

VexiiRiscv : A Debian demonstration

```
root@nexys:~# neofetch
```

```
      _ ,met$$$$$gg.
    ,g$$$$$$$$$$$$$$$P.
  ,g$$P"         ""Y$$".
 ,$$P'           `$$$ .
',$$P           ,ggs.   `$$b:
`d$$'           ,P"'    $$$
 $$P            d$'     $$P
 $$:            $$      -   ,d$$'
 $$;            Y$b._   _ ,d$P'
 Y$$           `."Y$$$$$P"'
`$$b           "-._
`Y$$
`Y$$
`$$b.
`Y$$b.
`"Y$b._
  ""
```

root@nexys

OS: Debian GNU/Linux trixie/sid riscv64

Kernel: 6.10.0-rc2+

Uptime: 1 min

Packages: 2101 (dpkg)

Shell: bash 5.2.15

Terminal: /dev/ttyLXU0

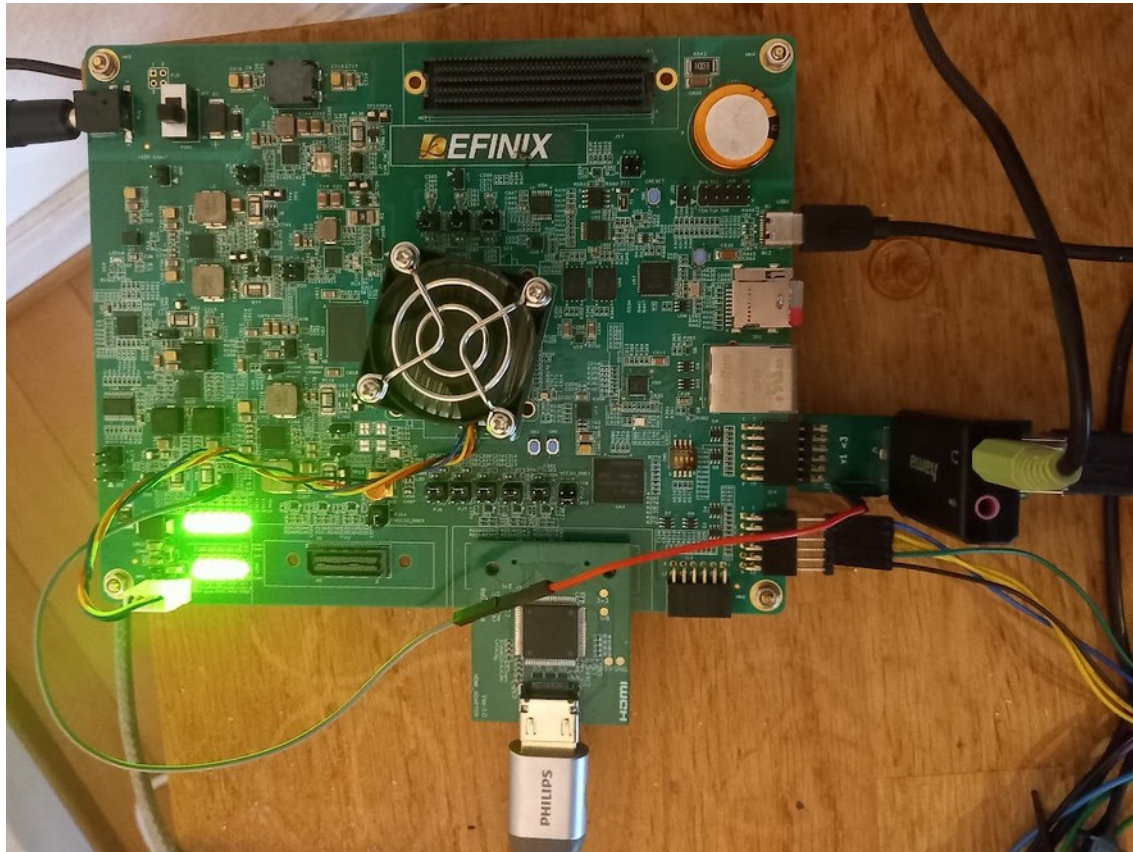
CPU: (4)

Memory: 81MiB / 987MiB



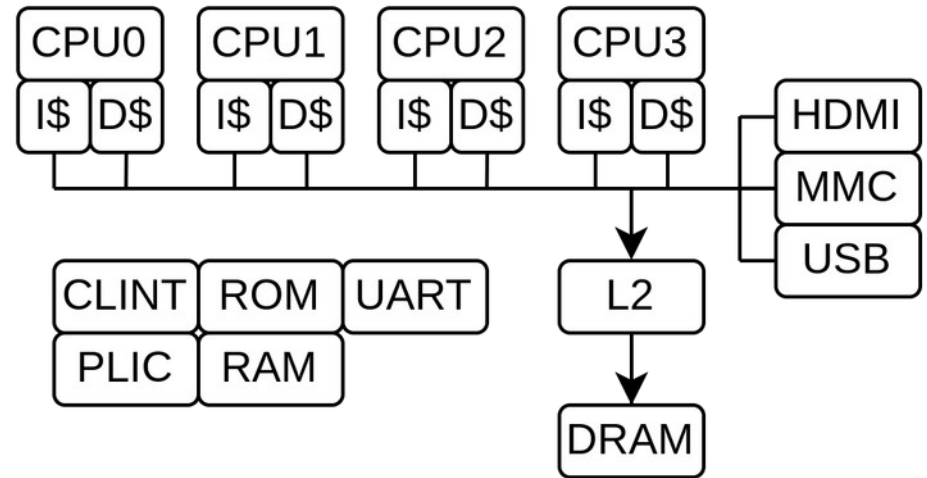
Whoami (FPGA + Board)

- Efinix Titanium Ti375 C529 Development Kit



Whoami (SoC)

- 4x VexiiRiscv @ 200MHz
 - RV64GC...
 - Single issue / in-order
 - BTB / GShare / RAS
 - Store buffer
 - Hardware prefetcher / non-blocking caches (I\$ D\$)
 - 16KB I\$, 16KB D\$ (per core)
- L2 (512KB, non-blocking, out-of-order)
- Tilelink bus (128 bits, memory coherent)
- SpinalHDL + Litex → Verilog



```
EFX_ADD           : 8679
EFX_LUT4          : 87660
    1-2 Inputs    : 12009
    3     Inputs  : 34680
    4     Inputs  : 40971
EFX_DSP48         : 68
EFX_FF            : 56329
EFX_SRL8          : 106
EFX_RAM10         : 1284
```

VexiiRiscv feature set

- Mainly :
 - RV32/RV64, optional MAFDCBSU
 - Can run linux / buildroot / Debian
 - Optional multi-issue
 - Optional late-ALU
 - Up to 5.24 coremark/Mhz 2.50 dhystone/Mhz
 - Free / open-source
- Aswell :
 - Optional I\$ D\$
 - Optional BTB / RAS / Gshare branch prediction
 - RISC-V official jtag debug
 - ...

Hardware description

ExecutePipelinePlugin

1..n

ExecuteLanePlugin

lane0 + lane1 (For a dual issue CPU)

1..n

LaneLayer

earlyAlu0 + lateAlu0 (in lane 0)

1..n

UopLayerSpec

ADD, OR, LW, SW, ...

completion
mayFlushUpTo
dontFlushFrom
decodings

1

1..n

RdSpec

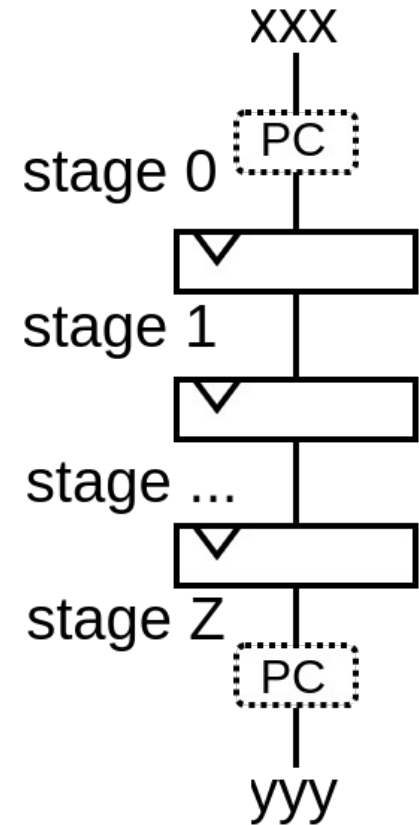
rf
data
..

RsSpec

rf
rs
from

IntRegFile
RS1
execute[0]

stage(0) (PC) := xxx
yyy := *stage*(Z) (PC)



plugins += new execute.fpu.FpuSqrtPlugin(*early0*)

Contributions

- Already a few
 - RVB (Andreas Wallner)
 - Compact divider (Thomas Kramer)
- Looking for help
 - Features additions
 - Alternative implementations
 - Performance improvements
 - Testing
 - targeted implementations (ASIC)
 - ...
- <https://github.com/SpinalHDL/VexiiRiscv>
- Thanks NLnet