RISC-V Model Documentation

Release unknown

Stefan Wallentowitz

Jun 11, 2020

Contents:

CHAPTER 1

Developer Handbook

1.1 Instructions

class riscvmodel.insn.**Instruction**
Base class for instructions

This is the abstract base class for all instruction. They are derived via their instruction type.

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

execute(*model: riscvmodel.model.State*)
Execute this instruction

Execute the instruction on the given model

Parameters **model** – RISC-V core model

Returns nothing

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionBType**(*rs1: int = None*, *rs2: int = None*, *imm: int = None*)
B-type instructions encode branches. Branches have two source registers that are compared. They then change the program counter by the immediate value.

Parameters

- **rs1** (*int*) – Source 1 for comparison
- **rs2** (*int*) – Source 2 for comparison

- **imm** (*int*) – Immediate for branch destination address calculation (13-bit, signed, 16-bit aligned)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionCBType**(*rd: int = None*, *imm: int = None*)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

class riscvmodel.insn.**InstructionCIType**(*rd: int = None*, *imm: int = None*)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionCRType**(*rd: int = None*, *rs: int = None*)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionCSSType**(*rs: int = None*, *imm: int = None*)

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionCType**

class riscvmodel.insn.**InstructionILType**(*rd: int = None*, *rs1: int = None*, *imm: int = None*)
I-type instruction specialization for stores. The produce a different assembler than the base class

Parameters

- **rd** (*int*) – Destination register
- **rs1** (*int*) – Source register 1
- **imm** – 12-bit signed immediate

class riscvmodel.insn.**InstructionISType**(*rd: int = None*, *rs1: int = None*, *shamt: int = None*)
I-Type instruction specialization for shifts by immediate. The immediate differs here (5-bit unsigned).

Parameters

- **rd** (*int*) – Destination register
- **rs1** (*int*) – Source register 1
- **imm** (*int*) – 12-bit signed immediate

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionIType**(*rd: int = None*, *rs1: int = None*, *imm: int = None*)
I-type instructions are registers that use one source register and an immediate to produce a new value for the destination register.

Two specialization exist for this class: *InstructionILType* for load instructions and *InstructionISType* for instructions that shift by an immediate value.

Parameters

- **rd** (*int*) – Destination register
- **rs1** (*int*) – Source register 1
- **imm** (*int*) – 12-bit signed immediate

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionJType**(*rd: int = None*, *imm: int = None*)
J-type instruction are used for jump and link instructions.

Parameters

- **rd** (*int*) – Destination register
- **imm** (*int*) – Immediate for the jump (21-bit, signed, 16-bit aligned)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionRType**(*rd: int = None*, *rs1: int = None*, *rs2: int = None*)
R-type instructions are 3-register instructions which use two source registers and write one output register.

Parameters

- **rd** (*int*) – Destination register
- **rs1** (*int*) – Source register 1
- **rs2** (*int*) – Source register 2

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionSType**(*rs1: int = None*, *rs2: int = None*, *imm: int = None*)
S-type instructions are used for stores. They don't have a destination register, but two source registers.

Parameters

- **rs1** (*int*) – Source register for base address
- **rs2** (*int*) – Source register for data
- **imm** (*int*) – Offset of store, for calculation of address relative to rs1

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

class riscvmodel.insn.**InstructionUType**(*rd: int = None*, *imm: int = None*)
U-type instructions are used for constant formation and set the upper bits of a register.

Parameters

- **rd** (*int*) – Destination register
- **imm** (*int*) – Immediate (20-bit, unsigned)

decode(*machinecode: int*)
Decode a machine code and configure this instruction from it.

Parameters **machinecode** (*int*) – Machine code as 32-bit integer

randomize(*variant: riscvmodel.variant.Variant*)
Randomize this instruction

This function randomizes the instance of an instruction according to the given variant.

Parameters **variant** – RISC-V ISA variant

Returns nothing

riscvmodel.insn.**get_insns**(*, *cls=None*)
Get all Instructions. This is based on all known subclasses of *cls*. If non is given, all Instructions are returned. Only such instructions are returned that can be generated, i.e., that have a mnemonic, opcode, etc. So other classes in the hierarchy are not matched.

Parameters **cls** (Instruction) – Base class to get list

Returns List of instruction classes

riscvmodel.insn.**get_mnenomics**()
Get all known mnemonics

Returns List of all known mnemonics

Return type List[str]

riscvmodel.insn.**isa**(*mnemonic: str*, *, *opcode: int*, *funct3: int = None*, *funct7: int = None*, *funct12: int = None*, *variant=Variant(intregs=32, xlen=32, extensions=Extensions(M=False, A=False, F=False, D=False, Q=False, C=False))*, *extension=None*)
Decorator for the instructions. The decorator contains the static information for the instructions, in particular the encoding parameters and the assembler mnemonic.

Parameters

- **mnemonic** – Assembler mnemonic
- **opcode** – Opcode of this instruction
- **funct3** – 3 bit function code on bits 14 to 12 (R-, I-, S- and B-type)
- **funct7** – 7 bit function code on bits 31 to 25 (R-type)
- **funct12** – 12 bit function code on bits 31 to 20

Returns Wrapper class that overwrites the actual definition and contains static data

riscvmodel.insn.**isaC**(*mnemonic: str*, *opcode: int*, ***, *funct3=None*, *funct4=None*, *funct6=None*, *variant=Variant(intregs=32, xlen=32, extensions=Extensions(M=False, A=False, F=False, D=False, Q=False, C=False))*, *extension=Extensions(M=False, A=False, F=False, D=False, Q=False, C=True)*)

Decorator for the instructions. The decorator contains the static information for the instructions, in particular the encoding parameters and the assembler mnemonic.

Parameters **mnemonic** – Assembler mnemonic

Returns Wrapper class that overwrites the actual definition and contains static data

riscvmodel.insn.**reverse_lookup**(*mnemonic: str*)

Find instruction that matches the mnemonic.

Parameters **mnemonic** – Mnemonic to match

Returns *Instruction* that matches or None

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

r

Index