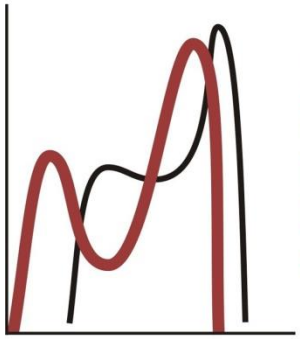


ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

имени И.И. Воровича —

Архитектура компьютера и операционные системы

Лекция 7. ISA MIPS

Андреева Евгения Михайловна
доцент кафедры информатики и вычислительного эксперимента

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

План лекции

- ISA
- Архитектура MIPS
- Домашнее задание

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Многоуровневая архитектура

5. ЯВУ	• Компиляторы, Библиотеки
4. Язык ассемблера	• Ассемблер, Линкер (компоновщик), Отладчик
3. Уровень ОС	• Этот уровень и ниже – системное программирование
2. Машинный код (Instruction Set Arch, ISA)	•ОЗУ, Системная шина, ЦП
1. Микрокод процессора (микроархитектура)	• Внутренняя шина, Тракт данных, АЛУ
0. Схемы цифровой логики	• Логические вентили и схемы
-1. Уровень физических устройств	• Сфера электронной техники и радиофизики

ISA (Instruction Set Architecture)

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

ISA включает:

- Модели памяти
- Режимы адресации
- Регистры
- Машинные команды (инструкции)
- Взаимодействие с внешними устройствами ввода/ вывода
- Различные типы внутренних данных (например, с плавающей запятой, целочисленные типы и т . д.)
- Обработчики прерываний и исключительных состояний

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Архитектура MIPS

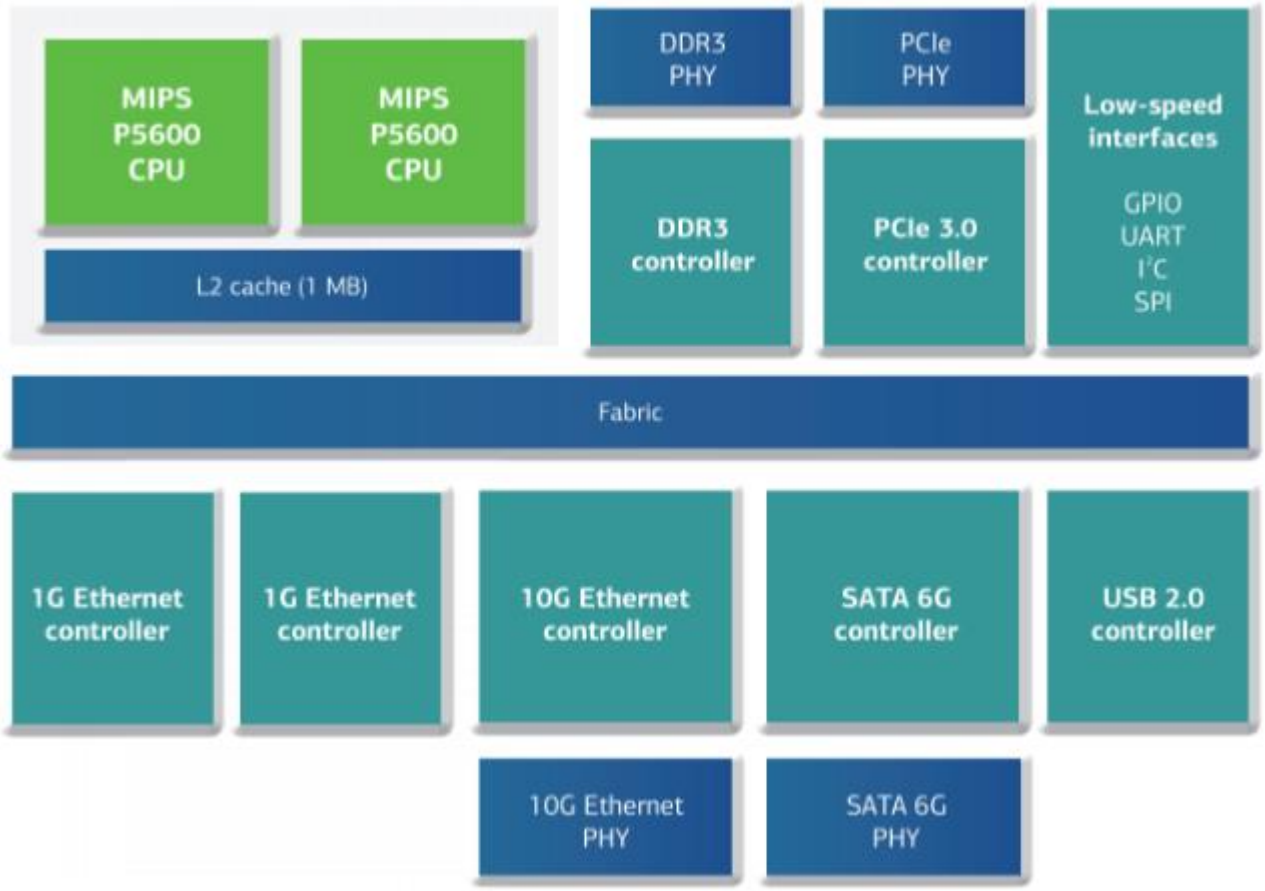
- Архитектура MIPS (Microprocessor without Interlocked Pipeline Stages) разработана Джоном Хеннесси и его коллегами в Стэнфорде в 1980-е годы в соответствии с концепцией RISC. Использовалась компаниями Silicon Graphics, Nintendo и Cisco.
- Д. Паттерсон, Дж. Хеннесси Архитектура компьютера и проектирование компьютерных систем
- Дэвид М. Харрис и Сара Л. Харрис Цифровая схемотехника и архитектура компьютера

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Baikal-T1

- Российская разработка на базе архитектуры MIPS.
- Разработчик - ОАО «Байкал Электроникс», основано компанией "Т-Платформы" в 2015 году

Блок-схема Baikal-T1



Инструкции

- Группы инструкций
  - арифметические
  - логические
  - перемещение данных
  - условные и безусловные

- Операнды инструкций (пример)

  add <приемник>, < операнд1 >, <операнд2>

  add rd, rs, rt

Регистры

Название	Номер	Применение
$zero	$0	всегда хранит 0
$at	$1	временный регистр для языка ассемблера
$v0—$v1	$2—$3	значения функций и выражений
$a0—$a3	$4—$7	аргументы функций
$t0—$t7	$8—$15	временные
$s0—$s7	$16—$23	сохраненные временные значения
$t8—$t9	$24—$25	временные
$k0—$k1	$26—$27	зарезервирована для ядра операционной системы
$gp	$28	глобальный указатель
$sp	$29	указатель стека
$fp	$30	указатель фрейма
$ra	$31	адрес возврата

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Отображение C в MIPS

- Код на C

- a=(b+c)–(d+e);

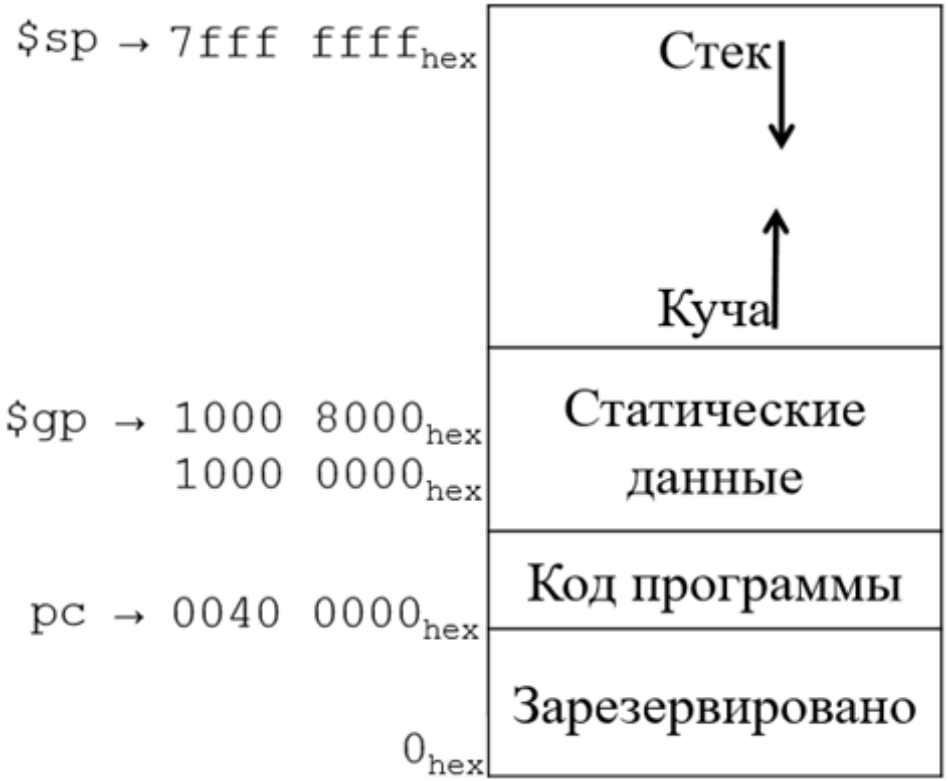
- Код на ассемблере MIPS – пусть a, … e находятся в регистрах $s0, … $s4

- add $t0, $s1, $s2
- add $t1, $s3, $s4
- sub $s0, $t0, $t1

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Модель памяти

- адресуется в байтах;
- требования по выравниванию (адрес кратен 4);
- составные типы данных размещаются в памяти;
- переключаемый порядок байтов;
- Load-Store - архитектура



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Режимы адресации

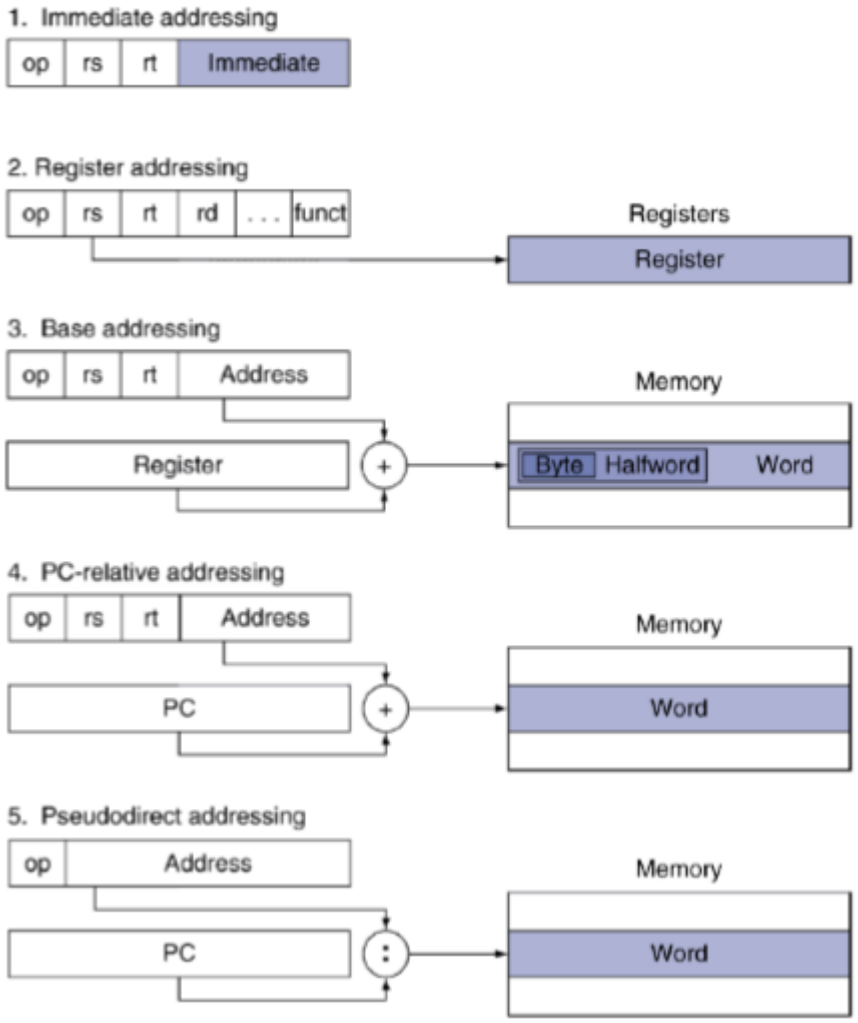
- Код на C
- a[3]=b+a[2];

- Код на ассемблере MIPS
  - пусть b находится в регистре $s2,
  - адрес начала массива a – в $s3
- lw $t0, 8($s3)
- add $t0, $s2, $t0
- sw $t0, 12($s3)

- Непосредственные операнды в MIPS
- addi $s0, $s2, -1

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Режимы адресации



- непосредственный (англ.: immediate)

```
addi $s0, $s2, -1
```

- регистровый (англ.: register-only)

```
add  $s0, $s1, $s2
```

- базовый (англ.: base)

```
add  $s0, $s1, 8($s2)
```

- относительно счетчика команд (англ.: PC-relative)

```
bne  $s3, $s4, Else
```

- псевдопрямой (англ.: pseudo-direct)

```
jal Sum
```

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Переходы

- Условные (branch)

beq rs, rt, L1
bne rs, rt, L1

- Безусловные (jump)

j L1
jal ProcAddr
jr $ra

- Сравнения

slt rd, rs, rt
slti rt, rs, const

- Set Less Than

R[rd] = (R[rs] < R[rt]) ? 1 : 0

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Переходы (пример)

- Код на C

```
if (i==j)
        a=b+c;
else
        a=b-c;
```

- Код на ассемблере MIPS
  - пусть a, b и c находятся в регистрах $s0-$s2,
    i – в $s3,
    j – в $s4

```
        bne  $s3, $s4, Else
        add  $s0, $s1, $s2
        j   Exit
Else: sub $s0, $s1, $s2
Exit: . . .
```

Типы инструкций

- R (register)
  - Операнды — 3 регистра, регистр назначения, первый аргумент и второй аргумент.
- I (immediate)
  - Операнды — 2 регистра и число.
- J (jump)
  - Операнд — 26 битный адрес, куда надо “прыгнуть”.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Код инструкций

Имя формата	Поля						Комментарий
	6 бит	5 бит	5 бит	5 бит	5 бит	6 бит	= 32 бита
R-формат	opcode	rs	rt	rd	shamt	функция	Арифметика
I-формат	opcode	rs	rt	адрес/непоср. операнд			Ветвления + непоср.оп.
J-формат	opcode	адрес для перехода (jump)					Безусловные п.

- Пример

```
add $t0, $s1, $s2
```

opcode	$s1	$s2	$t0	0	Add
0	17	18	8	0	32
000000	10001	10010	01000	00000	100000

Название	Номер
$0	0
$at	1
$v0-$v1	2–3
$a0-$a3	4–7
$t0-$t7	8–15
$s0-$s7	16–23
$t8-$t9	24–25
$k0-$k1	26–27
$gp	28
$sp	29
$fp	30
$ra	31

Таблица кодов инструкций

MIPS Reference Data

CORE INSTRUCTION SET

NAME, MNEMONIC		FOR-MAT	OPERATION (in Verilog)		OPCODE / FUNCT (Hex)
Add	`add`	R	R[rd] = R[rs] + R[rt]	(1)	$0 / 20_{hex}$
Add Immediate	`addi`	I	R[rt] = R[rs] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	`addiu`	I	R[rt] = R[rs] + SignExtImm	(2)	9_{hex}
Add Unsigned	`addu`	R	R[rd] = R[rs] + R[rt]		$0 / 21_{hex}$
And	`and`	R	R[rd] = R[rs] & R[rt]		$0 / 24_{hex}$
And Immediate	`andi`	I	R[rt] = R[rs] & ZeroExtImm	(3)	c_{hex}
Branch On Equal	`beq`	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4_{hex}
Branch On Not Equal	`bne`	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5_{hex}
Jump	`j`	J	PC=JumpAddr	(5)	2_{hex}
Jump And Link	`jal`	J	R[31]=PC+8;PC=JumpAddr	(5)	3_{hex}
Jump Register	`jr`	R	PC=R[rs]		$0 / 08_{hex}$
Load Byte Unsigned	`lbu`	I	R[rt]={24'b0,M[R[rs] +SignExtImm](7:0)}	(2)	24_{hex}
Load Halfword Unsigned	`lhu`	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25_{hex}

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Домашнее задание

- Подготовка к тесту по лекциям:
  - читать 2.1–2.3, 2.5-2.8, 2.10 [Паттерсон и Хеннесси]
  - стр. 379- 385 книги Таненбаума и Остина
  - Таблица кодов инструкций MIPS
- Подготовка к лабораторному занятию 6
- Приложение В книги Таненбаума и Остина.