

Университет ИТМО

Лабораторная работа #1

Вариант 1

Изучение микроархитектуры MIPS32

Выполнил  
Назарьев Сергей Сергеевич  
гр. Р3315

## Цель работы

1. Получить базовые навыки анализа цифровых схем, описанных с помощью языка Verilog HDL
2. Изучить особенности микроархитектуры MIPS32

## Задание

Определить, какие арифметические команды архитектуры MIPS32 реализованы в предоставленном процессорном ядре.

Дать краткое описание, мнемонику и структуру каждой арифметической команды.

Написать программу на ассемблере MIPS32 с использованием реализованных команд и продемонстрировать результат выполнения программы в симуляторе. В симуляции должен быть виден результат выполнения команд (код команды, изменения регистров, значений в ячейках памяти или транзакции на шине, ведущей к памяти).

## Результат

Мнемоника	Структура	Краткое описание
ADD ADDI	ADD <dst>, <reg1>, <reg2> -- R-TYPE ADDI <dst>, <reg1>, <imm1> -- I-TYPE	сложение значений двух регистров <i>reg1</i> и <i>reg2</i> (значений регистра <i>reg1</i> и числа <i>imm1</i> ) и сохранение результата в регистр <i>dst</i>
SUB	SUB <dst>, <reg1>, <reg2> -- R-TYPE	вычитание значения регистра <i>reg2</i> из <i>reg1</i> и сохранение результата в регистр <i>dst</i>
SLT SLTI	SLT <dst>, <reg1>, <reg2> -- R-TYPE SLTI <dst>, <reg1>, <imm1> -- I-TYPE	операция сравнения “меньше, чем”: результат булевой операции <i>reg1</i> < <i>reg2</i> (или <i>reg1</i> < <i>imm1</i> ), сохраняется в регистр <i>dst</i>

## Программа на MIPS Assembly Language:

```
.global entry

.data
    .word 0xABCD
    .word 0xDEAD

.text
.ent entry
entry:
    lw $t0, 0x200
    lw $t1, 0x201
test:  add $t0, $t0, 0x1
      beq $t0, $t1, exit
      j test
exit:  sub $t0, $t0, 0x1FFE
      sw $t0, 0x202
.end entry
```

## Получившийся листинг:

```
00000000 <entry>:
    0:  8c080200      lw      t0,512(zero)
    4:  8c090201      lw      t1,513(zero)

00000008 <test>:
    8:  21080001      addi   t0,t0,1
   c:  11090003      beq    t0,t1,1c <exit>
   10:  00000000      nop
   14:  08000002      j      8 <test>
   18:  00000000      nop

0000001c <exit>:
   1c:  2108e002      addi   t0,t0,-8190
   20:  ac080202      sw     t0,514(zero)
```

## Вывод

В рамках данной лабораторной работы я познакомился с форматами команд архитектуры MIPS, научился пользоваться системой симуляции и чуть лучше начал понимать синтаксис Verilog'a.