Московский физико-технический институт (государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Кафедра информатики и вычислительной техники

Поддержка исполнения расширения системы команд AES-NI в бинарном компиляторе x86->Эльбрус

Студент: Анохин К. А. 713 гр, ФРТК
Научный руководитель: к. ф.-м. н. Нейман-заде Мурад Искендер оглы
Консультант: к. ф.-м. н. Рожин А. Ф.

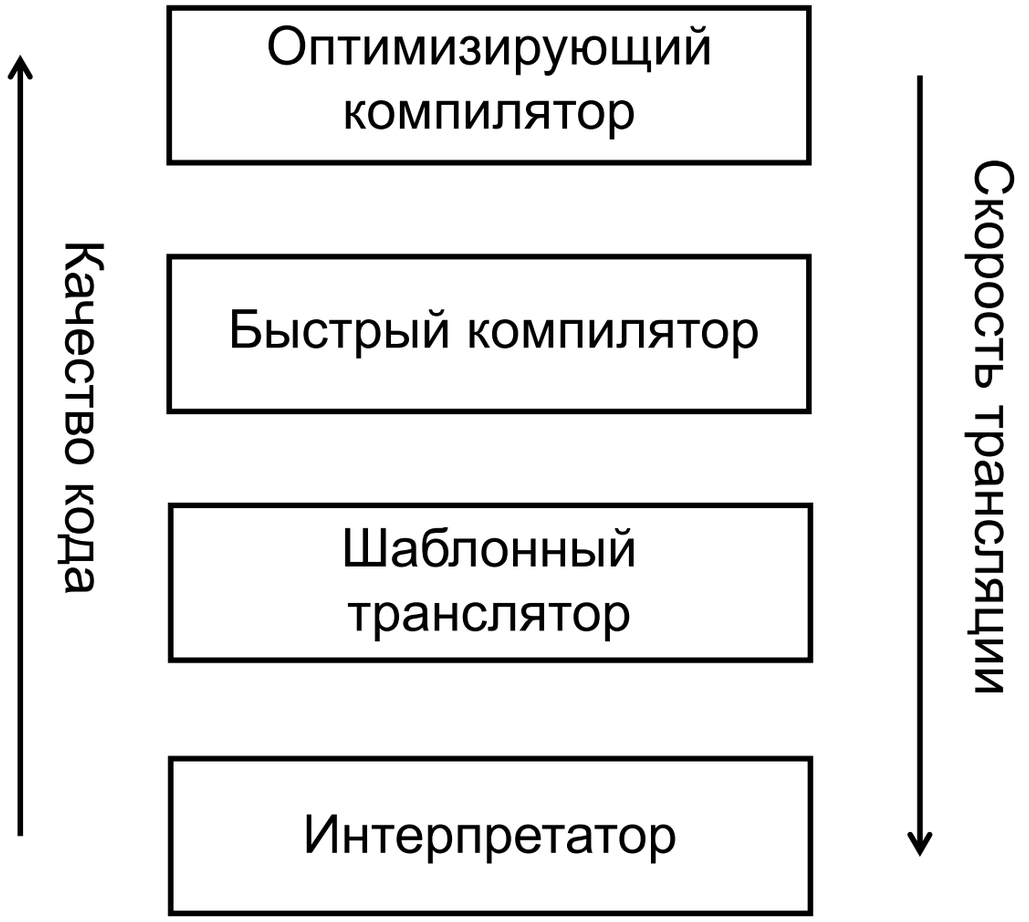
Москва 2021

Бинарный компилятор

Динамическая двоичная трансляция – это технология преобразования машинных кодов из одной системы команд в другую без использования исходных текстов программы и в реальном времени.

- Оптимизирующий компилятор – применяет полный набор оптимизаций, учитывающий все возможности аппаратуры
- Быстрый компилятор – применяет базовый набор оптимизаций
- Шаблонный транслятор – обрабатывает линейные участки, заменяя x86 инструкции на фиксированные наборы команд Эльбрус
- Интерпретатор – исполняет x86 инструкции пошагово

Переключение между уровнями происходит динамически при превышении определенного порога числа повторений кода



Алгоритм AES

AES (**A**dvanced **E**ncryption **S**tandard) – симметричный алгоритм блочного шифрования, принятый в качестве стандарта шифрования

Преимущества алгоритма:

- Устойчивость к атакам
- Высокая скорость шифрования
- Эффективное использование процессорных ресурсов за счет внутреннего параллелизма, присущего алгоритму

Расширение системы команд AES-NI

Расширение системы команд AES-NI для процессоров x86 – аппаратная реализация алгоритма AES для дополнительного повышения производительности приложений, работающих по данному алгоритму (OpenVPN, OpenSSH, The Bat)

AESENC (AES Encrypt Round) – выполнить один раунд шифрования
AESDEC (AES Decrypt Round) – выполнить один раунд дешифрования
AESENCLAST (AES Encrypt Last Round) – выполнить последний раунд шифрования
AESDECLAST (AES Decrypt Last Round) – выполнить последний раунд дешифрования
AESIMC (AES Inverse Mix Columns) – выполнить преобразование, обратное по отношению к операции перемешивания столбцов
AESKEYGENASSIST (AES Key Generation Assist) – способствовать генерации раундового ключа AES

Цель работы

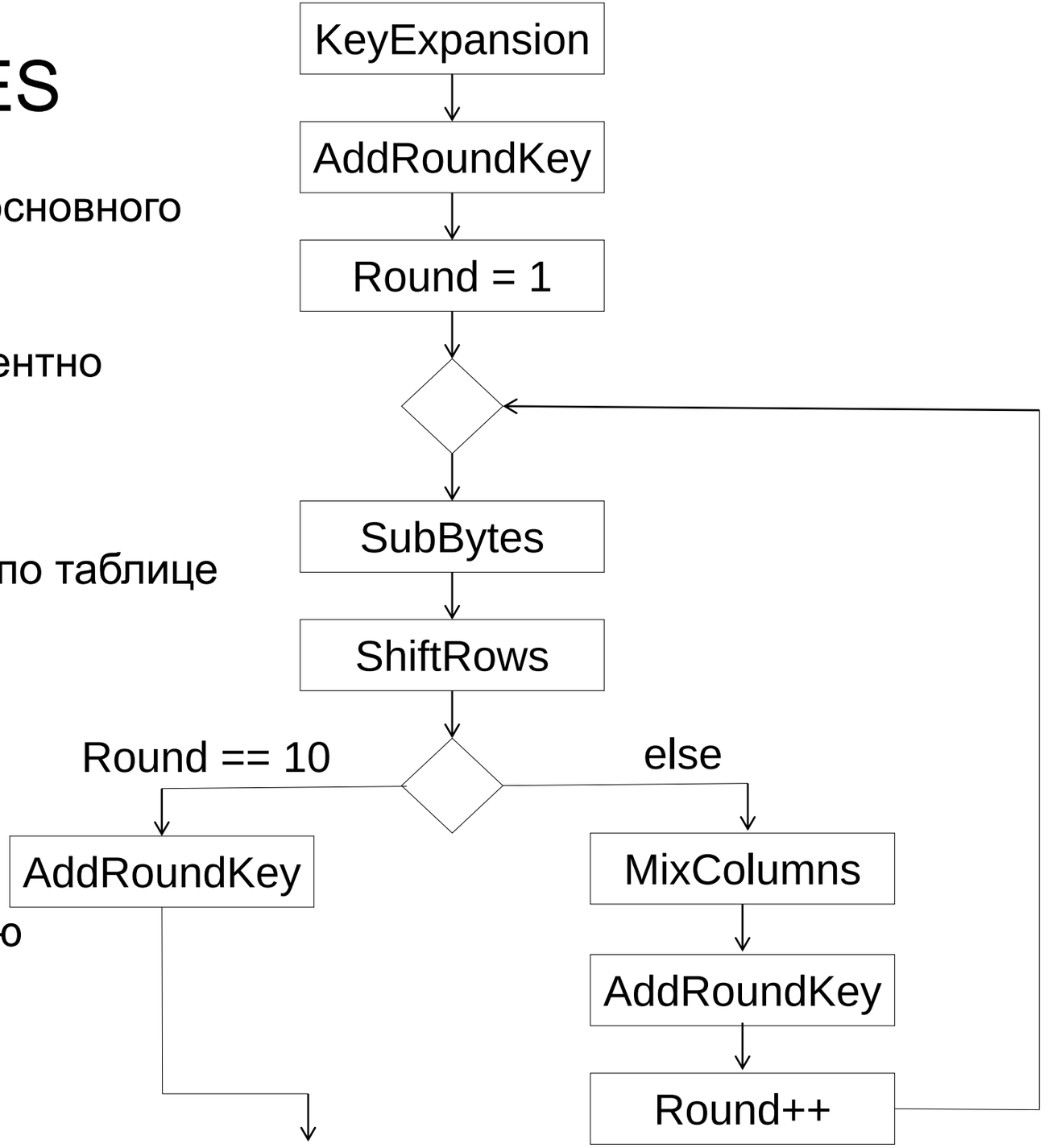
Реализовать поддержку расширения набора команд AES-NI в бинарном компиляторе x86->Эльбрус

Задачи:

- Реализовать на C++ логику инструкций AES-NI
- Реализовать поддержку инструкций на уровне интерпретатора
- Реализовать поддержку инструкций на уровне шаблонного транслятора
- Включить поддержку инструкций AES-NI в модели эмулируемого бинарным транслятором x86 процессора
- Протестировать корректность работы реализованных инструкций

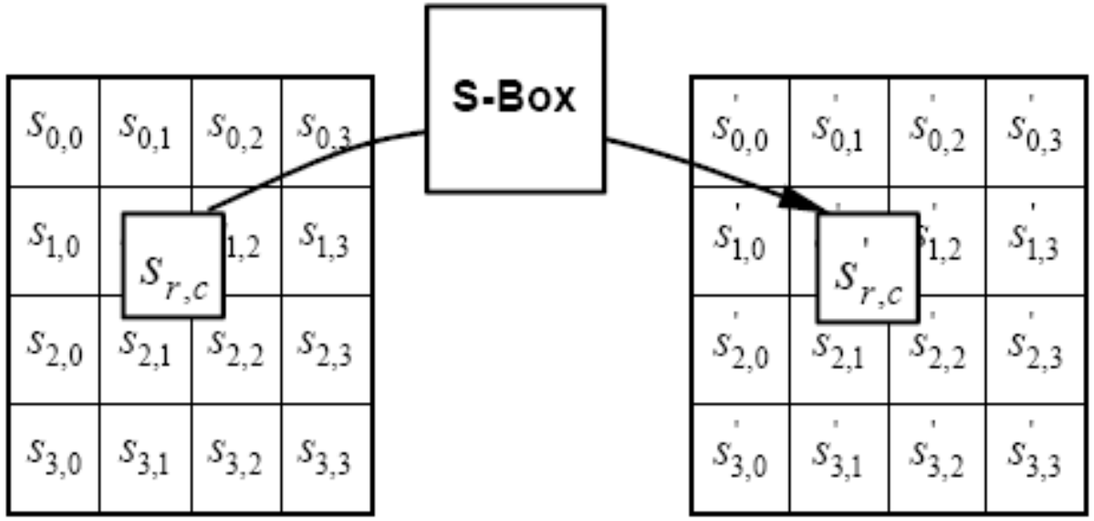
Описание алгоритма AES

- KeyExpansion – процедура расширения основного ключа для создания раундовых ключей
- AddRoundKey – раундовый ключ поэлементно добавляется к матрице state с помощью поразрядного XOR
- SubBytes - замена байтов матрицы state по таблице замен S-box
- ShiftRows – циклический сдвиг строк матрицы state
- MixColumns – умножение каждого столбца матрицы state на фиксированную матрицу

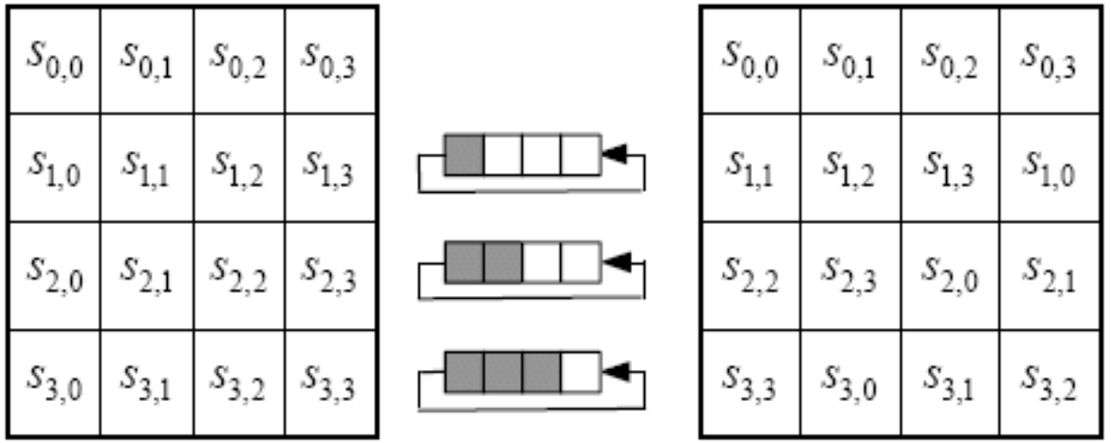


Преобразования алгоритма

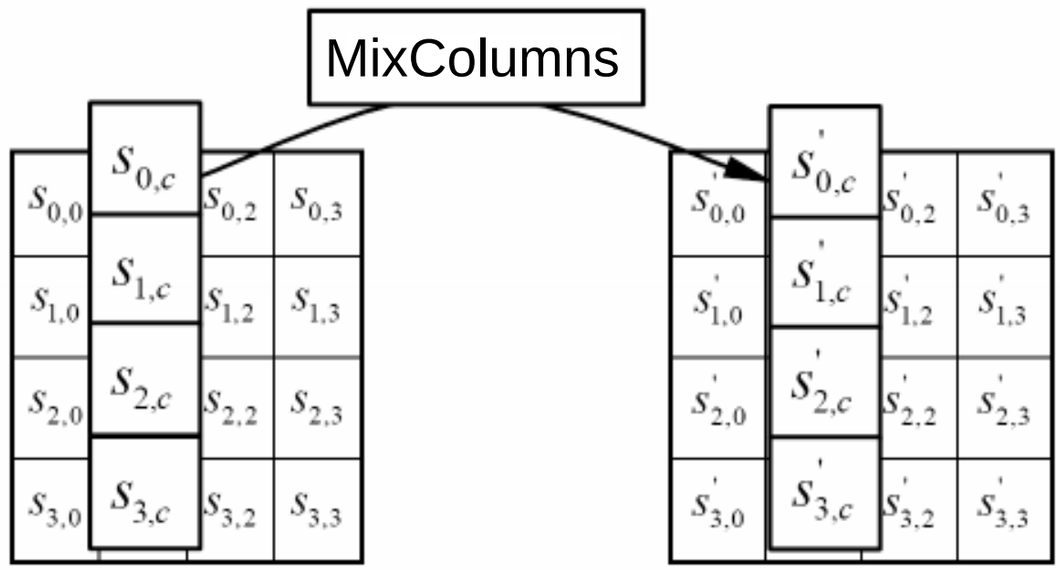
SubBytes



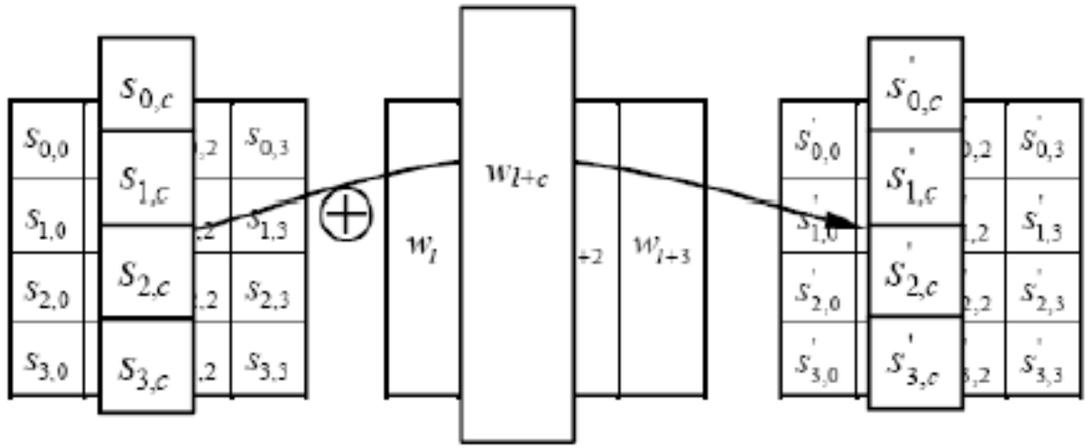
ShiftRows



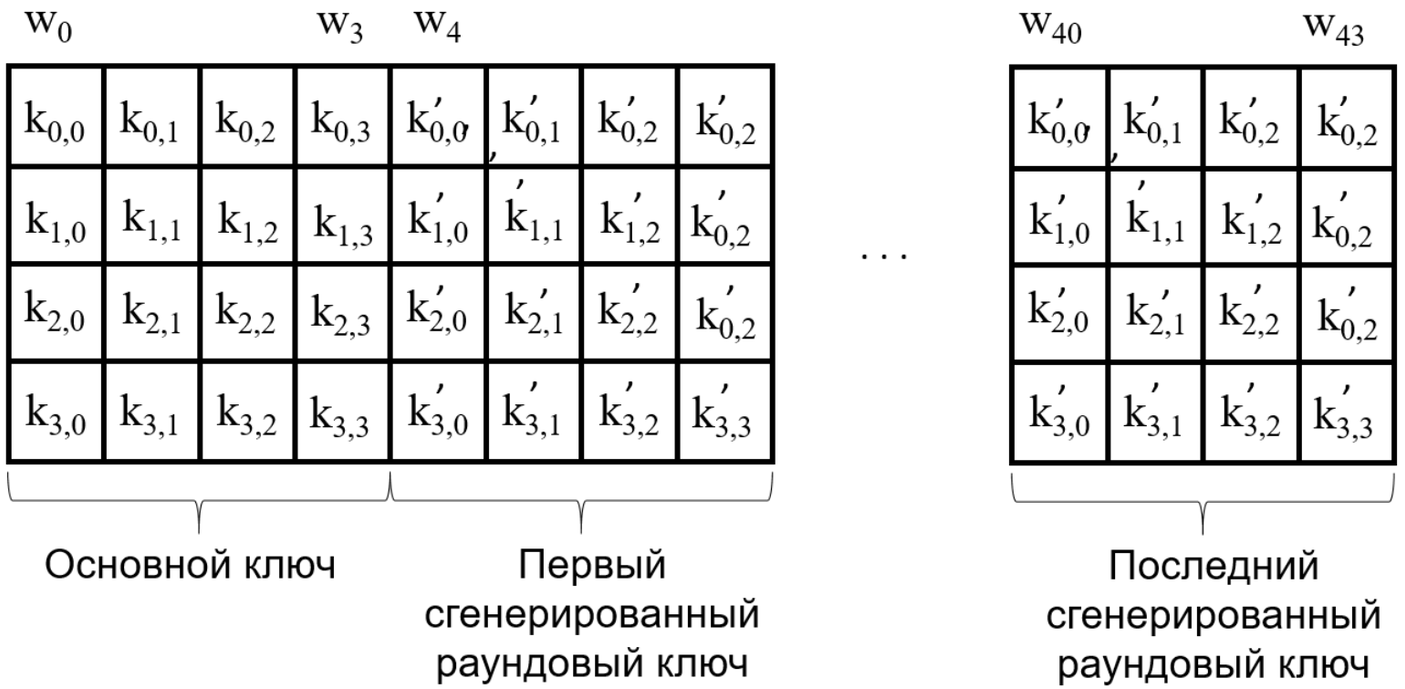
MixColumns



AddRoundKey



Процедура расширения основного ключа KeyExpansion



Если i больше 3 и кратно 4, то:

$w_i = SubBytes(RotWord(w_{i-1})) \text{ xor } Rcon(i/4)$

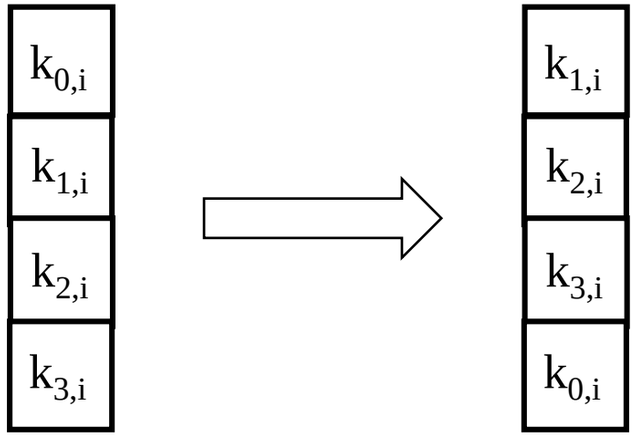
Если i больше 3 и не кратно 4, то:

$w_i = w_{i-4} \text{ xor } w_{i-1}$

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Массив констант Rcon

Операция RotWord:



Программная реализация набора инструкций AES-NI

Было реализовано 6 функций, реализующих семантику инструкций AES-NI

Произвести один раунд шифрования
aesenc (state, key) {
    SubBytes
    ShiftRows
    MixColumns
    AddRoundKey
}

Произвести последний раунд шифрования
aesenclast (state, key) {
    SubBytes
    ShiftRows
    AddRoundKey
}

Произвести обратный MixColumns
aesimc (state) {
    InverseMixColumns
}

Произвести один раунд дешифрования
aesdec (state, key) {
    AddRoundKey
    InverseMixColumns
    InverseShiftRows
    InverseSubBytes
}

Произвести последний раунд дешифрования
aesdeclast (state, key) {
    AddRoundKey
    InverseShiftRows
    InverseSubBytes
}

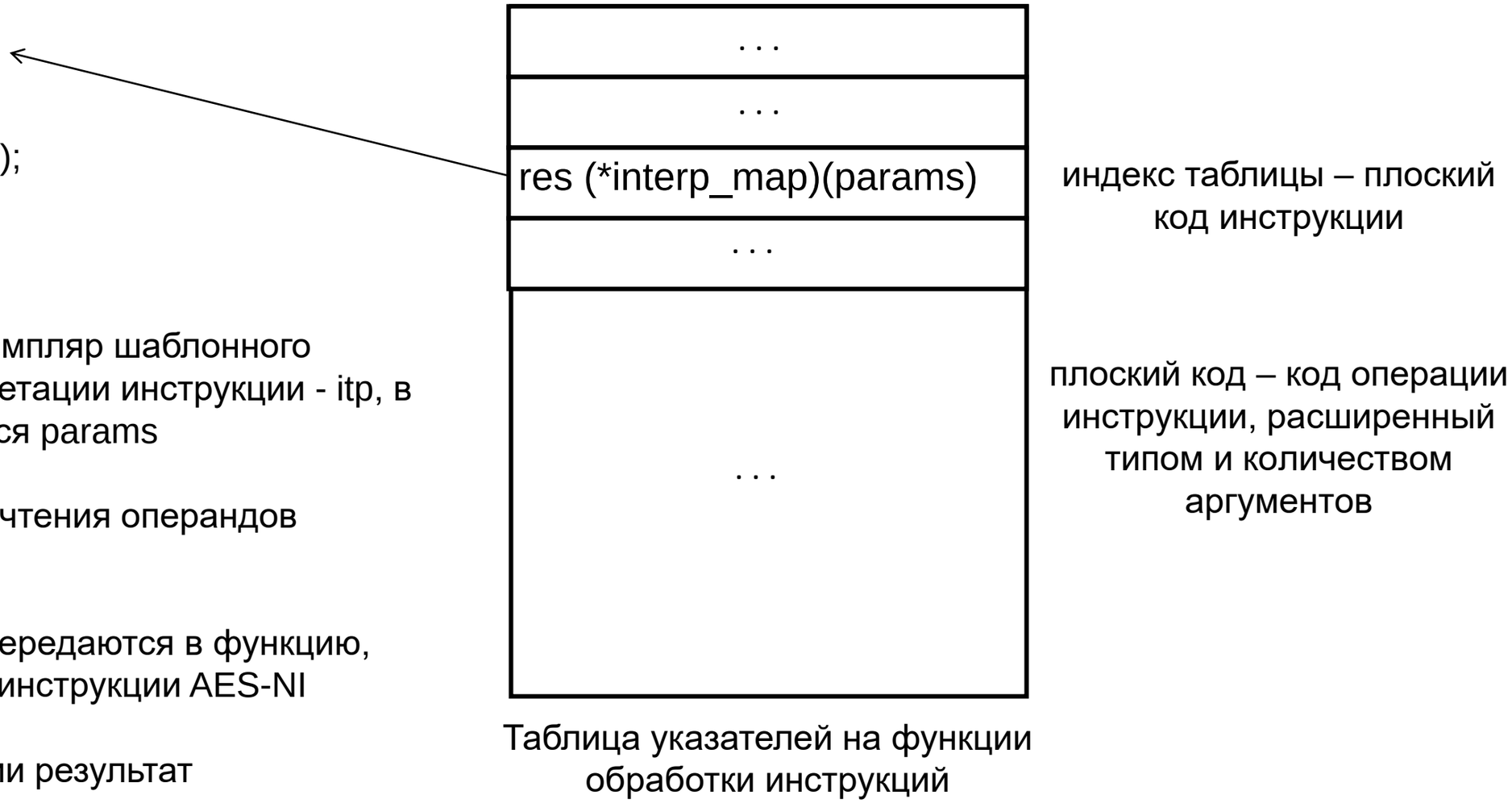
Сгенерировать раундовый ключ
aeskeygenassist (key) {
    KeyExpansion
}

Реализация поддержки инструкций в интерпретаторе

Было создано 6 функций-мапперов для каждой инструкции AES-NI

```
res interp_map (params)
{
    itp(params);
    itp.Read(state, key);
    result = faes(state, key);
    itp.Write(result);
    return itp.End();
}
```

- В функции создается экземпляр шаблонного класса контекста интерпретации инструкции - itp, в его конструктор передается params
- У itp вызываются методы чтения операндов инструкции
- Прочитанные операнды передаются в функцию, реализующую семантику инструкции AES-NI
- Возвращаемый из функции результат записывается в регистр назначения инструкции



Реализация поддержки инструкций в шаблонном трансляторе

Было создано 6 функций кодогенерации для каждой инструкции AES-NI

```
res tmpl_GenAes (args) {
    mapperHelper_t <tmpl_MCallWord_t> ctx (args);
    rgn_Mcall mcall = ctx.iword.getValue ();
    ctx.dispatch (E2K_REG, mcall);
    Q_reg src1 = ctx.readOpndV();
    Q_reg src2 = ctx.readOpndW();
    tcg_Gen_t < TCG_CH* >  g (ctx.offs);
                    …
    /* перемещение операндов в регистры архитектуры Эльбрус,
        вызов функции, запись результата */
                    …
     return ctx.mapperRet();
}
```

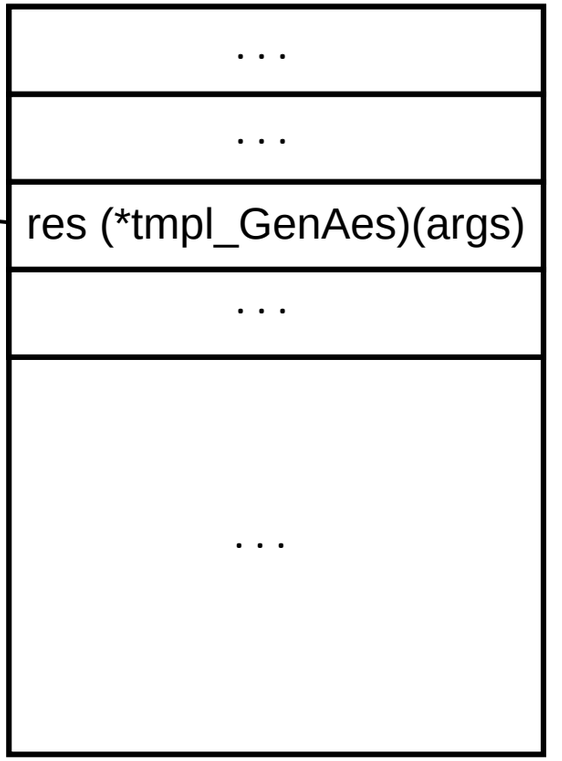


Таблица указателей на функции кодогенерации, таблица индексируется плоским кодом инструкции

- tmpl_MCallWord_t содержит информацию о соответствии между плоским кодом инструкции и функцией, реализующей семантику данной инструкции
- mcall – адрес вызываемой функции
- в регистр E2K_REG кладется адрес mcall
- tcg_Gen_t – класс-генератор команды архитектуры Эльбрус,
- TCG_CH* - флаги ALU-каналов

Включение поддержки инструкций AES-NI в модели эмулируемого бинарным транслятором x86 процессора

Проблема: если приложение предполагает использовать расширение системы команд AES-NI, и оно поддерживается на исходной платформе, то в бинарном компиляторе должна быть включена поддержка исполнения данных инструкций, в остальных случаях поддержка исполнения команд AES-NI в бинарном компиляторе не обязательна.

- Эмулируемый бинарным компилятором x86 процессор однозначно определяет поведение инструкции CPUID.
- Данная инструкция при значении регистра EAX = 0x00000001 выставляет в единицу 25 бит регистра ECX, если расширение AES-NI поддерживается на исходном процессоре, иначе выставляет 25 бит в ноль
- В бинарном компиляторе добавлена возможность опционального включения поддержки инструкций AES-NI в процессе эмуляции им x86 процессора.

Тестирование

В процессе тестирования был создан набор направленных тестов, проверяющих корректность встраивания инструкций AES-NI в бинарный компилятор

```
0x8048e3c <main>          movdqa 0x80e9930,%xmm1
0x8048e44 <main+8>        movdqa 0x80e9940,%xmm2
0x8048e4c <main+16>       aesdec %xmm2,%xmm1
0x8048e51 <main+21>       ret

Breakpoint 3, 0x08048e4c in main:
xmm1: v2_int64 = {0x63746f725d53475d, 0x7b5b546573745665}
xmm2: v2_int64 = {0x5b477565726f6e5d, 0x4869285368617929}

Breakpoint 4, 0x08048e51 in main:
xmm1: v2_int64 = {0xb58eb95eb730392a, 0x138ac342faea2787}
```

Результат запуска инструкции aesdec в отладчике gdb на процессоре архитектуры x86

```
_INTERNAL_12_symaddrs_cpp_mcst::faesdec(unsigned long long, unsigned long long, unsigned long long, unsigned long long)
        . . .
***** 0xa00a80.10190560 *****
HS    0 0c300012 :
ALS0  0 4ac000f0  PORd imm (0), %dbr0<%R14> (0xb58eb95eb730392a) -> %dg16<%G16> (0xb58eb95eb730392a)
ALS1  0 4ac001f1  PORd imm (0), %dbr1<%R15> (0x138ac342faea2787) -> %dg17<%G17> (0x138ac342faea2787)
***** 0xa00a90.10190561 *****
        . . .
```

Фрагмент трассы исполнения бинарного компилятора на симуляторе архитектуры Эльбрус: перенос результатов транслированной инструкции aesdec на регистры, хранящие x86-контекст

Результаты

- Реализована поддержка AES-инструкций на уровне интерпретатора
- Реализована поддержка AES-инструкций на уровне шаблонного транслятора
- Добавлена возможность опционального включения поддержки расширения AES-NI в бинарном компиляторе
- Проведено тестирование полученного решения