New issue

Jump to bottom

Offset typo in RV32I Conditional Branches #583

Open **GuillermoCallaghan** opened this issue on Sep 16, 2020 · 5 comments

GuillermoCallaghan commented on Sep 16, 2020 • edited

Hi everybody,
there seems to be a typo on

- RV32I Base Integer Instruction Set, Version 2.1
  - 2.5 Control Transfer Instructions
    - Conditional Branches

in page 22.

This is in /src/rv32.tex in the following lines: L898 (BEQ/BNE), L899 (BLT[U]) and L900 (BGE[U]).

The offset should be `offset[4:1|11]` and not `offset[11|4:1]` as shown in the figure.

Conditional Branches

All branch instructions use the B-type instruction format. The 12-bit B-immediate encodes signed offsets in multiples of 2 bytes. The offset is sign-extended and added to the address of the branch instruction to give the target address. The conditional branch range is ±4 KiB.

31	30 25	24 20	19 15	14 12	11 8	7	6 0
imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode
1	6	5	5	3	4	1	7
offset[12|10:5]		src2	src1	BEQ/BNE	offset[11|4:1]		BRANCH
offset[12|10:5]		src2	src1	BLT[U]	offset[11|4:1]		BRANCH
offset[12|10:5]		src2	src1	BGE[U]	offset[11|4:1]		BRANCH

David-Horner commented on Sep 16, 2020 Contributor

I believe this label is correct.

Compare with jal just above . it labels the 4 segments as offset [20:1].
Similarly,offset[11|4:1] labels the bits that are in the fields in
decreasing significance, so 11 comes first.
...

stnolting commented on Sep 16, 2020 • edited ▾ Contributor

I think the label is wrong.
Chapter 24 shows a list of all instruction encodings. And there we have a different (the correct) label:

RV32I Base Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU

GuillermoCallaghan commented on Sep 16, 2020 Author

Right, I see what you mean **@David-Horner**, however I find it a bit confusing, there is no need for the offset to be formatted there in decreasing order of significance, unless I am missing something.

I see there is another issue opened which is highlighting that the ISA Manual addresses offsets inconsistently (Issue 509).
The Compressed ISA seems not to be respecting that format for offset representations (which I personally like a bit more).

@stnolting Agree, I would also expect to see the label as `offset[4:1|11]` as shown in your figure.

allenjbaum commented on Sep 17, 2020

The JAL instruction lists each subfield separately (Imm[20], imm[10:1] as
opposed to concatenated, e.g. imm[20|10:11]), and then underneath, the
entire immediate in decreasing order of significance, so there is precedent
there also.
...

David-Horner commented on Sep 17, 2020 Contributor

Expressing concepts and issues in alternate terms can be useful and instructive.

I believe **@aswaterman** has done an excellent job in guiding this effort. He has weighed representations with wisdom.
Not all situations need extensive labelling or visual reinforcement/clarification/qualification and so many that could have been made are omitted.
Further, different contexts are best expressed with different representations even if they use similar techniques, e.g. labelling.

For the jal example, the label consolidates all four separate fields into one range, offset[20:1].
I believe the intent is obvious: to show that all these consecutive bits are contained within the combined fields.
It is also apparent that offset bit[0] is missing, reinforcing its intentional absence in the instruction.

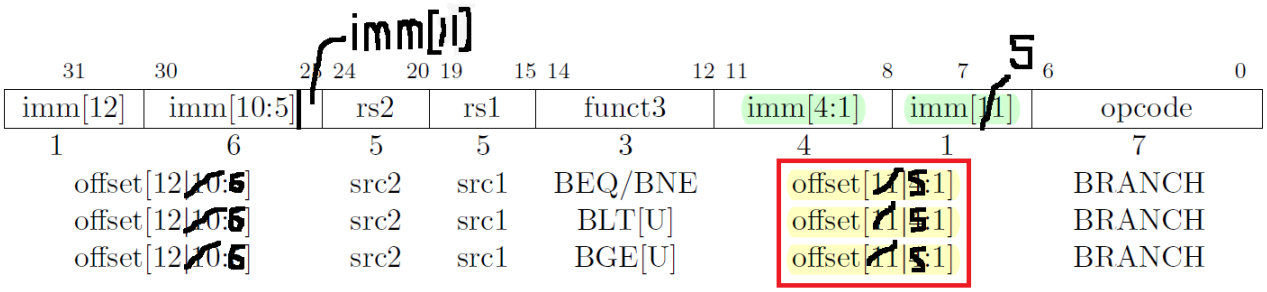
@stnolting: I note that the list of instructions entry for JAL dose not have labelling per se.
Instead, it actually consolidates the multiple fields into one entry that clearly identifies the specific fields (in their prescribed order) and their type (an immediate field).
This also is an excellent representation for its intended purpose, and kudos to Andrew for this labour of love.

@GuillermoCallaghan
Back to the initial concern.
First, I too had a suggestions during the early evolution of the now called "The RISC-V Instruction Set Manual Volume I: Unprivileged ISA".
None of mine were so neatly laid out with highlighted graphics; if they were, perhaps some may have been more readily adopted.
Thus I greatly appreciate your time and effort to clarify the text for others. It can be a low thanks job but it is very important.

The branch instruction does not have all the immediate fields adjacent to one another , so that the obvious "offset" label of
{12:1] is not easily annotated within the diagram. We can imagine an arrangement in which rs1,rs2 and funct3 were say at the high end of the instruction for with the label "offset[12:1]" encompassing all four fields would be an excellent consolidation , as was done for JAL.
However, in that the immediate value is physically spead over 2 sets of adjacent fields, two less obvious labels are required.
We can also imagine a scenario in which instruction bits 7 and 11 exchanged meaning;

Conditional Branches

All branch instructions use the B-type instruction format. The 12-bit B-immediate encodes signed offsets in multiples of 2 bytes. The offset is sign-extended and added to the address of the branch instruction to give the target address. The conditional branch range is ±4 KiB.



we would then have two nice labels identifying the bits in logical order [12:6] and [5:1].
Unfortunately for our graphics, we have the original layout, and the most consistent expression with [,that won't cause a reader to wonder if they misinterpreted,] the JAL label, is the one finally chosen.

We each have our preferences and perceptions of best representations, each valid for us.
In this particular case I believe that the best was made of a difficult presentation.

There may be other alternatives that might be more intuitive to some, e.g.

offset[12| ~~11~~| 10:5 |~~4:1~~] and **offset[~~12~~| 11| ~~10:5~~| 4:1]**

However, the labels would barely fit under their fields
and I expect this would not be universally embraced.

I appreciate that we corporately keep trying.

1

saahm mentioned this issue on Mar 18

Compressed Extension, Immediates, Offsets, etc. #830

Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

4 participants