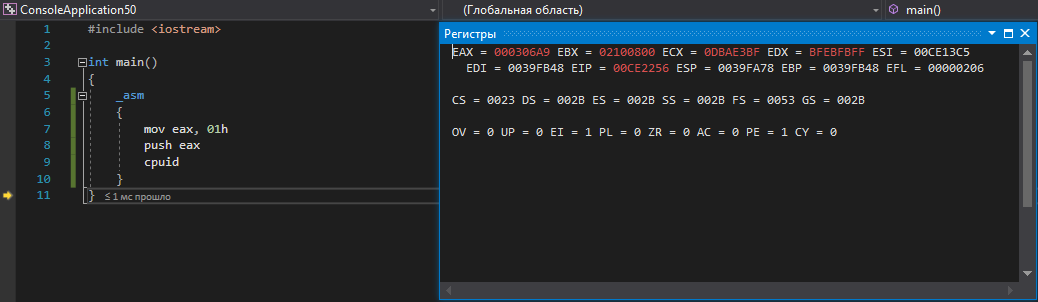


Рисунок 5 – Результат определения сигнатуры процессора

Также была получена дополнительная информация из регистра EBX =  02100800.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Значения | Биты | Описание |
| APIC ID | **0010** | 24-31 | 8-битный идентификатор, назначенный локальному модулю APIC процессора при включении питания |
| Count | **1010** | 16-23 | максимальное количество логических процессоров в одном модуле; для процессоров, поддерживающих технологию Multithreading |
| Chunks | **1000** | 8-15 | значение размера строки кэша (в 8-байтных блоках) |
| Brand ID | **0000** | 0-7 | имеет нулевое значение поля указывает на отсутствие поддержки таблицы брендов |

Таблица 2 – Содержимое регистра EBX после выполнения команды CPUID

Сигнатура процессора 00706E5, полученная из EAX соответсвует следующей модели:

000306A816 = 00000000 00000011 00000110 101010012

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Extended Model | Type | Family Code | Model No | Stepping ID | Description |
| 0 | 00 | 0011 | 6A | 9 | Intel(R) Core(TM) i3-8350K CPU @ 4.00GHz |

Таблица 3 – Сигнатура процессора

1. Получил флаги свойств.

\_asm {

mov EAX, 01h ; установка входного параметра

cpuid ; получение выходных параметров

mov esi, \_flags1

mov edi, \_flags2

mov[esi], EDX ; сохранение флагов свойств из EDX

mov[edi], ECX ; сохранение флагов свойств из ECX}

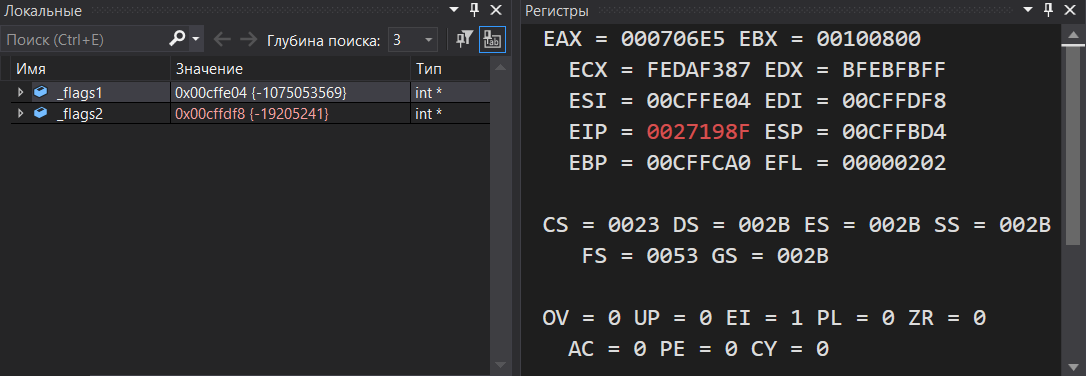


Рисунок 4 – Результат определения флагов свойств процессора

Флаги свойств процессора, полученные из регистров EDX и ECX имеют следующие значения:

EDX: BFEBFBFF 16 = 10111111111010111111101111111111 2

ECX: FEDAF387 16 = 11111110110110101111001110000111 2

В соответствии с таблицей флагов свойств, рассматриваемый процессор поддерживает следующие функции:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Флаги EDX | | | Флаги EСX | | |
| Имя | Описание | Поддержка | Имя | Описание | Поддержка |
|  | FPU | Floating-point Unit On-Chip | Да | SSE3 | Streaming SIMD Extensions 3 | Да |
|  | VME | Virtual Mode Extension | Нет | PCLMULDQ | PCLMULDQ Instruction | Да |
|  | DE | Debugging Extension | Да | DTES64 | 64-Bit Debug Store | Да |
|  | PSE | Page Size Extension | Да | MONITOR | MONITOR/MWAIT | Да |
|  | TSC | Time Stamp Counter | Да | DS-CPL | CPL Qualified Debug Store | Да |
|  | MSR | Model Specific Registers | Да | VMX | Virtual Machine Extensions | Да |
|  | PAE | Physical Address Extension | Да | SMX | Safer Mode Extensions | Да |
|  | MCE | Machine-Check Exception | Да | EIST | Enhanced Intel SpeedStep Technology | Нет |
|  | CX8 | CMPXCHG8 Instruction | Да | TM2 | Thermal Monitor 2 | Да |
|  | APIC | On-chip APIC Hardware | Да | SSSE3 | Supplemental Streaming SIMD Extensions 3 | Да |
|  |  | Reserved | -- | CNXT-ID | L1 Context ID | Нет |
|  | SEP | Fast System Call | Нет |  | Reserved | -- |
|  | MTRR | Memory Type Range Registers | Да | FMA | Fused Multiply Add | Да |
|  | PGE | Page Global Enable | Нет | CX16 | CMPXCHG16B | Нет |
|  | MCA | Machine-Check Architecture | Да | xTPR | xTPR Update Control | Да |
|  | CMOV | Conditional Move Instruction | Да | PDCM | Perfmon and Debug Capability | Нет |
|  | PAT | Page Attribute Table | Да |  | Reserved | Да |
|  | PSE-36 | 36-bit Page Size Extension | Да | PCID | Process Context Identifiers | Да |
|  | PSN | Processor serial number  is present and enabled | Да | DCA | Direct Cache Access | Да |
|  | CLFSH | CLFLUSH Instruction | Да | SSE4.1 | Streaming SIMD Extensions 4.1 | Да |
|  |  | Reserved | -- | SSE4.2 | Streaming SIMD Extensions 4.2 | Нет |
|  | DS | Debug Store | Нет | x2APIC | Extended xAPIC Support | Да |
|  | ACPI | Thermal Monitor and Software  Controlled Clock Facilities | Да | MOVBE | MOVBE Instruction | Да |
|  | MMX | MMX technology | Да | POPCNT | POPCNT Instruction | Да |
|  | FXSR | FXSAVE and FXSTOR Instructions | Да | TSC- DEADLINE | Time Stamp Counter Deadline | Нет |
|  | SSE | Streaming SIMD Extensions | Да | AES | AES Instruction Extensions | Нет |
|  | SSE2 | Streaming SIMD Extensions 2 | Да | XSAVE | XSAVE/XSTOR States | Нет |
|  | SS | Self-Snoop | Да | OSXSAVE | OS-Enabled Extended State  Management | Нет |
|  | HTT | Multi-Threading | Да | AVX | Advanced Vector Extensions | Да |
|  | TM | Thermal Monitor | Да | F16C | 16-bit floating-point conversion instructions | Да |
|  |  | Reserved | -- | RDRAND | RDRAND instruction supported | Да |
|  | PBE | Pending Break Enable | Да |  | Not Used | -- |

5 Анализ результатов и выводы

В данной лабораторной работе изучены способы идентификации микропроцессоров, совместимых с архитектурой семейства Intel x86-64; получены практические навыки по разработке низкоуровневых программ, определяющих тип установленного центрального процессора.

Во время выполнения работы были выполнены 4 задания, целью которых было убедиться в поддержке команды CPUID и определить максимальное значение параметра ее вызова, получить строку идентификации производителя процессора и сигнатуру процессора.

Также был выполнен анализ дополнительной информации о процессоре и получены флаги свойств, благодоря которым удалось составить список поддерживаемых процессором свойств.