F-CPU Architecture Guide

(draft version 0.19991113)

F-CPU Project

Mathias Brossard

F-CPU Architecture Guide(draft version 0.19991113)

by F-CPU Project, Mathias Brossard

Table of Contents

List of Tables

Forewords

This document is preliminary

Any part can be changed without notice.

I. Introduction

Table of Contents

Chapter 1. Instruction Formats

1.1. Instruction Formats

Table 1-1. Format 0

0-7	7-31
Opcode	Constant

Table 1-2. Format 1

0-7	7-25	26-31
Opcode	Constant	Register

Table 1-3. Format 2

0-7	8-19	20-25	26-31
Opcode	Constant	Register 1	Register 2

Table 1-4. Format 3

0-7	8-13	14-19	20-25	26-31
Opcode	Constant	Register 1	Register 2	Register 3

1.2. Size Flags

Table 1-5. Format 3

Flags	Size(byte)	Suffix	Name
00	1	B	Byte
01	2	D	Double-Byte
10	4	Q	Quad-Byte
11	8	(none)	Octa-Byte (Word)

II. Instruction Set

Table of Contents

Chapter 2. Data Manipulation

2.1. Arithmetic Instructions

2.1.1. add

add r1, r2, r3

add performs an integer addition of the two source operands (r1 + r2) and puts the result in destination operand (r3).

The size *flag* indicates that *add* performs the addition on the whole operands or only on a part of the operands.

The *SIMD* flag indicates that *add* performs multiple addition on parts of the operand (the size of these parts is defined by the *size* flags).

The *saturate* flag indicates that *add* does not “wrap” if the result is bigger than the size of the operands.

8-9	[qdb]	Defines the size parameter
10	[s]	Defines if the operation is SIMD
11	[m]	Defines if the operation should saturate
12		Unused
13		Unused

2.1.2. sub

sub r1, r2, r3

sub performs an integer substraction of the two source (r1 - r2) and puts the result in destination operand (r3).

The *size* flag indicates that *sub* performs the substraction on the whole operands or only on a part of the operands.

The *SIMD* flag indicates that *sub* performs multiple substraction on parts of the operand (the size of these parts is defined by the *size* flags).

The *floor* flag indicates that *sub* does not “wrap” if the second operand is bigger than the first one.

8-9	[qdb]	Defines the size parameter
10	[s]	Defines if the operation is SIMD
11	[f]	Defines if the operation should floor
12		Unused
13		Unused

2.1.3. mul

mul r1, r2, r3

mul performs an integer multiplication of the two source (r1 x r2) and puts the result in destination operand (r3).

The *size* flag indicates that *mul* performs the multiplication on the whole operands or only on a part of the operands. It only puts the lower part of the result.

The *SIMD* flag indicates that *mul* performs multiple multiplications on parts of the operand (the size of these parts is defined by the *size* flags).

The *sign* flag indicates that *mul* will consider the operands as signed by extending the MSB.

The *high* flag indicates that *mul* will will also store the higher part of the result in r3+1.

8-9	[qdb]	Defines the size parameter
10	[s]	Defines if the operation is SIMD
11	[a]	Defines if the operation is signed
12	[h]	Defines if the operation also stores the high part
13		Unused

Chapter 3. Load Store

3.1. load, store, loadi, storei...

load, store, loadi, storei...