Floating point instruction decoder spec plan

1. Overview

The Floating-Point Instruction Decoder (FP Decoder) is responsible for decoding RISC-V single-precision floating-point (F extension, version 2.2) instructions during the Decode stage of the pipeline. The decoder extracts instruction fields, identifies the floating-point operation class, and generates control signals required by the floating-point register file and memory subsystem. The module does not perform arithmetic or memory access; it only interprets the instruction and produces control and operand information for downstream stages.

This specification defines what the decoder must do, independent of its RTL implementation.

2. Standards and References

- RISC-V Instruction Set Manual, Volume I: Unprivileged ISA
- Chapter 20: "F" Extension for Single-Precision Floating-Point, Version 2.2
- IEEE 754-2008 Floating-Point Arithmetic Standard

3. Inputs

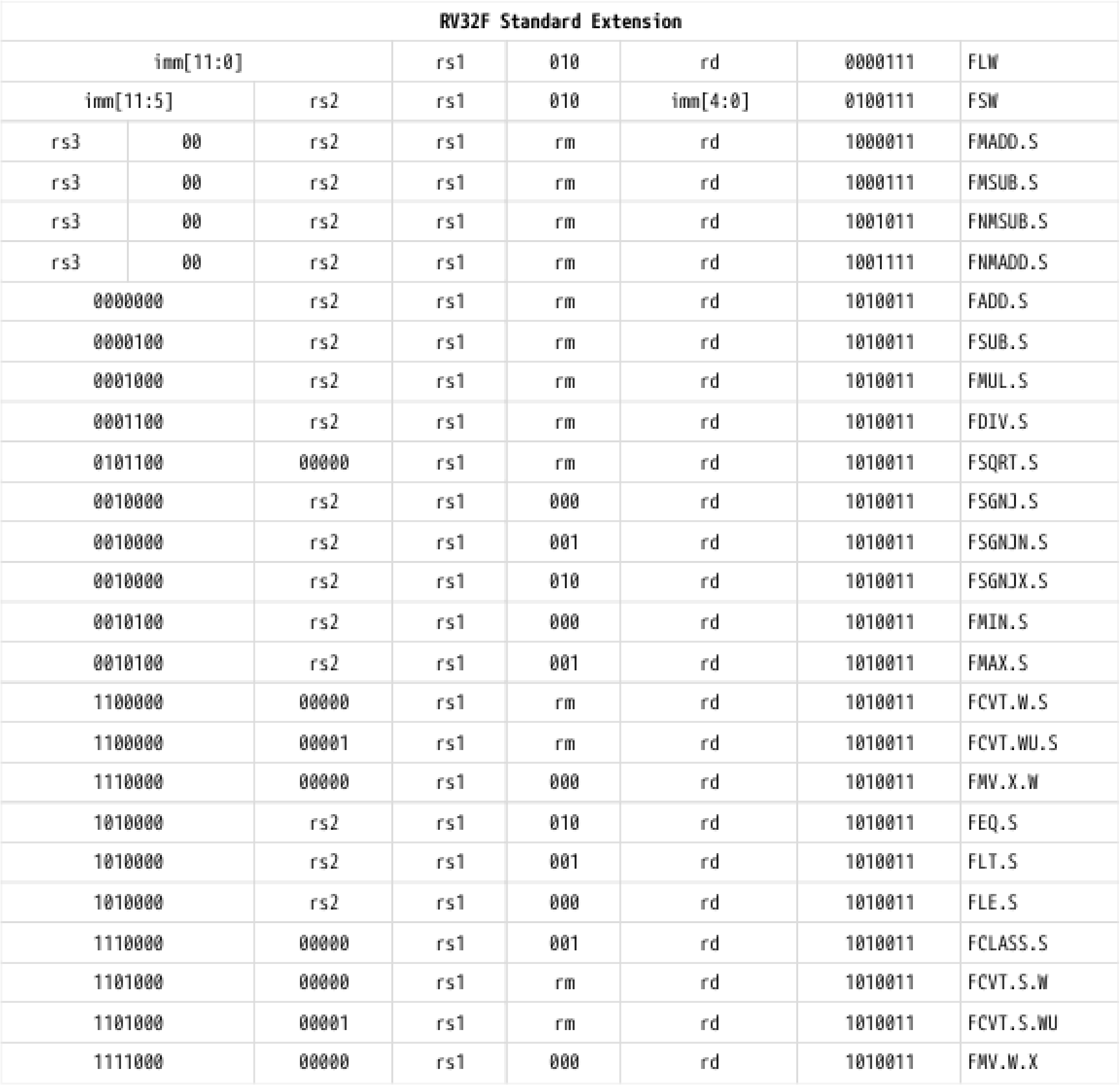
Signal	Width	Description
instr	32	32-bit instruction to be decoded
clk	1	Clock signal
rst	1	Synchronous reset; clears decoder outputs to a defined idle state.

4. Outputs

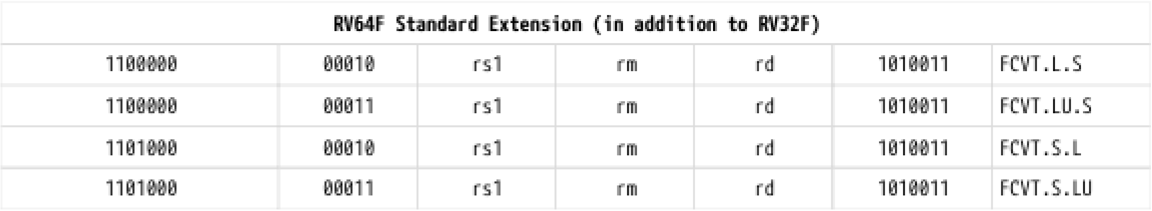
Signal	Width	Description
fp_op	7	Encoded floating-point operation derived from the instruction
function	5	Encoded function type derived from the instruction
fmt	2	Format field, always set to 00 when applicable
rs1	5	Source register index 1
rs2	5	Source register index 2

rs3	5	Source register index 3
rd	5	Destination register index
offset	12	Signed immediate offset for FP load/store instructions
fp_read	1	Asserted when the FP register file is read
fp_write	1	Asserted when the FP register file is written
eff_read	1	Asserted for floating-point load instructions (FLW)
int_read	1	Asserted when integer registers are read from.
int_write	1	Asserted when integer registers are written to.
eff_write	1	Asserted for floating-point store instructions (FSW)
rm	3	Rounding mode field extracted from the instruction

5. Each Instruction Layout



RV32F Standard Extension							
imm[11:0]			rs1	010	rd	0000111	FLW
imm[11:5]		rs2	rs1	010	imm[4:0]	0100111	FSW
rs3	00	rs2	rs1	rm	rd	1000011	FMADD.S
rs3	00	rs2	rs1	rm	rd	1000111	FMSUB.S
rs3	00	rs2	rs1	rm	rd	1001011	FNMSUB.S
rs3	00	rs2	rs1	rm	rd	1001111	FNMADD.S
0000000		rs2	rs1	rm	rd	1010011	FADD.S
0000100		rs2	rs1	rm	rd	1010011	FSUB.S
0001000		rs2	rs1	rm	rd	1010011	FMUL.S
0001100		rs2	rs1	rm	rd	1010011	FDIV.S
0101100		00000	rs1	rm	rd	1010011	FSQRT.S
0010000		rs2	rs1	000	rd	1010011	FSGNJ.S
0010000		rs2	rs1	001	rd	1010011	FSGNJN.S
0010000		rs2	rs1	010	rd	1010011	FSGNJX.S
0010100		rs2	rs1	000	rd	1010011	FMIN.S
0010100		rs2	rs1	001	rd	1010011	FMAX.S
1100000		00000	rs1	rm	rd	1010011	FCVT.W.S
1100000		00001	rs1	rm	rd	1010011	FCVT.WU.S
1110000		00000	rs1	000	rd	1010011	FMV.X.W
1010000		rs2	rs1	010	rd	1010011	FEQ.S
1010000		rs2	rs1	001	rd	1010011	FLT.S
1010000		rs2	rs1	000	rd	1010011	FLE.S
1110000		00000	rs1	001	rd	1010011	FCLASS.S
1101000		00000	rs1	rm	rd	1010011	FCVT.S.W
1101000		00001	rs1	rm	rd	1010011	FCVT.S.WU
1111000		00000	rs1	000	rd	1010011	FMV.W.X



RV64F Standard Extension (in addition to RV32F)						
1100000	00010	rs1	rm	rd	1010011	FCVT.L.S
1100000	00011	rs1	rm	rd	1010011	FCVT.LU.S
1101000	00010	rs1	rm	rd	1010011	FCVT.S.L
1101000	00011	rs1	rm	rd	1010011	FCVT.S.LU

6. Functional Behavior

6.1 Instruction Field Extraction

The decoder shall extract the following fields from the 32-bit instruction:

- Opcode
- Register indices (`rs1`, `rs2`, `rs3`, `rd`)
- Immediate fields for load/store instructions
- Rounding mode (`rm`)

For STORE-FP instructions, the immediate offset shall be reconstructed from split fields and sign-extended to 12 bits.

6.2 Instruction Classification

The decoder shall identify the instruction class based on opcode and function fields, including:

- Floating-point load and store instructions: FLW, FSW
- Floating-point computational instructions: FADD.S, FSUB.S, FMUL.S, FDIV.S, FSQRT.S, FMIN.S, FMAX.S.
- Fused multiply-add instructions: FMADD.S, FMSUB.S, FNMADD.S, FNMSUB.S
- Conversion and move instructions: FCVT.W.S, FCVT.L.S, FCVT.S.W, FCVT.S.L, FCVT.WU.S, FCVT.LU.S, FCVT.S.WU, FCVT.S.LU.
- Floating-point sign injection: FSGNJ.S, FSGNJN.S, FSGNJX.S.
- Bit pattern moving: FMV.X.W, FMV.W.X.
- Compare instructions: FEQ.S, FLT.S, FLE.S
- Classify instruction: FCLASS.S:

The decoded instruction class shall be encoded in the `fp_op` output.

6.3 Control Signal Generation

Based on the decoded instruction type, the decoder shall generate control signals as follows:

Floating-Point Load (FLW)

- `fp_write` = 1
- `fp_read` = 0
- `eff_read` = 1
- `eff_write` = 0
- int_read = 0
- int_write = 0

Floating-Point Store (FSW)

- `fp_write` = 0
- `fp_read` = 1
- `eff_read` = 0
- `eff_write` = 1
- int_read = 0
- int_write = 0

Floating-Point Arithmetic Instructions

- `fp_write` = 1
- `fp_read` = 1
- `eff_read` = 0
- `eff_write` = 0
- int_read = 0
- int_write = 0

Floating-Point Injection Instructions

- `fp_write` = 1
- `fp_read` = 1
- `eff_read` = 0
- `eff_write` = 0
- int_read = 0
- int_write = 0

Floating-Point Conversion/Class Instructions

- fp_write = 0 or 1
- fp_read = 0 or 1
- int_read = 0 or 1
- int_write = 0 or 1
- eff_read = 0
- eff_write = 0

Floating-Point Bit Patterns Movage Instructions

- fp_write = 0 or 1
- fp_read = 0 or 1
- int_read = 0 or 1
- int_write = 0 or 1
- eff_read = 0
- eff_write = 0

Floating-Point Compare Instructions

- fp_write = 0
- fp_read = 1
- int_read = 0
- int_write = 1
- eff_read = 0
- eff_write = 0

6.4 Rounding Mode Handling

The decoder shall output the 3-bit `rm` field directly from the instruction encoding.

- If `rm` = `111`, the instruction specifies dynamic rounding mode, which shall be resolved using the `frm` field of `fcsr` in a downstream stage.
- Reserved rounding mode encodings are passed through unchanged.

7. Effective Address Computation

Effective Address will NOT be computed during the decode phase. For load and store operations, the decoder outputs the offset, which will be used to compute the effective address as effective_address = value(rs1) + offset

8. Reset behaviour

When reset is asserted:

- All control outputs (fp_read, fp_write, eff_read, eff_write) will be deasserted
- Register index outputs will be set to zero.
- fp_op will be set to zero.

9. Summary

This specification defines the required behavior and interfaces of the Floating-Point Instruction Decoder. The decoder provides a clean separation between instruction interpretation and execution, enabling modular and verifiable floating-point pipeline design.