Верификация микропроцессоров и программно-аппаратных комплексов

Спецсеминар «Корректность программ»

к.ф.-м.н. А.С. Камкин, в.н.с. ИСП РАН
kamkin@ispras.ru

Корректность и надежность

- Корректность – соответствие системы налагаемым на нее требованиям
  - *Минимизация ошибок проектирования*
- Надежность – способность системы сохранять работоспособное состояние
  - *Минимизация сбоев при эксплуатации*
- Корректность $\vee$ $\neg$Надежность
  - *Ошибки проектирования – частая причина сбоев*

Цена ошибки проектирования

Вертолет Чинук CH-47, 1994
система управления двигателем

Боинг 757 (рейс 965), 1995
отображение информации

- Человеческие жизни
- > 100 млн долларов
- Потеря репутации

Комплекс ПВО Пэтриот, 1991
система наведения ракет

Ракета-носитель Ариан 5, 1996
несовместимость SW/HW

Ошибки в микропроцессорах

- Известные примеры ошибок
  - *Ошибка в реализации команды FDIV (Pentium, 1994 г.)*
    - Цена ошибки – 475 000 000 $ (замена микросхем)
  - *Ошибка в реализации кэширования (AMD Phenom, 2007 г.)*
- Общее число ошибок
  - *Ошибки, обнаруженные при проектировании*
    - 7855 ошибок (Pentium 4, 2000 г.)
  - *Ошибки, обнаруженные при эксплуатации*
    - 153 ошибки (Core i7, 2011 г.)

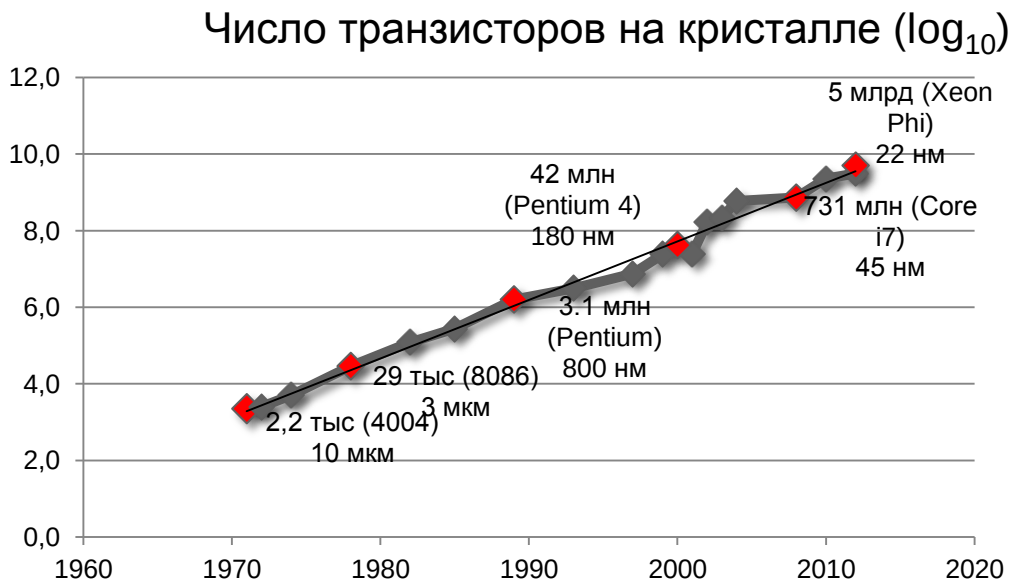
Сложность микропроцессоров

Микропроцессор

- > 10^9 транзисторов
- > 10^7 строк на Verilog
- ≈ 4-16 ядер

Проектирование

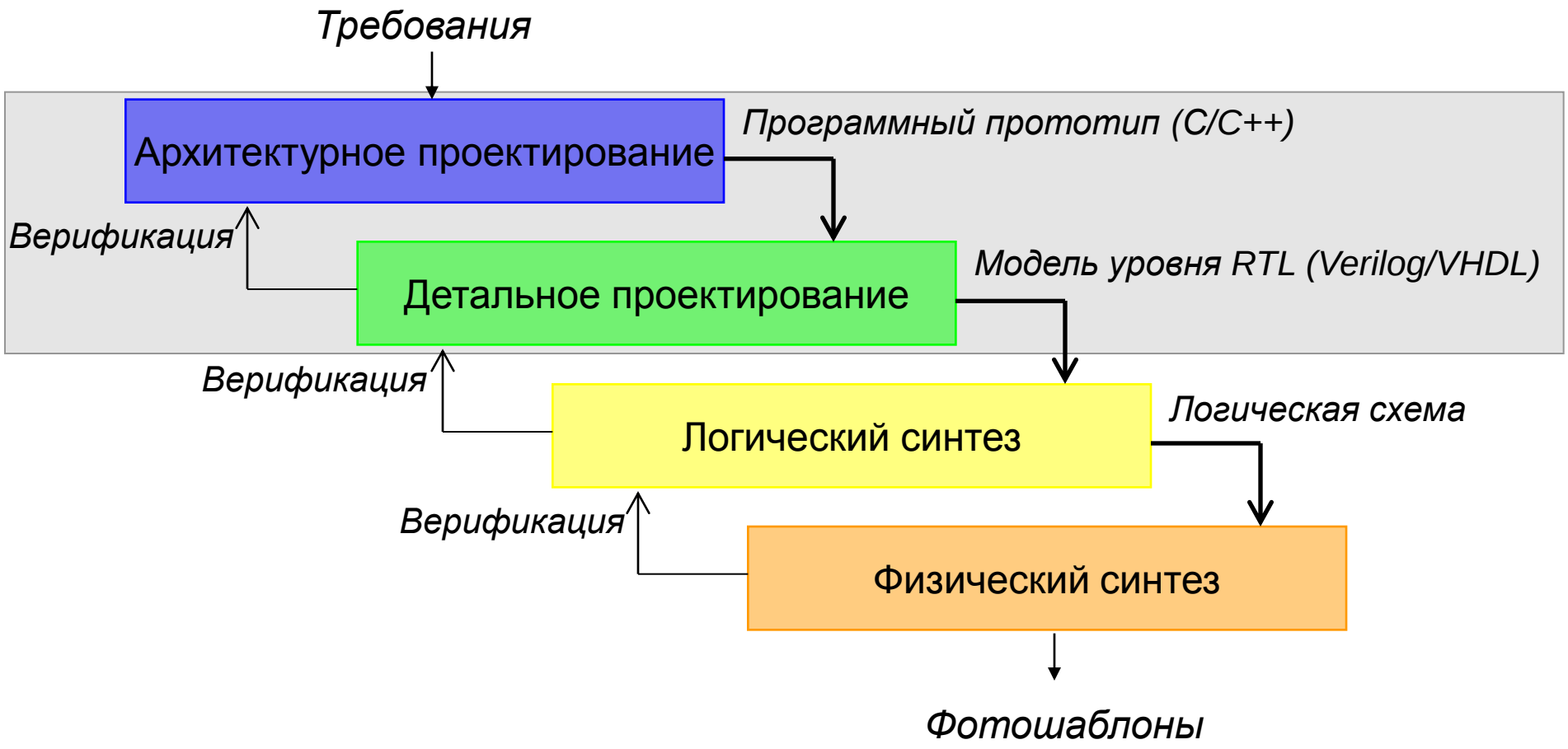
- > 500 разработчиков
- ≈ 3-4 года работы
- > 100 млн долларов



Верификация – 50-80% затрат на проектирование микропроцессора

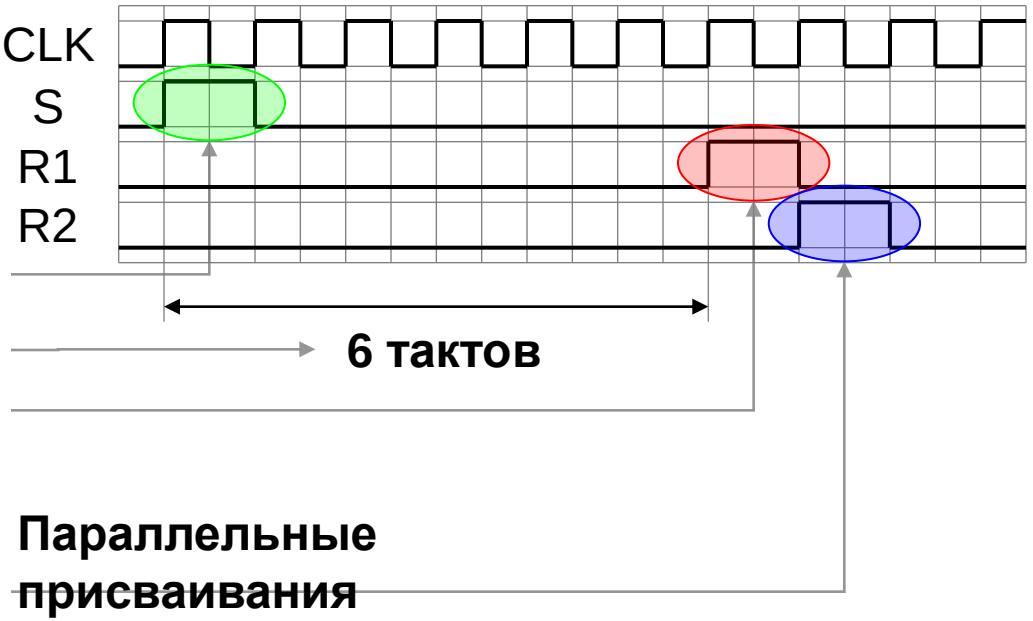
- > 10 000 ошибок проектирования (в спецификация и исходном коде)
- > 100 инженеров-верификаторов (разработка моделей и тестов, верификация)
- > 1000 компьютеров (круглосуточное моделирование и прогон тестов)

Проектирование микропроцессоров



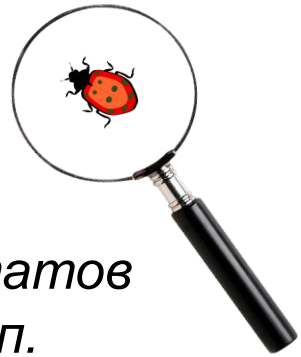
Языки описания аппаратуры (HDL)

```
// Код Verilog
always @(posedge CLK)
begin
    if(state == 2'h0 && S) begin
        state <= 2'h1;
        delay <= 3'h6;
    end
    else if(state == 2'h1) begin
        delay <= delay - 3'h1;
        if(delay == 3'h0) begin
            state <= 2'h2;
            R1    <= 1'h1;
        end
    end
    else if(state == 2'h2) begin
        state <= 2'h3;
        R1    <= 1'h0;
        R2    <= 1'h1;
    end
    else if(state == 2'h3) begin
        state <= 2'h0;
        R2    <= 1'h0;
    end
end
```



Поведение цифровой аппаратуры

Верификация микропроцессоров

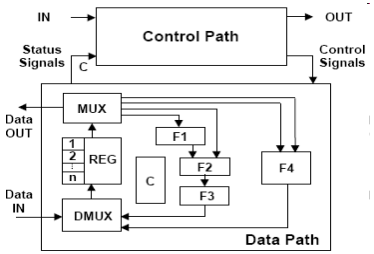


- Верификация – это проверка корректности, соответствия реализации ее спецификации
  - *Экспертиза – непосредственный анализ результатов проектирования: инспекция кода программы и т.п.*
  - *Статический анализ кода – автоматическая проверка заданных правил, поиск ошибок по шаблонам и т.п.*
  - *Тестирование – оценка корректности системы по результатам ее работы в некоторых ситуациях*
  - *Формальная верификация – строгое доказательство корректности формальной модели системы*

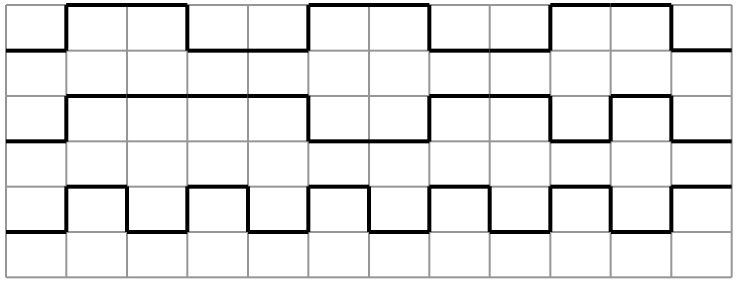
Модульная и системная верификация

Модульная верификация

Проверяется модель отдельного модуля

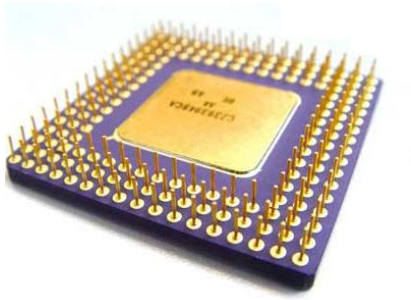


через входные и выходные сигналы



Системная верификация

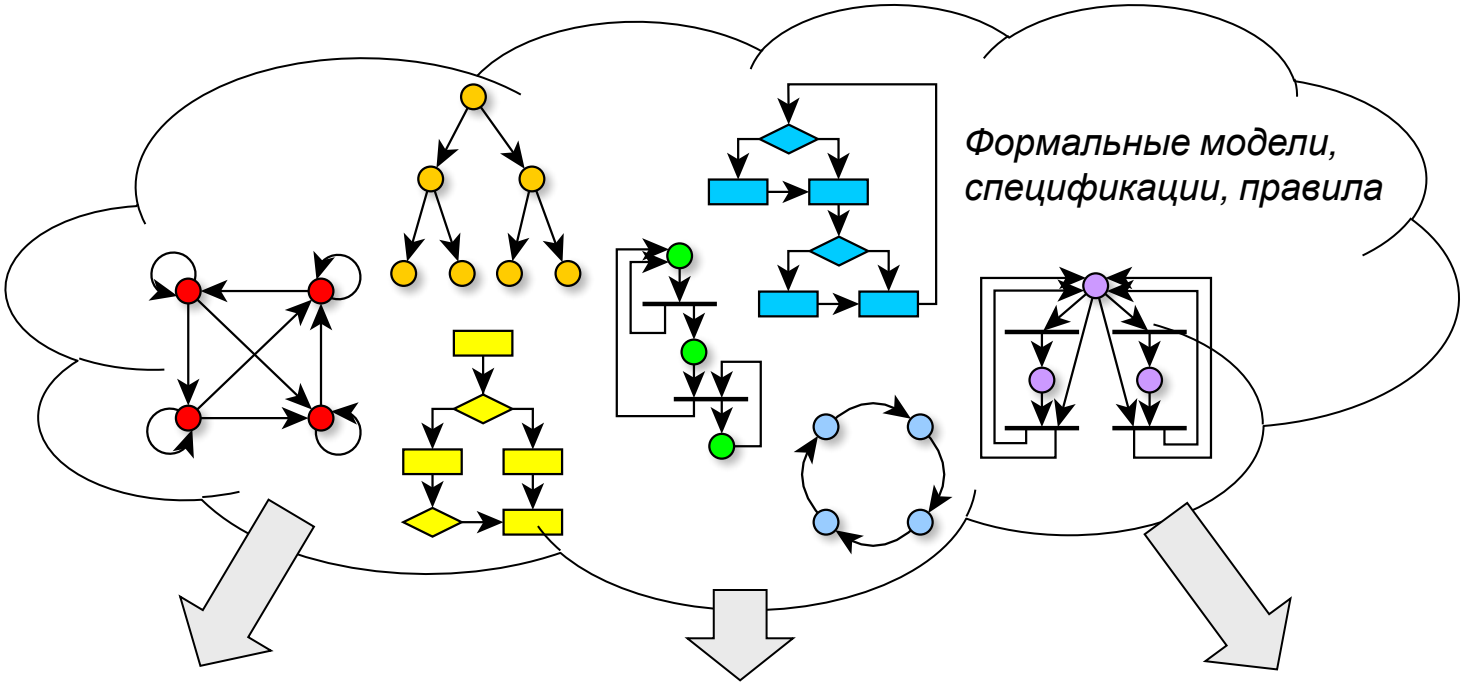
Проверяется модель всего микропроцессора



с помощью тестовых программ

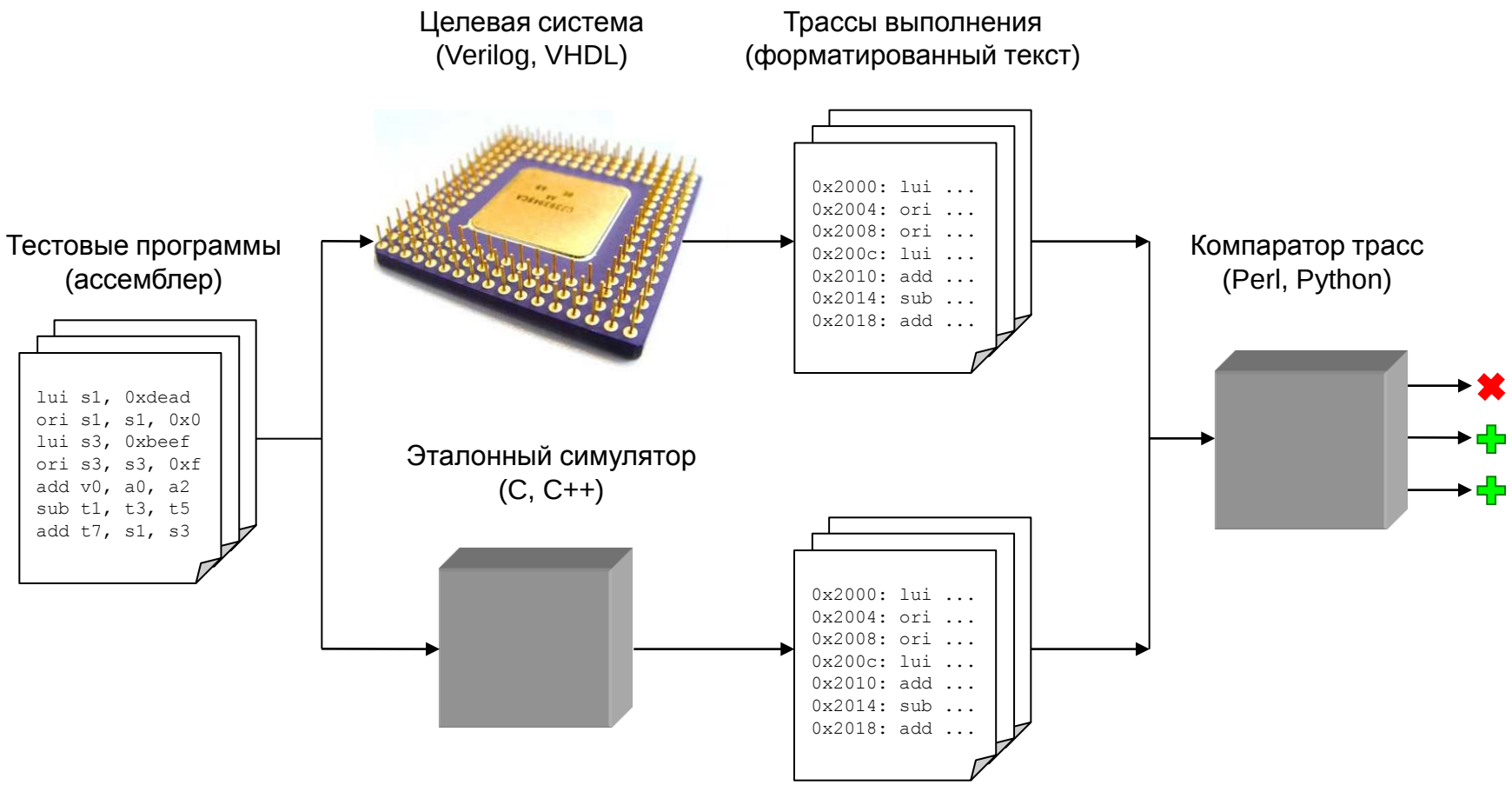
```
lui   s1, 0x2779
ori   s1, s1, 0xc8b9
lui   s3, 0x4ee
ori   s3, s3, 0xf798
add   v0, a0, a2
sub   t1, t3, t5
add   t7, s1, s3
```

Верификация на основе моделей

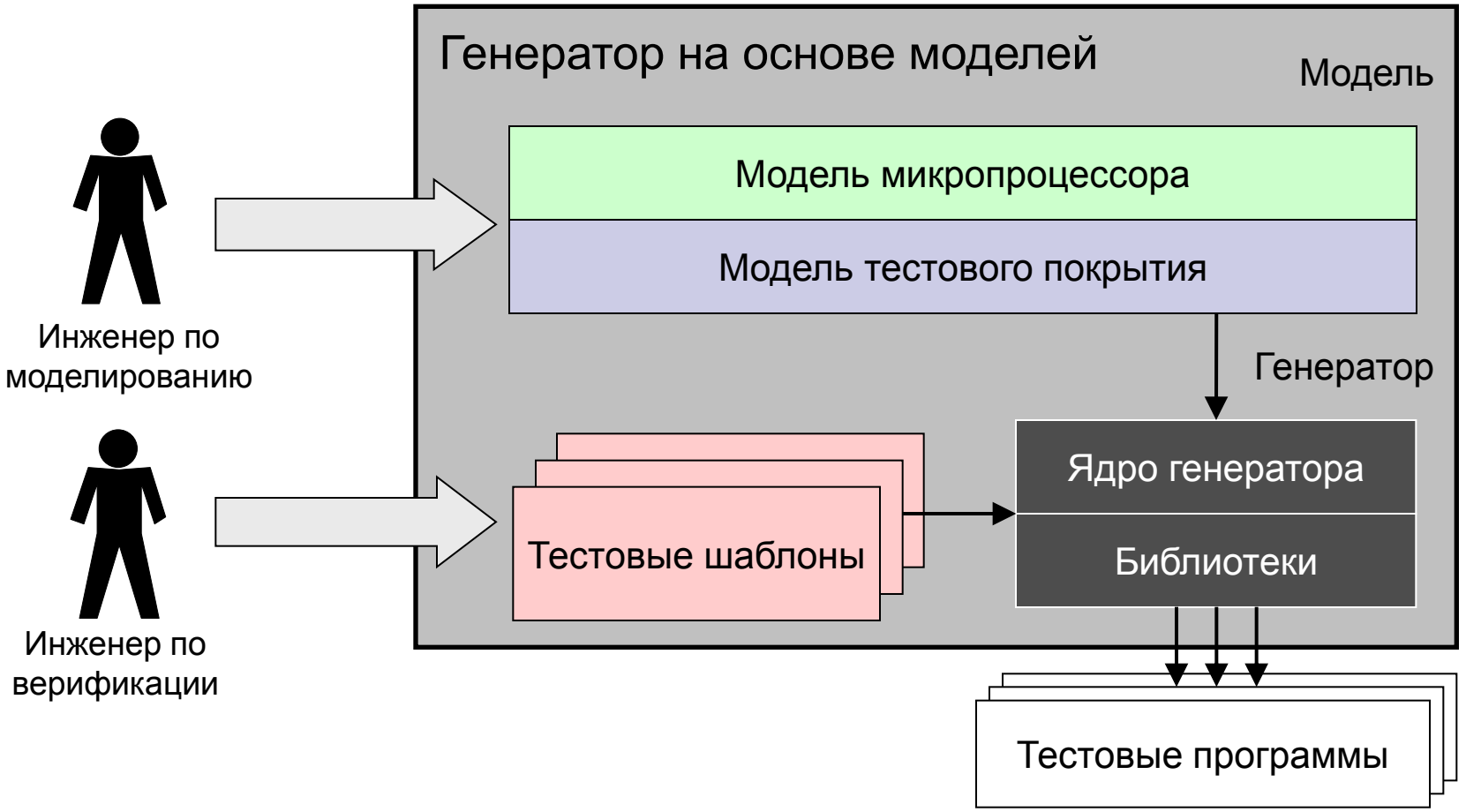


Генерация тестов	Проверка поведения	Проверка моделей	Дедуктивный анализ	Статический анализ
конечные автоматы *цепи Маркова*	*пред- и постусловия* *исполнимые модели*	*системы переходов* *временные логики*	*аксиомы* *правила вывода*	*потоки управления* *шаблоны ошибок*

Генерация тестовых программ



Генерация программ на основе моделей



Проект MicroTESK

- **Micro**processor **TE**sting and **S**pecification **K**it
  - *Спецификация системы команд на nML*
  - *Спецификация подсистемы памяти на MMUSL*
  - *Описание шаблонов тестовых программ на Ruby*
  - *Комбинаторная генерация тестовых программ*
  - *Генерация тестовых данных на основе ограничений*
- http://forge.ispras.ru/projects/microtesk

Спецификация системы команд

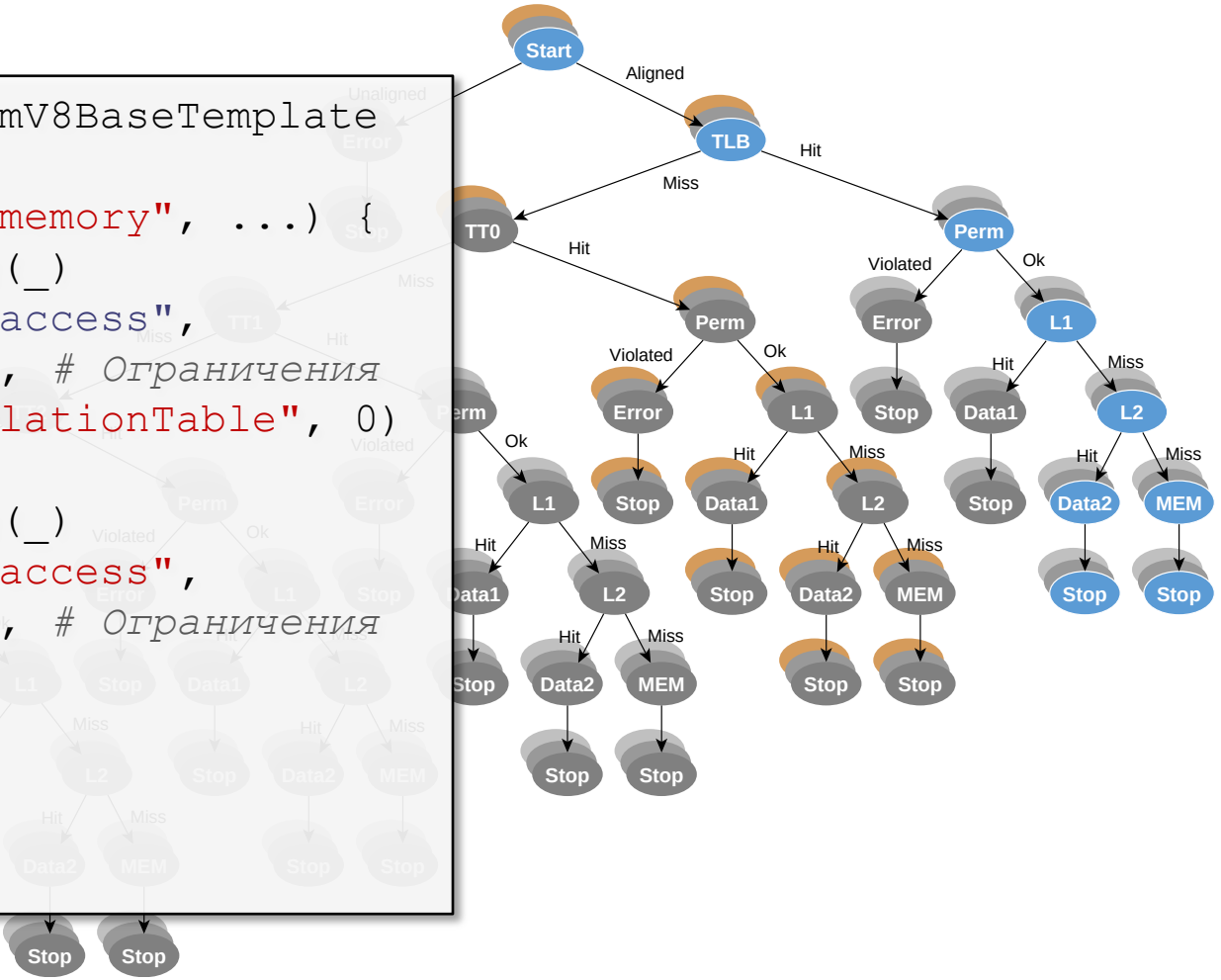
```
// Команда загрузки данных в регистр (Load-Acquire Register)
op LDAR(rt: REG, rn: REG)
  syntax =   // Ассемблерный формат
    format("ldar %s, [%s, #0]", rt.syntax, rn.syntax)
  image  =   // Бинарная кодировка
    format("11001000110111111111111%5s%5s", rn.image, rt.image)
  action = { // Семантика команды
    va = rn; // Чтение виртуального адреса из регистра
    ...
    if va<...> != 0 then
      exception("AlignmentFault");
    endif;
    temp = MEM[va >> 3]; // Доступ к памяти (обращение к MMU)
    rt   = temp<...>;    // Запись данных в регистр
      ...
    endif;
  }
```

Спецификация системы команд

```
mmu MMU(va: VA) = (data: 64)
  read = { // Операция чтения данных из памяти
    if va.acctype != AccType_PTW then
      pa = AArch64TranslateAddress(va); // Трансляция адреса
    else
      pa = SEG(va); // Неотображаемый адрес
    endif;
    if L1(pa).hit then ... else ...
      if va.acctype != AccType_PTW then
        temp = AArch64MemSingleRead(va);
      else
        temp = MEM(pa);
      endif;
    ...
    endif;
    data = temp<...>; // Возвращаемые данные
  }
  write = { ... } // Операция записи данных в память
}
```

Описание тестовых шаблонов

```
class MmuTemplate < ArmV8BaseTemplate
  def run
    block(:engine => "memory", ...) {
      ldar reg(_), reg(_)
        do situation("access",
          miss ("TLB"), # Ограничения
          hit  ("TranslationTable", 0)
        end
      stlr reg(_), reg(_)
        do situation("access",
          hit  ("TLB"), # Ограничения
          miss ("L1"),
        end
    }
  end
end
```



Разработки ИСП РАН

- Инструменты верификации микропроцессоров
  - *MicroTESK – генерация тестовых программ*
  - *C++TESK – разработка тестовых систем*
  - *Retrascope – статический анализ HDL-описаний*
- Инструменты верификации системного ПО
  - *CTESK – разработка тестовых систем*
  - *LDV – формальная верификация драйверов Linux*
  - *KEDR – динамический анализ модулей ядра*

Исследования ИСП РАН

- Моделирование и спецификация аппаратуры
- Имитационная верификация на основе моделей
- Генерация тестовых программ на основе моделей
- Статический анализ HDL-описаний
- Обратная инженерия моделей аппаратуры
- Верификация реализаций протоколов когерентности
- Формальная верификация моделей аппаратуры

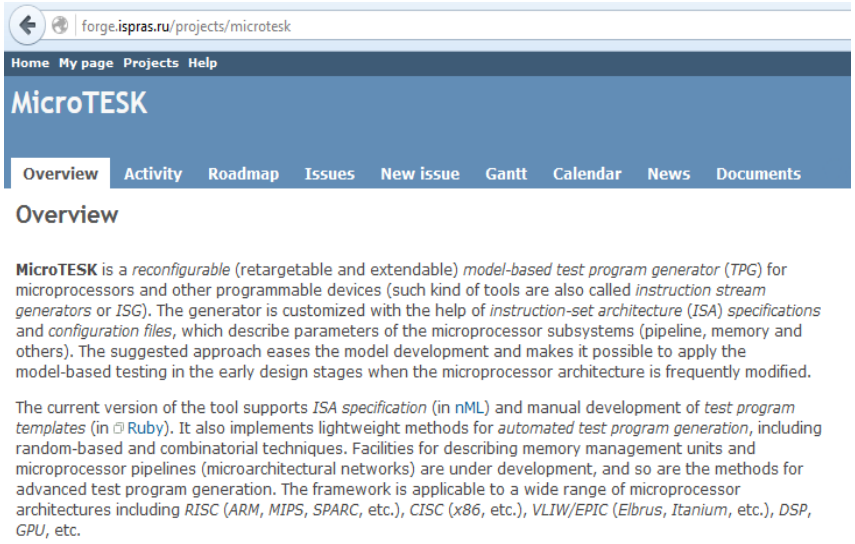
«Горячие» темы

- Online-генерация тестовых программ
- Обратная инженерия (абстракция) HDL-описаний
- Аппаратное ускорение солверов / пруверов
- Высокоуровневый синтез аппаратуры
- Формальная верификация аппаратуры и протоколов
- Поиск «закладок» в HDL-описаниях

Лекции и семинары ИСП РАН

- Дедуктивная верификация программ (ВМиК)
- Верификация программного обеспечения (МФТИ)
- Верификация моделей программ (ВМиК)
- Тестирование на основе моделей (ВМиК)
- Корректность программ (ВМиК, МФТИ)

Контакты



Александр Камкин: kamkin@ispras.ru

http://www.facebook.com/MicroTESK

Спасибо! Вопросы?

```
        .org 0x50000
_start: nop
        adr  x0, page_table
        msr  ttbr0_el1, x0
        movz x0, #0x8082, LSL #16
        movk x0, #0x3219, LSL #0
        msr  tcr_el1, x0
        isb  #0
        movz x0, #0x30c5, LSL #16
        movk x0, #0x0831, LSL #0
        msr  sctlr_el1, x0
        isb  #0
        movz x0, #0x0020, LSL #16
        movk x0, #0x4000, LSL #0
        ldar x1, [x0, #0]
        hlt  #0xf00d
```