



ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно  
&& видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо

# Архитектура MIPS

Регистры процессора



Имеет RISC архитектуру

*RISC (reduced instruction set computer) - компьютер с набором простых и быстрых команд*

Особенности:

- ✓ Фиксированная длина машинных инструкций (например, 32 бита) и простой формат команды
- ✓ Большое количество регистров общего назначения (32 и более)

Используется в различных маршрутизаторах, принтерах, роботах, игровых приставках (Sony Play Station)

Особенности:

- ✓ Фиксированная длина машинных инструкций (например, 32 бита) и простой формат команды
- ✓ Большое количество регистров общего назначения (32 и более)

Стандартный MIPS процессор содержит 32 регистра для работы с целыми числами и 32 регистра для работы с плавающей арифметикой

Номер	Название	Описание
0	\$zero	Всегда содержит ноль
1	\$at	(assembler temporary) Зарезервировано для сборки
2,3	\$v0,\$v1	(values) Для возвращаемых функциями значений
4-7	\$a0-\$a3	(arguments) Первые четыре параметра для функции
8-15	\$t0-\$t7	(temporaries) Для временных значений
16-23	\$s0-\$s7	(saved values) Для постоянных значений
24,25	\$t8,\$t9	(temporaries) Для временных значений
26,27	\$k0,\$k1	Зарезервированы для обработчика (ядро ОС)
28	\$gp	(global pointer) Указывает на глобальную область
29	\$sp	(stack pointer) Значение равно верхнему адресу стека.
30	\$s8/\$fp	(saved values/frame pointer) Для постоянных значений/Указатель на фрейм
31	\$ra	(return address) Возвращаемый адрес

## Эмулятор MIPS

<http://courses.missouristate.edu/KenVollmar/MARS/download.htm>

D:\Languages\MIPS\mips1.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Edit Execute**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	10: lb \$t0, x
	0x00400004	0x0292359	lb \$s,0x00002359(\$1)	
	0x00400008	0x3c011001	lui \$1,0x00001001	11: lb \$t0, array1
	0x0040000c	0x02920000	lb \$s,0x00000000(\$1)	

  

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00636261	0x00000002	0x00000003	0x00000004	0x00000005	0x00000006	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 [data] Hexadecimal Addresses Hexadecimal Values ASCII

**Registers** Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$fp	28	0x00000000
\$sp	29	0x7fffffc0
\$gp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

**Mars Messages Run I/O**

```

Error in D:\Languages\MIPS\mips1.asm line 11 column 10:      lb $t0, array1[x]
Invalid language element: array1[x]
Error in D:\Languages\MIPS\mips1.asm line 11 column 10:      lb $t0, array1[x]
Invalid language element: array1[x]
Assemble: operation completed with errors.

Assemble: assembling D:\Languages\MIPS\mips1.asm

Assemble: operation completed successfully.
    
```

Clear

# Типы данных в MIPS-ассемблере

Нет различия знаковых и беззнаковых переменных

MIPS	Размер
X: .byte 0	8 бит
X: .half 0	16 бит
X: .word 0	32 бита
X: .dword 0	64 бита

- ✓ Метки используются для того, чтобы давать имена участкам памяти
- ✓ Различают метки данных и метки инструкций
- ✓ Секция данных начинается с `0x10010000`
- ✓ Секция кода начинается с `0x00400000`

# Типы инструкций

- ✓ Тип R (register) – в роли операндов используются три регистра:
  - `$rd` – регистр назначения
  - `$rs` – первый аргумент
  - `$rt` – второй аргумент
- ✓ Тип I (immediate) – в роли операндов используются два регистра и число
  - `$rd` – регистр назначения
  - `$rs` – первый аргумент
  - `I` – число
- ✓ Тип J (jump) – в роли операндов используется только один операнд – 26 битный адрес

Вопросы???





Пакулов Артур

[A.Pakulov.Otus@Gmail.com](mailto:A.Pakulov.Otus@Gmail.com)

Спасибо  
за внимание!

