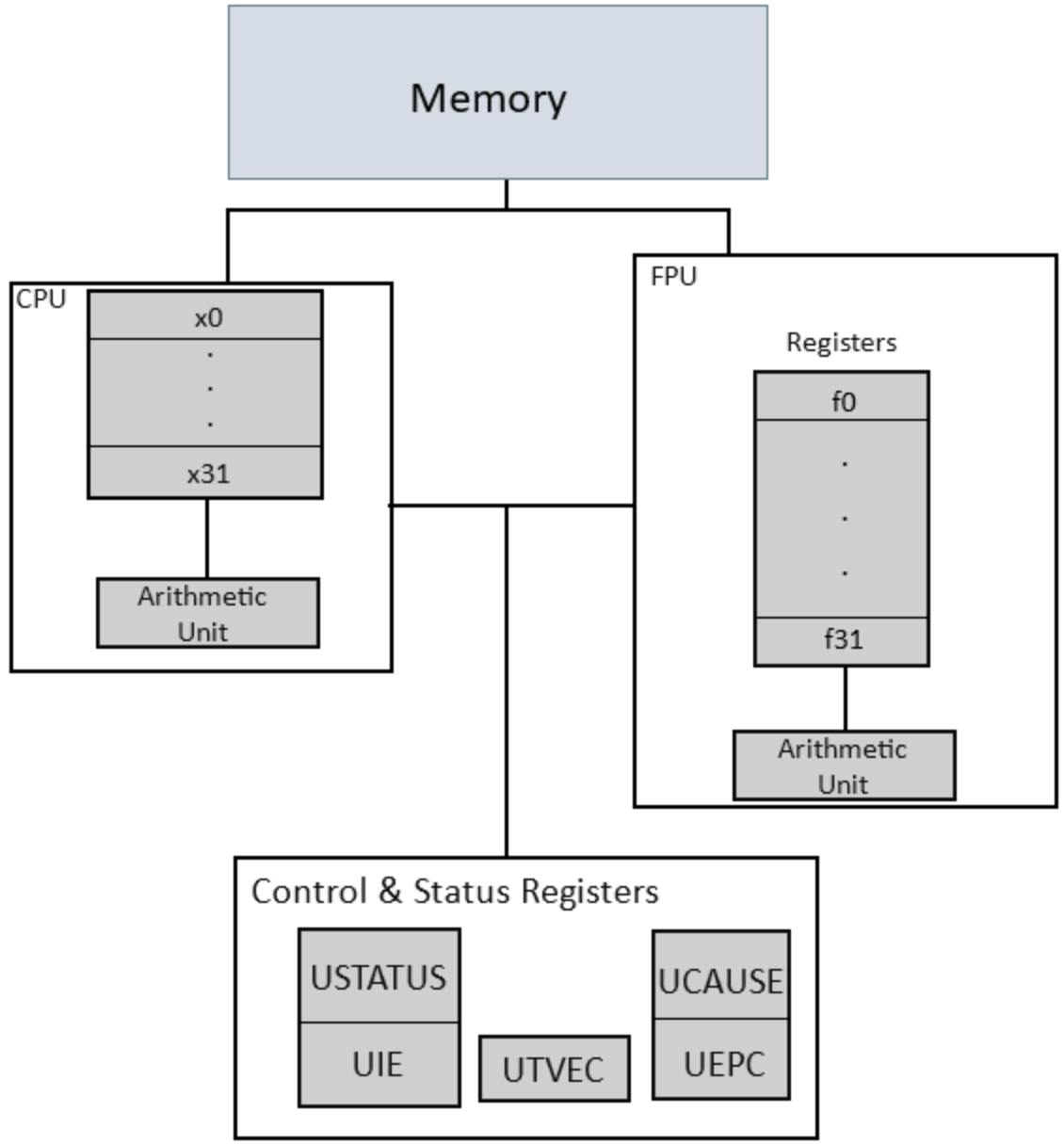
Topic V25

Floating Point in RISC-V


Memory
CPU
x0
x31
Arithmetic Unit
FPU
Registers
f0
f31
Arithmetic Unit
Control & Status Registers
USTATUS
UIE
UTVEC
UCAUSE
UEPC

Addition of signed magnitude numbers

Operation		A > B	B > A
(+A) + (+B)	+ (A + B)		
(+A) + (-B)		+ (A – B)	- (B – A)
(-A) + (+B)		- (A – B)	+ (B – A)
(-A) + (-B)	- (A + B)		
(+A) – (+B)		+ (A – B)	- (B – A)
(+A) – (-B)	+ (A + B)		
(-A) – (+B)	- (A + B)		
(-A) – (-B)		- (A – B)	+ (B – A)

- for addition – if signs differ subtract smaller from larger, otherwise add the fractions
  – if signs differ, sign of result is sign of larger

- for subtraction – if signs are the same subtract smaller from larger, otherwise add

Hardware for handling just the fractions

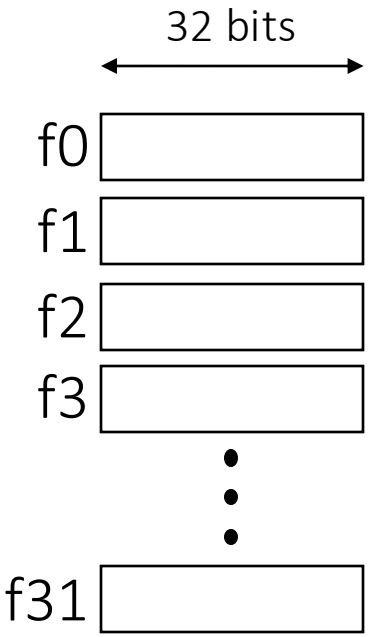
- two registers – one for each fraction
- two flip-flops – one for each sign
- adder
- complementer, with “+1” on the output
- comparator, to decide which sign to select
- various muxes and control logic

Floating Point in RISC-V

RISC-V supports the IEEE 754 single-precision and double-precision formats

Three architecture extensions: F, D, Q

F extension - floats



f0 is **not** hardwired to the value 0

Chapter 12

"D" Standard Extension for Double-Precision Floating-Point, Version 2.2

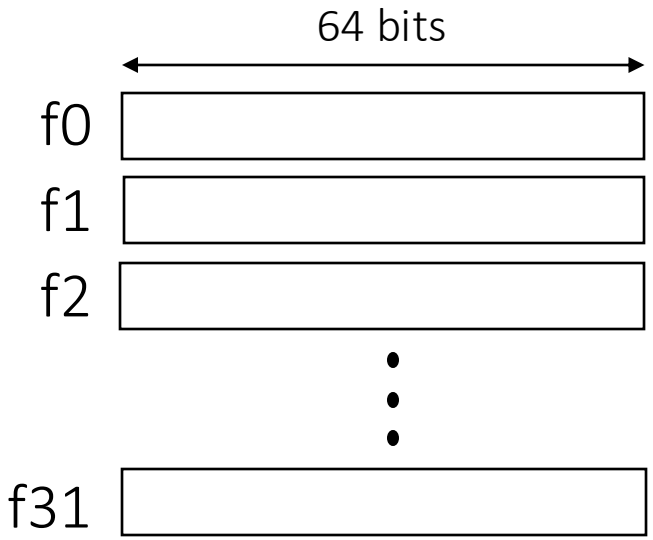
This chapter describes the standard double-precision floating-point instruction-set extension, which is named "D" and adds double-precision floating-point computational instructions compliant with the IEEE 754-2008 arithmetic standard. The D extension depends on the base single-precision instruction subset F.

12.1 D Register State

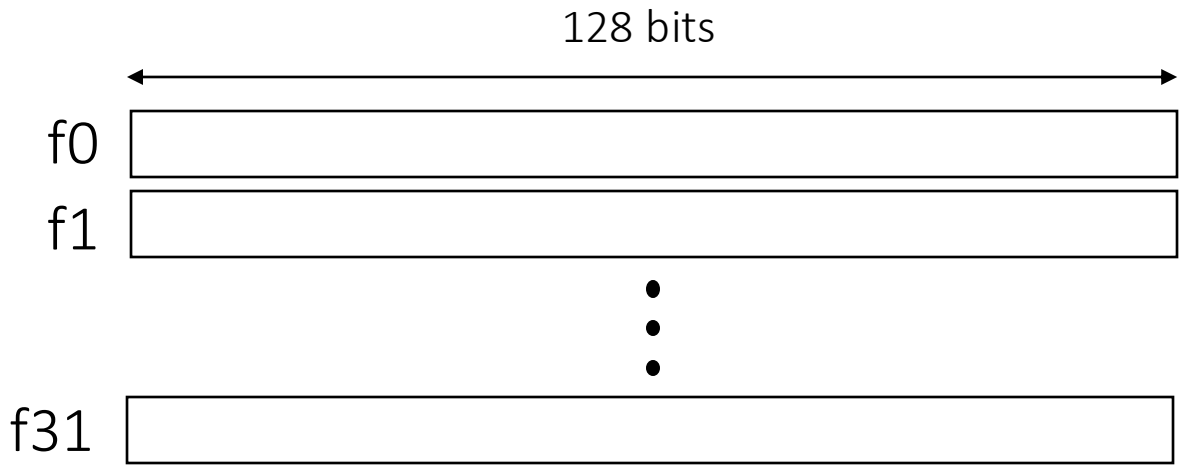
The D extension widens the 32 floating-point registers, `f0`–`f31`, to 64 bits (FLEN=64 in Figure 11.1). The `f` registers can now hold either 32-bit or 64-bit floating-point values as described below in Section 12.2.

FLEN can be 32, 64, or 128 depending on which of the F, D, and Q extensions are supported. There can be up to four different floating-point precisions supported, including H, F, D, and Q.

D extension - doubles



Q extension - quads



Floating Point load/store in RISC-V

Single-precision floating point value

```
flw     rd, imm(rs1)          # F[rd] = M[R[rs1]+imm]
```

Destination is FP register

Address is rs1 + 12-bit signed imm

```
fsw     rs2, imm(rs1)         # M[R[rs1]+imm] = F[rs2]
```

Source is FP register

Floating Point Instructions in RISC-V

fadd.s	fadd.d	FP addition single or double

Floating Point Instructions in RISC-V

fadd.s	fadd.d	FP addition single or double
fsub.s	fsub.d	FP subtraction single or double
fmul.s	fmul.d	FP multiplication single or double
fdiv.s	fdiv.d	FP division single or double
fx.s	fx.d	FP comparison (x = eq, lt, le), single or double, an integer register is set to 0 if the comparison is false otherwise it is set to 1. Paired with an integer branch instruction beq/bne to perform FP branches.
fcvt.s.w	fcvt.d.w	Converts a 32-bit signed integer in integer register rs1 to a single or double-precision floating point value in FP register rd
fcvt.w.s	fcvt.w.d	Converts a single or double-precision floating point value in FP register rs1 to a signed 32-bit integer in integer register rd

Floating Point Instructions in RISC-V

What does the following assembly code do?

```
flw      f4, 4(sp)
flw      f6, 8(sp)
fadd.s   f2, f4, f6
fsw      f2, 12(sp)
```

Reads two 32-bit floating point values from the stack, adds them, and stores the result into the stack