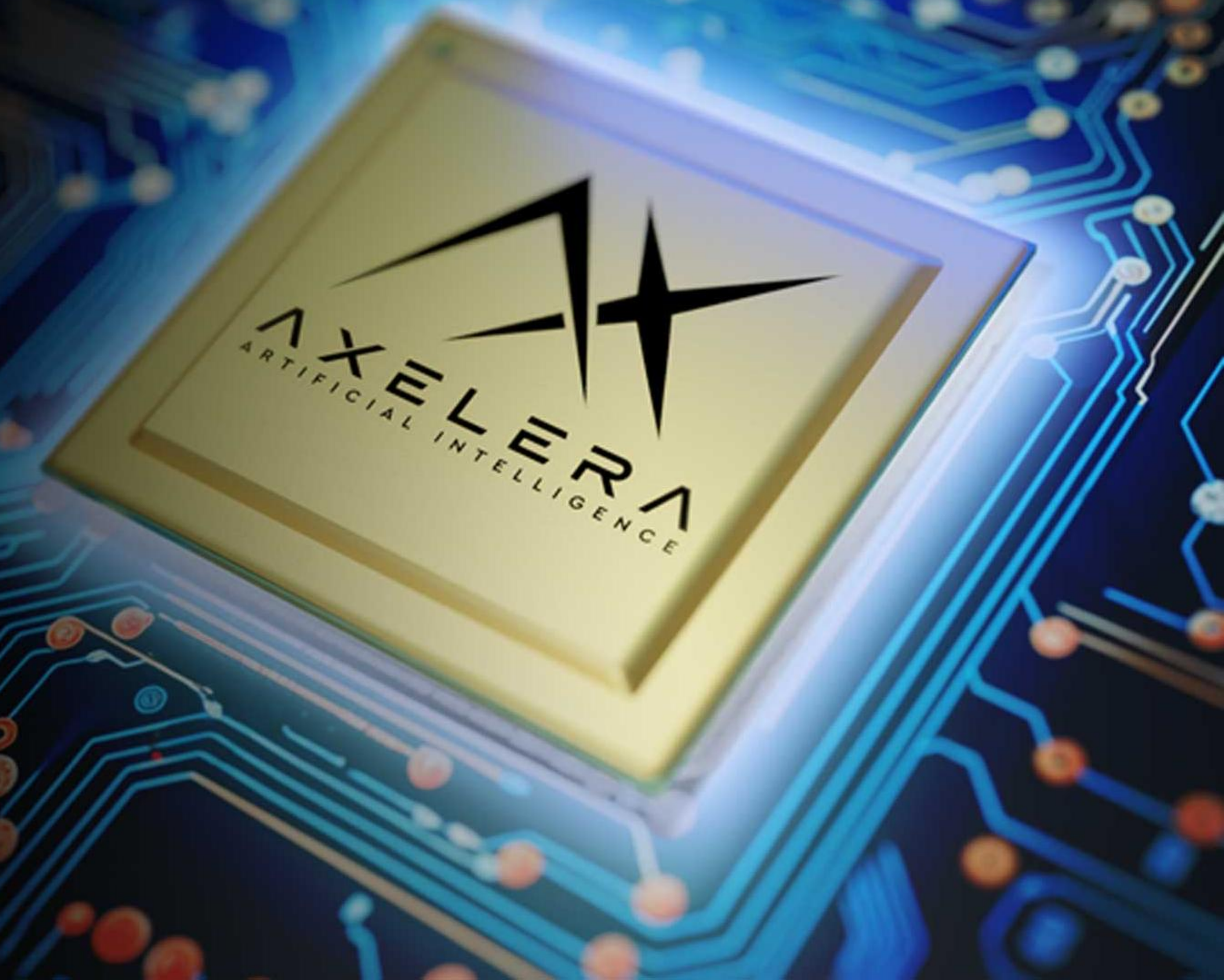
Datacenter Performance. Edge Efficiency.
Accelerating Inference, Everywhere.



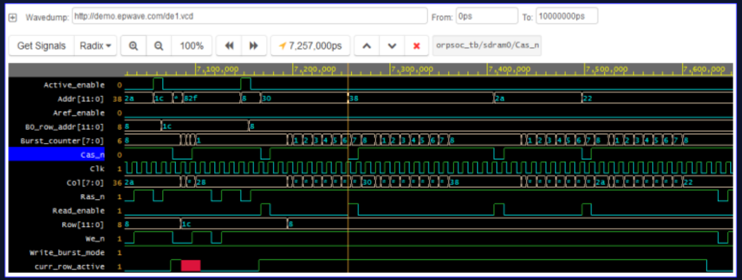
In-house tool development for offline debugging

Presenter: Jerome Sauger

23/09/2025

- Top-level verification flow
- Setup

- Our top-level tests and drivers are written in C and run on several platforms:



Simulation

Emulation

Spike, RISC-V ISA Simulator

- Top-level verification flow
- Debugging options

- Online debugging (i.e. gdb)
  - Doable for spike and emulator
  - Requires a functioning debug infrastructure at the top-level
  - Emulator resources are scarce ➔ the less time a user spends on it, the better

- Offline debugging
  - Waves:
    - CPU registers are not always easy to locate
    - Tedious to navigate
    - Get slower to generate as the size of design increases

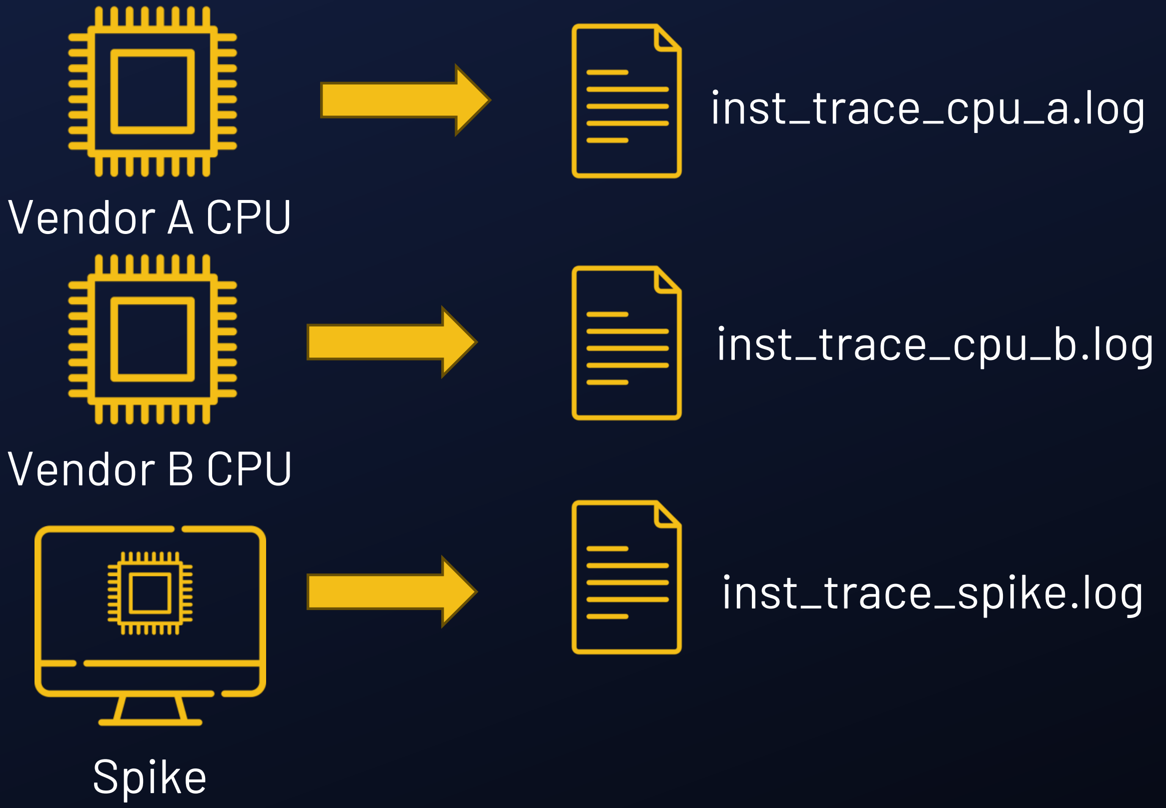
Most viable option

  - CPU trace logs
    - Automatically generated by Spike
    - Can be generated on the other platforms by binding instruction tracers to all CPUs

AXELERA ARTIFICIAL INTELLIGENCE

Parsing the trace logs

- Constraints
- Several trace formats coexist depending on the CPU or the platform

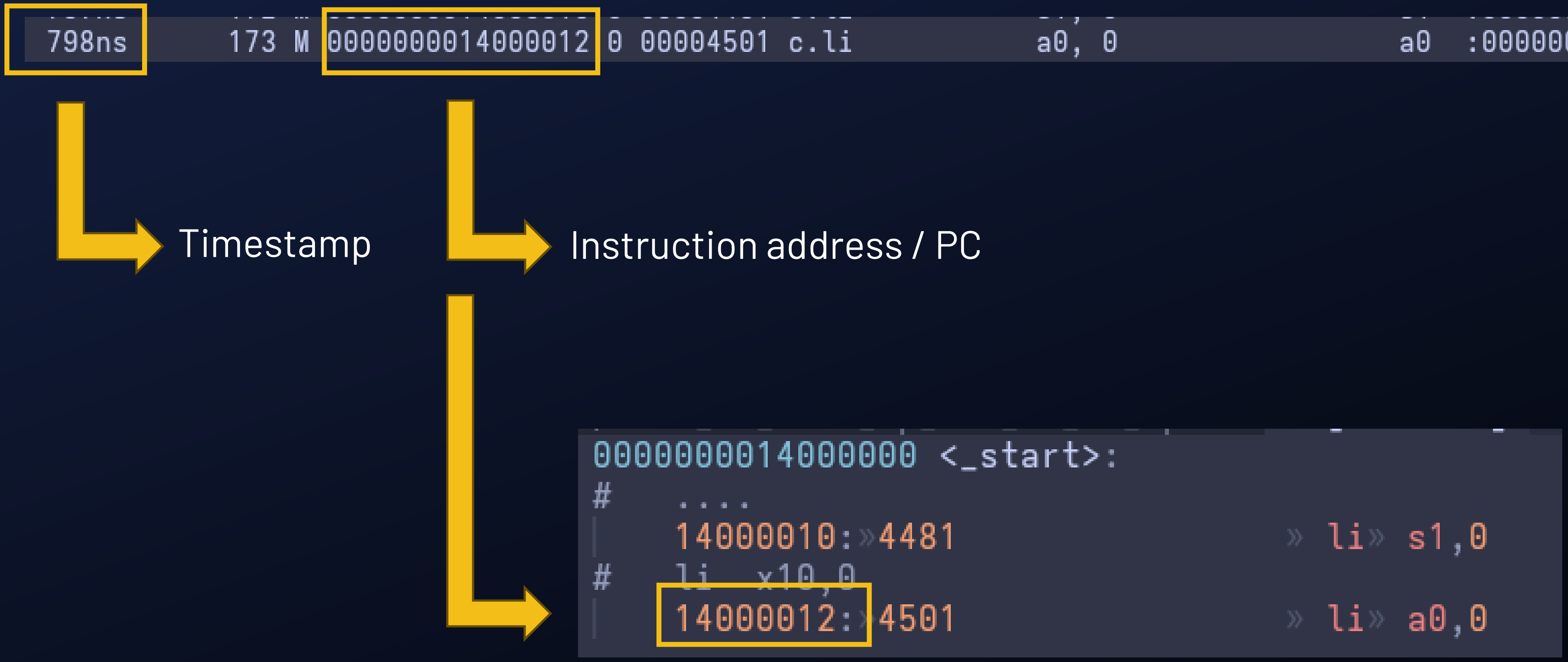


- Commercial tools exist but they require licensing and do not support these formats

➔ Creating our own tool was the way to go

Parsing the trace logs

- Trace format



```
798ns      173 M 0000000014000012 0 00004501 c.li          a0, 0                a0  :0000000000000000
```

```
0000000014000000 <_start>:
#    ....
      14000010:»4481                   » li» s1,0
#     li  x10,0
      14000012:»4501                   » li» a0,0
```

Can be matched against the disassembly of the original elf

- Parsing the trace logs
- Reconstituting the execution flow

Read the instruction address

```
1797ns       1373  M  0000000014017faa
```

```
9688ns      10846  M  00000000140180c6
```

Check if it matches a function address

```
0000000014017faa <main>:
    14017faa:»1101
```

We entered a function

Check if it matches a "ret" instruction

```
0000000014017faa <main>:
  # etc ....
    140180c6:»8082                      » ret
```

We exited a function

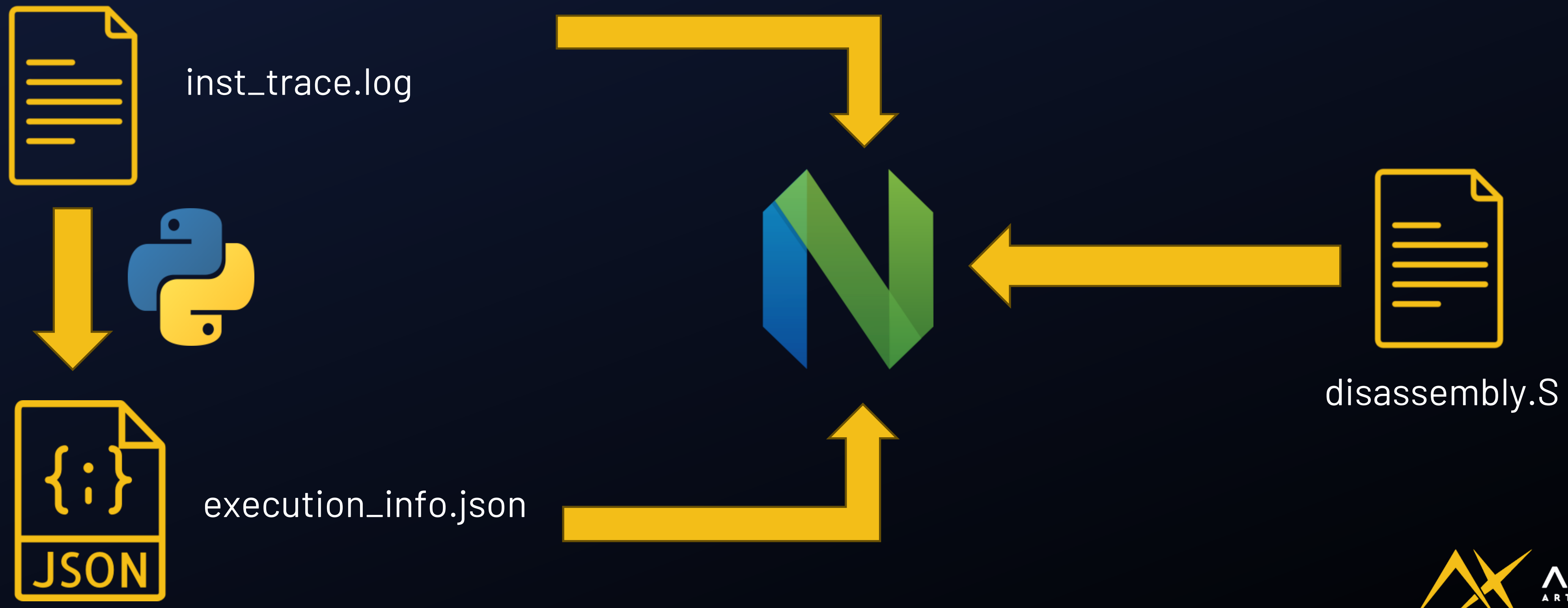
- ## Parsing the trace logs
- Quick recap

- Through simple parsing of the trace logs and the disassembly file, we have obtained:
  - The function call tree
  - Profiling information
    - How many times a function was invoked
    - How much time each invocation took
    - When each call happened

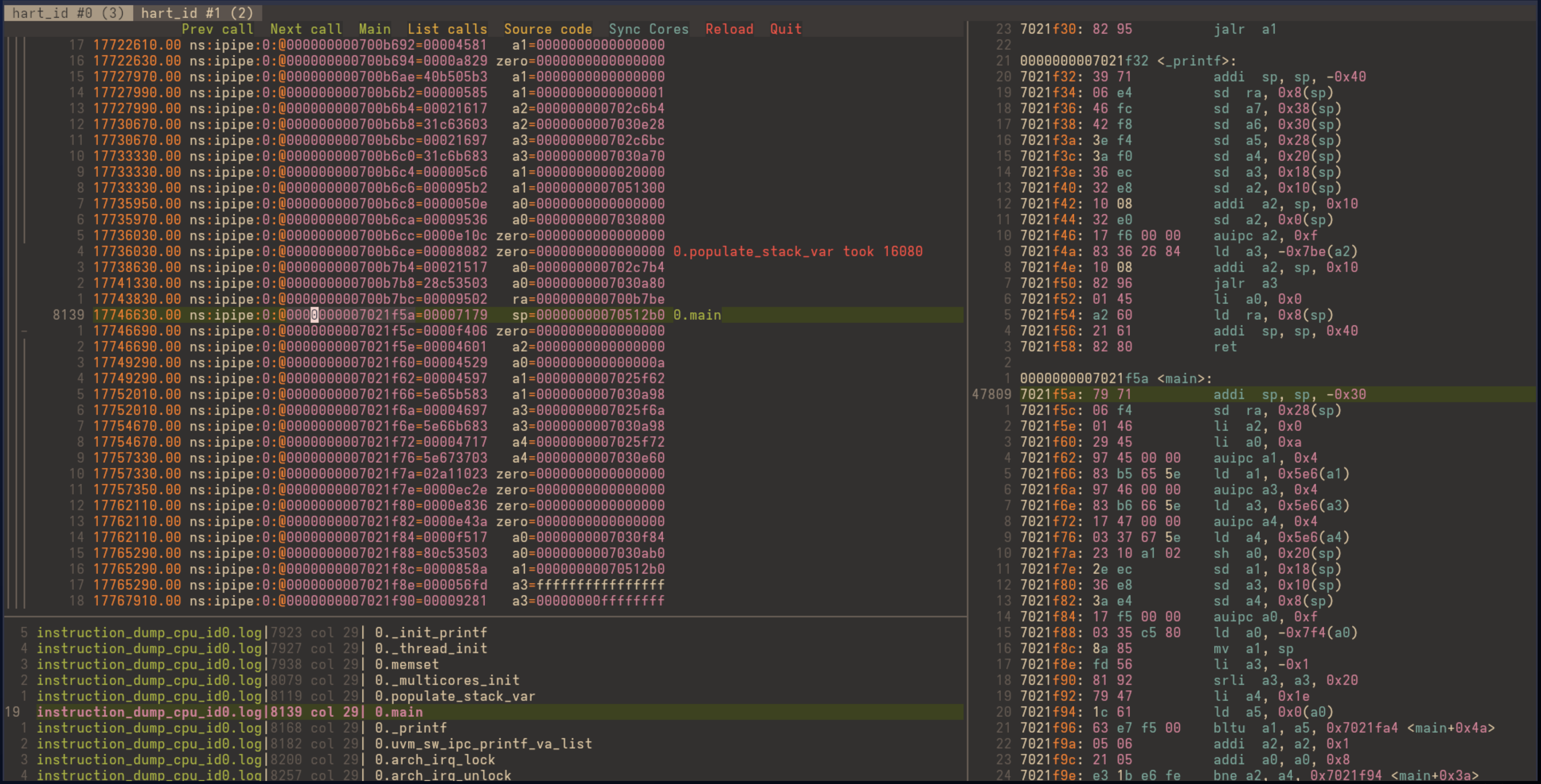
Displaying the information

- Leveraging neovim capabilities

- Neovim
  - Is installed on all our machines
  - Runs in the terminal
  - Is easily scriptable (lua)
- Final setup:



- Displaying the information
- Overview



```
17722610.00 ns:ipipe:0:@000000000700b692=00004581   a1=0000000000000000
17722630.00 ns:ipipe:0:@000000000700b694=0000a829 zero=0000000000000000
17727970.00 ns:ipipe:0:@000000000700b6ae=40b505b3   a1=0000000000000000
17727990.00 ns:ipipe:0:@000000000700b6b2=00000585   a1=0000000000000001
17727990.00 ns:ipipe:0:@000000000700b6b4=00021617   a2=000000000702c6b4
17730670.00 ns:ipipe:0:@000000000700b6b8=31c63603   a2=0000000007030e28
17730670.00 ns:ipipe:0:@000000000700b6bc=00021697   a3=000000000702c6bc
17733330.00 ns:ipipe:0:@000000000700b6c0=31c6b683   a3=0000000007030a70
17733330.00 ns:ipipe:0:@000000000700b6c4=000005c6   a1=0000000000020000
17733330.00 ns:ipipe:0:@000000000700b6c6=000095b2   a1=0000000007051300
17735950.00 ns:ipipe:0:@000000000700b6c8=0000050e   a0=0000000000000000
17735970.00 ns:ipipe:0:@000000000700b6ca=00009536   a0=0000000007030800
17736030.00 ns:ipipe:0:@000000000700b6cc=0000e10c zero=0000000000000000
17736030.00 ns:ipipe:0:@000000000700b6ce=00008082 zero=0000000000000000 0.populate_stack_var took 16080
17738630.00 ns:ipipe:0:@000000000700b7b4=00021517   a0=000000000702c7b4
17741330.00 ns:ipipe:0:@000000000700b7b8=28c53503   a0=0000000007030a80
17743830.00 ns:ipipe:0:@000000000700b7bc=00009502   ra=000000000700b7be
17746630.00 ns:ipipe:0:@0000000007021f5a=00007179   sp=00000000070512b0 0.main
17746690.00 ns:ipipe:0:@0000000007021f5c=0000f406 zero=0000000000000000
17746690.00 ns:ipipe:0:@0000000007021f5e=00004601   a2=0000000000000000
17749290.00 ns:ipipe:0:@0000000007021f60=00004529   a0=000000000000000a
17749290.00 ns:ipipe:0:@0000000007021f62=00004597   a1=0000000007025f62
17752010.00 ns:ipipe:0:@0000000007021f66=5e65b583   a1=0000000007030a98
17752010.00 ns:ipipe:0:@0000000007021f6a=00004697   a3=0000000007025f6a
17754670.00 ns:ipipe:0:@0000000007021f6e=5e66b683   a3=0000000007030a98
17754670.00 ns:ipipe:0:@0000000007021f72=00004717   a4=0000000007025f72
17757330.00 ns:ipipe:0:@0000000007021f76=5e673703   a4=0000000007030e60
17757330.00 ns:ipipe:0:@0000000007021f7a=02a11023 zero=0000000000000000
17757350.00 ns:ipipe:0:@0000000007021f7e=0000ec2e zero=0000000000000000
17762110.00 ns:ipipe:0:@0000000007021f80=0000e836 zero=0000000000000000
17762110.00 ns:ipipe:0:@0000000007021f82=0000e43a zero=0000000000000000
17762110.00 ns:ipipe:0:@0000000007021f84=0000f517   a0=0000000007030f84
17765290.00 ns:ipipe:0:@0000000007021f88=80c53503   a0=0000000007030ab0
17765290.00 ns:ipipe:0:@0000000007021f8c=0000858a   a1=00000000070512b0
17765290.00 ns:ipipe:0:@0000000007021f8e=000056fd   a3=ffffffffffffffff
17767910.00 ns:ipipe:0:@0000000007021f90=00009281   a3=00000000ffffffff
```

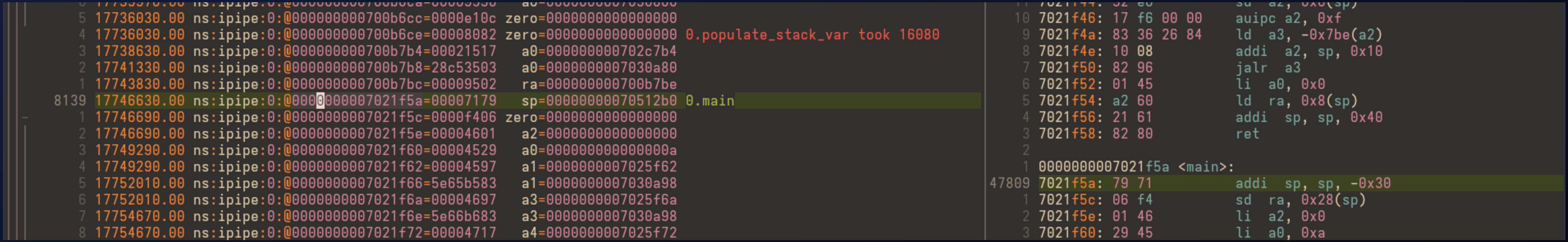
```
instruction_dump_cpu_id0.log|7923 col 29| 0._init_printf
instruction_dump_cpu_id0.log|7927 col 29| 0._thread_init
instruction_dump_cpu_id0.log|7938 col 29| 0.memset
instruction_dump_cpu_id0.log|8079 col 29| 0._multicores_init
instruction_dump_cpu_id0.log|8119 col 29| 0.populate_stack_var
instruction_dump_cpu_id0.log|8139 col 29| 0.main
instruction_dump_cpu_id0.log|8168 col 29| 0._printf
instruction_dump_cpu_id0.log|8182 col 29| 0.uvm_sw_ipc_printf_va_list
instruction_dump_cpu_id0.log|8200 col 29| 0.arch_irq_lock
instruction_dump_cpu_id0.log|8257 col 29| 0.arch_irq_unlock
```

```
7021f30: 82 95          jalr  a1

000000007021f32 <_printf>:
7021f32: 39 71          addi  sp, sp, -0x40
7021f34: 06 e4          sd  ra, 0x8(sp)
7021f36: 46 fc          sd  a7, 0x38(sp)
7021f38: 42 f8          sd  a6, 0x30(sp)
7021f3a: 3e f4          sd  a5, 0x28(sp)
7021f3c: 3a f0          sd  a4, 0x20(sp)
7021f3e: 36 ec          sd  a3, 0x18(sp)
7021f40: 32 e8          sd  a2, 0x10(sp)
7021f42: 10 08          addi  a2, sp, 0x10
7021f44: 32 e0          sd  a2, 0x0(sp)
7021f46: 17 f6 00 00    auipc a2, 0xf
7021f4a: 83 36 26 84    ld  a3, -0x7be(a2)
7021f4e: 10 08          addi  a2, sp, 0x10
7021f50: 82 96          jalr  a3
7021f52: 01 45          li  a0, 0x0
7021f54: a2 60          ld  ra, 0x8(sp)
7021f56: 21 61          addi  sp, sp, 0x40
7021f58: 82 80          ret

000000007021f5a <main>:
7021f5a: 79 71          addi  sp, sp, -0x30
7021f5c: 06 f4          sd  ra, 0x28(sp)
7021f5e: 01 46          li  a2, 0x0
7021f60: 29 45          li  a0, 0xa
7021f62: 97 45 00 00    auipc a1, 0x4
7021f66: 83 b5 65 5e    ld  a1, 0x5e6(a1)
7021f6a: 97 46 00 00    auipc a3, 0x4
7021f6e: 83 b6 66 5e    ld  a3, 0x5e6(a3)
7021f72: 17 47 00 00    auipc a4, 0x4
7021f76: 03 37 67 5e    ld  a4, 0x5e6(a4)
7021f7a: 23 10 a1 02    sh  a0, 0x20(sp)
7021f7e: 2e ec          sd  a1, 0x18(sp)
7021f80: 36 e8          sd  a3, 0x10(sp)
7021f82: 3a e4          sd  a4, 0x8(sp)
7021f84: 17 f5 00 00    auipc a0, 0xf
7021f88: 03 35 c5 80    ld  a0, -0x7f4(a0)
7021f8c: 8a 85          mv  a1, sp
7021f8e: fd 56          li  a3, -0x1
7021f90: 81 92          srli  a3, a3, 0x20
7021f92: 79 47          li  a4, 0x1e
7021f94: 1c 61          ld  a5, 0x0(a0)
7021f96: 63 e7 f5 00    bltu  a1, a5, 0x7021fa4 <main+0x4a>
7021f9a: 05 06          addi  a2, a2, 0x1
7021f9c: 21 05          addi  a0, a0, 0x8
7021f9e: e3 1b e6 fe    bne a2, a4, 0x7021f94 <main+0x3a>
```

Displaying the information

- Features

- Automated synchronization between trace and disassembly

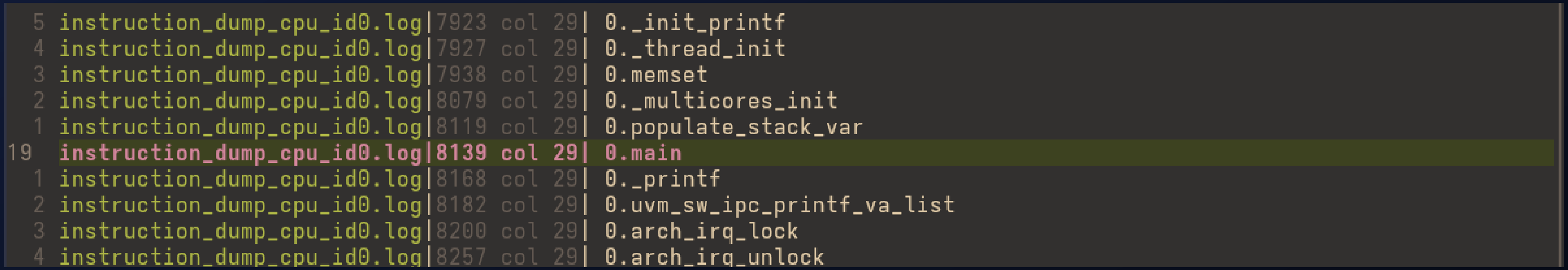


```
17736030.00 ns:ipipe:0:@000000000700b6cc=0000e10c zero=0000000000000000
17736030.00 ns:ipipe:0:@000000000700b6ce=00008082 zero=0000000000000000 0.populate_stack_var took 16080
17738630.00 ns:ipipe:0:@000000000700b7b4=00021517   a0=000000000702c7b4
17741330.00 ns:ipipe:0:@000000000700b7b8=28c53503   a0=0000000007030a80
17743830.00 ns:ipipe:0:@000000000700b7bc=00009502   ra=000000000700b7be
17746630.00 ns:ipipe:0:@000000007021f5a=00007179   sp=00000000070512b0 0.main
17746690.00 ns:ipipe:0:@0000000007021f5c=0000f406 zero=0000000000000000
17746690.00 ns:ipipe:0:@0000000007021f5e=00004601   a2=0000000000000000
17749290.00 ns:ipipe:0:@0000000007021f60=00004529   a0=000000000000000a
17749290.00 ns:ipipe:0:@0000000007021f62=00004597   a1=0000000007025f62
17752010.00 ns:ipipe:0:@0000000007021f66=5e65b583   a1=0000000007030a98
17752010.00 ns:ipipe:0:@0000000007021f6a=00004697   a3=0000000007025f6a
17754670.00 ns:ipipe:0:@0000000007021f6e=5e66b683   a3=0000000007030a98
17754670.00 ns:ipipe:0:@0000000007021f72=00004717   a4=0000000007025f72
```

```
7021f46: 17 f6 00 00    auipc a2, 0xf
7021f4a: 83 36 26 84    ld  a3, -0x7be(a2)
7021f4e: 10 08          addi  a2, sp, 0x10
7021f50: 82 96          jalr  a3
7021f52: 01 45          li  a0, 0x0
7021f54: a2 60          ld  ra, 0x8(sp)
7021f56: 21 61          addi  sp, sp, 0x40
7021f58: 82 80          ret

000000007021f5a <main>:
7021f5a: 79 71          addi  sp, sp, -0x30
7021f5c: 06 f4          sd  ra, 0x28(sp)
7021f5e: 01 46          li  a2, 0x0
7021f60: 29 45          li  a0, 0xa
```

- Displaying the information
- Features

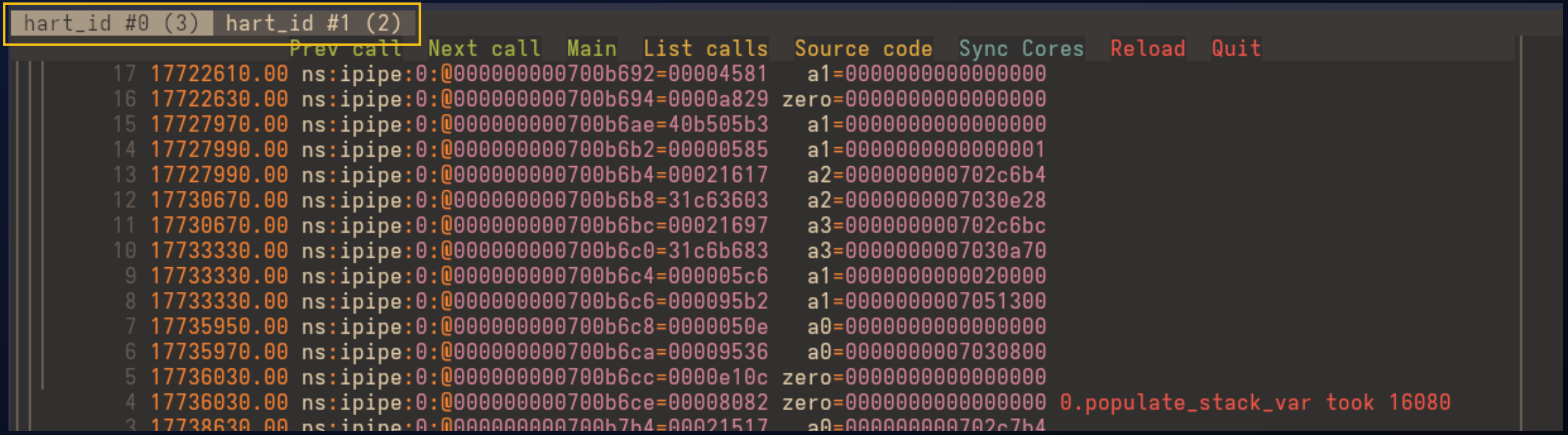
- Possibility to navigate from one function to another



```
instruction_dump_cpu_id0.log|7923 col 29| 0._init_printf
instruction_dump_cpu_id0.log|7927 col 29| 0._thread_init
instruction_dump_cpu_id0.log|7938 col 29| 0.memset
instruction_dump_cpu_id0.log|8079 col 29| 0._multicores_init
instruction_dump_cpu_id0.log|8119 col 29| 0.populate_stack_var
instruction_dump_cpu_id0.log|8139 col 29| 0.main
instruction_dump_cpu_id0.log|8168 col 29| 0._printf
instruction_dump_cpu_id0.log|8182 col 29| 0.uvm_sw_ipc_printf_va_list
instruction_dump_cpu_id0.log|8200 col 29| 0.arch_irq_lock
instruction_dump_cpu_id0.log|8257 col 29| 0.arch_irq_unlock
```

- Displaying the information
- Features

- Visualize multiple CPU traces at the same time
- Possibility to synchronize the cursors

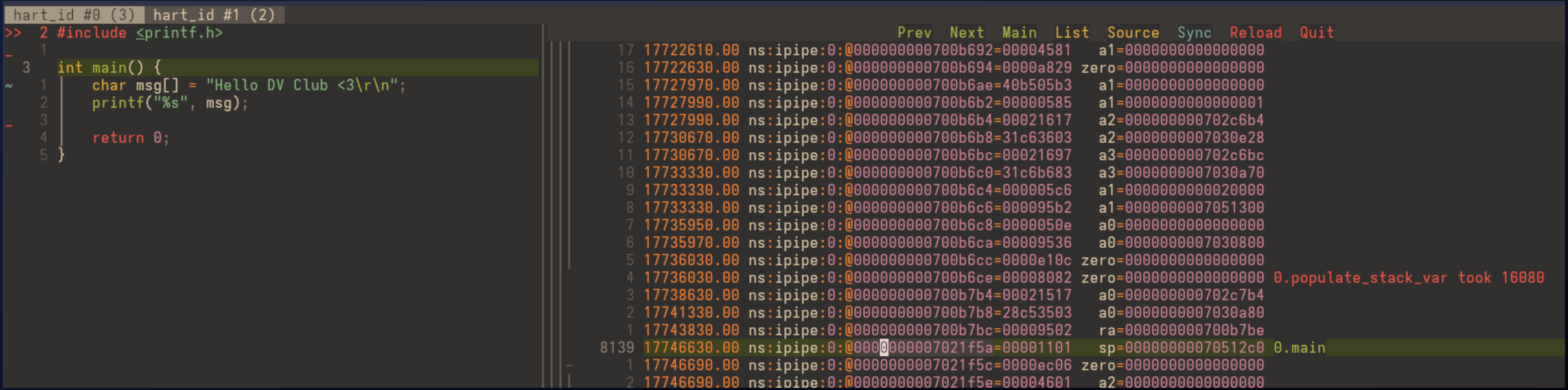


```
17722610.00 ns:ipipe:0:@000000000700b692=00004581   a1=0000000000000000
17722630.00 ns:ipipe:0:@000000000700b694=0000a829 zero=0000000000000000
17727970.00 ns:ipipe:0:@000000000700b6ae=40b505b3   a1=0000000000000000
17727990.00 ns:ipipe:0:@000000000700b6b2=00000585   a1=0000000000000001
17727990.00 ns:ipipe:0:@000000000700b6b4=00021617   a2=00000000702c6b4
17730670.00 ns:ipipe:0:@000000000700b6b8=31c63603   a2=000000007030e28
17730670.00 ns:ipipe:0:@000000000700b6bc=00021697   a3=00000000702c6bc
17733330.00 ns:ipipe:0:@000000000700b6c0=31c6b683   a3=000000007030a70
17733330.00 ns:ipipe:0:@000000000700b6c4=000005c6   a1=0000000000020000
17733330.00 ns:ipipe:0:@000000000700b6c6=000095b2   a1=000000007051300
17735950.00 ns:ipipe:0:@000000000700b6c8=0000050e   a0=0000000000000000
17735970.00 ns:ipipe:0:@000000000700b6ca=00009536   a0=000000007030800
17736030.00 ns:ipipe:0:@000000000700b6cc=0000e10c zero=0000000000000000
17736030.00 ns:ipipe:0:@000000000700b6ce=00008082 zero=0000000000000000 0.populate_stack_var took 16080
17738630.00 ns:ipipe:0:@000000000700b7b4=00021517   a0=00000000702c7b4
```

AXELERA
ARTIFICIAL INTELLIGENCE

- Viewing source code

- The source code line for an instruction can be obtained with **riscv64-unknown-elf-addr2line**



```c
#include <printf.h>

int main() {
    char msg[] = "Hello DV Club <3\r\n";
    printf("%s", msg);

    return 0;
}
```

```
17722610.00 ns:ipipe:0:@000000000700b692=00004581   a1=0000000000000000
17722630.00 ns:ipipe:0:@000000000700b694=0000a829 zero=0000000000000000
17727970.00 ns:ipipe:0:@000000000700b6ae=40b505b3   a1=0000000000000000
17727990.00 ns:ipipe:0:@000000000700b6b2=00000585   a1=0000000000000001
17727990.00 ns:ipipe:0:@000000000700b6b4=00021617   a2=000000000702c6b4
17730670.00 ns:ipipe:0:@000000000700b6b8=31c63603   a2=0000000007030e28
17730670.00 ns:ipipe:0:@000000000700b6bc=00021697   a3=000000000702c6bc
17733330.00 ns:ipipe:0:@000000000700b6c0=31c6b683   a3=0000000007030a70
17733330.00 ns:ipipe:0:@000000000700b6c4=000005c6   a1=0000000000020000
17733330.00 ns:ipipe:0:@000000000700b6c6=000095b2   a1=0000000007051300
17735950.00 ns:ipipe:0:@000000000700b6c8=0000050e   a0=0000000000000000
17735970.00 ns:ipipe:0:@000000000700b6ca=00009536   a0=0000000007030800
17736030.00 ns:ipipe:0:@000000000700b6cc=0000e10c zero=0000000000000000
17736030.00 ns:ipipe:0:@000000000700b6ce=00008082 zero=0000000000000000 0.populate_stack_var took 16080
17738630.00 ns:ipipe:0:@000000000700b7b4=00021517   a0=000000000702c7b4
17741330.00 ns:ipipe:0:@000000000700b7b8=28c53503   a0=0000000007030a80
17743830.00 ns:ipipe:0:@000000000700b7bc=00009502   ra=000000000700b7be
17746630.00 ns:ipipe:0:@00000000007021f5a=00001101   sp=00000000070512c0 0.main
17746690.00 ns:ipipe:0:@0000000007021f5c=0000ec06 zero=0000000000000000
17746690.00 ns:ipipe:0:@0000000007021f5e=00004601   a2=0000000000000000
```

Conclusion

- C tests are best debugged with CPU traces
- Perks of developing our tool:
  - The initial version took only a week of development
  - Free
  - Useful to everyone doing firmware
  - Tailored to our use cases
  - Not tied to any simulator/specific tool

AXELERA
ARTIFICIAL INTELLIGENCE

Thank You!