

# Моделирование центрального процессора с помощью интерпретации

Курс «Программное моделирование вычислительных систем»

Григорий Речистов  
[grigory.rechistov@phystech.edu](mailto:grigory.rechistov@phystech.edu)

21 февраля 2015 г.

- 1 Конвейер процессора
- 2 Fetch
- 3 Decode
- 4 Execute
- 5 Write Back
- 6 Исключения
- 7 Advance PC
- 8 Улучшенные схемы
- 9 Литература

# На прошлой лекции

Требования к симуляторам:

- точность,
- скорость,
- совместимость/расширяемость

# Вопросы

- В чём измеряется скорость симуляции?

# Вопросы

- В чём измеряется скорость симуляции?
- Как соотносятся скорости функционального и потактового симуляторов?

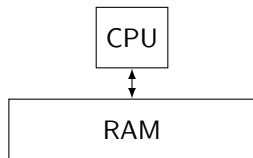
# Вопросы

- В чём измеряется скорость симуляции?
- Как соотносятся скорости функционального и потактового симуляторов?
- Для чего необходимо предоставлять API симулятора?

# Вопросы

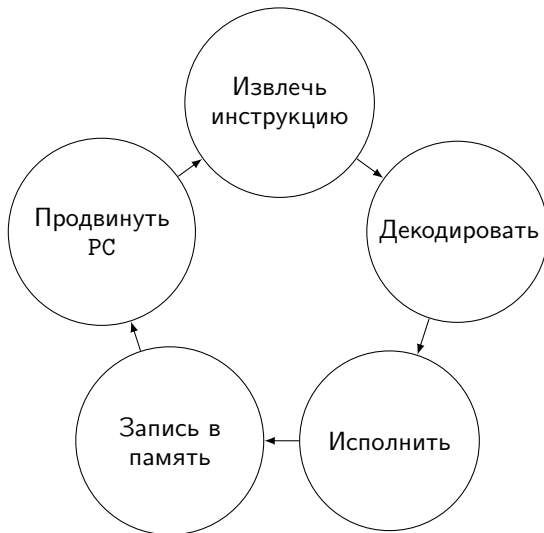
- В чём измеряется скорость симуляции?
- Как соотносятся скорости функционального и потактового симуляторов?
- Для чего необходимо предоставлять API симулятора?  
Чтобы пользователи могли с ним поиграть.

# Моделируемая система





# Конвейер процессора



# Переключаемый интерпретатор (switched)

```
while (run) {  
    raw_code = fetch(PC);  
    (opcode, operands) = decode(raw_code);  
    switch (opcode) {  
        case opcode1:  
            func1(operands); PC++; break;  
  
        case opcode2:  
            func2(operands); PC++; break;  
  
        /*...*/  
    }  
}
```

# Чтение инструкции из памяти

```
data = mem[pc];
```

# Чтение инструкции из памяти

```
data = mem[pc];  
Скорее,  
data = read_mem(pc);
```

# Чтение инструкции из памяти

```
data = mem[pc];
```

Скорее,

```
data = read_mem(pc);
```

И не забыть про преобразование адресов:

```
paddr = v2p(pc); // pc - vaddr
```

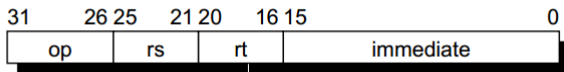
```
data = mem[paddr];
```

# Декодирование

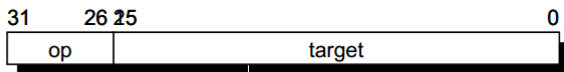
Перевод данных об инструкции из машинного представления (последовательность байт) во внутреннее (высокоуровневое), удобное для последующего использования

# Пример: MIPS

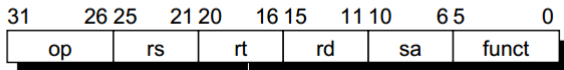
## I-Type (Immediate)



## J-Type (Jump)



## R-Type (Register)



MIPS Technologies Inc. MIPS32 4K™ Processor Core Family Software User's Manual - 2002

## Пример: код 1

```
#define BITS(v, s, e) (v >> s) & ((1 << (e-s+1)) - 1)

typedef struct decode {
    uint32_t op;
    uint32_t rs;
    uint32_t rt;
    int32_t  imm;
    int32_t  tgt;
    /* ... */
} decode_t;

static inline
int32_t sign_extend(uint32_t v, int width)
    { /* ... */ }
```



## Пример: код 2

```
decode_t decode(uint32_t raw) {
    uint32_t op = BITS(raw, 26, 31);
    uint32_t rs = BITS(raw, 21, 25);
    uint32_t rt = BITS(raw, 16, 20);
    int32_t imm = sign_extend(BITS(raw, 0, 15));
    int32_t tgt = sign_extend(BITS(raw, 0, 25));
    /* ... */
    uint32_t funct = BITS(raw, 0, 5);

    return (decode_t){op, rs, rt, /* ... */, funct};
}
```

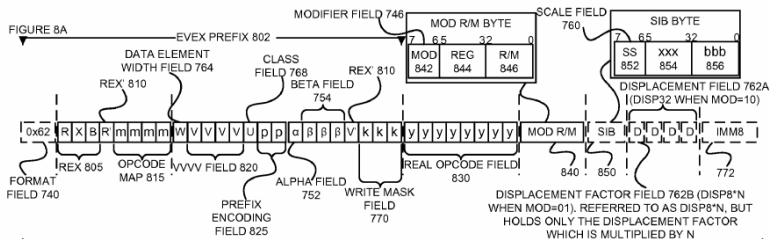
# Более сложный пример: Intel IA-64 2.3

FP Arithmetic	F1	8 - D	x	sf			f <sub>4</sub>		f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
Fixed Multiply Add	F2	E	x	x <sub>2</sub>			f <sub>4</sub>		f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
FP Select	F3	E	x				f <sub>4</sub>		f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
FP Compare	F4	4	r <sub>b</sub>	sf	r <sub>a</sub>	p <sub>2</sub>			f <sub>3</sub>		f <sub>2</sub>		t <sub>a</sub>	p <sub>1</sub>	qp
FP Class	F5	5			fc <sub>2</sub>	p <sub>2</sub>			fclass <sub>7c</sub>		f <sub>2</sub>		t <sub>a</sub>	p <sub>1</sub>	qp
FP Recip Approx	F6	0 - 1	q	sf	x	p <sub>2</sub>			f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
FP Recip Sqrt App	F7	0 - 1	q	sf	x	p <sub>2</sub>			f <sub>3</sub>				f <sub>1</sub>		qp
FP Min/Max/Pcmp	F8	0 - 1		sf	x	x <sub>6</sub>			f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
FP Merge/Logical	F9	0 - 1			x	x <sub>6</sub>			f <sub>3</sub>		f <sub>2</sub>		f <sub>1</sub>		qp
Convert FP to Fixed	F10	0 - 1		sf	x	x <sub>6</sub>					f <sub>2</sub>		f <sub>1</sub>		qp
Convert Fixed to FP	F11	0			x	x <sub>6</sub>					f <sub>2</sub>		f <sub>1</sub>		qp
FP Set Controls	F12	0		sf	x	x <sub>6</sub>		omask <sub>7c</sub>		amask <sub>7b</sub>					qp
FP Clear Flags	F13	0		sf	x	x <sub>6</sub>									qp
FP Check Flags	F14	0	s	sf	x	x <sub>6</sub>									qp
Break	F15	0	i		x	x <sub>6</sub>									qp
Nop/Hint	F16	0	i		x	x <sub>6</sub>	y								qp
Break	X1	0	i	x <sub>3</sub>		x <sub>6</sub>									qp
Move Imm <sub>64</sub>	X2	6	i				imm <sub>9d</sub>		imm <sub>5c</sub>	i <sub>c</sub>	v <sub>c</sub>	imm <sub>7b</sub>		r <sub>1</sub>	qp
Long Branch	X3	C	i	d	wh								p	btype	qp
Long Call	X4	D	i	d	wh								p	b <sub>1</sub>	qp
Nop/Hint	X5	0	i	x <sub>3</sub>		x <sub>6</sub>	y								qp

40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

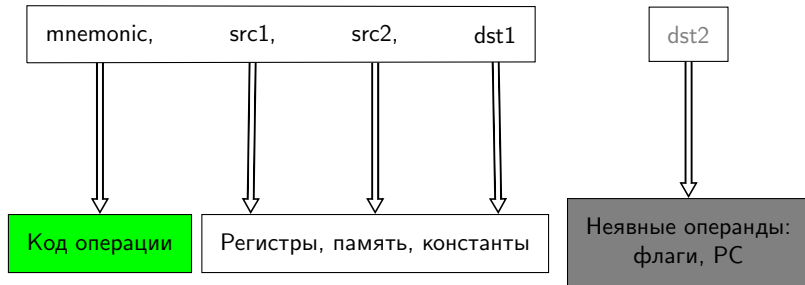
Intel Corporation. Intel® Itanium® Architecture Software Developer's Manual, p 3:296

# Пример посложнее: Intel IA-32



J.C.S. Adrian et al. Systems, Apparatuses, and Methods for Blending Two Source Operands into a Single Destination Using a Writemask. US Patent Application Publication. № 2012/0254588 A1

# Что извлекать из машинного кода инструкции



На выходе декодера:

- Успех, неуспех, недостаточно данных
- Для успеха: длина инструкции

# Декодирование

- Код декодера редко пишется вручную, чаще он генерируется по описанию.
- A5 YX 0Z 00  $\Rightarrow$  MOD RX, RY, RZ
- В общем случае: классическая задача построения синтаксического анализатора.
- На практике: специализированные инструменты и языки.

# Декодирование: суровая реальность

- Переменная длина инструкций. IA-32: от 8 до 120 бит. Сколько байт пытаться декодировать за один раз?
- Зависимость смысла от префикса, режима работы процессора. Пример: 0x40 - 0x4f в IA-32/Intel 64/AMD64
- Полное несоответствие какому-либо здравому смыслу

# Дизассемблирование

- Дизассемблирование — перевод инструкций из машинного представление в понятный человеку вид (мнемонику)
- (За)кодирование (encoding) — перевод инструкций из мнемонической записи в машинный код

# Исполнение

- Базовая единица — функция-эмулятор одной инструкции (service routine).
- Функция-эмуляторы пишутся на языке высокого уровня → переносимость кода между хозяйскими платформами, компиляторами.



## Пример: ADD reg reg reg

```
typedef struct {uint32_t regs[16];} cpu_t;
```

```
void add32_rrr(cpu_t *cpu, int src1, int src2, int dst) {  
    cpu->regs[dst] = cpu->regs[src1] + cpu->regs[src2];  
}
```

## Пример: ADD reg reg reg

```
typedef struct {uint32_t regs[16];} cpu_t;

void add32_rrr(cpu_t *cpu, int src1, int src2, int dst) {
    cpu->regs[dst] = cpu->regs[src1] + cpu->regs[src2];

    cpu->z_flag = cpu->regs[dst] == 0;
    cpu->n_flag = cpu->regs[dst] & (1 << 31);
    cpu->o_flag = cpu->regs[dst] <
        MAX(cpu->regs[src1], cpu->regs[src2]);
    cpu->c_flag = calc_c_flag(cpu->regs[src1],
        cpu->regs[src2]);
}
```

## Пример посложнее: IA-32 CALL

Figure 1 shows the results of the regression analysis. The dependent variable is the number of employees who are not satisfied with their working conditions. The independent variable is the number of employees who are satisfied with their working conditions. The regression line shows a negative correlation between the two variables. The regression equation is  $y = -0.15x + 1.5$ , where  $y$  is the number of employees who are not satisfied with their working conditions and  $x$  is the number of employees who are satisfied with their working conditions. The regression coefficient is  $-0.15$ , which indicates that for every unit increase in the number of employees who are satisfied with their working conditions, the number of employees who are not satisfied with their working conditions decreases by  $0.15$  units. The intercept is  $1.5$ , which indicates that if the number of employees who are satisfied with their working conditions is zero, the number of employees who are not satisfied with their working conditions is  $1.5$  units.

593

```

1  # Import the data
2  # Import the data
3  # Import the data
4  # Import the data
5  # Import the data
6  # Import the data
7  # Import the data
8  # Import the data
9  # Import the data
10 # Import the data
11 # Import the data
12 # Import the data
13 # Import the data
14 # Import the data
15 # Import the data
16 # Import the data
17 # Import the data
18 # Import the data
19 # Import the data
20 # Import the data
21 # Import the data
22 # Import the data
23 # Import the data
24 # Import the data
25 # Import the data
26 # Import the data
27 # Import the data
28 # Import the data
29 # Import the data
30 # Import the data
31 # Import the data
32 # Import the data
33 # Import the data
34 # Import the data
35 # Import the data
36 # Import the data
37 # Import the data
38 # Import the data
39 # Import the data
40 # Import the data
41 # Import the data
42 # Import the data
43 # Import the data
44 # Import the data
45 # Import the data
46 # Import the data
47 # Import the data
48 # Import the data
49 # Import the data
50 # Import the data
51 # Import the data
52 # Import the data
53 # Import the data
54 # Import the data
55 # Import the data
56 # Import the data
57 # Import the data
58 # Import the data
59 # Import the data
60 # Import the data
61 # Import the data
62 # Import the data
63 # Import the data
64 # Import the data
65 # Import the data
66 # Import the data
67 # Import the data
68 # Import the data
69 # Import the data
70 # Import the data
71 # Import the data
72 # Import the data
73 # Import the data
74 # Import the data
75 # Import the data
76 # Import the data
77 # Import the data
78 # Import the data
79 # Import the data
80 # Import the data
81 # Import the data
82 # Import the data
83 # Import the data
84 # Import the data
85 # Import the data
86 # Import the data
87 # Import the data
88 # Import the data
89 # Import the data
90 # Import the data
91 # Import the data
92 # Import the data
93 # Import the data
94 # Import the data
95 # Import the data
96 # Import the data
97 # Import the data
98 # Import the data
99 # Import the data
100 # Import the data

```

594

1. **Identify the main components of the system.**  
 2. **Define the system boundaries.**  
 3. **Identify the inputs and outputs of the system.**  
 4. **Identify the internal processes of the system.**  
 5. **Identify the external environment of the system.**  
 6. **Identify the stakeholders of the system.**  
 7. **Identify the constraints of the system.**  
 8. **Identify the risks of the system.**  
 9. **Identify the opportunities of the system.**  
 10. **Identify the challenges of the system.**  
 11. **Identify the goals of the system.**  
 12. **Identify the objectives of the system.**  
 13. **Identify the strategies of the system.**  
 14. **Identify the tactics of the system.**  
 15. **Identify the techniques of the system.**  
 16. **Identify the tools of the system.**  
 17. **Identify the methods of the system.**  
 18. **Identify the procedures of the system.**  
 19. **Identify the processes of the system.**  
 20. **Identify the functions of the system.**  
 21. **Identify the roles of the system.**  
 22. **Identify the responsibilities of the system.**  
 23. **Identify the tasks of the system.**  
 24. **Identify the activities of the system.**  
 25. **Identify the events of the system.**  
 26. **Identify the incidents of the system.**  
 27. **Identify the problems of the system.**  
 28. **Identify the issues of the system.**  
 29. **Identify the concerns of the system.**  
 30. **Identify the questions of the system.**  
 31. **Identify the answers of the system.**  
 32. **Identify the solutions of the system.**  
 33. **Identify the results of the system.**  
 34. **Identify the outcomes of the system.**  
 35. **Identify the impacts of the system.**  
 36. **Identify the effects of the system.**  
 37. **Identify the consequences of the system.**  
 38. **Identify the implications of the system.**  
 39. **Identify the significance of the system.**  
 40. **Identify the importance of the system.**  
 41. **Identify the value of the system.**  
 42. **Identify the worth of the system.**  
 43. **Identify the merit of the system.**  
 44. **Identify the advantage of the system.**  
 45. **Identify the benefit of the system.**  
 46. **Identify the gain of the system.**  
 47. **Identify the profit of the system.**  
 48. **Identify the success of the system.**  
 49. **Identify the achievement of the system.**  
 50. **Identify the accomplishment of the system.**  
 51. **Identify the fulfillment of the system.**  
 52. **Identify the realization of the system.**  
 53. **Identify the attainment of the system.**  
 54. **Identify the completion of the system.**  
 55. **Identify the conclusion of the system.**  
 56. **Identify the end of the system.**  
 57. **Identify the close of the system.**  
 58. **Identify the finish of the system.**  
 59. **Identify the termination of the system.**  
 60. **Identify the cessation of the system.**  
 61. **Identify the discontinuation of the system.**  
 62. **Identify the discontinuity of the system.**  
 63. **Identify the interruption of the system.**  
 64. **Identify the disturbance of the system.**  
 65. **Identify the disruption of the system.**  
 66. **Identify the interference of the system.**  
 67. **Identify the obstruction of the system.**  
 68. **Identify the hindrance of the system.**  
 69. **Identify the impediment of the system.**  
 70. **Identify the obstacle of the system.**  
 71. **Identify the barrier of the system.**  
 72. **Identify the block of the system.**  
 73. **Identify the stop of the system.**  
 74. **Identify the halt of the system.**  
 75. **Identify the pause of the system.**  
 76. **Identify the break of the system.**  
 77. **Identify the crack of the system.**  
 78. **Identify the split of the system.**  
 79. **Identify the divide of the system.**  
 80. **Identify the separate of the system.**  
 81. **Identify the detach of the system.**  
 82. **Identify the disconnect of the system.**  
 83. **Identify the disengage of the system.**  
 84. **Identify the disjoin of the system.**  
 85. **Identify the disjoin of the system.**  
 86. **Identify the disjoin of the system.**  
 87. **Identify the disjoin of the system.**  
 88. **Identify the disjoin of the system.**  
 89. **Identify the disjoin of the system.**  
 90. **Identify the disjoin of the system.**  
 91. **Identify the disjoin of the system.**  
 92. **Identify the disjoin of the system.**  
 93. **Identify the disjoin of the system.**  
 94. **Identify the disjoin of the system.**  
 95. **Identify the disjoin of the system.**  
 96. **Identify the disjoin of the system.**  
 97. **Identify the disjoin of the system.**  
 98. **Identify the disjoin of the system.**  
 99. **Identify the disjoin of the system.**  
 100. **Identify the disjoin of the system.**

595

[illegible]

596

[illegible]

597

[illegible]

598

[illegible]

599

<b>Question</b>		<p>At a certain temperature, the following reaction is spontaneous:</p> $2\text{Fe}^{3+}(\text{aq}) + 3\text{Zn}(\text{s}) \rightarrow 2\text{Fe}(\text{s}) + 3\text{Zn}^{2+}(\text{aq})$ <p>Which of the following is true?</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Response(s)</b>	<b>Correct Answer(s)</b>	<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>
<b>Helping</b>		<p>At least one of the half-reactions is spontaneous.</p> <p>At least one of the half-reactions is non-spontaneous.</p> <p>The standard cell potential is positive.</p> <p>The standard cell potential is negative.</p> <p>The standard cell potential is zero.</p>

600

## Запись результата в память

```
write_mem(cpu, dst_addr, data);
```

# Запись результата в память

```
write_mem(cpu, dst_addr, data);
```

- Невыровненный адрес
- Граница страниц
- Запись в память «только для чтения»
- Исключения для всего или части записи/чтения

# Уточнённый цикл работы процессора



# Классификация

- Interrupts (термин из документации IA-64) — вмешательство, перерыв, приостановка
- Exception — синхронное исключение, без повторения текущей инструкции
- Fault — синхронное, с повторением текущей инструкции
- Trap — синхронное, без повторения, намеренно вызванное
- Interrupt — внешнее асинхронное прерывание
- Abort — внешнее асинхронное с отсутствием информации о точке возврата

# Продвижение PC

- Для большинства команд: увеличение счетчика на длину обработанной инструкции.  
`cpu->pc += instr_length;`



# Продвижение PC

- Для большинства команд: увеличение счетчика на длину обработанной инструкции.  
`cpu->pc += instr_length;`  
Бывают исключения: REP MOVSB в IA-32.

# Продвижение PC

- Для большинства команд: увеличение счетчика на длину обработанной инструкции.  
`cpu->pc += instr_length;`  
Бывают исключения: REP MOVS в IA-32.
- Явное изменение PC — команды управления исполнением:
  - (Без)условные (не)прямой прыжок/переход
  - Вызов/возврат из процедуры
  - Системный вызов

# Преимущества и недостатки интерпретации

- Пишется на языках высокого уровня: код переносим
- Простая структура: надёжность, расширяемость, переиспользование

# Преимущества и недостатки интерпретации

- Пишется на языках высокого уровня: код переносим
- Простая структура: надёжность, расширяемость, переиспользование
- (Очень) низкая скорость работы

# Куда тратится время?

```
start: interruption = false;
while (!interruption) {
    raw_code = fetch(PC);
    (opcode, operands) = decode(raw_code); // <-- здесь
    switch (opcode) { // <-- здесь
        case opcode1:
            func1(operands); PC++; break;
        case opcode2:
            func2(operands); PC++; break;
        /*...*/
    }
}
handle_interruption();
goto start;
```

# Сцепленная интерпретация (threaded interpretation)

Вместо возвращения к началу цикла «прыгаем» прямо на исполнение следующей инструкции

```
func0: /* simulate instr0 */; PC++;  
    next_opcode = decode(fetch(PC));  
    goto func_ptr[next_opcode];  
func1: /* simulate instr1 */; PC++;  
    next_opcode = decode(fetch(PC));  
    goto func_ptr[next_opcode];  
func2: /* simulate instr2 */; PC++;  
    next_opcode = decode(fetch(PC));  
    goto func_ptr[next_opcode];
```

<http://stackoverflow.com/questions/11227809/>

[why-is-processing-a-sorted-array-faster-than-an-unsorted-array](#)

# Кэширующая интерпретация

- В большинстве случаев в код гостевого приложения неизменен
- Велика вероятность того, что инструкции с некоторыми PC будут исполнены много раз (*задача*)
- Зачем каждый раз их декодировать?
- Заводим таблицу соответствия «адрес инструкции → декодированный результат»

# Кэширующая интерпретация

```
while (!interruption) {  
    if (operation = cache[PC]); // короткий путь  
    else { // не в кэше, длинный путь  
        operation = decode(fetch(PC));  
        cache[PC] = operation; // на будущее  
    }  
    switch (operation) {  
        /* ... */  
    }  
}
```






# Кэширующая интерпретация

- Ёмкость любого кэша ограничена, старые данные надо выбрасывать
- Необходимо следить за неизменностью исходного кода, иначе сохранённое соответствие будет неверным

# Итоги

- Фазы исполнения: F, D, E, W, A
- Decoder, disassembler, encoder
- Переключаемый (switched) И.
- Сцепленный (threaded) И.
- Кэширующий И.
- Ситуации: interrupt, trap, exception, fault, abort

# Литература I

-  D. Mihoka , S. Shwartsman. Virtualization Without Direct Execution or Jitting: Designing a Portable Virtual Machine Infrastructure <http://bochs.sourceforge.net/>
-  Fredrik Larsson, Peter Magnusson, Bengt Werner. SimGen: Development of Efficient Instruction Set Simulators <ftp://ftp.sics.se/pub/SICS-reports/Reports/SICS-R--97-03--SE.ps.Z>
-  Yair Lifshitz, Robert Cohn, Inbal Livni, Omer Tabach, Mark Charney, Kim Hazelwood. Zsim: A Fast Architectural Simulator for ISA Design-Space Exploration <http://www.cs.virginia.edu/kim/docs/wish11zsim.pdf>

## Литература II



Префиксы в системе команд IA-32

<http://habrahabr.ru/company/intel/blog/200598/>



Программная симуляция микропроцессора. Коробка передач

<http://habrahabr.ru/company/intel/blog/202926/>

# На следующей лекции

Моделирование архитектурного состояния

# Спасибо за внимание!

*Замечание:* все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.