Assignment 4 - Code Generation

A few words about the MIPS architecture

Compiler Construction

Fall 2012

Sven van Haastregt

LIACS

Leiden University

The MIPS Machine

- Highlights:
  - Word-addressable (word = 4 bytes)
  - Has 32 general purpose registers **$0**, **$1**, …, **$31**
  - Each register also has a name, e.g.: **$29 = $sp** (stack pointer)
  - Most instructions have the following form:

opcode destination, source1, source2

MIPS Registers

- Provides 32 registers, but some are reserved or have a special meaning:

`$0`:	Always 0
`$1`:	Reserved for assembler
`$2`:	Function return value
`$26-28`:	Reserved for OS etc.
`$29`:	Stack pointer
`$30`:	Frame pointer
`$31`:	Return address

MIPS Instructions

- Examples of MIPS instructions:

```
add   $2, $3, $4
sub   $4, $4, $3
li    $4, 123
addi  $4, $0, 123
beq   $2, $3, label
lw    $4, x
```

MIPS Addressing Modes

- MIPS is a load/store architecture: memory can be accessed only by load & store instructions.
- Data must be aligned properly.
- Main addressing modes:

`lw`	**`$3,($2)`**	Store *memory*(**$2**) into register **$3**
`lw`	**`$3,4($2)`**	Store *memory*(**4+$2**) into register **$3**
`lw`	**`$3,200`**	Store *memory*(**200**) into register **$3**
`lw`	**`$3,x`**	Store value of *x* into register **$3**

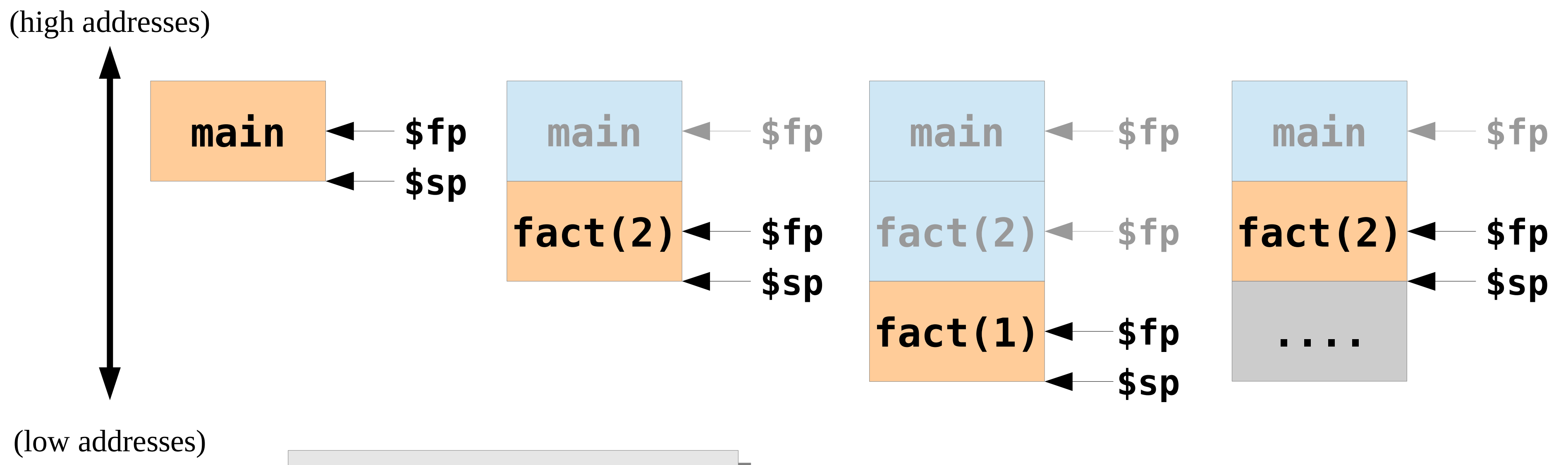
MIPS - Floating point

- The MIPS architecture has a separate coprocessor which deals with floating point arithmetic.
- Provides 32 new registers: **`$f0`** - **`$f31`**
- Single (32-bit) & double (64-bit) precision
- Examples:

```
li.s   $f2, 8.32
li.d   $f4, 3.14159265
add.d  $f6, $f4, $f4
```

Stack Frames

(also known as "Activation records")



```
unsigned fact(unsigned x) {
  if (x == 1)
    return 1;
  else
    return fact(x-1)*x;
}
```