Роман Артемьев

Новосибирский центр информационных технологий
«УНИПРО»

Joker<?>, 14.10.2016

План доклада

Введение

FAQ по Эльбрусу

Предыстория

Шаблонный интерпретатор

Деоптимизация

JIT-компилятор

Производительность

Будущее Эльбруса

Введение

Этот доклад для вас, если:

Введение

Этот доклад для вас, если:

- Ваш продукт будет (уже) работать на Эльбрусе

Введение

Этот доклад для вас, если:

- Ваш продукт будет (уже) работать на Эльбрусе
- Вы хотите знать, что такое Эльбрус

Введение

Этот доклад для вас, если:

- Ваш продукт будет (уже) работать на Эльбрусе
- Вы хотите знать, что такое Эльбрус
- Вам интересны доклады о внутренностях JVM

Введение

Этот доклад для вас, если:

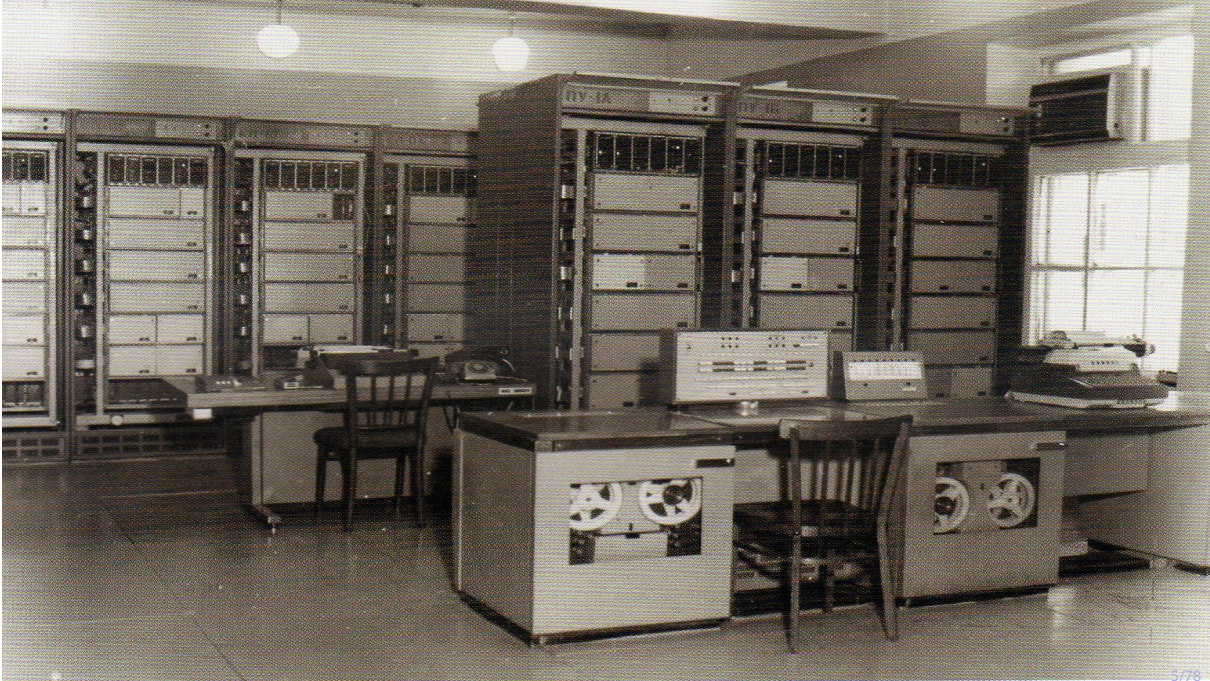
- Ваш продукт будет (уже) работать на Эльбрусе
- Вы хотите знать, что такое Эльбрус
- Вам интересны доклады о внутренностях JVM
- Вам интересно, что ”под капотом” у процессора

Введение

Этот доклад для вас, если:

- Ваш продукт будет (уже) работать на Эльбрусе
- Вы хотите знать, что такое Эльбрус
- Вам интересны доклады о внутренностях JVM
- Вам интересно, что ”под капотом” у процессора
- Вам интересно, как портируется JVM

FAQ по Эльбрусу


IIV-1A
IIV-1B


Joker<?>
Главная Java-конференция в России
Joker<?>
Student Edition
МЦСТ
МЦСТ
ЭЛЬБРУС

FAQ: Зачем?!

FAQ: Зачем?!

Развитие российской электроники

FAQ: Зачем?!

Развитие российской электроники

Защита от кибер-угроз

FAQ: Зачем?!

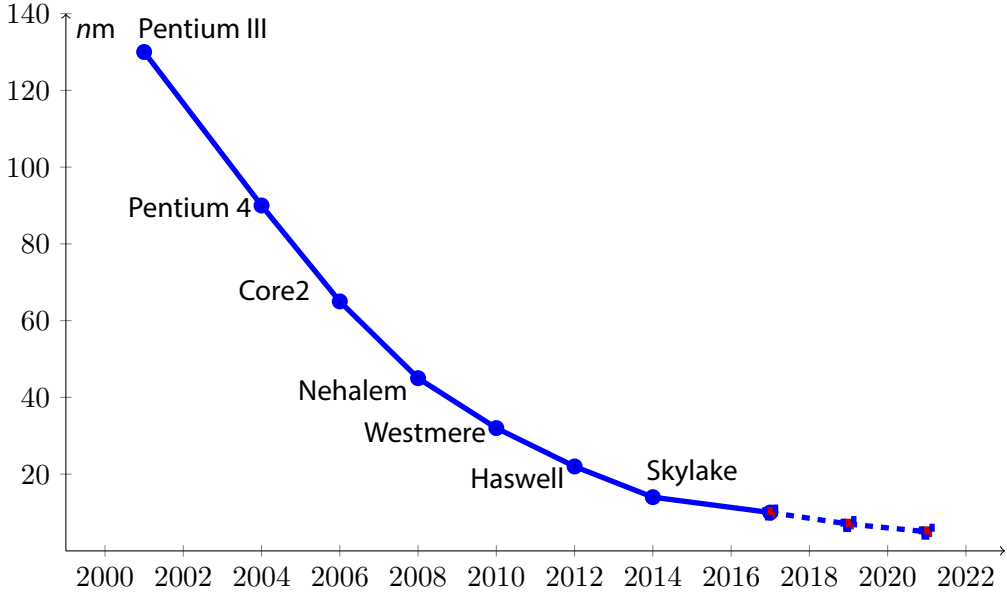
Развитие российской электроники

Защита от кибер-угроз

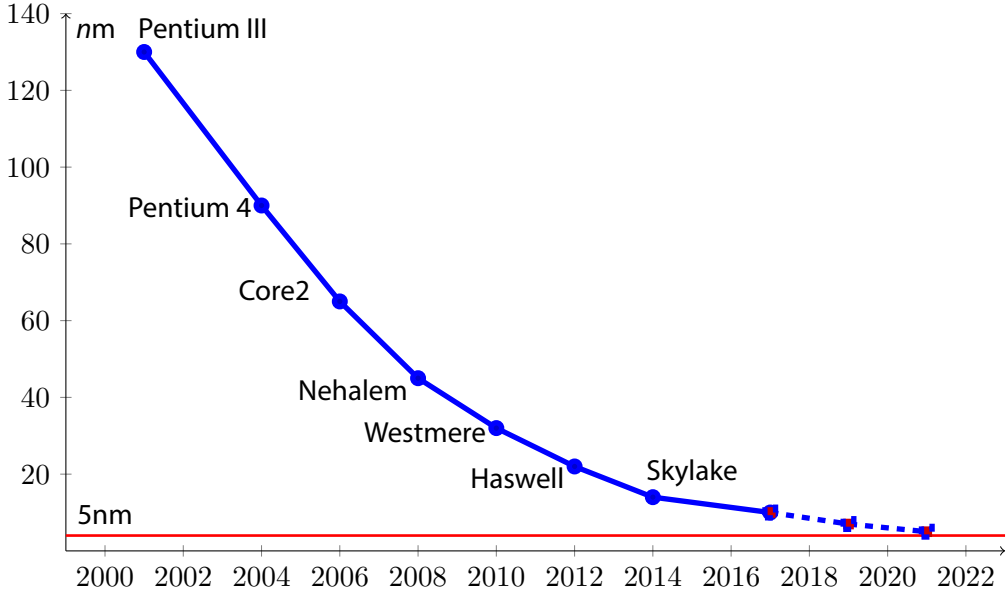
Создание российской альтернативы

FAQ: Закон Мура

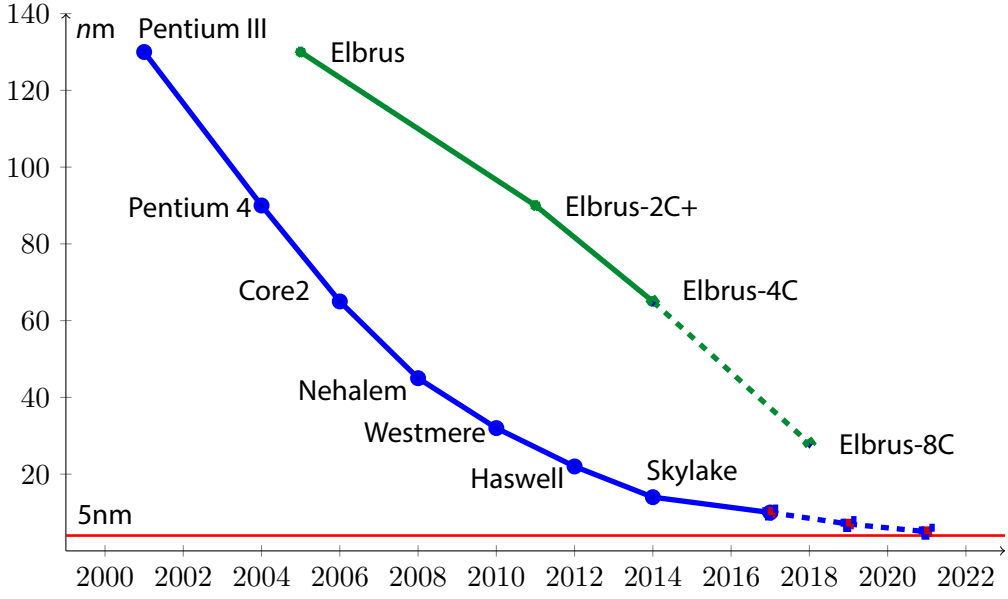
FAQ: Закон Мура



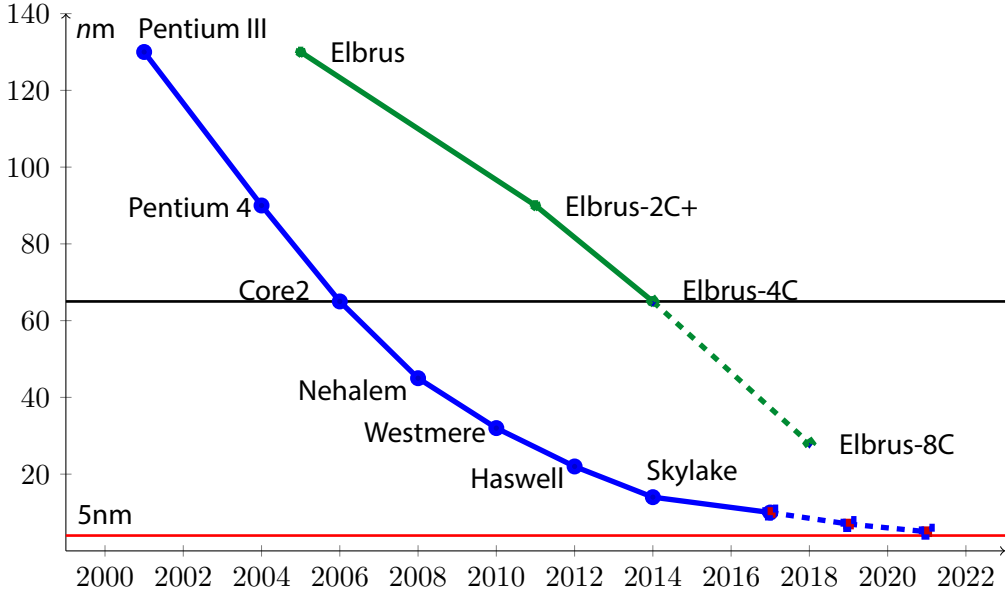
FAQ: Закон Мура



FAQ: Закон Мура



FAQ: Закон Мура



FAQ: Архитектура

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт
- Много регистров (192+32+32)

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт
- Много регистров (192+32+32)
- Явная спекулятивность

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт
- Много регистров (192+32+32)
- Явная спекулятивность
- Условное исполнение ”почти” любых инструкций

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт
- Много регистров (192+32+32)
- Явная спекулятивность
- Условное исполнение ”почти” любых инструкций
- 3 аппаратных стека (Безопасно)

FAQ: Архитектура

- VLIW (Very Long Instruction Word) до 23-х операций за такт
- Много регистров (192+32+32)
- Явная спекулятивность
- Условное исполнение ”почти” любых инструкций
- 3 аппаратных стека (Безопасно)
- Нет предсказателя переходов

FAQ: Ассемблер

FAQ: Ассемблер

```
#include <math.h>

struct Point {
  float x;
  float y;
};

float floatmath(Point* a, Point* b) {
  return ::sqrt((a->x * b->x * a->y * b->y));
}
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

```
setwd wsz = 0x4
return %ctpr3
ldw,0 %dr0, 0x0, %r0
ldw,2 %dr0, 0x4, %r1
ldw,3 %dr1, 0x0, %r2
ldw,5 %dr1, 0x4, %r3

fmuls,0 %r0, %r1, %r0
fmuls,3 %r2, %r3, %r1

fmuls,0 %r0, %r1, %r0

fsqrts,0 %r0, %r0
ct %ctpr3
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

```
setwd wsz = 0x4
return %ctpr3
ldw,0 %dr0, 0x0, %r0
ldw,2 %dr0, 0x4, %r1
ldw,3 %dr1, 0x0, %r2
ldw,5 %dr1, 0x4, %r3

fmuls,0 %r0, %r1, %r0
fmuls,3 %r2, %r3, %r1

fmuls,0 %r0, %r1, %r0

fsqrts,0 %r0, %r0
ct %ctpr3
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

```
setwd wsz = 0x4
return %ctpr3
ldw,0 %dr0, 0x0, %r0
ldw,2 %dr0, 0x4, %r1
ldw,3 %dr1, 0x0, %r2
ldw,5 %dr1, 0x4, %r3

fmuls,0 %r0, %r1, %r0
fmuls,3 %r2, %r3, %r1

fmuls,0 %r0, %r1, %r0

fsqrts,0 %r0, %r0
ct %ctpr3
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

```
setwd wsz = 0x4
return %ctpr3
ldw,0 %dr0, 0x0, %r0
ldw,2 %dr0, 0x4, %r1
ldw,3 %dr1, 0x0, %r2
ldw,5 %dr1, 0x4, %r3

fmuls,0 %r0, %r1, %r0
fmuls,3 %r2, %r3, %r1

fmuls,0 %r0, %r1, %r0

fsqrts,0 %r0, %r0
ct %ctpr3
```

FAQ: Ассемблер

```
return ::sqrt((a->x * b->x * a->y * b->y));
```

```
movss (%rdi), %xmm1
mulss 4(%rdi), %xmm1
mulss (%rsi), %xmm1
mulss 4(%rsi), %xmm1
sqrtss %xmm1, %xmm0
ucomiss %xmm0, %xmm0
jp .L8
ret
.L8:
movaps %xmm1, %xmm0
pushq %rax
call sqrtf
popq %rdx
ret
```

```
setwd wsz = 0x4
return %ctpr3
ldw,0 %dr0, 0x0, %r0
ldw,2 %dr0, 0x4, %r1
ldw,3 %dr1, 0x0, %r2
ldw,5 %dr1, 0x4, %r3

fmuls,0 %r0, %r1, %r0
fmuls,3 %r2, %r3, %r1

fmuls,0 %r0, %r1, %r0

fsqrts,0 %r0, %r0
ct %ctpr3
```

Компилятор - это ВСЕ!!!

Предыстория

Предыстория

12.2012 Начало проекта "Java для Эльбруса"

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

11.2015 Non-LLVM JIT, ”Java 2.0”

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

11.2015 Non-LLVM JIT, ”Java 2.0”

12.2015 Порт C2-компилятора, Java 8, ”Java 2.1”

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

11.2015 Non-LLVM JIT, ”Java 2.0”

12.2015 Порт C2-компилятора, Java 8, ”Java 2.1”

03.2016 Развитие JVM, JBreak, ”Java 3.0”

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

11.2015 Non-LLVM JIT, ”Java 2.0”

12.2015 Порт C2-компилятора, Java 8, ”Java 2.1”

03.2016 Развитие JVM, JBreak, ”Java 3.0”

06.2016 Сжатие кучи, интринсики, оптимизированный рантайм, ”Java 3.1”

Предыстория

12.2012 Начало проекта ”Java для Эльбруса”

11.2013 Первая версия ”Java 1.0” на основе Zero и Shark (LLVM)

11.2015 Non-LLVM JIT, ”Java 2.0”

12.2015 Порт C2-компилятора, Java 8, ”Java 2.1”

03.2016 Развитие JVM, JBreak, ”Java 3.0”

06.2016 Сжатие кучи, интринсики, оптимизированный рантайм, ”Java 3.1”

XX.2016 Шаблонный интерпретатор, G1, ”Java 3.2”

Тестовый стенд: CPU

Параметры

Тестовый стенд: CPU

Параметры

- 65нм

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня
- 64 бит

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня
- 64 бит

x86

Intel Quad Q9300 (2.50 GHz)
TDP: 95W

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня
- 64 бит

x86

Intel Quad Q9300 (2.50 GHz)
TDP: 95W

Elbrus

Elbrus 4C (800 MHz)
TDP: 45W

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня
- 64 бит

x86

Intel Quad Q9300 (2.50 GHz)
TDP: 95W

Elbrus

Elbrus 4C (800 MHz)
TDP: 45W

Отношение частоты: $2500/800 = 3.125$
Отношение TDP: $95/45 \approx 2.1$

Тестовый стенд: JRE

Тестовый стенд: JRE

x86

```
$ java -version
$ openjdk version "1.8.0_102"
$ OpenJDK Runtime Environment (build 1.8.0_102-8u102-b14)
$ OpenJDK 64-Bit Server VM (build 25.102-b14, mixed mode)
```

Тестовый стенд: JRE

x86

```
$ java -version
$ openjdk version "1.8.0_102"
$ OpenJDK Runtime Environment (build 1.8.0_102-8u102-b14)
$ OpenJDK 64-Bit Server VM (build 25.102-b14, mixed mode)
```

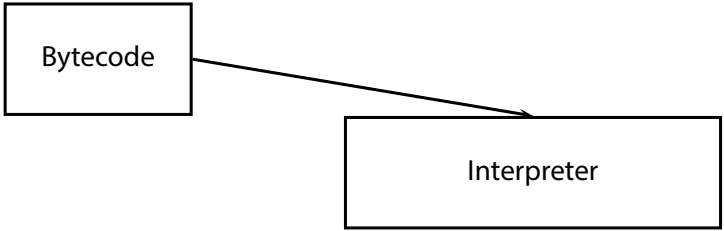
Elbrus

```
$ java -version
$ openjdk version "1.8.0_102"
$ OpenJDK Runtime Environment (build 1.8.0_102-8u102-b14)
$ OpenJDK 64-Bit Server VM (RVM 3.2) (25.102-b14, mixed mode)

$ uname -sr Linux 3.14.73
```

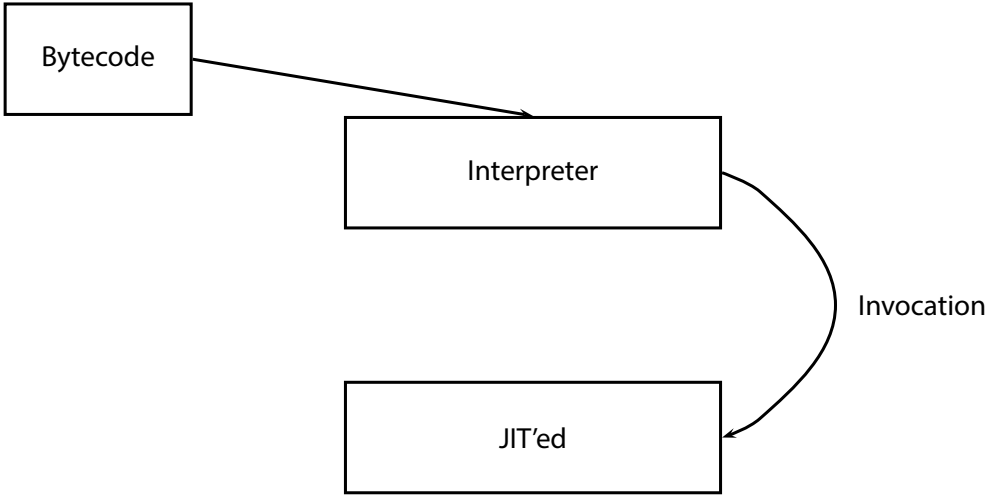
Цикл работы программы

Цикл работы программы

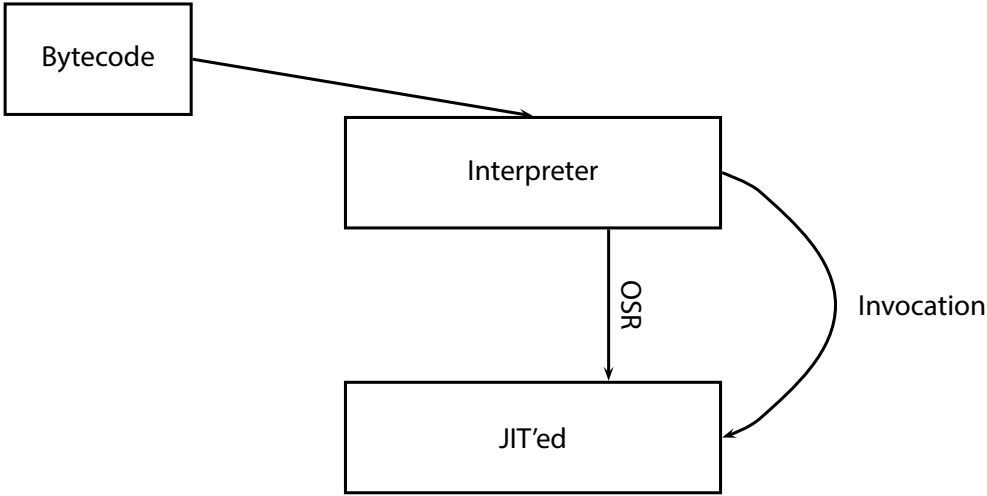
Цикл работы программы



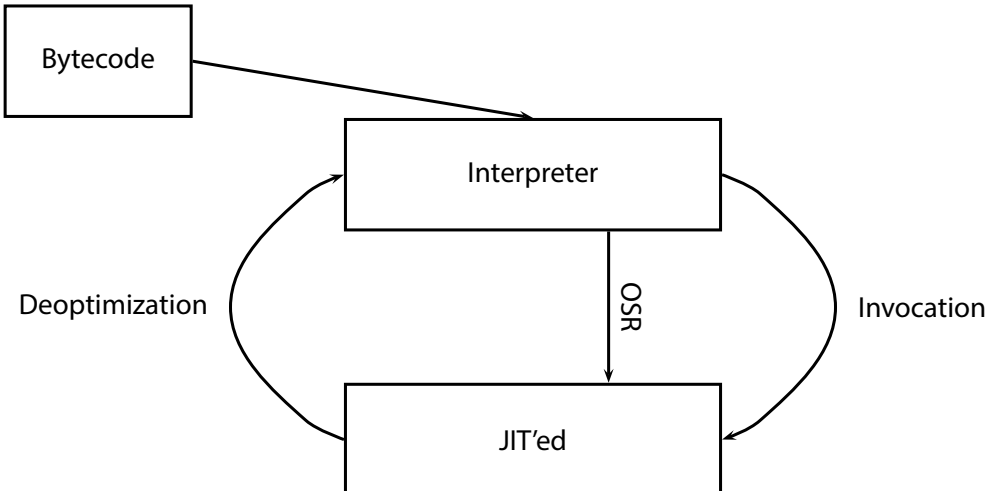
Цикл работы программы



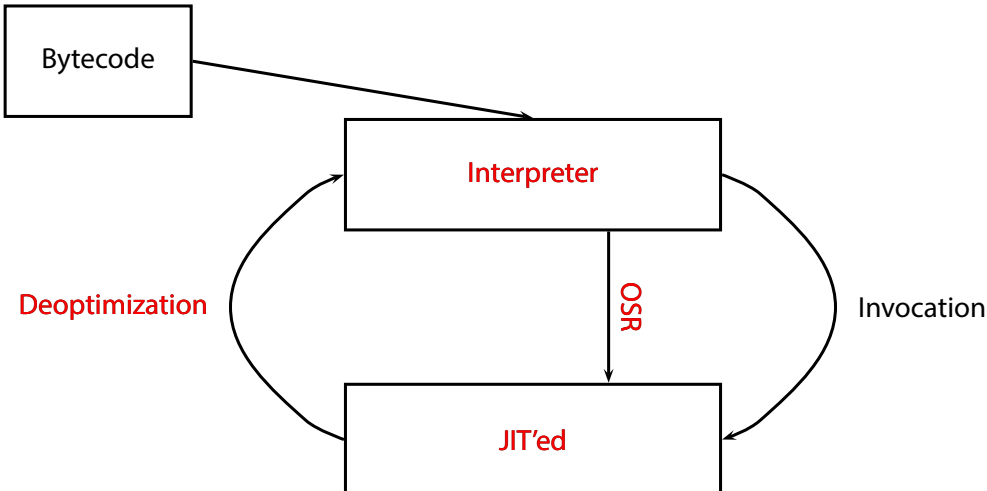
Цикл работы программы



Цикл работы программы



Цикл работы программы



Интерпретатор

TINT: Обзор

TINT: Обзор

CPP Интерпретатор

TINT: Обзор

CPP Интерпретатор

Написан на C++

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

Полностью написан на ассемблере

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

Полностью написан на ассемблере

Непортируем

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

Полностью написан на ассемблере

Непортируем

Генерируется при старте VM

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

Полностью написан на ассемблере

Непортируем

Генерируется при старте VM

Содержит оптимизации

TINT: Обзор

CPP Интерпретатор

Написан на C++

Работает на любой архитектуре

Единственный интерпретатор для Zero

Упразднен в Java 9 (кроме Zero)

Шаблонный Интерпретатор (TINT)

Полностью написан на ассемблере

Непортируем

Генерируется при старте VM

Содержит оптимизации

TINT быстрее CPP примерно в 2.2 раза для x86

TINT: Обзор

Оптимизации

TINT: Обзор

Оптимизации

- Регистры

TINT: Обзор

Оптимизации

- Регистры
- Top Of Stack

TINT: Обзор

Оптимизации

- Регистры
- Top Of Stack
- Неявная обработка исключений

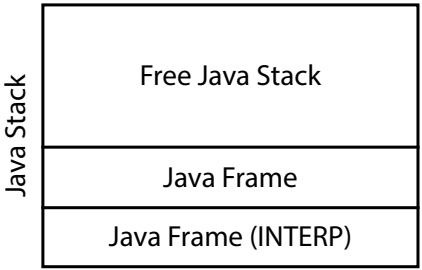
TINT: Обзор

Оптимизации

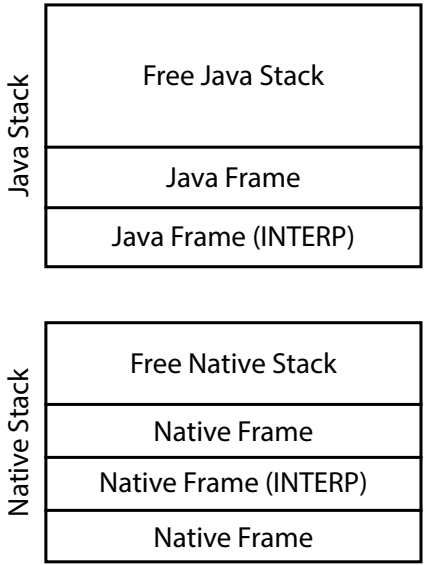
- Регистры
- Top Of Stack
- Неявная обработка исключений
- ”rewriting” байткодов

TINT: Stack

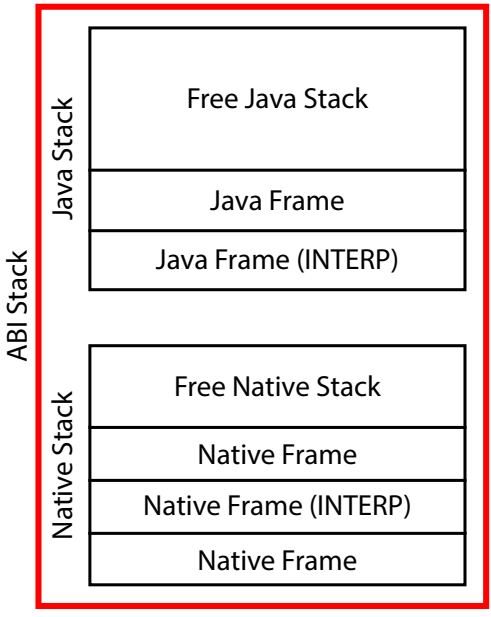
TINT: Stack



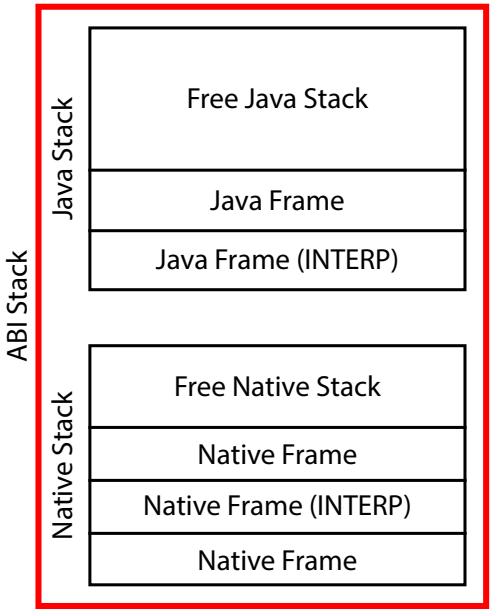
TINT: Stack

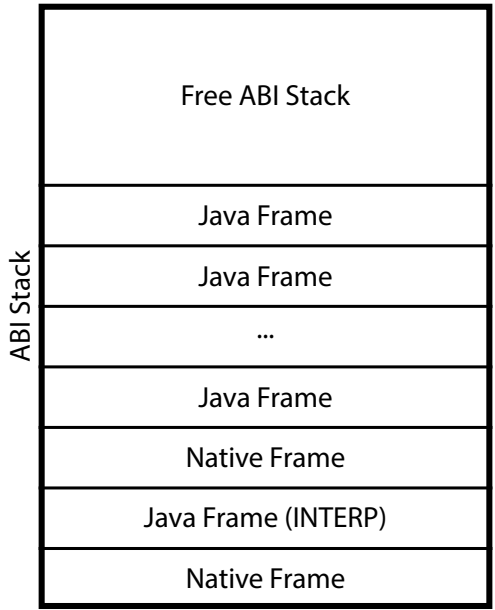

Java Stack
Free Java Stack
Java Frame
Java Frame (INTERP)
Native Stack
Free Native Stack
Native Frame
Native Frame (INTERP)
Native Frame

TINT: Stack

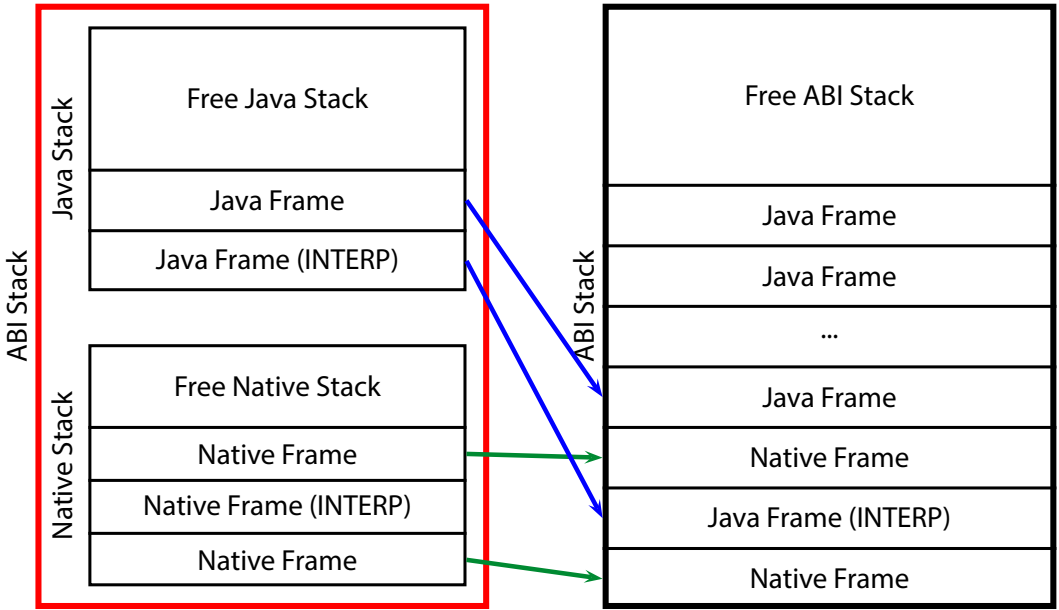

ABI Stack
Java Stack
Free Java Stack
Java Frame
Java Frame (INTERP)
Native Stack
Free Native Stack
Native Frame
Native Frame (INTERP)
Native Frame

TINT: Stack

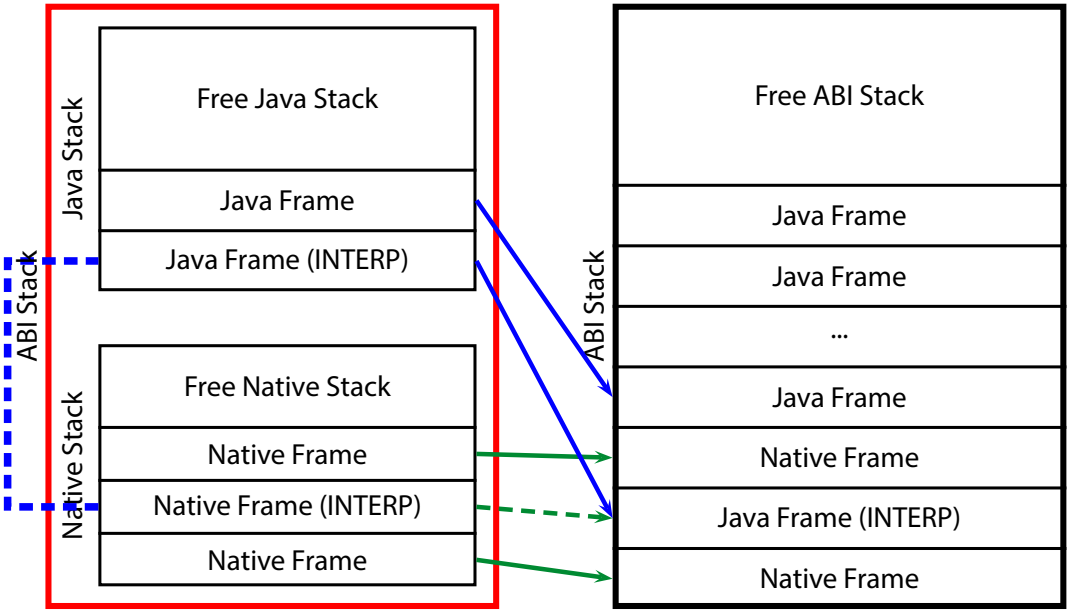

ABI Stack
Java Stack
Free Java Stack
Java Frame
Java Frame (INTERP)
Native Stack
Free Native Stack
Native Frame
Native Frame (INTERP)
Native Frame


ABI Stack
Free ABI Stack
Java Frame
Java Frame
...
Java Frame
Native Frame
Java Frame (INTERP)
Native Frame

TINT: Stack


ABI Stack
Java Stack
Free Java Stack
Java Frame
Java Frame (INTERP)
Native Stack
Free Native Stack
Native Frame
Native Frame (INTERP)
Native Frame
ABI Stack
Free ABI Stack
Java Frame
Java Frame
...
Java Frame
Native Frame
Java Frame (INTERP)
Native Frame

TINT: Stack


Java Stack
Free Java Stack
Java Frame
Java Frame (INTERP)
ABI Stack
Native Stack
Free Native Stack
Native Frame
Native Frame (INTERP)
Native Frame
ABI Stack
Free ABI Stack
Java Frame
Java Frame
...
Java Frame
Native Frame
Java Frame (INTERP)
Native Frame

TINT: Неявные исключения

- NullPointerException (SIGSEGV)
- StackOverflowException (SIGSEGV)
- ArithmeticException (SIGFPE)
- AbstractMethodError (SIGSEGV)

TINT: putfield VS fast_putfield

`putfield:`

TINT: putfield VS fast_putfield

putfield:

```
dispatch VOID_TOS
switch (type) {
   case INT:    ...
   case FLOAT:  ...
   case DOUBLE: ...
   case OBJECT: ...
   case LONG:   ...
   case SHORT:  ...
   case CHAR:   ...
   case BYTE:   ...
   case BOOL:   ...
}
```

TINT: putfield VS fast_putfield

putfield:

```
dispatch VOID_TOS                dispatch {I|F|D|A|L|S|C|B|B}_TOS
switch (type) {
   case INT:    ...              fast_iputfield: ...
   case FLOAT:  ...              fast_fputfield: ...
   case DOUBLE: ...              fast_dputfield: ...
   case OBJECT: ...              fast_aputfield: ...
   case LONG:   ...              fast_lputfield: ...
   case SHORT:  ...              fast_sputfield: ...
   case CHAR:   ...              fast_cputfield: ...
   case BYTE:   ...              fast_bputfield: ...
   case BOOL:   ...              fast_bputfield: ...
}
```

TINT: putfield VS fast_putfield

putfield:

```
dispatch VOID_TOS
switch (type) {
   case INT:    ...
   case FLOAT:  ...
   case DOUBLE: ...
   case OBJECT: ...
   case LONG:   ...
   case SHORT:  ...
   case CHAR:   ...
   case BYTE:   ...
   case BOOL:   ...
}
```

```
dispatch {I|F|D|A|L|S|C|B|B}_TOS

fast_iputfield: ...
fast_fputfield: ...
fast_dputfield: ...
fast_aputfield: ...
fast_lputfield: ...
fast_sputfield: ...
fast_cputfield: ...
fast_bputfield: ...
fast_bputfield: ...
```

TINT: ”rewriting” байткодов

Original	Patched
`_push_i`	`_push_i`
`_putfield int_field`	`_fast_iputfield`

TINT: ”rewriting” байткодов

Original	Patched
`_push_i`	`_push_i`
`_putfield int_field`	`_fast_iputfield`
`...`	`...`
`_aload_0`	`_fast_faccess_0`
`_getfield float`	

TINT: Эльбрус

TINT: Эльбрус

- Написан на VLIW-ассемблере

TINT: Эльбрус

- Написан на VLIW-ассемблере
- Регистровый кеш

TINT: Эльбрус

- Написан на VLIW-ассемблере
- Регистровый кеш
- Неявный ArrayIndexOutOfBoundsException

TINT: Эльбрус

- Написан на VLIW-ассемблере
- Регистровый кеш
- Неявный ArrayIndexOutOfBoundsException
- Быстрее обычного в 3.1 раза

TINT: Регистровый кеш

data	x86	Elbrus
local[0 ... 3]	stack	register
local[4 ... N]	stack	stack
Top of Stack	register	register
BC stream	stack	register
CP Cache	stack	register
stack ptr	register	register
frame ptr	register	register
Profile Data	stack	register

TINT: AIOOBE

ArrayIndexOutOfBoundsException

TINT: AIOOBE

ArrayIndexOutOfBoundsException

```
ldw [%array_reg + length_off], %length_reg ; SIGSEGV (npe)
```

TINT: AIOOBE

ArrayIndexOutOfBoundsException

```
ldw [%array_reg + length_off], %length_reg ; SIGSEGV (npe)

cmp %length_reg, %index_reg, %pred          ; unsigned <=
```

TINT: AIOOBE

ArrayIndexOutOfBoundsException

```
ldw [%array_reg + length_off], %length_reg ; SIGSEGV (npe)

cmp %length_reg, %index_reg, %pred          ; unsigned <=

mov NULL, %array_reg ? %pred
mov 0,    %index_reg ? %pred
```

TINT: AIOOBE

ArrayIndexOutOfBoundsException

```
ldw [%array_reg + length_off], %length_reg ; SIGSEGV (npe)

cmp %length_reg, %index_reg, %pred          ; unsigned <=

mov NULL, %array_reg ? %pred
mov 0,    %index_reg ? %pred

ldd [%array_reg + %index_reg], %result_reg ; SIGSEGV (aioobe)
```

TINT: Сравнение

SpecJVM98: java -Xint

	Elbrus					
Test (сек)	CPP	TINT	Speedup			
compress	1555	512	3.0			
jess	340	109	3.1			
db	663	218	3.0			
javac	387	123	3.2			
mpegaudio	1281	447	2.9			
mtrt	362	109	3.3			
jack	212	69	3.1			
<geomean>			3.1			

TINT: Сравнение

SpecJVM98: java -Xint

	Elbrus					
Test (сек)	CPP	TINT	Speedup			
compress	1555	512	3.0			
jess	340	109	3.1			
db	663	218	3.0			
javac	387	123	3.2			
mpegaudio	1281	447	2.9			
mtrt	362	109	3.3			
jack	212	69	3.1			
<geomean>			3.1			

TINT: Сравнение

SpecJVM98: java -Xint

	Elbrus					
Test (сек)	CPP	TINT	Speedup	x86	x86*FREQ	e2k/x86
compress	1555	512	3.0	52.1	163	3.1
jess	340	109	3.1	16.1	50	2.2
db	663	218	3.0	28.0	88	2.5
javac	387	123	3.2	18.0	56	2.2
mpegaudio	1281	447	2.9	42.6	133	3.4
mtrt	362	109	3.3	20.9	65	1.7
jack	212	69	3.1	12.2	38	1.8
<geomean>			3.1			2.3

Деоптимизация

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
```

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
// ...
static int test_static(int i) {
  return (new JokerTest()).test(i);
}
```

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
// ...
static int test_static(int i) {
  return (new JokerTest()).test(i);
}
@Inline
public int test(int i) {
  if (i == 30_000) {
    System.out.println("!!!Trap!!!");
  }
  return i / 5;
}
```

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
// ...
static int test_static(int i) {
  return (new JokerTest()).test(i);
}
@Inline
public int test(int i) {
  if (i == 30_000) {
    System.out.println("!!!Trap!!!");
  }
  return i / 5;
}
```

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
// ...
static int test_static(int i) {
  return (new JokerTest()).test(i);
}
@Inline
public int test(int i) {
  if (i == 30_000) {  // [15_000 -> else, 0 -> then]
    System.out.println("!!!Trap!!!");
  }
  return i / 5;
}
```

Java: JokerTest.java

```
static int test_static_OPT(int i) {  // i ∈ [0, 50_000)
  if (i == 30_000) goto trap;

  return (((long)i * 0x66666667) >> 33) - (i >> 31);

  trap: @UncommonTrap<untaken_branch>@

}
```

Деоптимизация: Overview

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

- Кооперативная (Uncommon Trap)

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

- Кооперативная (Uncommon Trap)
  - Недостигнутые пути (if-else, Null Check, etc.)

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

- Кооперативная (Uncommon Trap)
  - Недостигнутые пути (if-else, Null Check, etc.)
- Принудительная (неявно, на Safepoint‘ах)

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

- Кооперативная (Uncommon Trap)
  - Недостигнутые пути (if-else, Null Check, etc.)
- Принудительная (неявно, на Safepoint‘ах)
  - Нарушение зависимостей метода

Деоптимизация: Overview

Что делает?

”На лету” подменяет JIT-метод интерпретатором ($\neg OSR$)

Для чего?

Позволяет выполнять ”агрессивные” оптимизации

Какая бывает?

- Кооперативная (Uncommon Trap)
  - Недостигнутые пути (if-else, Null Check, etc.)
- Принудительная (неявно, на Safepoint‘ах)
  - Нарушение зависимостей метода
  - Обработка исключений

Java: JokerTest.java

```
static int  sum;
for (int i = 0; i < 50_000; ++i) {
  sum += test_static(i);
}
// ...
static int test_static(int i) {
  return (new JokerTest()).test(i);
}
@Inline
public int test(int i) {
  if (i == 30_000) {
    System.out.println("!!!Trap!!!");
  }
  return i / 5;
}
```

java -XX:+PrintCompilation JokerTest

```
6157 1 JokerTest::test_static (12 bytes) made not entrant
!!!Trap!!!
```

java -XX:+PrintCompilation -XX:+TraceDeoptimization JokerTest

```
Uncommon trap occurred in JokerTest::test_static
 reason=unstable_if action=reinterpret unloaded_class_index=-1
6157 1 JokerTest::test_static (12 bytes) made not entrant
Virtual frames (innermost first):
 0 - frame: JokerTest.test(JokerTest.java:14) - if_icmpne @ bci 4


 1 - frame: JokerTest.test_static(JokerTest.java:23) - IV @ bci 8



!!!Trap!!!
```

java -XX:+PrintCompilation -XX:+TraceDeoptimization JokerTest

```
Uncommon trap occurred in JokerTest::test_static
 reason=unstable_if action=reinterpret unloaded_class_index=-1
6157 1 JokerTest::test_static (12 bytes) made not entrant
Virtual frames (innermost first):
 0 - frame: JokerTest.test(JokerTest.java:14) - if_icmpne @ bci 4

 1 - frame: JokerTest.test_static(JokerTest.java:23) - IV @ bci 8

!!!Trap!!!
```

java -XX:+PrintCompilation -XX:+TraceDeoptimization JokerTest

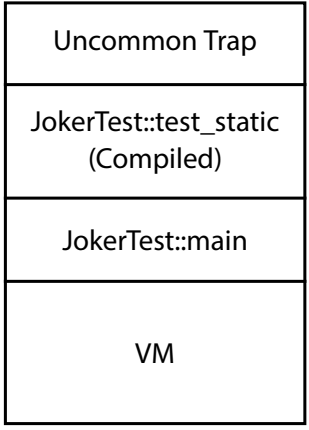
```
Uncommon trap occurred in JokerTest::test_static
 reason=unstable_if action=reinterpret unloaded_class_index=-1
6157 1 JokerTest::test_static (12 bytes) made not entrant
Virtual frames (innermost first):
 0 - frame: JokerTest.test(JokerTest.java:14) - if_icmpne @ bci 4

 1 - frame: JokerTest.test_static(JokerTest.java:23) - IV @ bci 8

!!!Trap!!!
```

java -XX:+PrintCompilation -XX:+TraceDeoptimization JokerTest

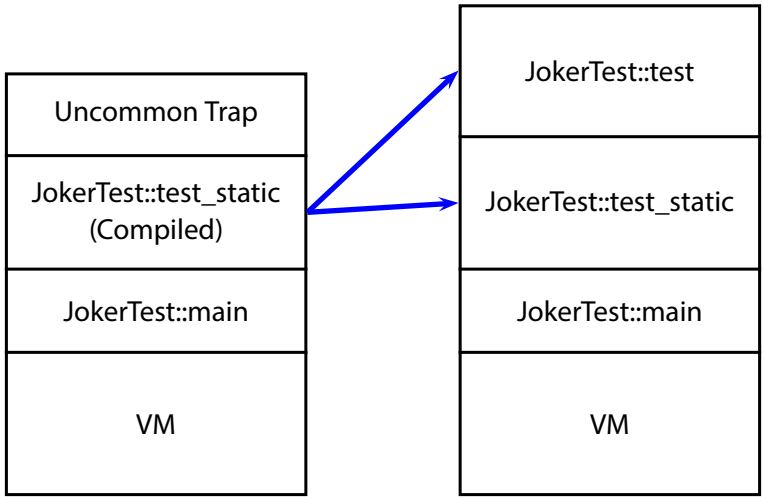
```
Uncommon trap occurred in JokerTest::test_static
 reason=unstable_if action=reinterpret unloaded_class_index=-1
6157 1 JokerTest::test_static (12 bytes) made not entrant
Virtual frames (innermost first):
 0 - frame: JokerTest.test(JokerTest.java:14) - if_icmpne @ bci 4

 if (i == 30_000) {
 1 - frame: JokerTest.test_static(JokerTest.java:23) - IV @ bci 8
 return (new JokerTest()).test(i);


!!!Trap!!!
```

Фреймы

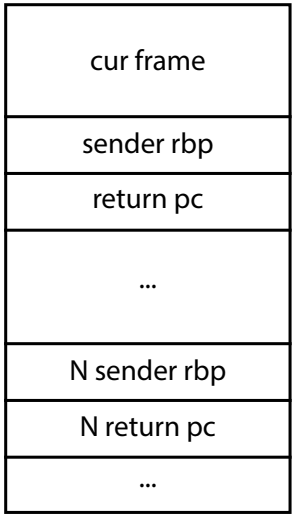
Фреймы

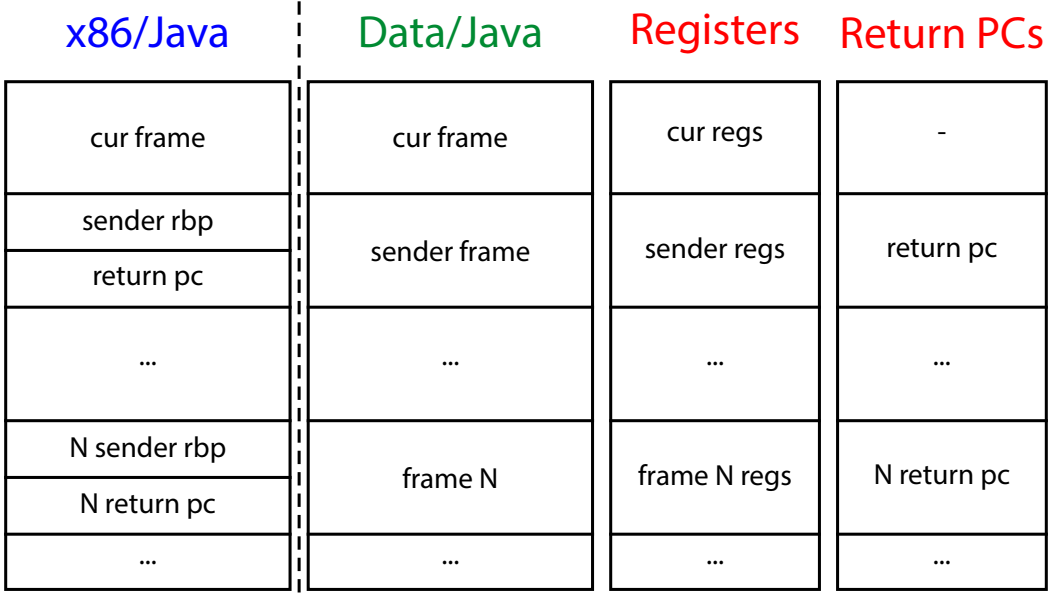

Uncommon Trap
JokerTest::test_static
(Compiled)
JokerTest::main
VM

Фреймы

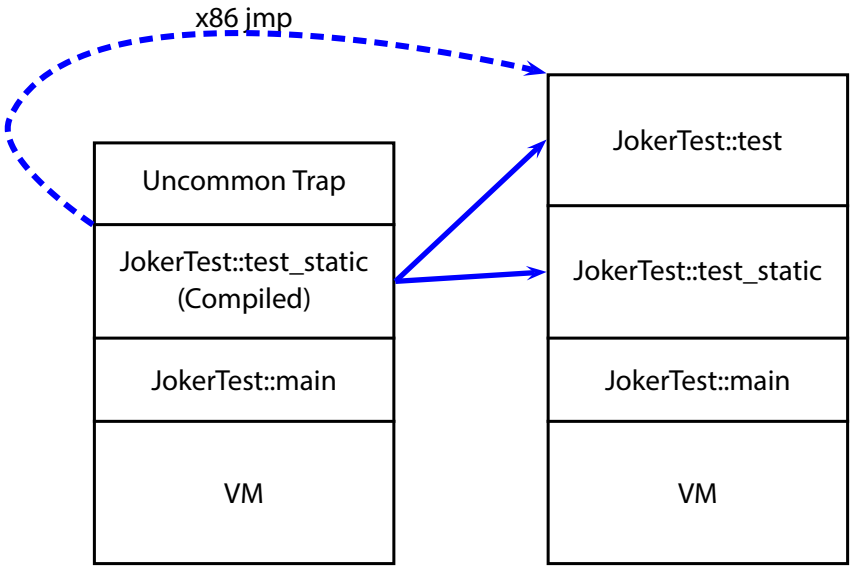

Uncommon Trap
JokerTest::test_static
(Compiled)
JokerTest::main
VM
JokerTest::test
JokerTest::test_static
JokerTest::main
VM

x86/Java

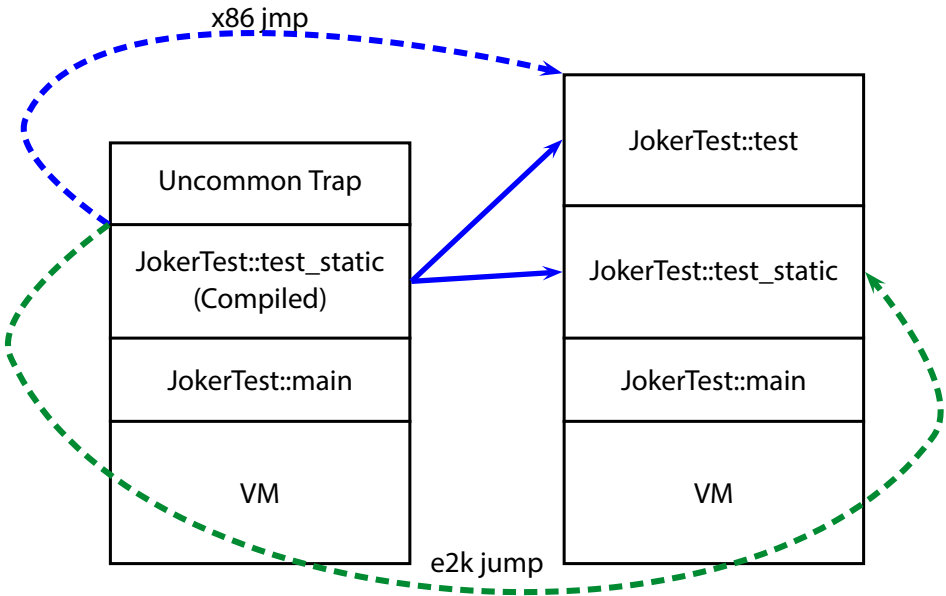



x86/Java
Data/Java
Registers
Return PCs
cur frame
sender rbp
return pc
...
N sender rbp
N return pc
...
cur frame
sender frame
...
frame N
...
cur regs
sender regs
...
frame N regs
...
-
return pc
...
N return pc
...

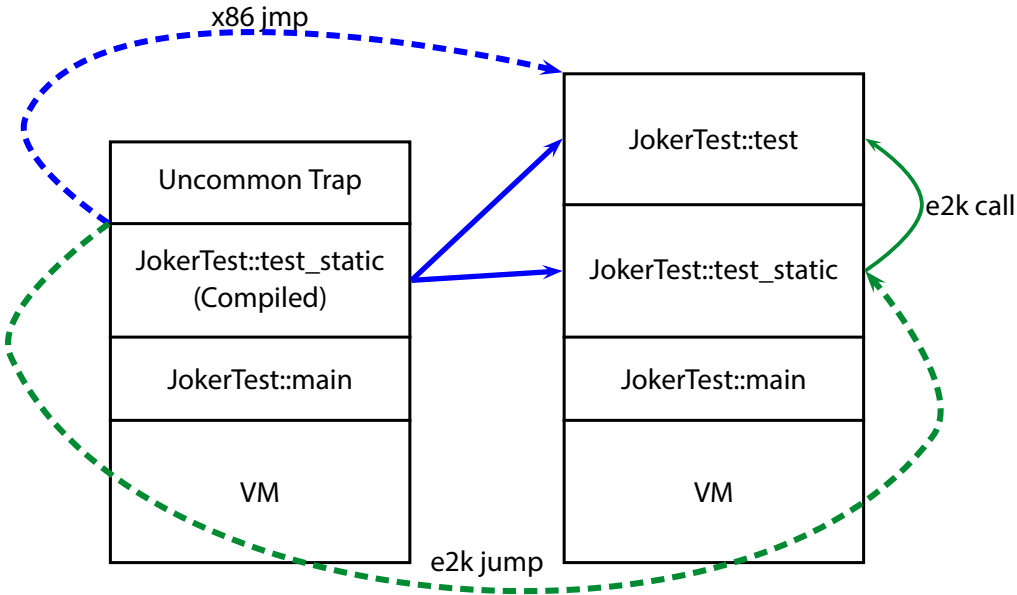
Фреймы


x86 jmp
Uncommon Trap
JokerTest::test_static
(Compiled)
JokerTest::main
VM
JokerTest::test
JokerTest::test_static
JokerTest::main
VM

Фреймы


x86 jmp
Uncommon Trap
JokerTest::test_static
(Compiled)
JokerTest::main
VM
JokerTest::test
JokerTest::test_static
JokerTest::main
VM
e2k jump

Фреймы


x86 jmp
Uncommon Trap
JokerTest::test_static
(Compiled)
JokerTest::main
VM
JokerTest::test
JokerTest::test_static
JokerTest::main
VM
e2k call
e2k jump

JIT-компилятор

JIT: Обзор

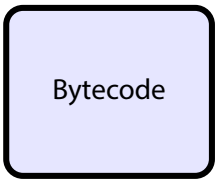
C2 компилятор

x86, SPARC, POWER-PC, ...

JIT: Обзор

C2 компилятор

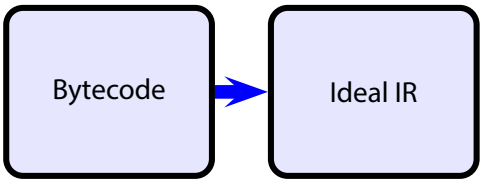
x86, SPARC, POWER-PC, ...



JIT: Обзор

C2 компилятор

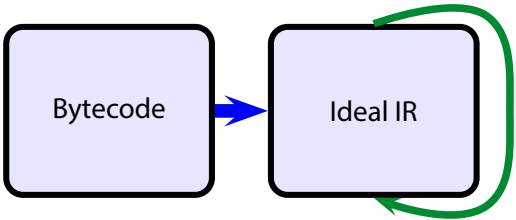
x86, SPARC, POWER-PC, ...



JIT: Обзор

C2 компилятор

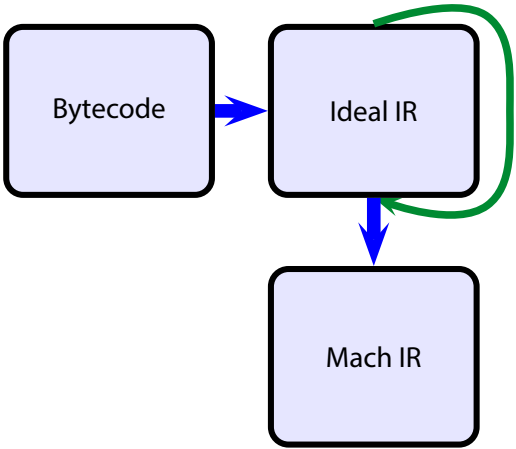
x86, SPARC, POWER-PC, ...



JIT: Обзор

C2 компилятор

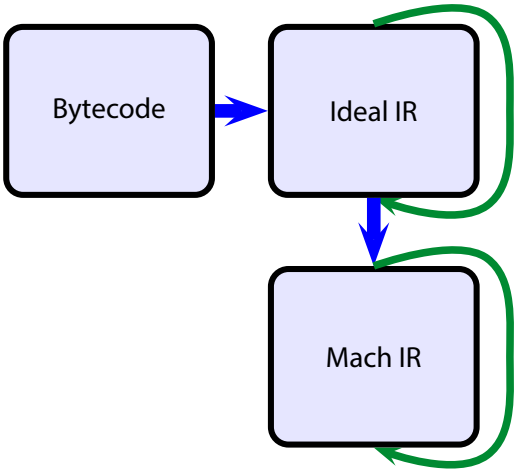
x86, SPARC, POWER-PC, ...



JIT: Обзор

C2 компилятор

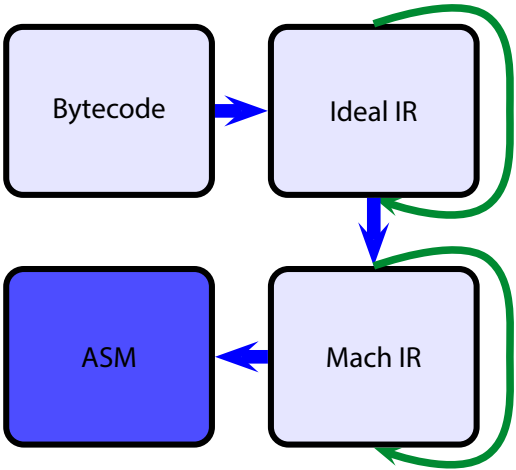
x86, SPARC, POWER-PC, ...



JIT: Обзор

C2 компилятор

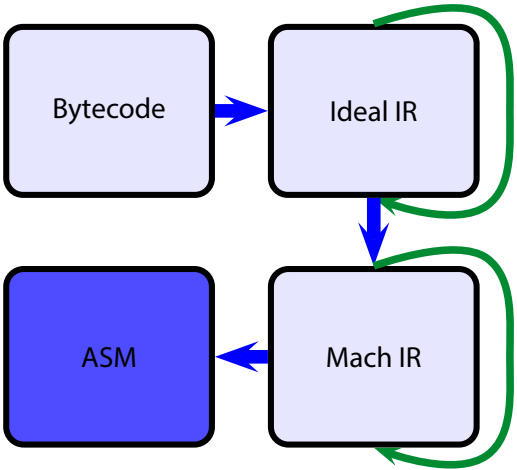
x86, SPARC, POWER-PC, ...



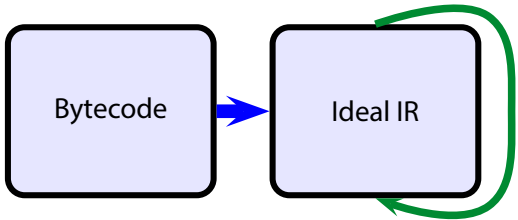
JIT: Обзор

C2 компилятор

x86, SPARC, POWER-PC, ...



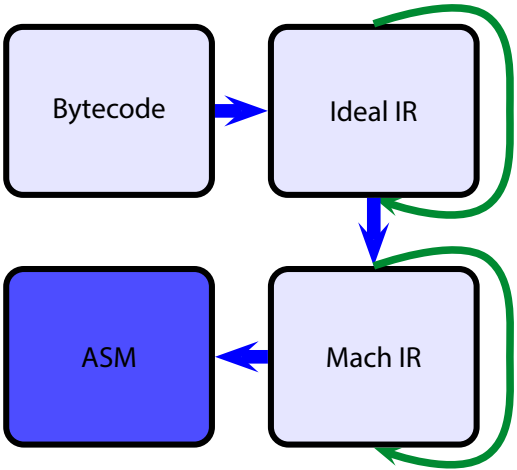
Elbrus



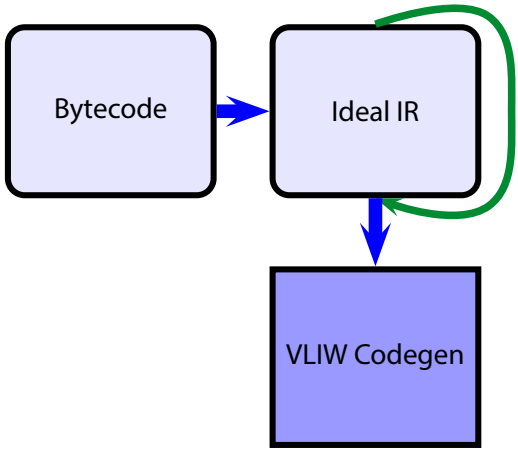
JIT: Обзор

C2 компилятор

x86, SPARC, POWER-PC, ...



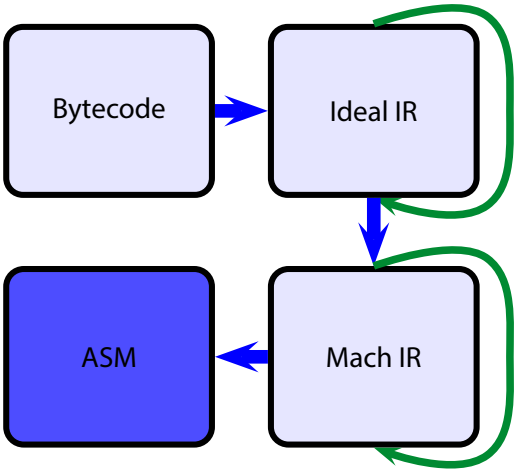
Elbrus



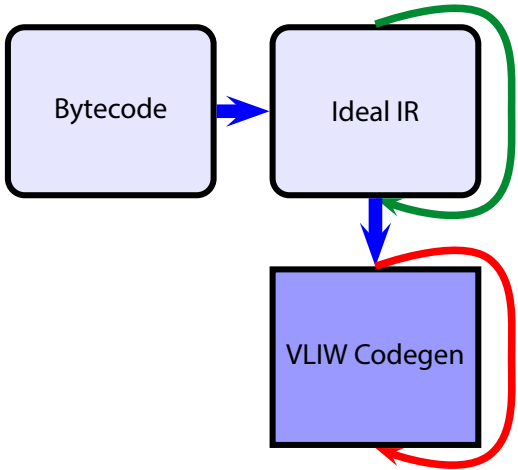
JIT: Обзор

C2 компилятор

x86, SPARC, POWER-PC, ...



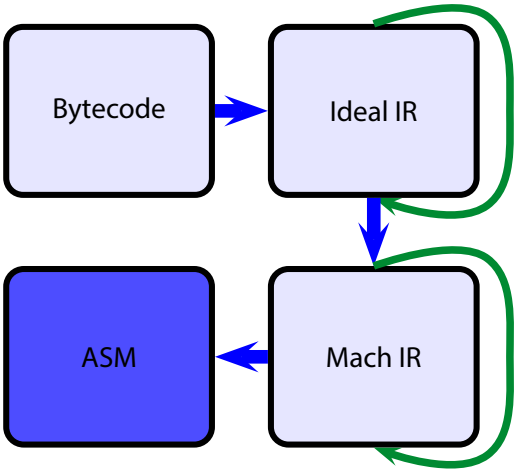
Elbrus



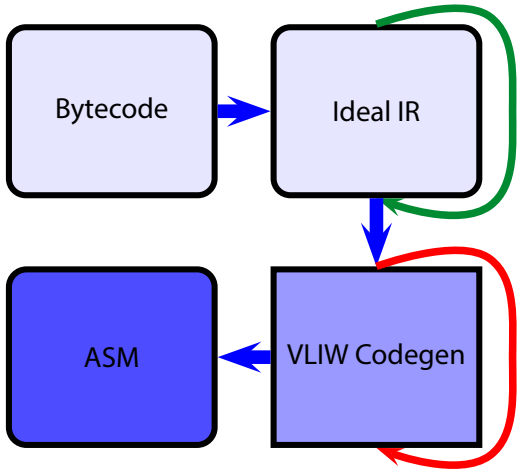
JIT: Обзор

C2 компилятор

x86, SPARC, POWER-PC, ...



Elbrus



JIT: Пример

JIT: Пример

```
class Point {
  public float x;
  public float y;
}

static float codegen_demo(Point a, Point b, float d) {
  float ret = (float)Math.sqrt(a.x * a.y * b.x * b.y);
  if (d > 0.0f) {
    ret += d;
  }
  return ret;
}
```

JIT: Java => C2 => Codegen

JIT: Java => C2 => Codegen

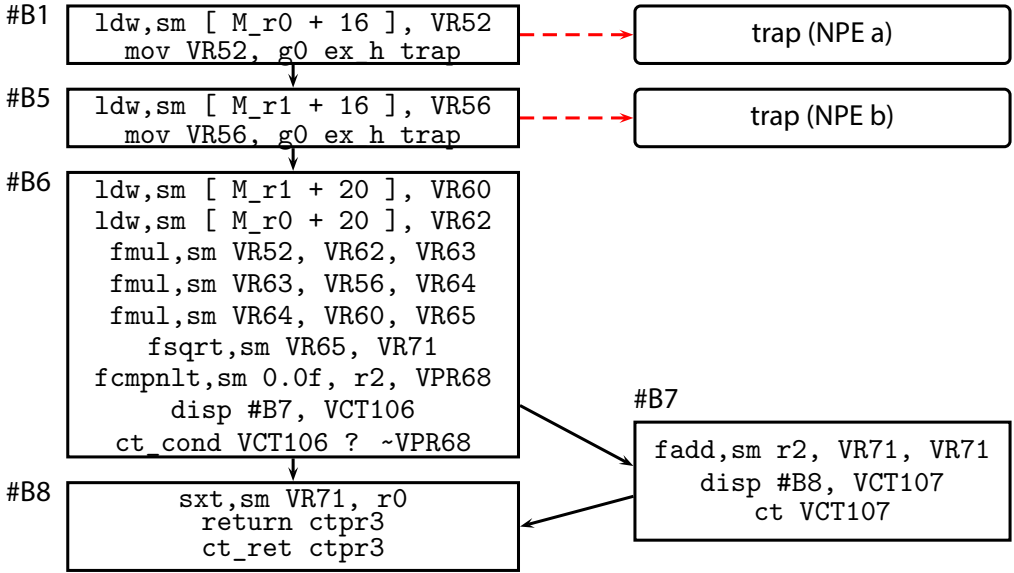
```
static float codegen_demo(
  Point a, Point b,
  float d) {

  float ret = (float)Math.sqrt(
    a.x * a.y * b.x * b.y);
  if (d > 0.0f) {
    ret += d;
  }
  return ret;
}
```

JIT: Java => C2 => Codegen

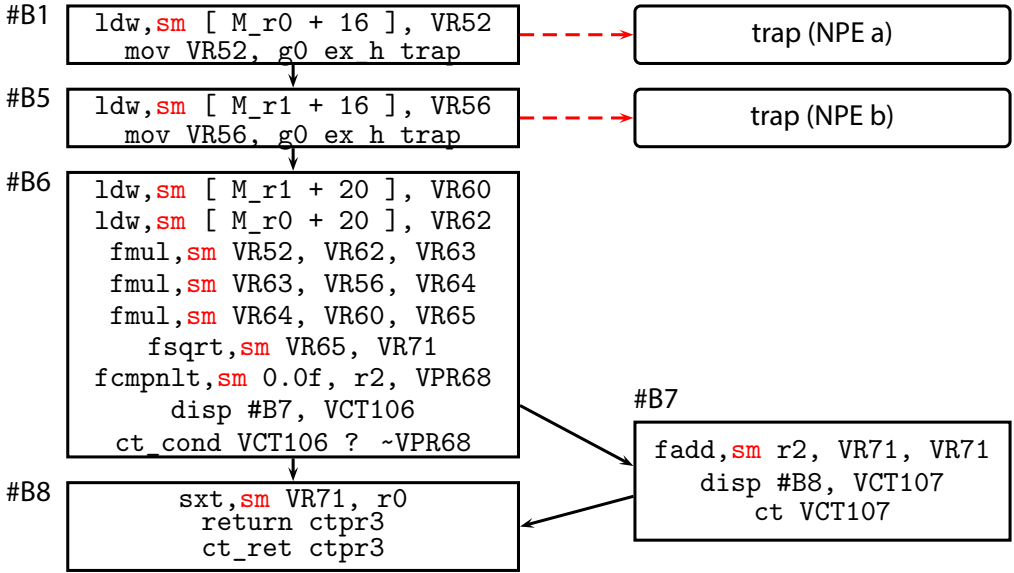
```
static float codegen_demo(
  Point a, Point b,
  float d) {

  float ret = (float)Math.sqrt(
    a.x * a.y * b.x * b.y);
  if (d > 0.0f) {
    ret += d;
  }
  return ret;
}
```

```
#B1: ldw,sm    [ M_r0 + 16 ], VR52
     mov       VR52, g0 ex_h trap
#B5: ldw,sm    [ M_r1 + 16 ], VR56
     mov       VR56, g0 ex_h trap
#B6: ldw,sm    [ M_r1 + 20 ], VR60
     ldw,sm    [ M_r0 + 20 ], VR62
     fmul,sm   VR52, VR62, VR63
     fmul,sm   VR63, VR56, VR64
     fmul,sm   VR64, VR60, VR65
     fsqrt,sm  VR65, VR71
     fcmpnlt,sm 0.0f, r2, VPR68
     disp   #B8, VCT106
     ct_cond VCT106 ? VPR68
#B7: fadd,sm   r2, VR71, VR71
#B8: sxt,sm    VR71, r0
     return ctpr3
     ct_ret ctpr3
```

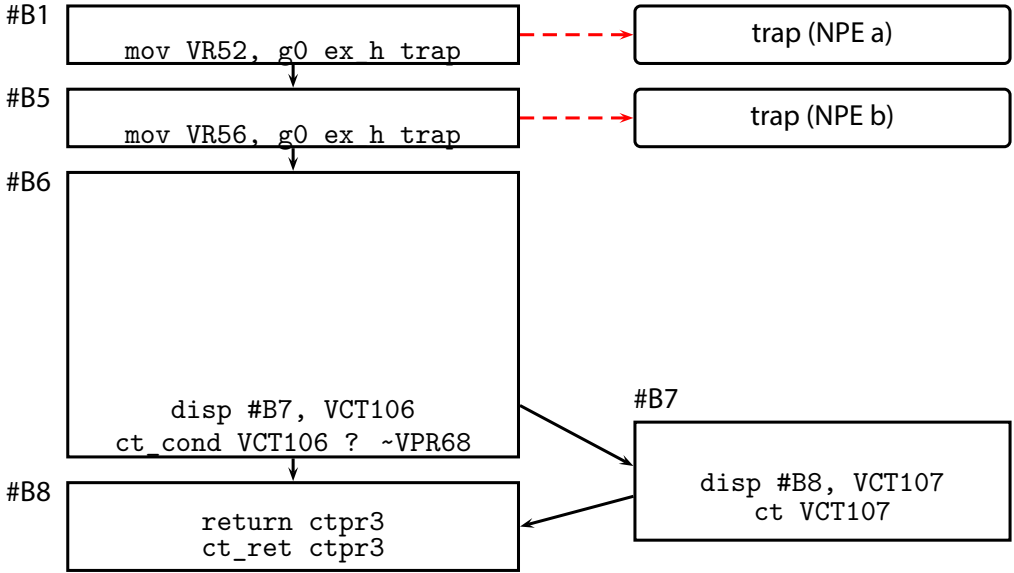
JIT: Java => C2 => Codegen



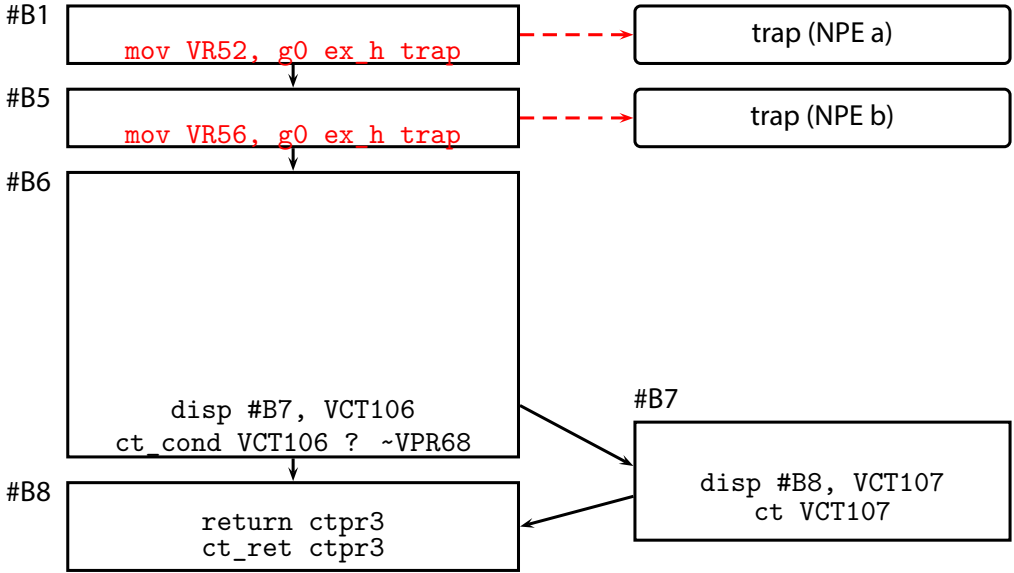
JIT: Java => C2 => Codegen



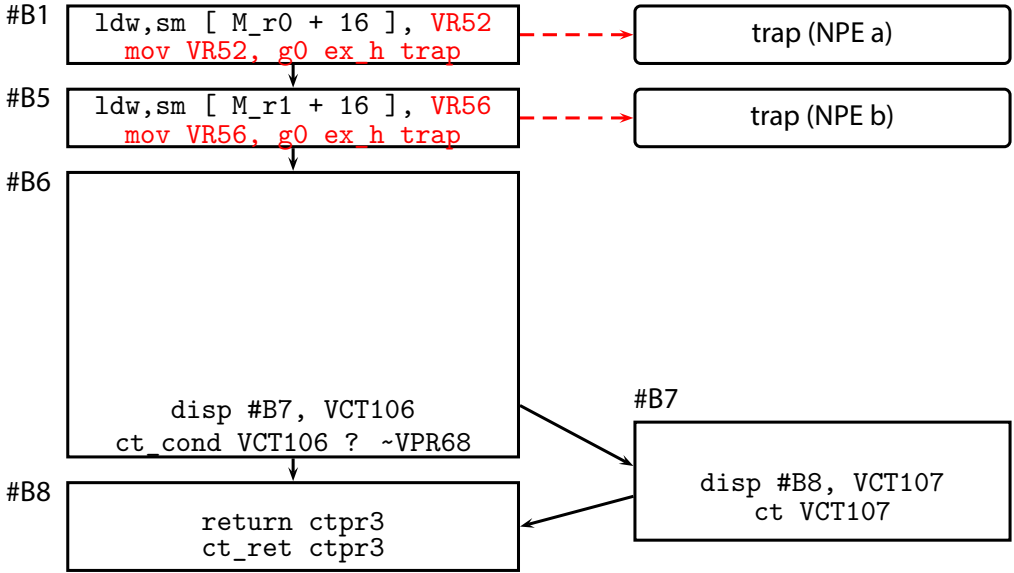
JIT: Java => C2 => Codegen



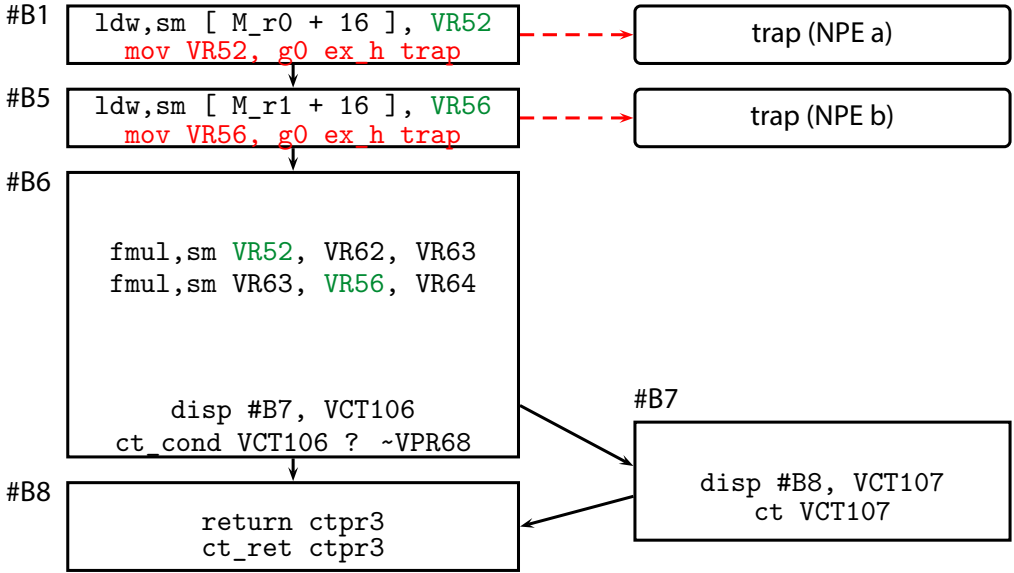
JIT: Java => C2 => Codegen



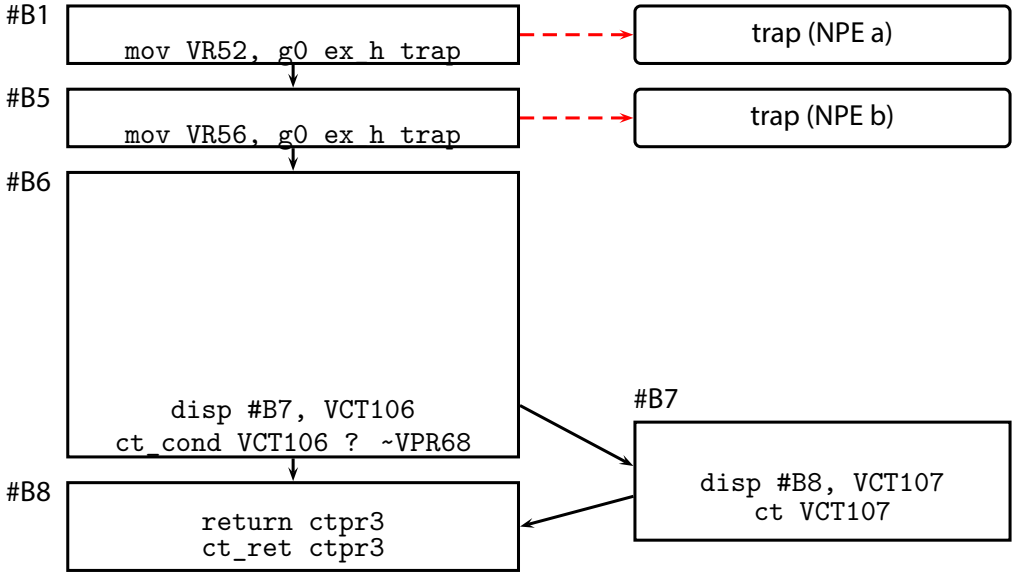
JIT: Java => C2 => Codegen



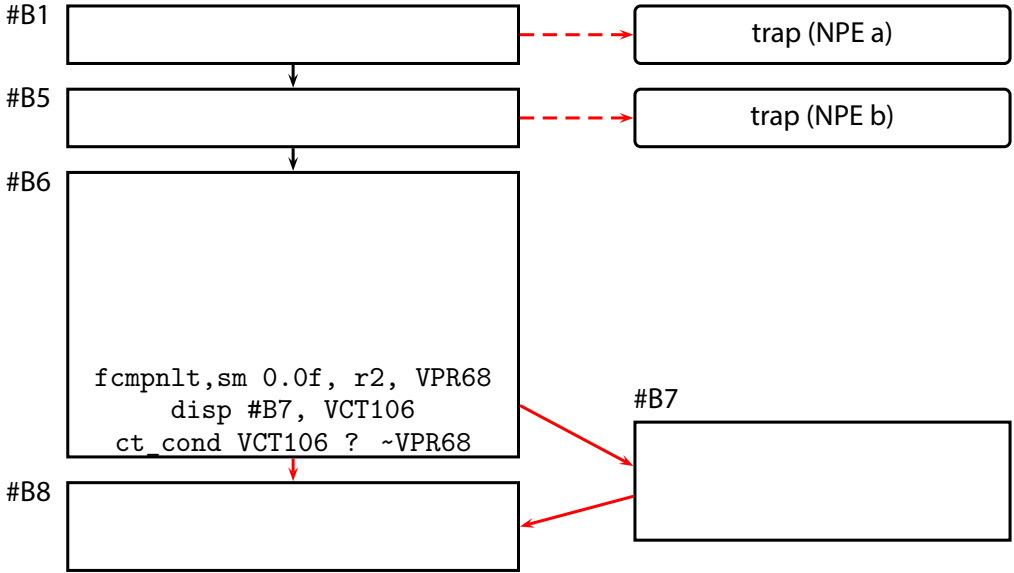
JIT: Java => C2 => Codegen



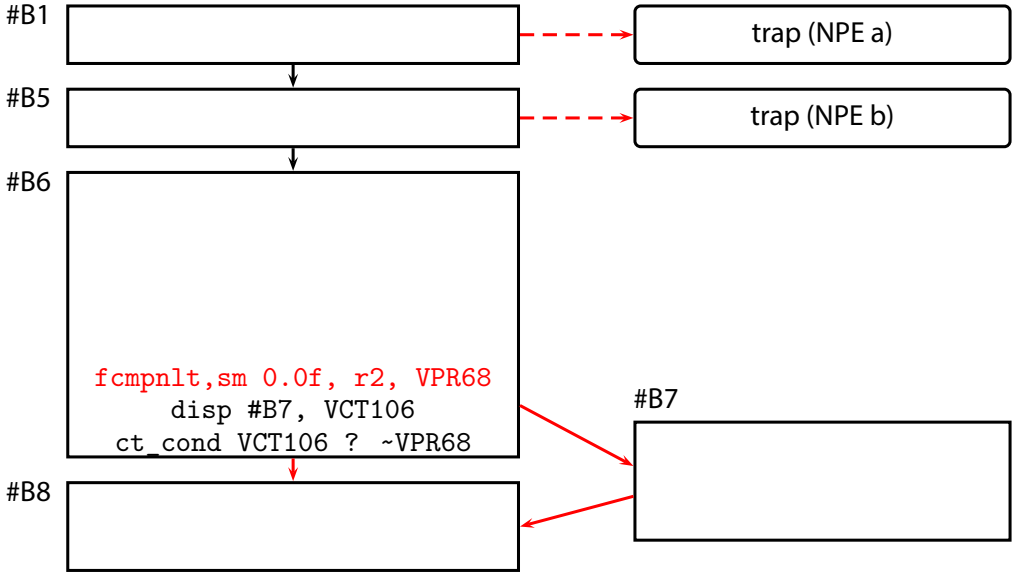
JIT: Java => C2 => Codegen



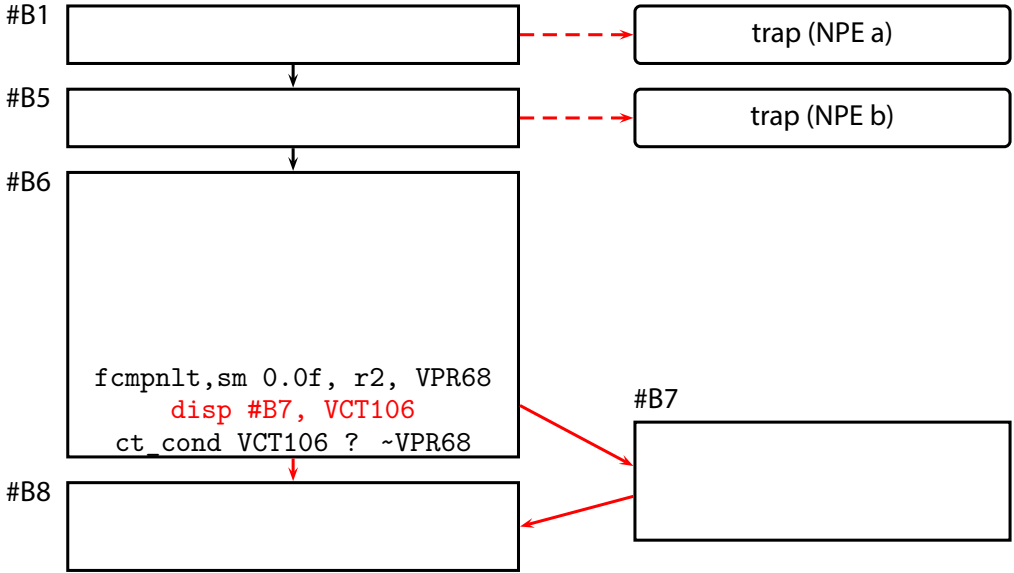
JIT: Java => C2 => Codegen



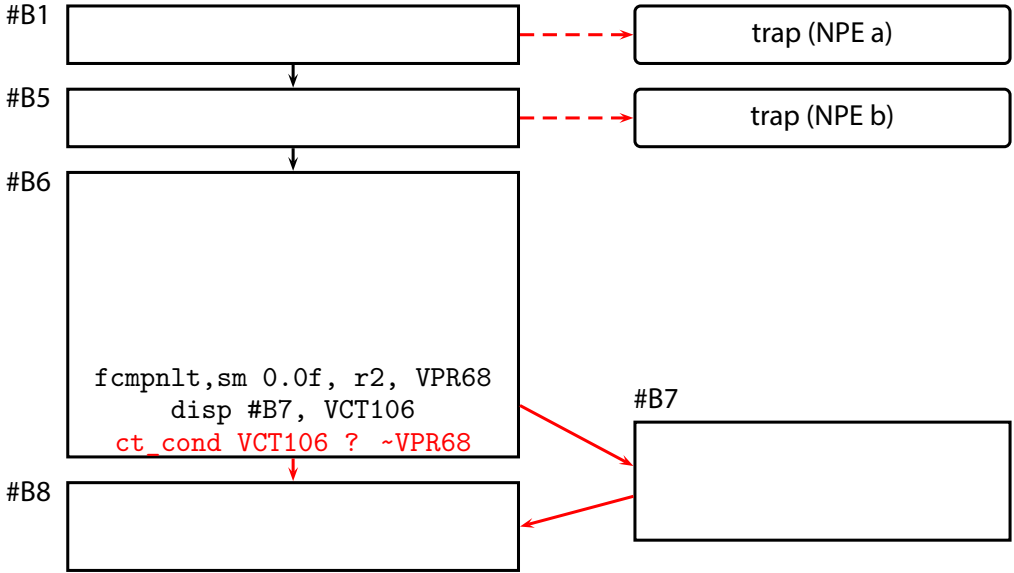
JIT: Java => C2 => Codegen



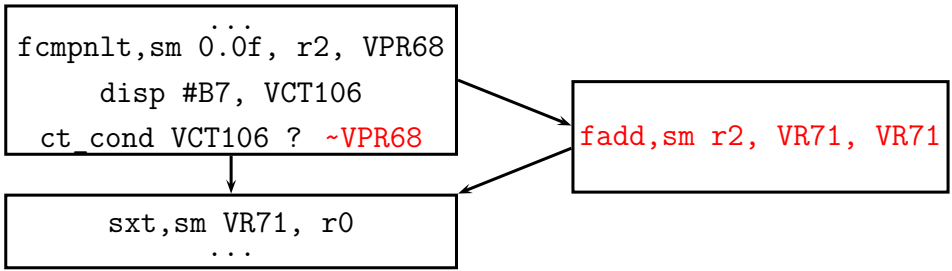
JIT: Java => C2 => Codegen



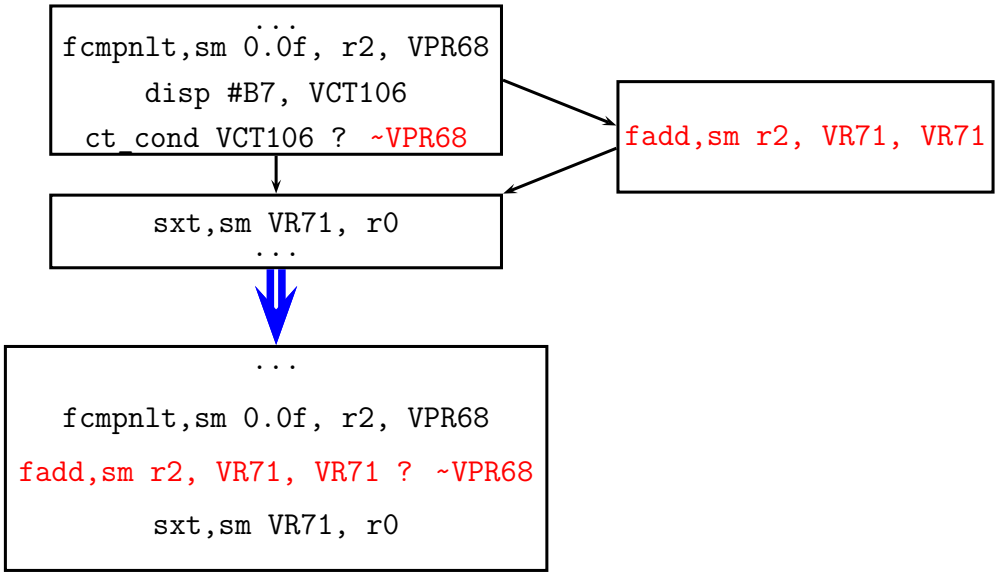
JIT: Java => C2 => Codegen



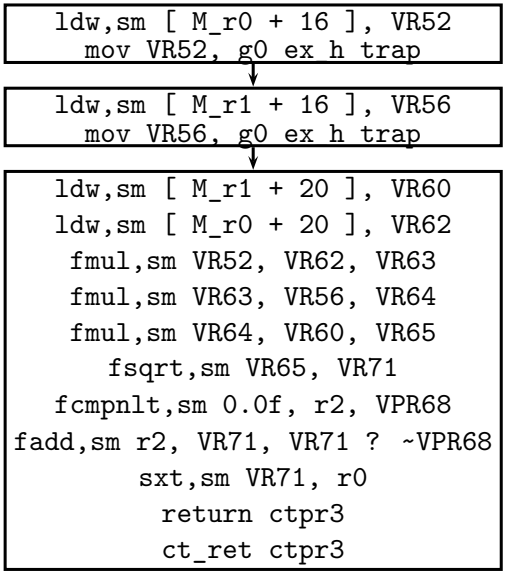
JIT: If conversion



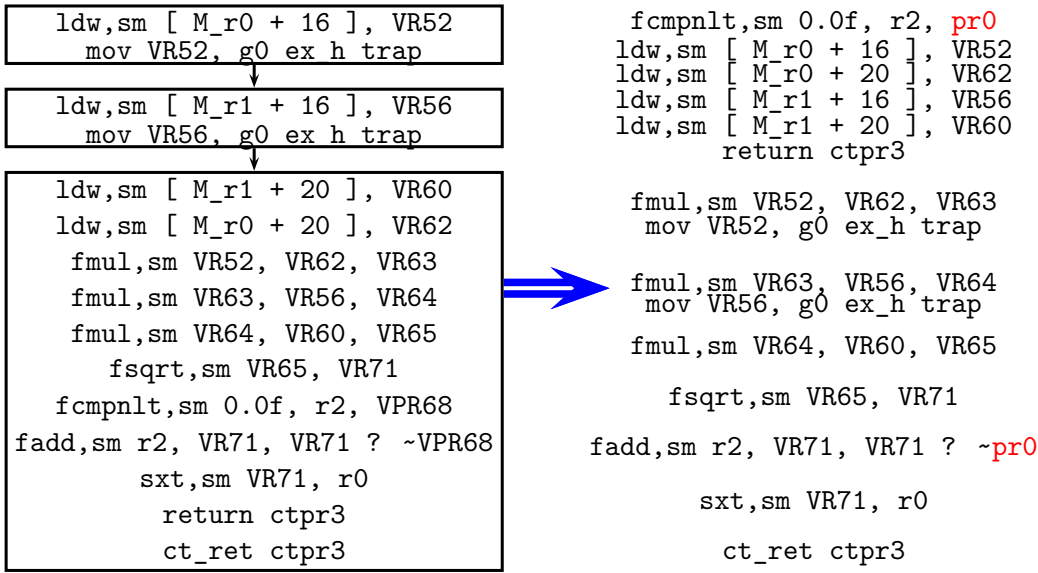
JIT: If conversion



JIT: Планировщик



JIT: Планировщик



JIT: Распределение регистров

```
    fcmpnlt,sm 0.0f, r2, pr0
   ldw,sm [ M_r0 + 16 ], VR52
   ldw,sm [ M_r0 + 20 ], VR62
   ldw,sm [ M_r1 + 16 ], VR56
   ldw,sm [ M_r1 + 20 ], VR60
          return ctpr3

    fmul,sm VR52, VR62, VR63
     mov VR52, g0 ex_h trap

    fmul,sm VR63, VR56, VR64
     mov VR56, g0 ex_h trap

    fmul,sm VR64, VR60, VR65

       fsqrt,sm VR65, VR71

fadd,sm r2, VR71, VR71 ? ~pr0

         sxt,sm VR71, r0

          ct_ret ctpr3
```

JIT: Распределение регистров

```
fcmpnlt,sm 0.0f, r2, pr0
ldw,sm [ M_r0 + 16 ], VR52
ldw,sm [ M_r0 + 20 ], VR62
ldw,sm [ M_r1 + 16 ], VR56
ldw,sm [ M_r1 + 20 ], VR60
       return ctpr3

fmul,sm VR52, VR62, VR63
 mov VR52, g0 ex_h trap

fmul,sm VR63, VR56, VR64
 mov VR56, g0 ex_h trap

fmul,sm VR64, VR60, VR65

   fsqrt,sm VR65, VR71

fadd,sm r2, VR71, VR71 ? ~pr0

      sxt,sm VR71, r0

       ct_ret ctpr3
```

```
fcmpnlt,sm 0.0f, r2, pr0
ldw,sm [ M_r0 + 16 ], r3
ldw,sm [ M_r0 + 20 ], r4
ldw,sm [ M_r1 + 16 ], r5
ldw,sm [ M_r1 + 20 ], r6
       return ctpr3

fmul,sm r3, r4, r4
mov r3, g0 ex_h trap

fmul,sm r5, r6, r6
mov r5, g0 ex_h trap

fmul,sm r4, r6, r0

   fsqrt,sm r0, r0

fadd,sm r2, r0, r0 ? ~pr0

      sxt,sm r0, r0

       ct_ret ctpr3
```

Производительность

Тестовый стенд: CPU

Параметры

- 65нм
- 4 ядра без HT
- Нет кеша 3 уровня
- 64 бит

x86

Intel Quad Q9300 (2.50 GHz)
TDP: 95W

Elbrus

Elbrus 4C (800 MHz)
TDP: 45W

Отношение частоты: $2500/800 = 3.125$
Отношение TDP: $95/45 \approx 2.1$

Тесты: specjbb2005

Java 3.0	Java 3.2	Boost +%	x86	x86/FREQ	x86/e2k
18077	25247	31.32%	65682	21018	0.83

Table: specjbb2005, server, ops

Тесты: specjvm98

Test	Java 3.0	Java 3.2	Speedup	x86	x86*FREQ	e2k/x86
compress	17.4	10.6	1.6	2.4	7.5	1.4
jess	6.0	3.8	1.6	0.5	1.6	2.4
db	20.9	16.8	1.2	5.0	15.6	1.1
javac	10.0	6.7	1.5	1.6	5.0	1.3
mpegaudio	12.7	7.1	1.8	1.3	4.1	1.8
mtrt	1.9	1.2	1.5	0.3	0.9	1.3
jack	20.3	4.4	4.6	0.8	2.5	1.8
<geomean>			1.8			1.5

Table: specjvm98, server, сек

Тесты: specjvm2008

Test	Java 3.0	Java 3.2	Boost +%	x86	x86/FREQ	x86/e2k
compiler	24.9	46.2	85.6%	241.7	77.3	1.7
compress	20.4	28.5	39.5%	147.4	47.2	1.7
crypto	20.5	33.8	64.8%	152.0	48.6	1.4
derby	37.2	50.5	35.8%	209.9	67.2	1.3
mpegaudio	12.6	18.0	42.6%	86.1	27.5	1.5
scimark.large	3.9	4.8	23.6%	17.5	5.6	1.2
scimark.small	16.4	22.4	36.3%	164.5	52.7	2.4
serial	12.4	17.1	37.7%	88.7	28.4	1.7
startup	2.3	3.1	33.9%	22.5	7.2	2.3
sunflow	8.3	11.7	40.8%	49.5	15.8	1.4
xml	26.6	52.6	97.7%	308.6	98.7	1.9
Общий счет	13.1	19.3	47.5%	98.6	31.5	1.6

Table: specjvm2008, server, ops/m

Будущее платформы

Будущее платформы

- Java 9

Будущее платформы

- Java 9
- Развитие JIT’a

Будущее платформы

- Java 9
- Развитие JIT’а
  - If conversion
  - Оптимизация работы с памятью
  - Конвейеризация циклов
  - Еще что-нибудь

Будущее платформы

- Java 9
- Развитие JIT'а
  - If conversion
  - Оптимизация работы с памятью
  - Конвейеризация циклов
  - Еще что-нибудь
- Стать сертифицированной реализацией Java

Будущее платформы

- Java 9
- Развитие JIT'а
  - If conversion
  - Оптимизация работы с памятью
  - Конвейеризация циклов
  - Еще что-нибудь
- Стать сертифицированной реализацией Java
- **Победить x86**

Q/A