Using Custom Instructions with the GDB Debugger

17th Feb 2022, Limerick, Ireland.

Using Custom Instructions with the GDB Debugger

Custom instructions are typically used to move critical portions of an application implementation from software to the hardware domain with a resulting performance boost, power saving and code density improvement. It's all about CPU cycles and doing more in less! Custom instructions typically require a new instruction mnemonic to invoke and you can think of this as an "API" to call the encapsulated hardware functionality. The RISC-V ISA was designed from day one to be extendable to support custom instruction enhancements and we have recently worked on a GDB patch to allow these custom instruction mnemonics to be defined in an XML file which GDB can read and use to display the instruction (and parameters) in a human-readable format. So instead of seeing this in GDB:

```
0x00000000 <+0>:     addi      sp,sp,-16
0x00000004 <+4>:     sw        s0,12(sp)
0x00000008 <+8>:     addi      s0,sp,16
0x0000000c <+12>:    .4byte    0xa5221e3 ;?
0x00000010 <+16>:    .4byte    0x12211fb ;?
0x00000014 <+20>:    .4byte    0x19453db ;?
```

You see this:

```
0x00000000 <+0>:     addi      sp,sp,-16
0x00000004 <+4>:     sw        s0,12(sp)
0x00000008 <+8>:     addi      s0,sp,16
0x0000000c <+12>:    cust1     gp, tp, t0 ;😎
0x00000010 <+16>:    cust2     gp, tp, 18 ;😎
0x00000014 <+20>:    cust3     t2, 25(fp) ;😎
```

i.e. GDB now clearly shows your custom instruction mnemonics allowing you to debug them more effectively. As you would expect, our ***RiscFree™* Debugger** can read and use this XML file too.

Thanks for reading.

Hugh @ Ashling.