



# Функциональная верификация микропроцессоров

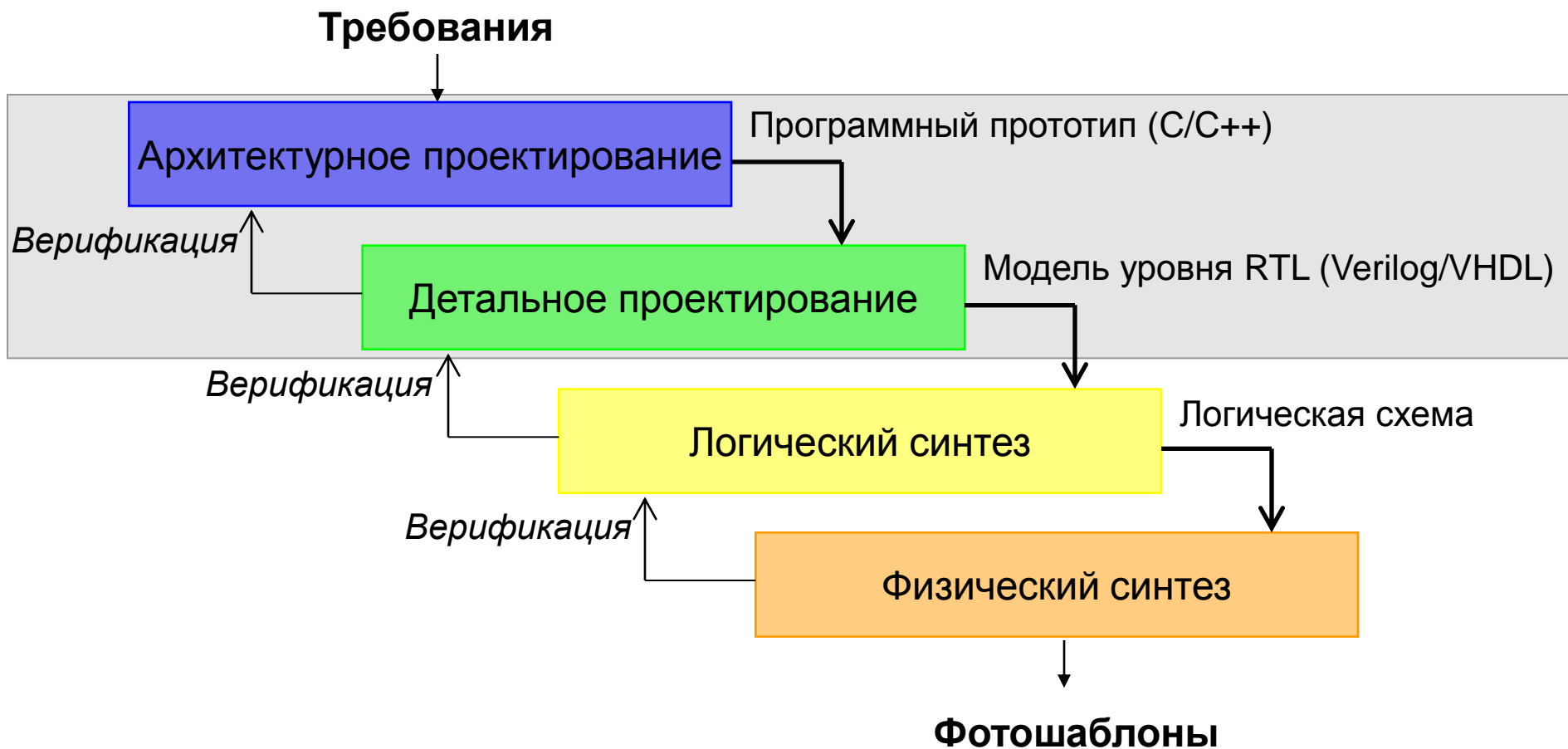
А.С. Камкин  
[kamkin@ispras.ru](mailto:kamkin@ispras.ru)



# Содержание лекции

- Введение
  - Проектирование микропроцессоров
  - Языки описания аппаратуры
  - Уровни верификации
- Основные подходы
  - Имитационная верификация (тестирование)
  - Генерация тестовых программ
  - Формальная верификация
- Проекты ИСП РАН
  - Проект C++TESK
  - Проект MicroTESK
  - Открытые проблемы

# Проектирование микропроцессоров



# Ошибки проектирования

Число ошибок  $\approx 10\ 000$  (Pentium 4)


Цена ошибки  $\approx 500\ 000\ 000$  \$ (Pentium)

Сложность микропроцессоров растет

9/9

0800 Anikan started  
 1000 " stopped - anikan ✓ { 1.2700 9.032 847 025  
 13:00 (032) MP-MC ~~1.982177000~~ 9.037 846 945 connect  
 (033) PRO 2 2.130476415 (033) 4.615925059(.2)  
 connect 2.130676415  
 Relays 6-2 in 033 failed special speed test  
 in factory " " 11,000 test.

1100 Started Cosine Tape (Sine check)  
 1525 Started Multi + Adder Test

1545  Relay #70 Panel F  
 (moth) in relay.

165/160 First actual case of bug being found.  
 700 Anikan started.  
 700 closed down.

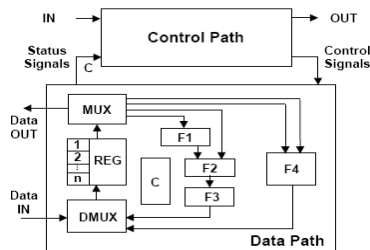
Relay 3145  
 Relay 3376



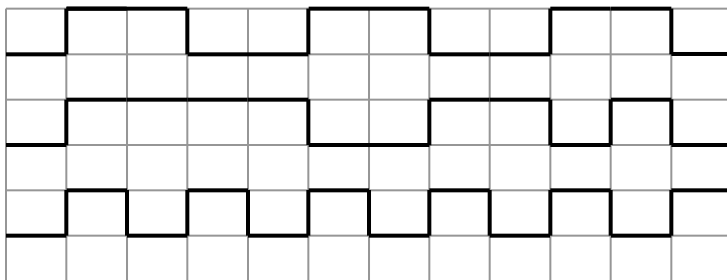
# Модульная и системная верификация

## Модульная верификация

Проверяется модель отдельного модуля

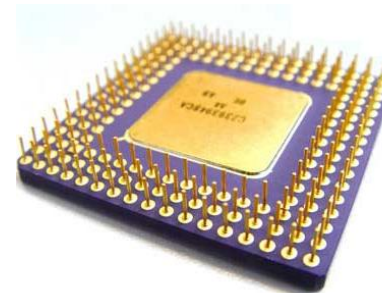


через входные и выходные сигналы



## Системная верификация

Проверяется модель всего микропроцессора

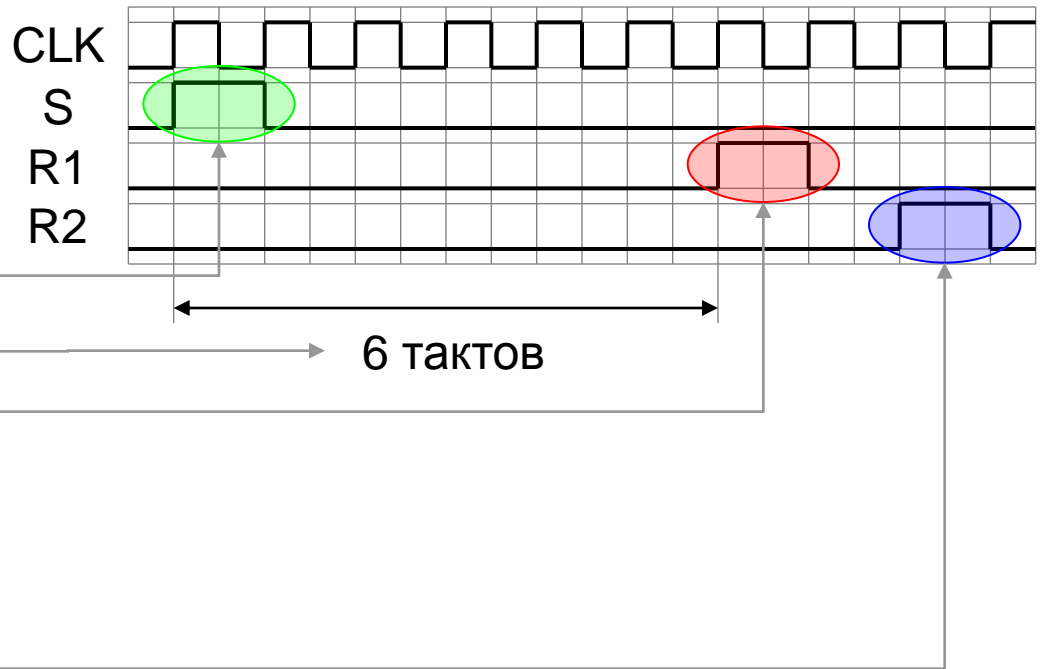


с помощью тестовых программ

```
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
```

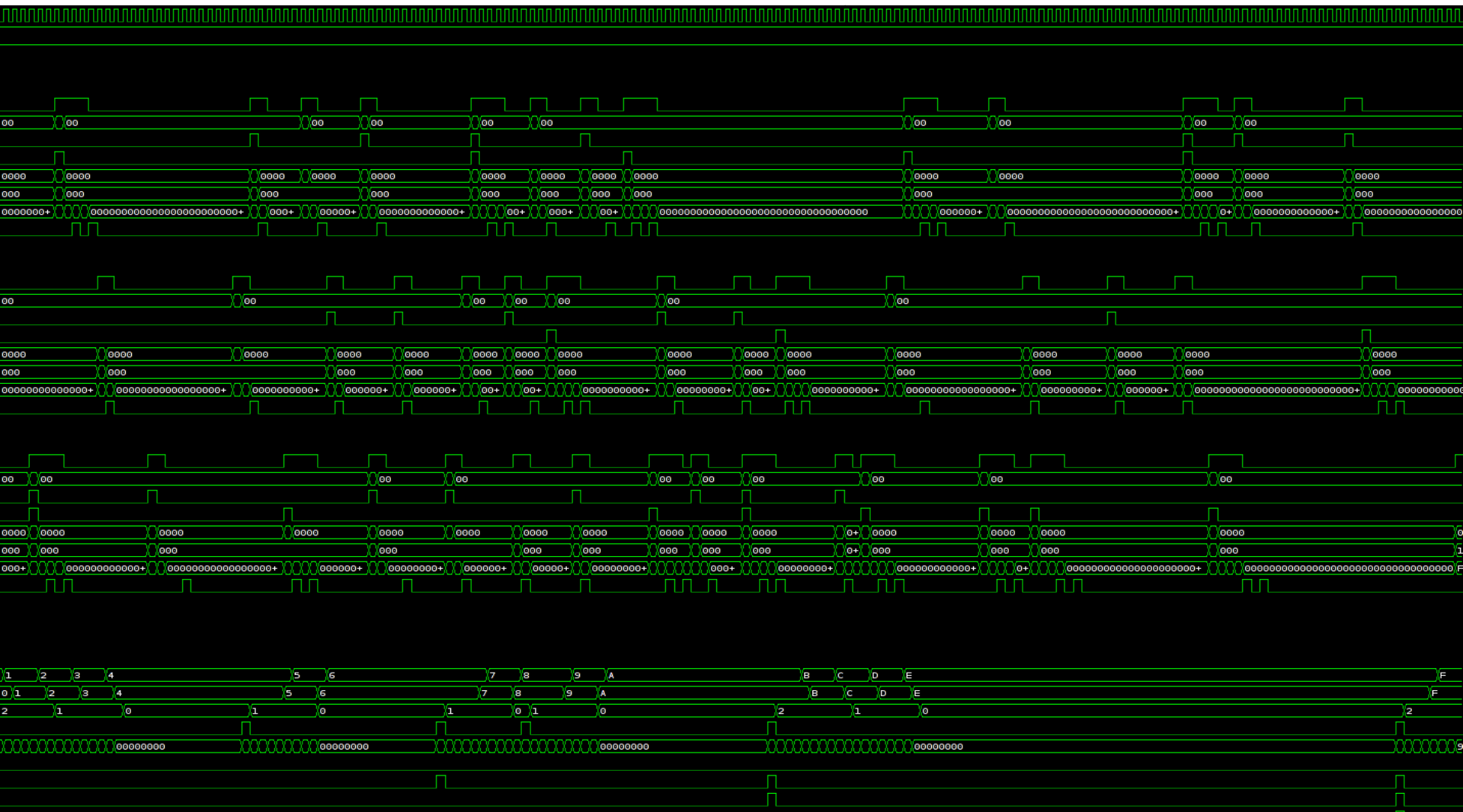
# Языки описания аппаратуры (HDL)

```
input S;  
output R1, R2;  
void design() {  
    while(true) {  
        wait(S);  
        delay(6);  
        R1 = 1;  
        delay(1);  
        R1 = 0;  
        delay(1);  
        R2 = 1;  
        delay(1);  
        R2 = 0;  
        V1 = 1;  
    }  
}
```



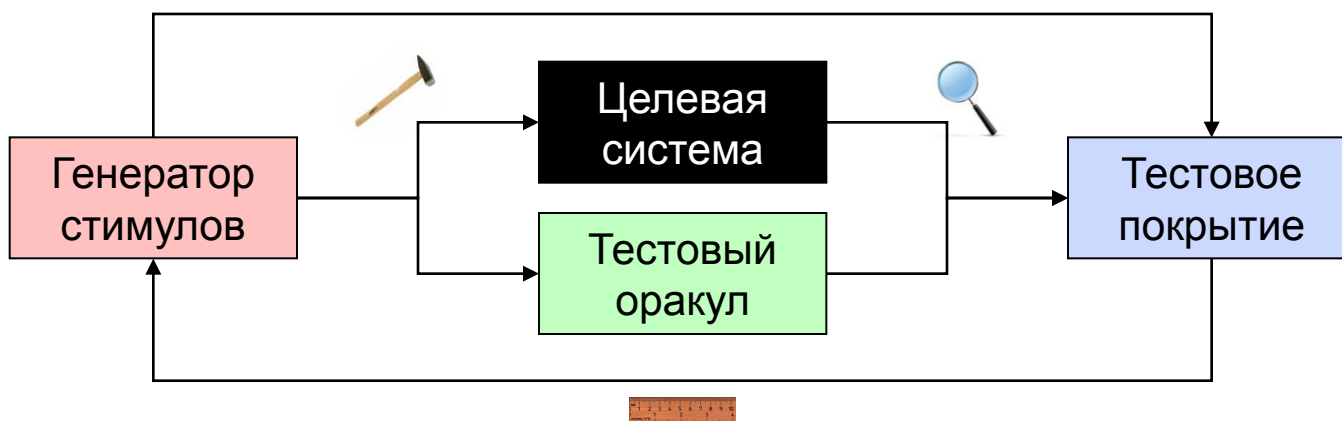
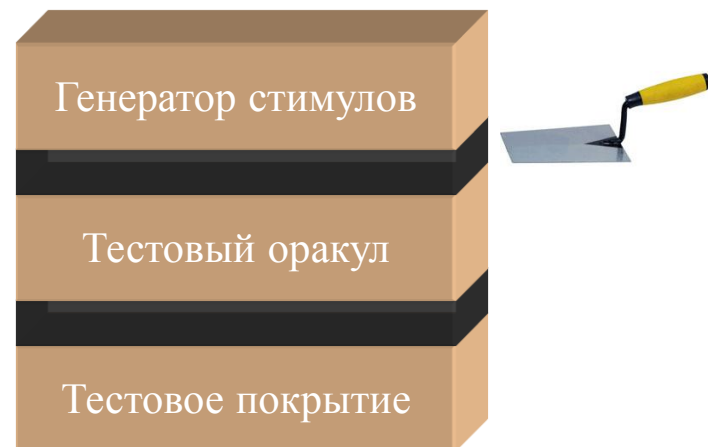
Одновременные присваивания

# Имитационная верификация



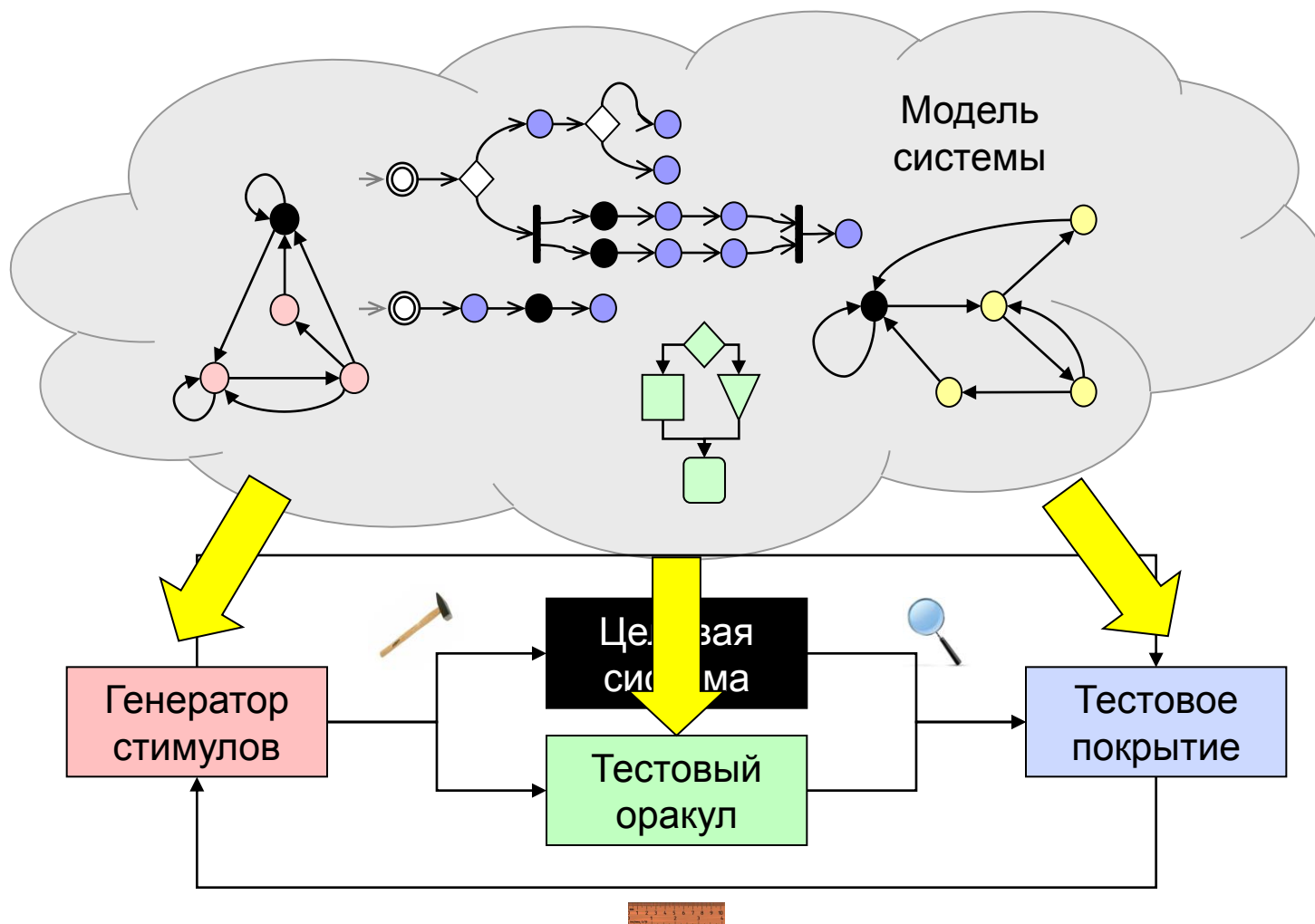
# Задачи имитационной верификации

- Генерация стимулов
- Проверка реакций
- Оценка покрытия

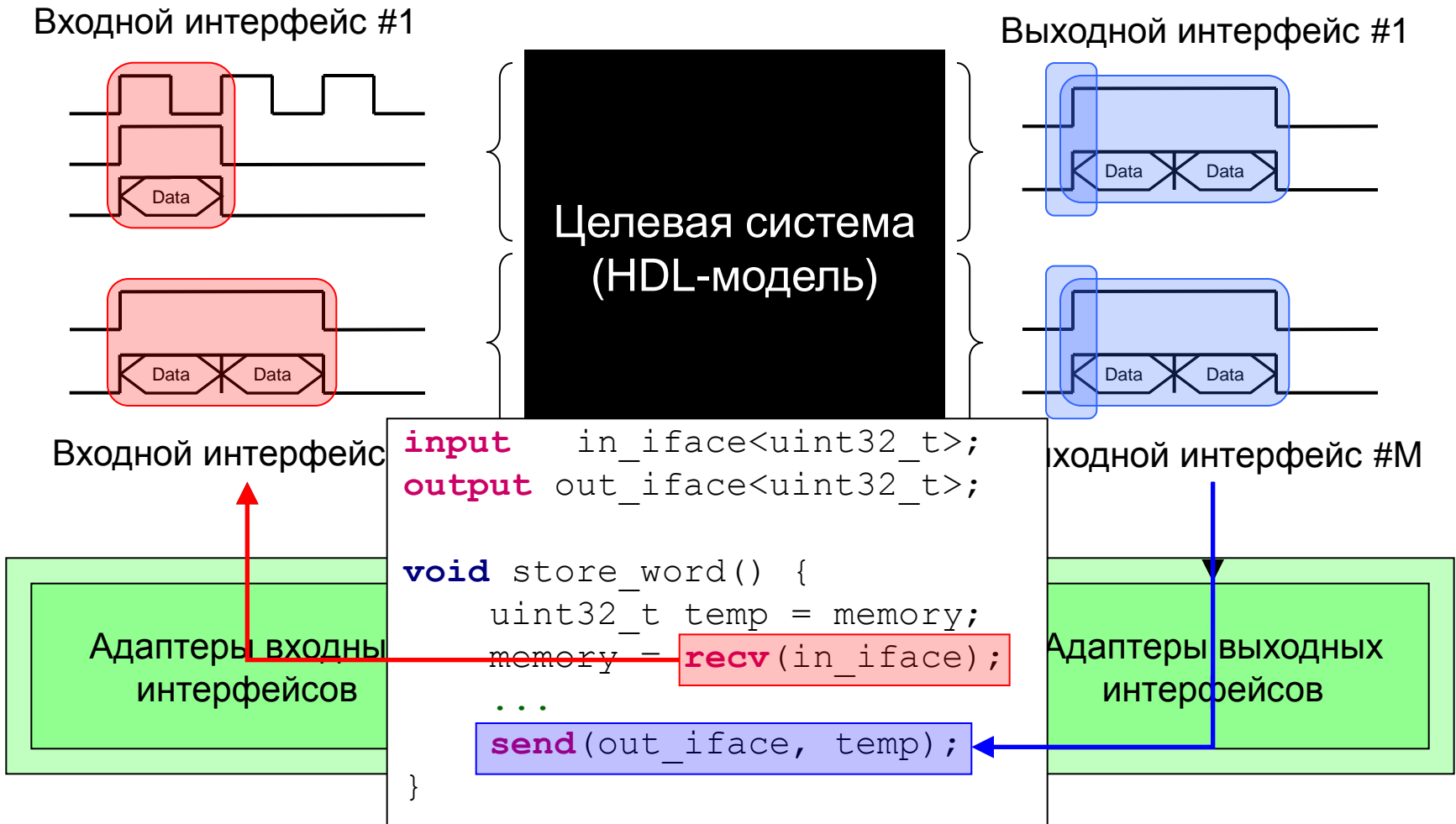




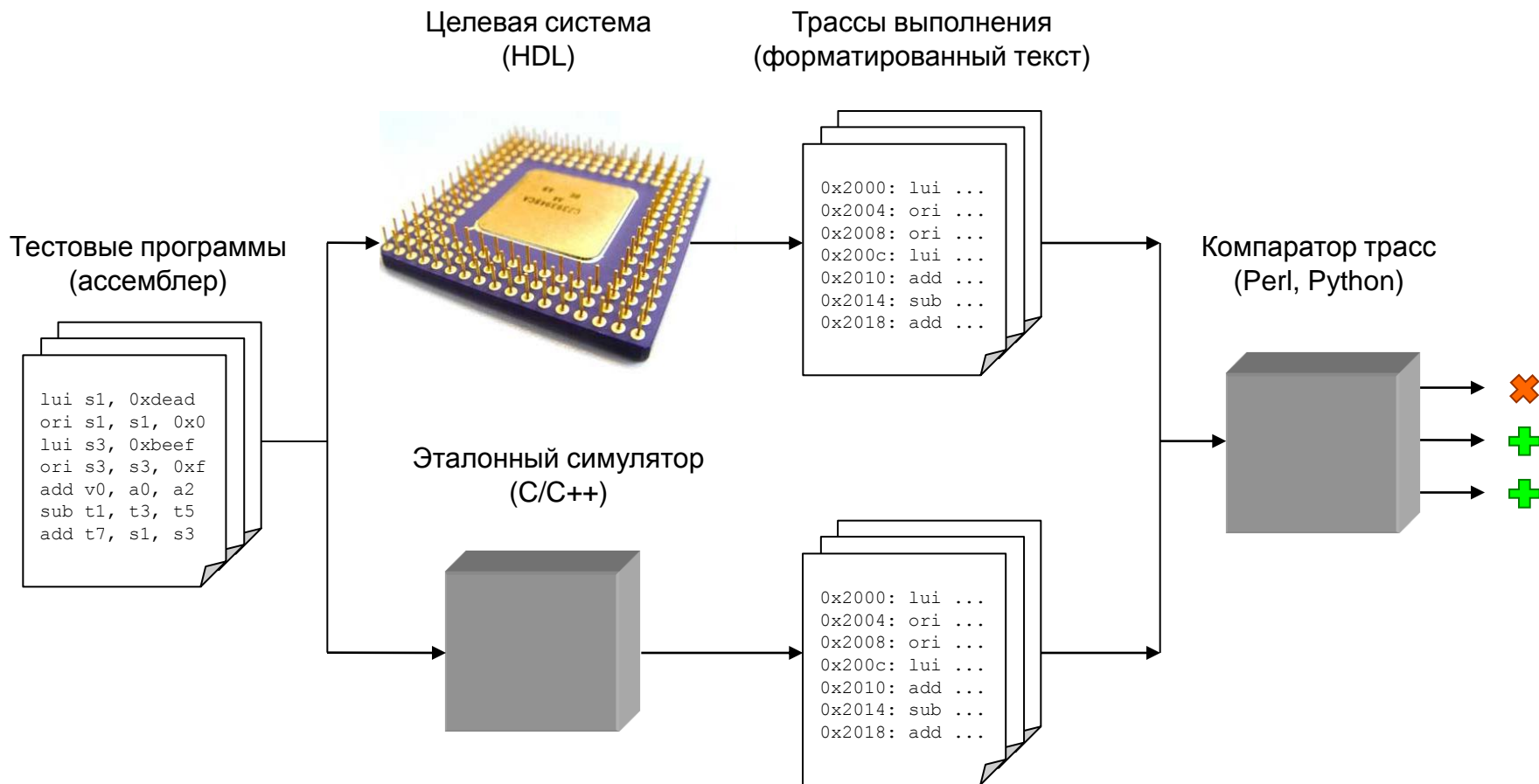
# Тестирование на основе моделей



# Адаптеры интерфейсов



# Генерация тестовых программ



# Генерация тестов на основе моделей

## Шаблон тестовой программы

```
Variable: addr = 0x100
Variable: reg
Bias: Resource-Dependency(GPR) = 30
Bias: Alignment(4) = 50

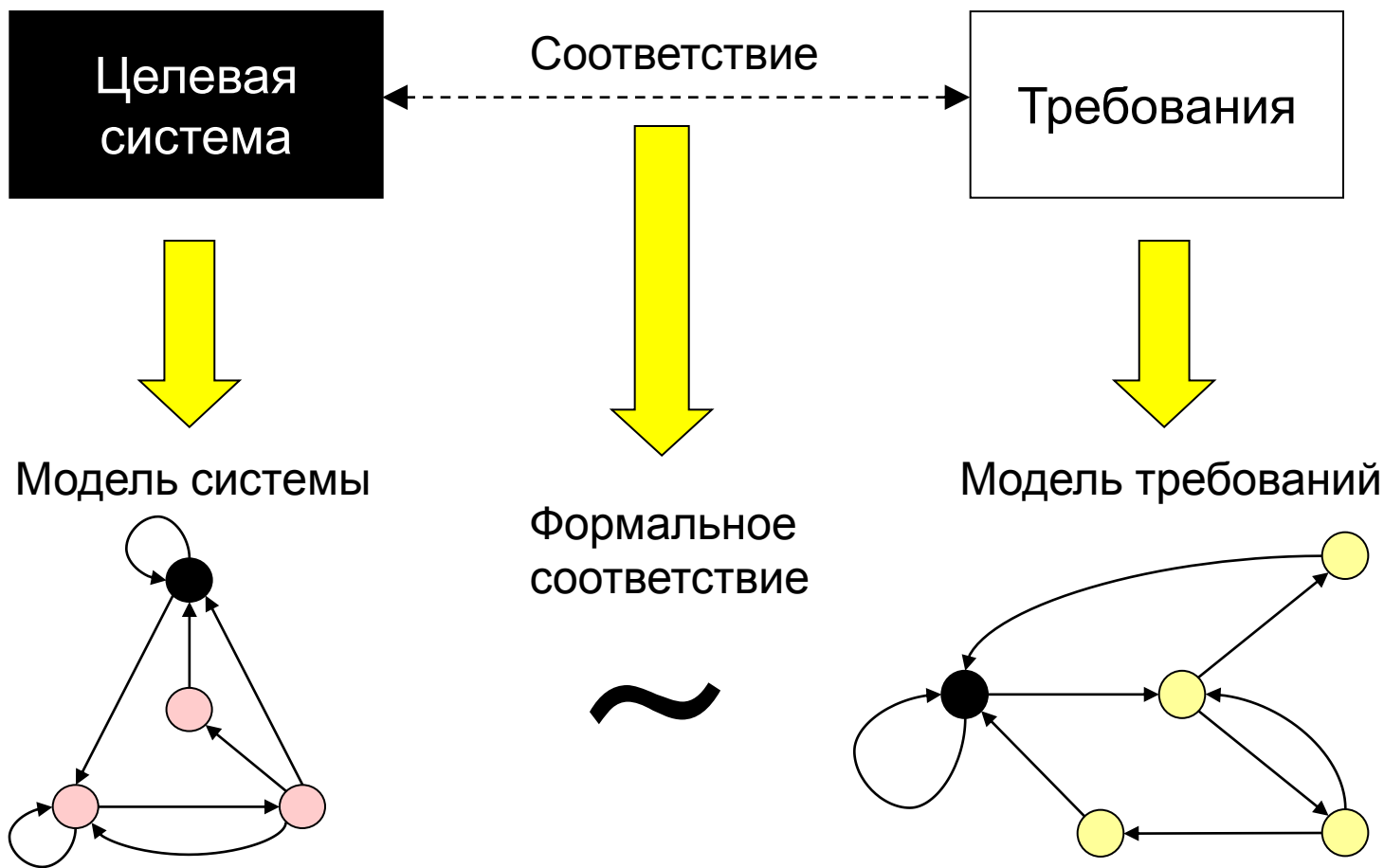
Instruction: Load R5 <- ?
  Bias: Alignment(16) = 100
Repeat (addr < 0x200)
  Instruction: Store reg -> addr
  Select
    Instruction: Add ? <- reg + ?
    Bias: SumZero
    Instruction: Sub ? <- ? - ?
  addr = addr + 0x10
```

## Тестовая программа

```
Resource Initial Values:
R6 = 8, R3 = - 25, ..., R17 = - 16
100 = 7, 110 = 25, ..., 1F0 = 16

Instructions:
500: Load R5 <- FF0
:
504: Store R4 -> 100
508: Sub R5 <- R6 - R4
50C: Store R4 -> 110
510: Add R6 <- R4 + R3
:
57C: Store R4 -> 1F0
580: Add R9 <- R4 + R17
```

# Формальная верификация



# Проект C++TESK

```

191
192
193 bool CacheMediator::IsCacheProcessExists()
194 {
195     return Cache::cacheProcessExists;
196 }
197
198 DEFINE_PROCESS(CacheMediator::serialize_loadd_iface)
199 {
200     DEBUG_PRINT(DEBUG_USER, "Call stimulus serialize_loadd_iface()");
201     InputData msg = CAST_MESSAGE(InputData);
202     START();
203     CAPTURE_IFACE();
204     // Setting dcache_req_in to 1.
205     Inputs.L2_control_inputs |= 0x400;
206     // Setting dcache_addr_req_in to 1.
207     Inputs.L2_control_inputs |= 0x800;
208     // Setting DPA_addr_in
209     Inputs.DPA_addr_in = msg.DPA_addr_in;
210     CYCLE();
211     RELEASE_IFACE();
212     DEBUG_PRINT(DEBUG_USER, "****serialize_loadd_iface 1
                
```

## Coverage report: coverage :: Coverage of bank numbers

Generated: Tue Oct 18 12:17:42 MSD 2011

**Summary** | **Details** | **Requirements Reports**

Report Page

Aspect: coverage  
Coverage name: Coverage of bank numbers

Coverage elements	Hits
(iface=iface_out_mrj cycle=8)	1
(iface=iface_out_mrj cycle=8)	0
(iface=iface_out_mrj cycle=8)	0
(iface=iface_out_mrj cycle=8)	0

Домашняя страница | Проекты | Помощь

## ISPRAS C++TESK Testing Toolkit

Обзор | Активность | Оперативный план | Задачи | Новости | Документы | Wiki | Форумы | **Файлы**

### Файлы

Файл	Дата	Размер	Скачиваний
<b>1.0</b>			
cpptestek-toolkit-src-1.0.1-beta-110415.tar.gz	15.04.2011 16:17	916,33 КБ	38
cpptestek-toolkit-src-1.0.10-beta-110824.tar.gz	24.08.2011 18:20	11,91 МБ	9
cpptestek-toolkit-src-1.0.11-beta-110901.tar.gz	01.09.2011 16:36	12,64 МБ	8
cpptestek-toolkit-src-1.0.12-beta-110906.tar.gz	06.09.2011 16:09	12,64 МБ	18
cpptestek-toolkit-src-1.0.13-beta-110928.tar.gz	28.09.2011 17:07	10,58 МБ	9
cpptestek-toolkit-src-1.0.2-beta-110504.tar.gz	04.05.2011 15:14	3,78 МБ	18
cpptestek-toolkit-src-1.0.3-beta-110510.tar.gz	10.05.2011 22:32	5,41 МБ	23
cpptestek-toolkit-src-1.0.4-beta-110520.tar.gz	20.05.2011 19:31	7,57 МБ	12
cpptestek-toolkit-src-1.0.5-beta-110528.tar.gz	28.05.2011 19:22	8,73 МБ	17
cpptestek-toolkit-src-1.0.6-beta-110621.tar.gz	21.06.2011 21:10	9,22 МБ	8
cpptestek-toolkit-src-1.0.7-beta-110625.tar.gz	25.06.2011 19:10	9,97 МБ	10
cpptestek-toolkit-src-1.0.8-beta-110702.tar.gz	02.07.2011 15:09	9,99 МБ	18
cpptestek-toolkit-src-1.0.9-beta-110727.tar.gz	28.07.2011 12:28	11,86 МБ	9

# Проект MicroTESK

MicroTESK - Komdiv64 Test Program Generator - <Unknown>.section

File Generation Window Help

Komdiv64

- cp1
  - arithmetic
    - abs.s
    - abs.d
    - abs.ps
    - addsub.ps
    - add.s
    - add.d
    - add.ps
    - div.s
    - div.d
    - madd.s
    - madd.d
    - madd.ps
    - maddsub1.ps
    - maddsub2.ps
    - msub.s
    - msub.d
    - msub.ps
    - mul.s
    - mul.d
    - mul.ps
    - neg.s
    - neg.d
    - neg.ps
    - nmadd.s
    - nmadd.d
    - nmadd.ps
    - nmsub.s
    - nmsub.d
    - nmsub.ps
    - recip.s

Group ARITHMETIC (cp1.arithmetic)

Test	Subgroup or Instruction	Equivalence Class	Situations
abs.s	abs.s	ABS_Equivalence_Class	1/1
abs.d	abs.d	ABS_Equivalence_Class	1/1
abs.ps	abs.ps	ABS_Equivalence_Class	1/1
addsub.ps	addsub.ps	ADD_Equivalence_Class	1/1
add.s	add.s	ADD_Equivalence_Class	1/1
add.d	add.d	ADD_Equivalence_Class	1/1
add.ps	add.ps	ADD_Equivalence_Class	1/1
div.s	div.s	DIV_Equivalence_Class	1/1
div.d	div.d	DIV_Equivalence_Class	1/1
madd.s	madd.s	MADD_Equivalence_Class	1/1
madd.d	madd.d	MADD_Equivalence_Class	1/1
madd.ps	madd.ps	MADD_Equivalence_Class	1/1
maddsub1.ps	maddsub1.ps	MADD_Equivalence_Class	1/1
maddsub2.ps	maddsub2.ps	MADD_Equivalence_Class	1/1
msub.s	msub.s	MSUB_Equivalence_Class	1/1
msub.d	msub.d	MSUB_Equivalence_Class	1/1
msub.ps	msub.ps	MSUB_Equivalence_Class	1/1
mul.s	mul.s	MUL_Equivalence_Class	1/1
mul.d	mul.d	MUL_Equivalence_Class	1/1
mul.ps	mul.ps	MUL_Equivalence_Class	1/1
neg.s	neg.s	NEG_Equivalence_Class	1/1
neg.d	neg.d	NEG_Equivalence_Class	1/1
neg.ps	neg.ps	NEG_Equivalence_Class	1/1
nmadd.s	nmadd.s	NMADD_Equivalence_Class	1/1
nmadd.d	nmadd.d	NMADD_Equivalence_Class	1/1
nmadd.ps	nmadd.ps	NMADD_Equivalence_Class	1/1
nmsub.s	nmsub.s	NMSUB_Equivalence_Class	1/1
nmsub.d	nmsub.d	NMSUB_Equivalence_Class	1/1
nmsub.ps	nmsub.ps	NMSUB_Equivalence_Class	1/1
recip.s	recip.s	RECIP_Equivalence_Class	1/1

MicroTESK - Options For Section section

MicroTESK - Settings

Generator Console

```

Generating file: test_00000.S
Creating package: test_00000
Moving file: test_00000.S to C:\Documents and Settings\root\Desktop\tests\test_00000
Generating file: test_00001.S
Creating package: test_00001
Moving file: test_00001.S to C:\Documents and Settings\root\Desktop\tests\test_00001
Generating file: test_00002.S
Creating package: test_00002
Moving file: test_00002.S to C:\Documents and Settings\root\Desktop\tests\test_00002
  
```

Top Groups: 0/0 (100%) Leaf Groups: 0/0 (100%) Inst

Top Groups: 0/0 (100%) Leaf Groups: 11/11 (100%) Instructions: 191/191 (100%) Situations: 191/191 (100%)

# Открытие проблемы

- Статический анализ HDL-кода, использование статического анализа в верификации
- Использование методов формальной верификации в имитационной верификации
- Распараллеливание процесса верификации, обработка графов сверхвысокой размерности
- Автоматическая локализация и диагностика ошибок в HDL-моделях



# Контакты

- Институт системного программирования РАН (ИСП РАН)  
<http://www.ispras.ru>
- Верификация микропроцессоров @ ИСП РАН  
<http://hardware.ispras.ru>
- А.С. Камкин, с.н.с. ИСП РАН  
[kamkin@ispras.ru](mailto:kamkin@ispras.ru)



**Спасибо  
Вопросы?**