

Open source CPU and SoC design: The flow, the challenges and a perspective

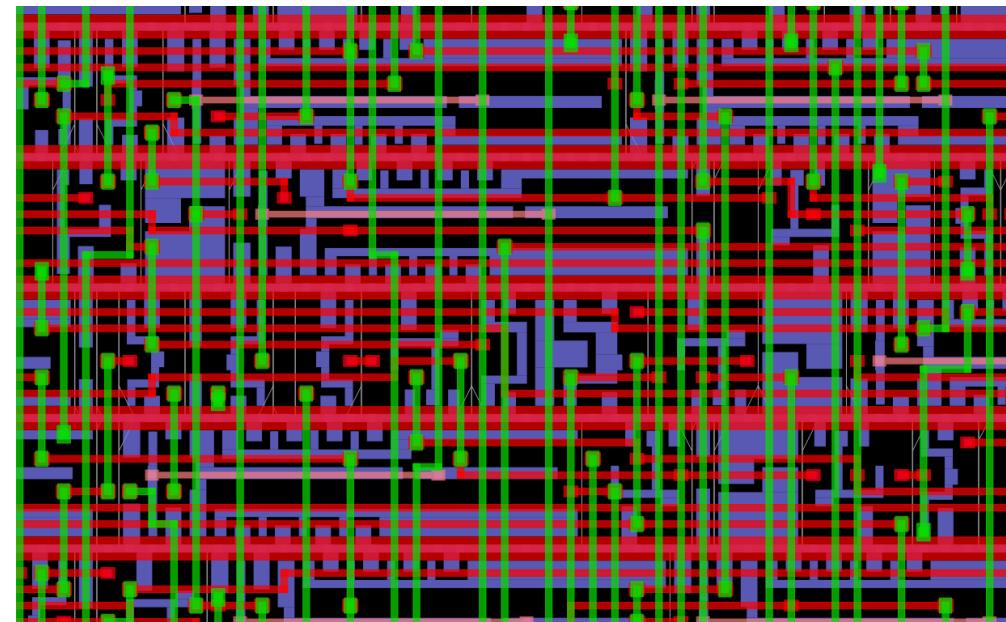
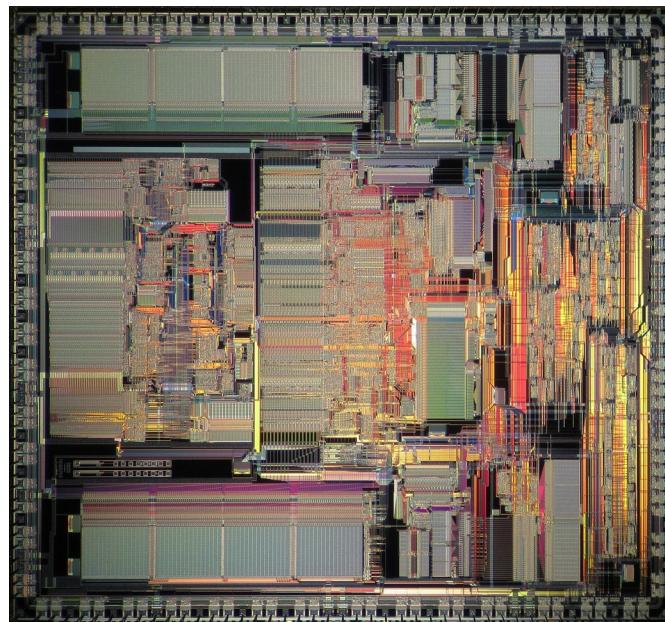
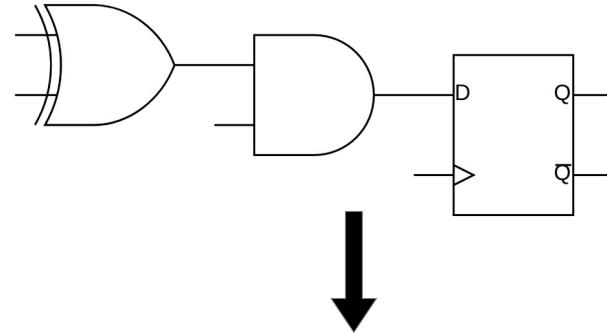
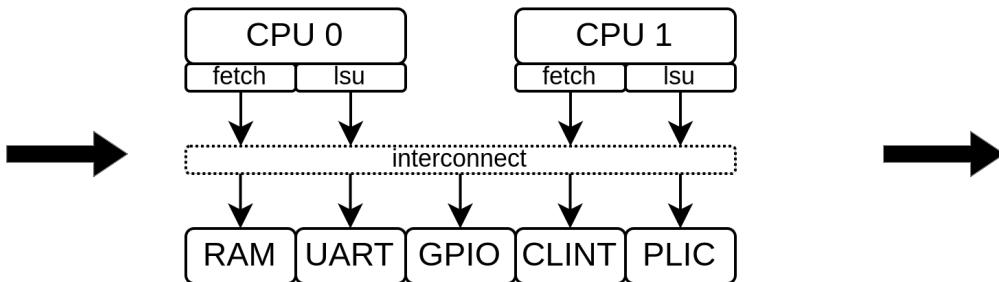
```
[1377614.579833] rcu: INFO: rcu_sched self-detected stall on CPU
[1377614.579845] rcu: 18.....: (2099 ticks this GP) idle=54e/1/0x4000000000000002
[1377614.579895] (t=2100 jiffies g=155867385 q=20879)
[1377614.579898] Task dump for CPU 18:
[1377614.579899] CPU 1/KVM R running task 0 1030947 256019 0x06000004
[1377614.579902] Call Trace:
[1377614.579912] (<0000001f1f4b4f52>) show_stack+0x7a/0xc0)
[1377614.579918] (<0000001f1ec8e96c>) sched_show_task.part.0+0xdc/0x100
[1377614.579919] (<0000001f1f4b7248>) rcu_dump_cpu_stacks+0xc0/0x100
[1377614.579924] (<0000001f1ecdd10c>) rcu_sched_clock_irq+0x75c/0x980
[1377614.579926] (<0000001f1eceb26c>) update_process_times+0x3c/0x80
[1377614.579931] (<0000001f1ecfcfea>) tick_sched_handle.isra.0+0x4a/0x70
[1377614.579932] (<0000001f1ecfd28e>) tick_sched_timer+0x5e/0xc0
[1377614.579933] (<0000001f1ecec294>) __hrtimer_run_queues+0x114/0x2f0
[1377614.579935] (<0000001f1ececfdc>) hrtimer_interrupt+0x12c/0x2a0
[1377614.579938] (<0000001f1ebecb6a>) do_IRQ+0xaa/0xb0
[1377614.579942] (<0000001f1f4c6d08>) ext_int_handler+0x130/0x134
[1377614.579945] (<0000001f1ec0af10>) ptep_zap_key+0x40/0x60
```

Background / whoami

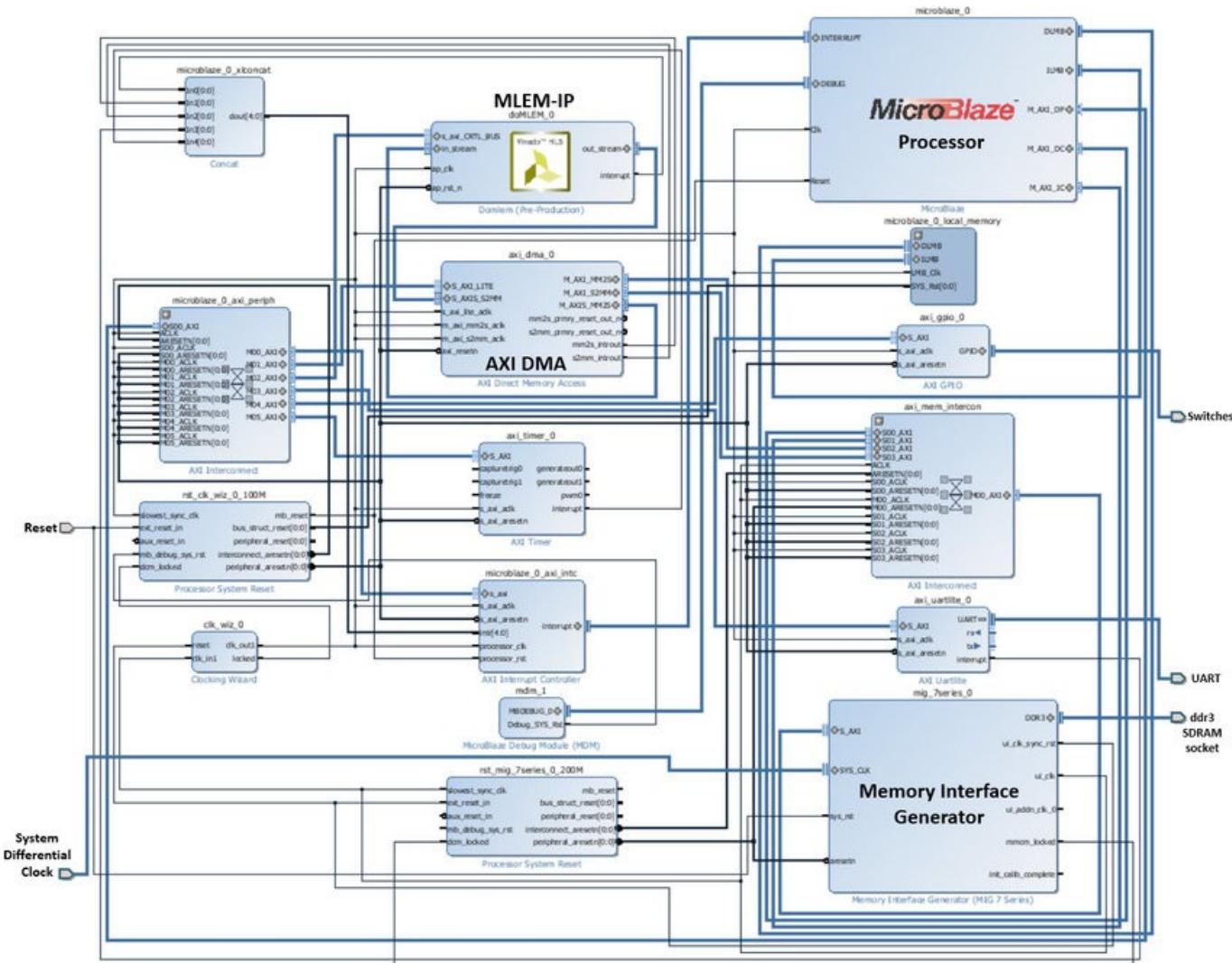
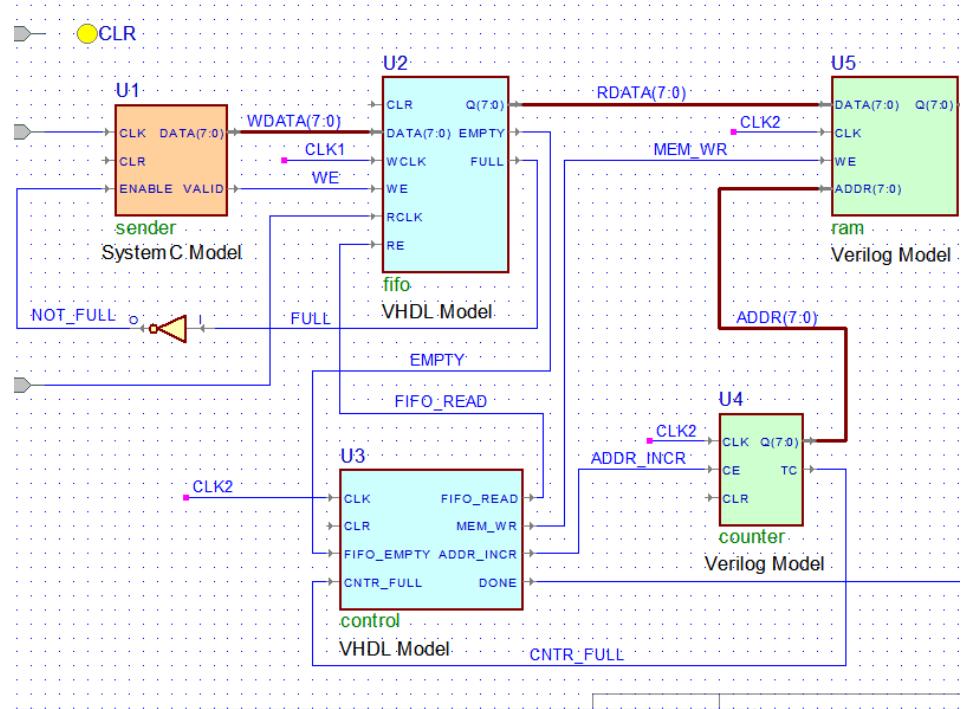
- Dolu1990 on github, independent dev
- Software / Hardware background
 - Industrial system / Electronic degree
- Active on open/free project
 - SpinalHDL (2015) : Hardware Description Library
 - VexRiscv (2017) NaxRiscv(2021) VexiiRiscv (2023) : RISC-V CPUs
- Roadmap for this talk
 - Big introduction to hardware design
 - Issues / challenges

Digital Hardware design

Human



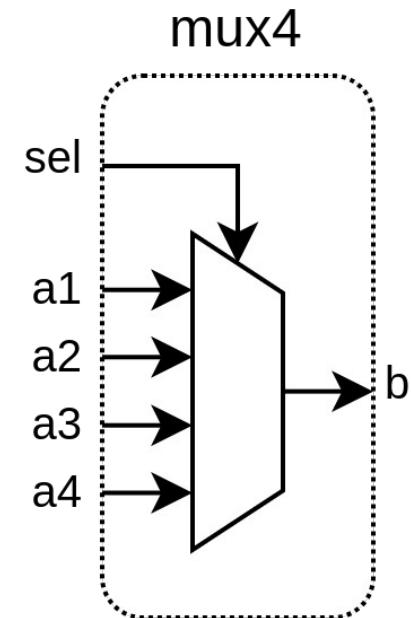
Schematic design



HDL based design

```
library IEEE;
use IEEE.std_logic_1164.all;
entity mux4 is
  port(
    a1      : in  std_logic_vector(2 downto 0);
    a2      : in  std_logic_vector(2 downto 0);
    a3      : in  std_logic_vector(2 downto 0);
    a4      : in  std_logic_vector(2 downto 0);
    sel     : in  std_logic_vector(1 downto 0);
    b       : out std_logic_vector(2 downto 0)
  );
end mux4;

architecture rtl of mux4 is
  -- declarative part: empty
begin
  p_mux : process(a1,a2,a3,a4,sel)
  begin
    case sel is
      when "00" => b <= a1 ;
      when "01" => b <= a2 ;
      when "10" => b <= a3 ;
      when others => b <= a4 ;
    end case;
  end process p_mux;
end rtl;
```



HDL based design

- VHDL / [System]Verilog
 - Industry standard / teached in universities
 - Will get you a job
 - Unproductive / verbose / limited / cursed
 - Throw enough manpower until the work is done
 - Will not attract people with software background

```
def isOdd(value : Int) : Boolean = {
    if (value == 1) return true
    if (value == 2) return false
    if (value == 3) return true
    if (value == 4) return false
    if (value == 5) return true
    if (value == 6) return false
    if (value == 7) return true
    if (value == 8) return false
    if (value == 9) return true
```

Open-source alternatives

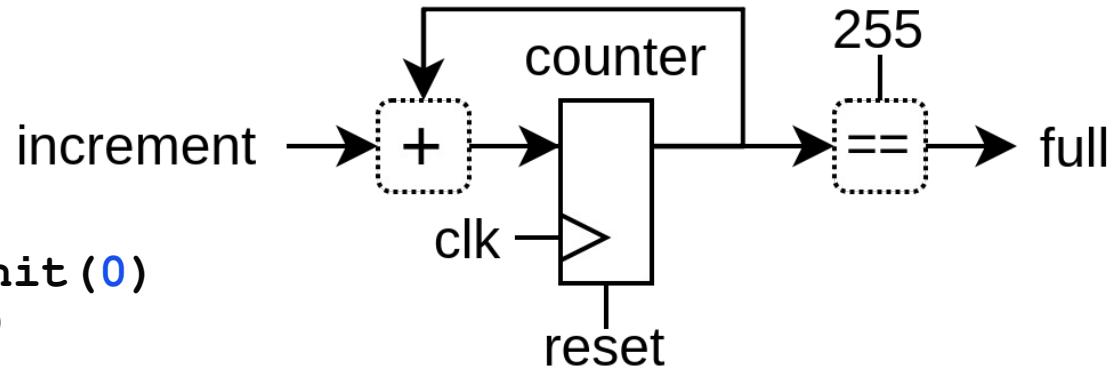
- SpinalHDL / Chisel / Migen / Amaranth / ...
 - Embedded in general purpose programming languages



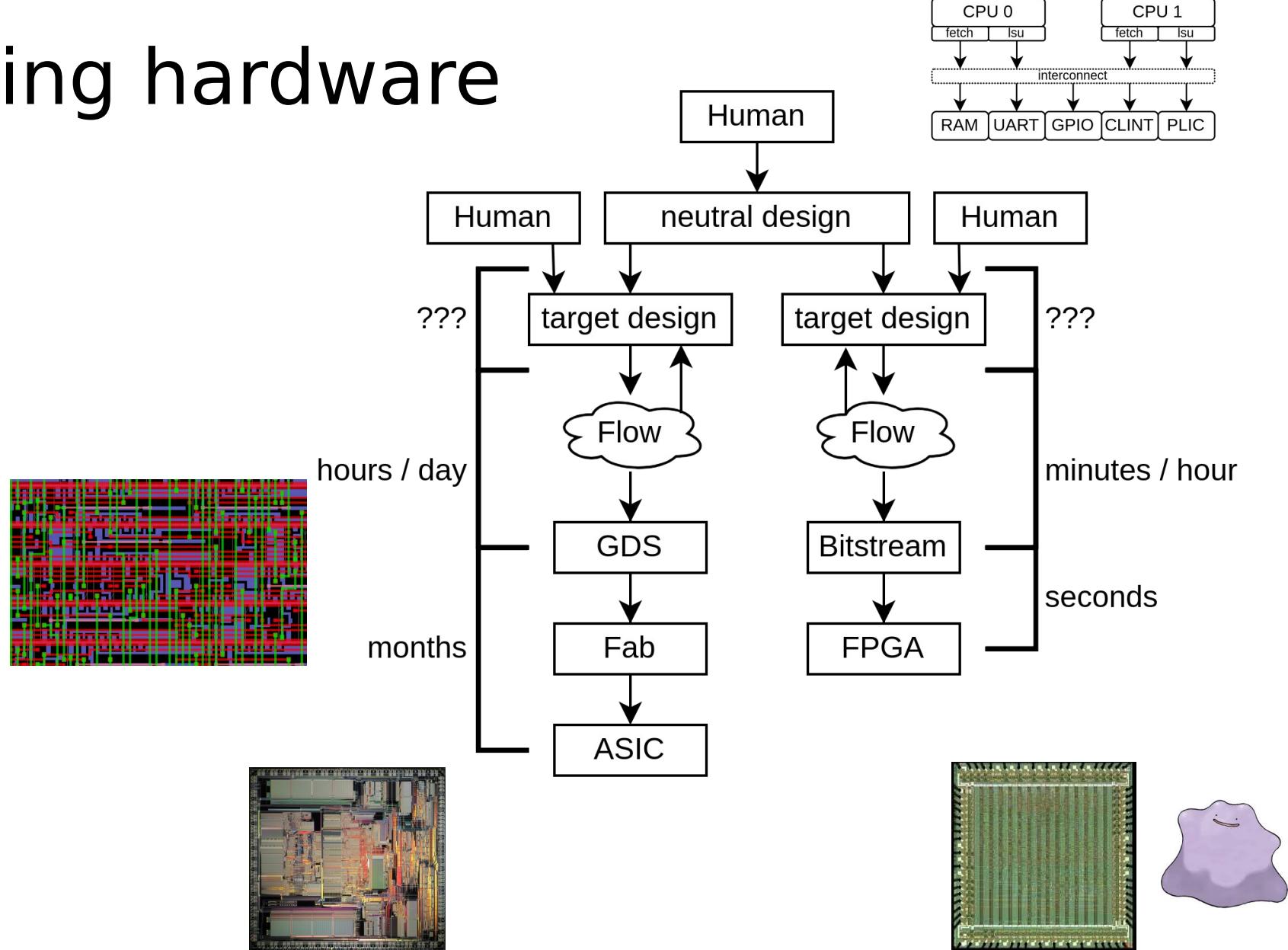
```
import spinal.core._

object MyMain extends App{
  SpinalVerilog(new Timer)
}

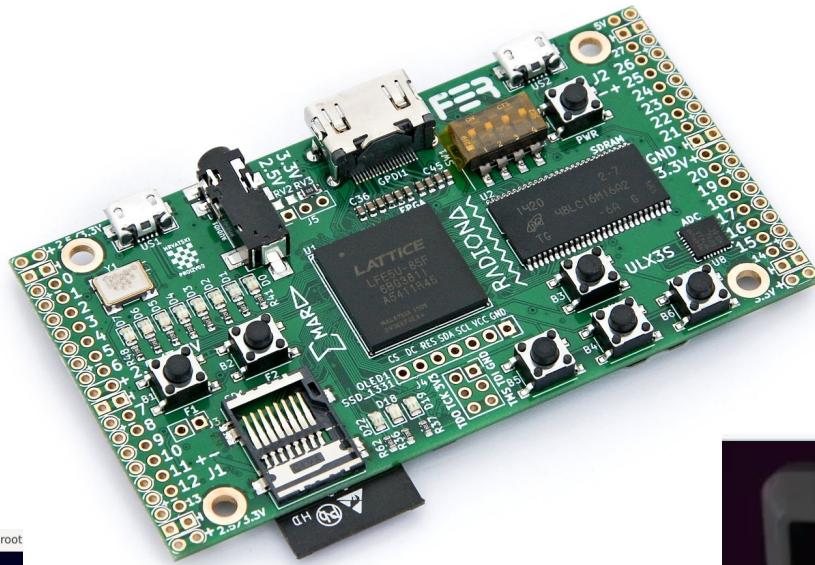
class Timer extends Component {
  val increment = in(Bool())
  val counter  = Reg(UInt(8 bits)).init(0)
  val full     = out(counter === 255)
  when(increment){
    counter := counter + 1
  }
}
```



Deploying hardware



FPGA => easy for open-source



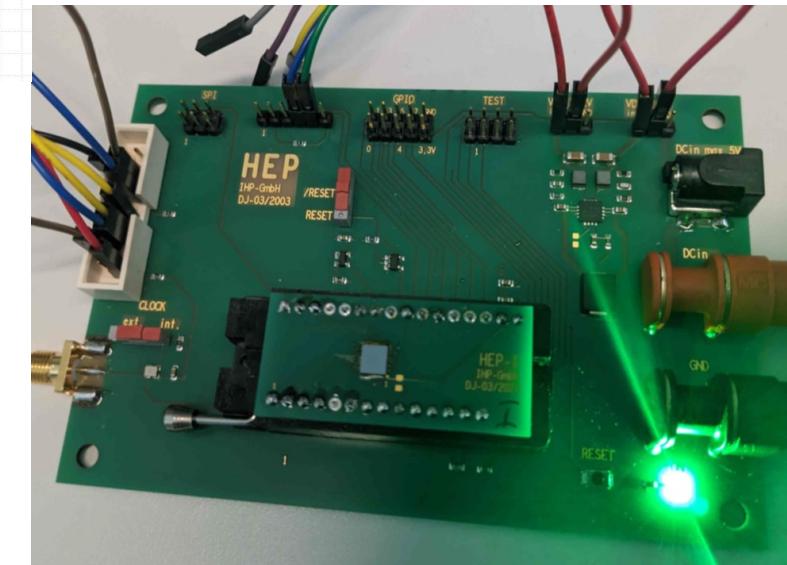
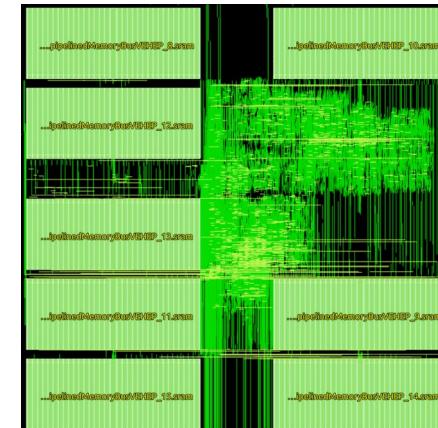
ASIC

- It is difficult
 - ASIC cost
 - Very stressful (one shot)
 - Slow iteration rate
 - Very big / complex toolchain
 - Culture of secrecy / NDA
- Open-source push
 - Openroad / Coriolis / Klayout / Magic
 - Sky130 / Tinytapout

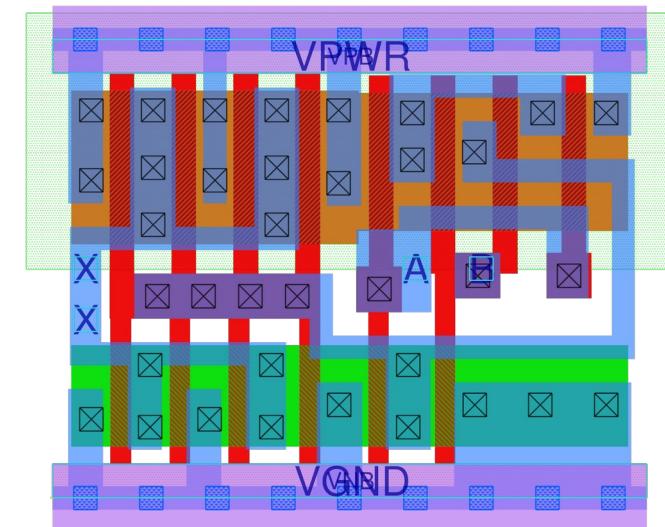
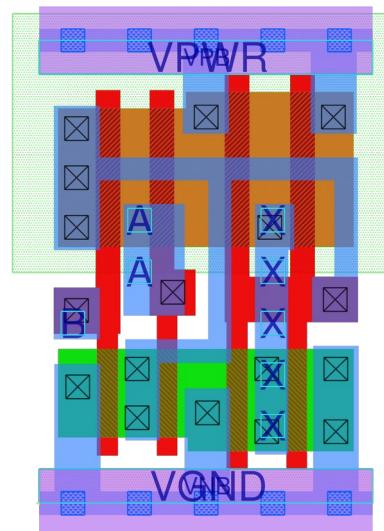
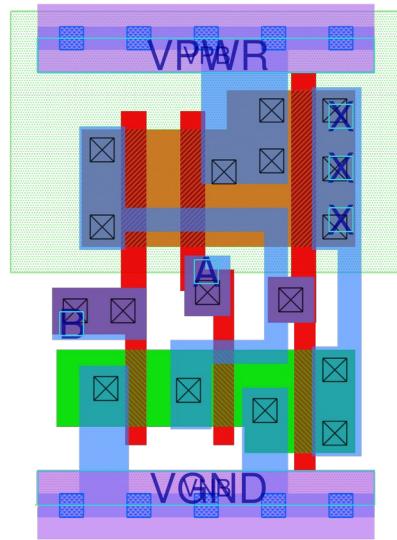
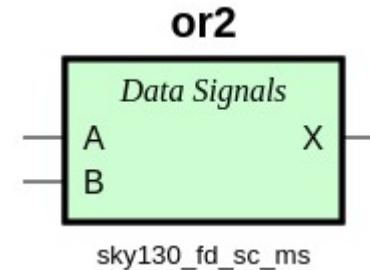
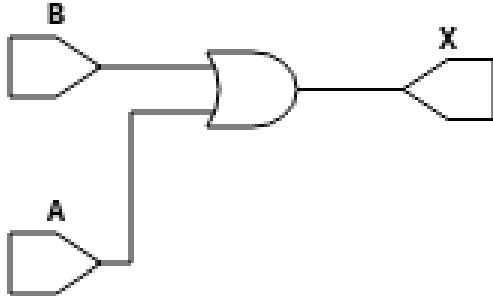
<https://tinytapeout.com/>



<https://github.com/HEP-Alliance/hsm>
<https://hep-alliance.org/Project/>



process design kit (PDK)



OpenROAD

File View Tools Windows Options Help

Fit Find Inspect Timing

Display Control

- Layers
- li1
- mcon
- met1
- via
- met2
- via2
- met3
- via3
- met4
- via4
- met5

- Nets

- Instances

- Blockages

- Rulers

- Rows

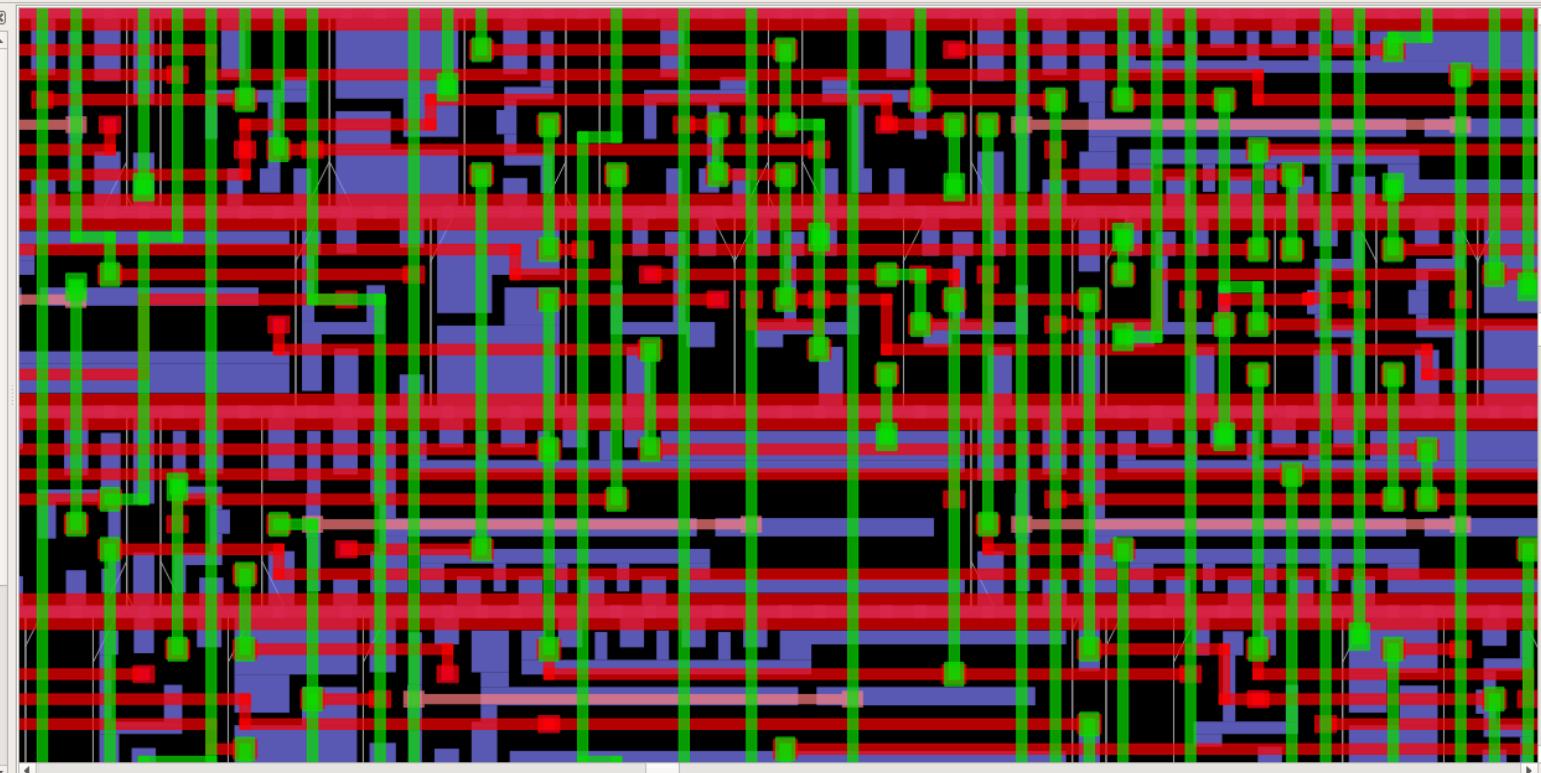
- Pins

- Tracks

- Misc

- Instances

Names	<input type="checkbox"/>
Pins	<input checked="" type="checkbox"/>
Pin Na...	<input type="checkbox"/>
Blocka...	<input checked="" type="checkbox"/>
Scale bar	<input type="checkbox"/>
Fills	<input type="checkbox"/>
Access no...	<input type="checkbox"/>



Inspector

Name	Value
Type	Net
Name	integer_RegFilePlugin_logic_regfile_nax
Block	nax
Signal type	SIGNAL
Source type	NONE
Wire type	ROUTED
Special	False
Dont Touch	False
ITerms	5 items
1	ALU0_ExecutionUnitBase_pipeline_e
2	EU0_ExecutionUnitBase_pipeline_ex
3	integer_RegFilePlugin_logic_regfile_
4	integer_RegFilePlugin_logic_regfile_
5	integer_RegFilePlugin_logic_regfile_
BTerms	0 items
BBox	(795.46,1091.405), (853.735,1195.1)

Timing Route Guides Inspector Hierarchy Browser Timing Report

Scripting

```
[WARNING GUI-0076] QXcbConnection: XCB error: 2 (BadValue), sequence: 49453, resource id: 1952, major code: 130 (Unknown), minor code: 3
[WARNING GUI-0076] QXcbConnection: XCB error: 2 (BadValue), sequence: 49458, resource id: 1952, major code: 130 (Unknown), minor code: 3
[WARNING GUI-0076] QXcbConnection: XCB error: 2 (BadValue), sequence: 49462, resource id: 1952, major code: 130 (Unknown), minor code: 3
```

Idle TCL commands

integer_RegFilePlugin_logic_regfile_latches.io_writes_1_data\[2\]_sky130_fd_sc_hd_inv_2_Y_A

825.877, 1177.295

File View Tools Windows Options Help

Fit Find Inspect Timing

Display Control

Layers

- li1
- mcon
- met1
- via
- met2
- via2
- met3
- via3
- met4**
- via4
- met5

Nets

Instances

Blockages

Rulers

Rows

Pins

Tracks

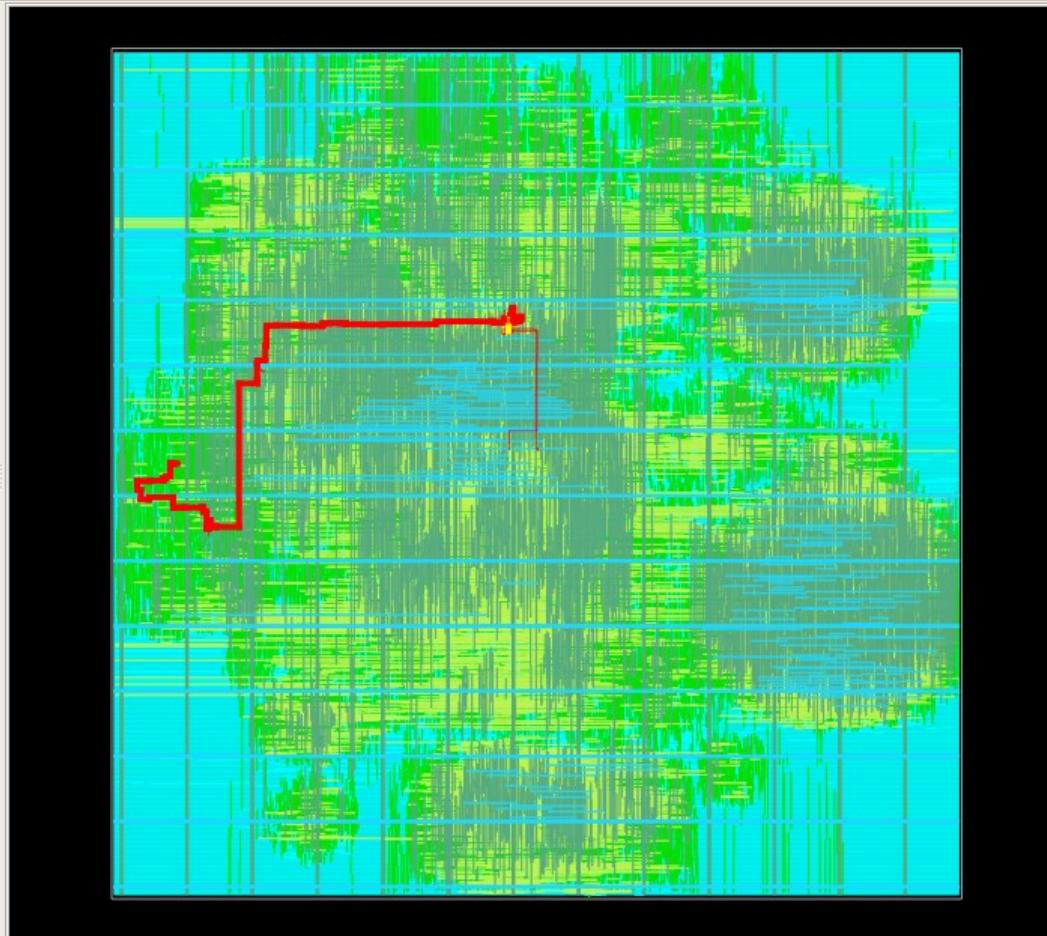
Misc

- Instances**
- Names**
- Pins**
- Pin Na...**
- Blocka...**

Scale bar

Fills

Access no...



Timing Report

Settings

Update

Setup Hold

Capture Clock	Required	Arrival	Slack	
clk	5.690	9.385	-3.694	EU0_ExecutionUnitB...
clk	5.690	9.370	-3.680	EU0_ExecutionUnitB...

Data Path Details

Capture Path Details

Pin	Fanout	t	Time	Delay	Slew	Load
clk	4	↑	0.000	0.000	1.261	
clock network delay			3.299	3.299		
EU0_ExecutionUnitBase_pipeline...		↑	3.299	0.001	0.074	
EU0_ExecutionUnitBase_pipeline...	6	↑	0.639	-2.661	0.362	0.129
EU0_ExecutionUnitBase_pipeline...		↑	0.643	0.005	0.362	
EU0_ExecutionUnitBase_pipeline...	10	↓	0.791	0.147	0.122	0.102
EU0_ExecutionUnitBase_pipeline...		↓	0.794	0.003	0.122	

Inspector

Hierarchy Browser

Timing Report

Scripting

```
0.16 9.38 ^ Lsu2Plugin_logic_sharedPip_stages_0_MMU_L0_HITS_PRE_VALID[1]_sky130_fd_sc_hd_a21oi_4_Y/Y (sky130_fd_sc_hd_a21oi_2)
0.00 9.38 ^ Lsu2Plugin_logic_sharedPip_stages_0_MMU_L0_HITS_PRE_VALID[1]_sky130_fd_sc_hd_dfxtpl_2_D/D (sky130_fd_sc_hd_dfxtpl_1)
  9.38 data arrival time
```

Idle

TCL commands

met4

2231.832, 1230.868

Open-source push for CPU

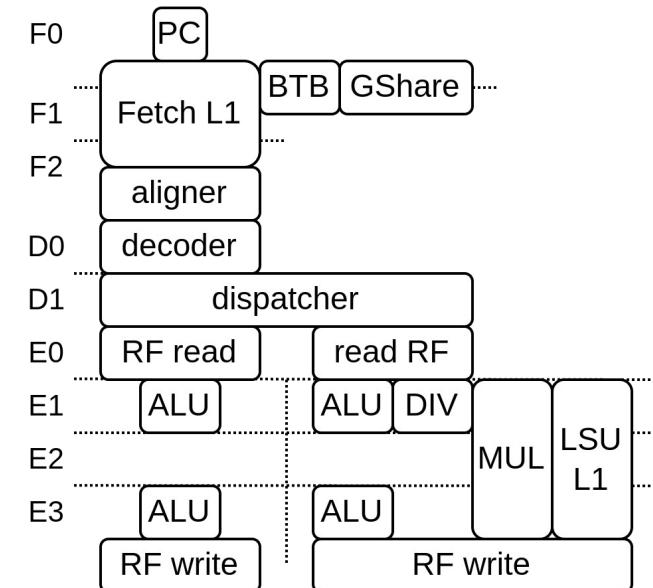
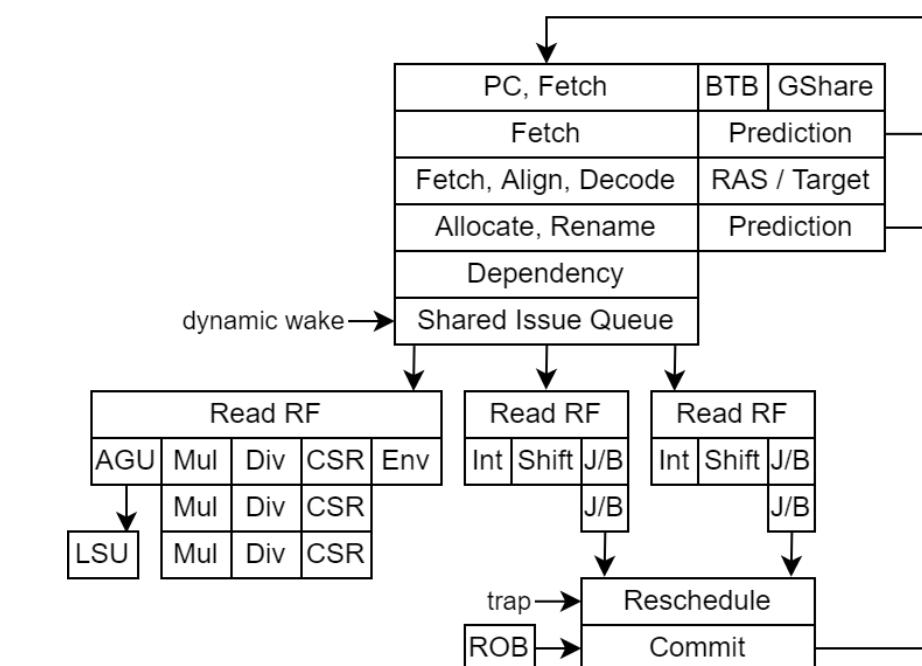
- RISC-V 
 - slowly changing the hardware culture
 - Open-source specification != Open-source implementation
 - Often hidden to the user
 - Getting ride of the ARM “tax”
 - Breakout of walled garden
 - Customizing things

bigPULP biRISC-V BOOM CV32E40P CVA6 DarkRISCV E203 Freedom FWRISC
FWRISC-S Ibex KLESSYDRA-F03 KLESSYDRA-T02 KLESSYDRA-T03
KLESSYDRA-T13 Kronos Leros lipsi Lizard Maestro Minerva MR1 mriscv NaxRiscv
NEORV32 OpenPiton NutShell patmos PicoRV32 PULP Rattlesnake Reindeer ReonV
RISCV-CLaSH riscv-mini Riscy RiscyOO Rocket RPU RSD RV01 RV12 Sail RISC-V
SCR1 SERV Shakti C-Class Shakti E-Class Sodor SSRV Starsea Steel SweRV
SweRV EH2 SweRV EL2 Taiga Tiny Risc-V VexRiscv VexiRiscv WARP-V

RV32I Base Instruction Set				
imm[31:12]		rd		0110111
imm[31:12]		rd		0010111
imm[20 10:1 11 19:12]		rd		1101111
imm[11:0]	rs1	000	rd	1100111
imm[12 10:5]	rs2	rs1	000	imm[4:1 11]
imm[12 10:5]	rs2	rs1	001	imm[4:1 11]
imm[12 10:5]	rs2	rs1	100	imm[4:1 11]
imm[12 10:5]	rs2	rs1	101	imm[4:1 11]
imm[12 10:5]	rs2	rs1	110	imm[4:1 11]
imm[12 10:5]	rs2	rs1	111	imm[4:1 11]
imm[11:0]	rs1	000	rd	0000011
imm[11:0]	rs1	001	rd	0000011
imm[11:0]	rs1	010	rd	0000011
imm[11:0]	rs1	100	rd	0000011
imm[11:0]	rs1	101	rd	0000011
imm[11:5]	rs2	rs1	000	imm[4:0]
imm[11:5]	rs2	rs1	001	imm[4:0]
imm[11:5]	rs2	rs1	010	imm[4:0]

CPU design is hard

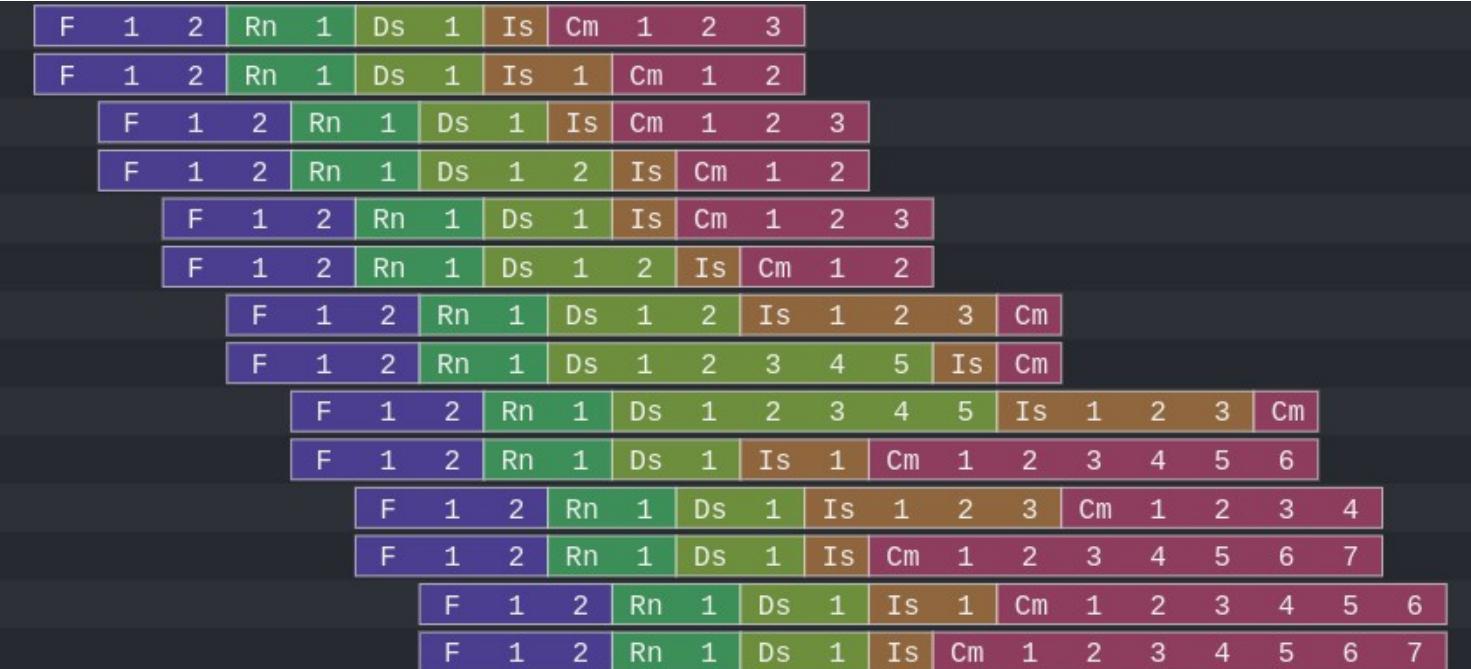
- Horror stories
- Design space is very large / trade-off
- Nuances => radically different outcomes
- Latency / bandwidth sensitive combo
- There should be no “low hanging fruit”
- Many tricks
 - Inaccurate / pessimistic approaches
 - Deflecting the bullet
 - ...
- Hard to debug



```

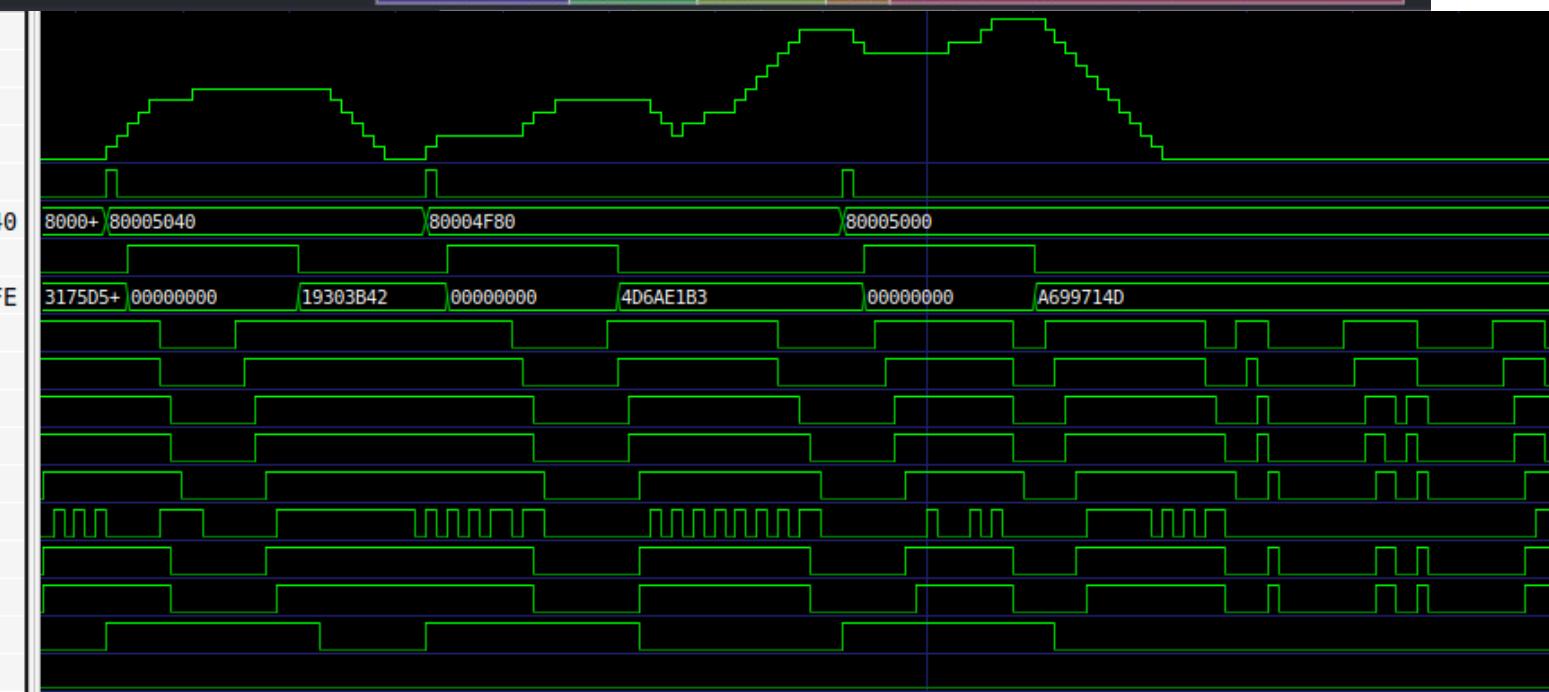
0x80006c58: addi    a7, a2, 1
0x80006c5c: j      pc + 0xffbac
0x80006808: slli   a2, a4, 2
0x8000680c: addi   a4, a2, 64
0x80006810: addi   a2, sp, 16
0x80006814: add    a4, a4, a2
0x80006818: lw     a2, 4032(a4)
0x8000681c: addi   a2, a2, 1
0x80006820: sw     a2, -64(a4)
0x80006824: bnez  a6, pc + 4004
0x800067c8: lbu    a4, 1(a7)
0x800067cc: addi   a2, a7, 1
0x800067d0: beq   a6, t5, pc + 1148
0x800067d4: addi   t1, a6, 4048

```



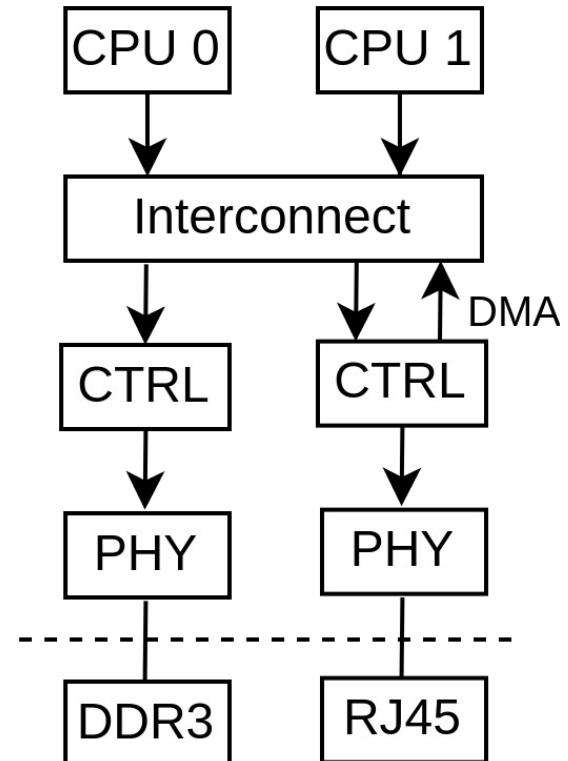
LsuPlugin_logic_storeBuffer_ops_occupancy[5:0] =00

LsuL1Plugin_logic_bus_read_cmd_valid	=0
LsuL1Plugin_logic_bus_read_cmd_payload_address[31:0]	=80004F40
LsuL1Plugin_logic_bus_read_rsp_valid	=0
LsuL1Plugin_logic_bus_read_rsp_payload_data[31:0]	=021288FE
fetch_logic_ctrls_0_up_valid	=1
fetch_logic_ctrls_1_up_valid	=1
fetch_logic_ctrls_2_up_valid	=1
decode_ctrls_0_up_valid	=1
decode_ctrls_1_up_valid	=1
DispatchPlugin_logic_candidates_0_ctx_valid	=1
DispatchPlugin_logic_candidates_1_ctx_valid	=1
DispatchPlugin_logic_candidates_2_ctx_valid	=1
LsuL1Plugin_logic_refill_slots_0_valid	=0
LsuL1Plugin_logic_refill_slots_1_valid	=0



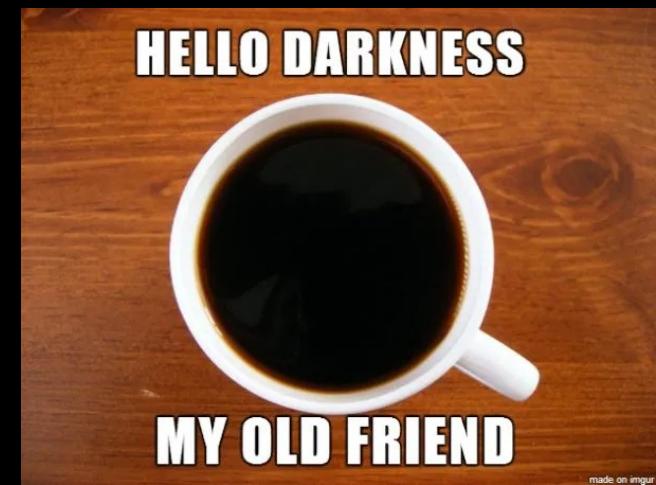
SoC design

- Memory interconnect and coherency
 - ARM is king (AXI ACE / CHI)
 - Tilelink / Wishbone / ...
- Controllers / Peripherals / Acceleration
 - Simples : GPIO / UART / SPI / Ethernet / SDCARD / DRAM
 - Scary : PCIE / USB
 - Mighty : GPU 
 - IO mapped / Direct memory access
- Physical layer (PHY)
 - May require specialized skills (ASIC)
 - May be integrated (FPGA)
 - May be externalized (PHY Chip)



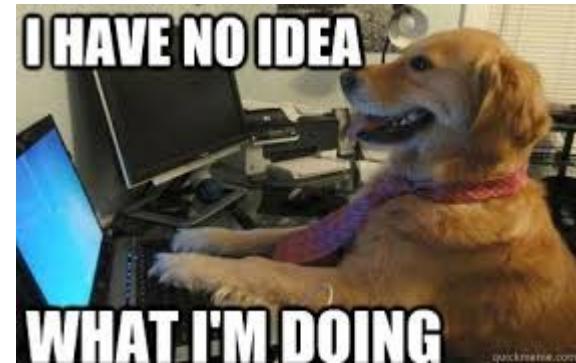
So it is hard

```
[1377614.579833] rcu: INFO: rcu_sched self-detected stall on CPU
[1377614.579845] rcu: 18-....: (2099 ticks this GP) idle=54e/1/0x4000000000000002
[1377614.579895] (t=2100 jiffies g=155867385 q=20879)
[1377614.579898] Task dump for CPU 18:
[1377614.579899] CPU 1/KVM R running task 0 1030947 256019 0x06000004
[1377614.579902] Call Trace:
[1377614.579912] ([<0000001f1f4b4f52>] show_stack+0x7a/0xc0)
[1377614.579918] [<0000001f1ec8e96c>] sched_show_task.part.0+0xdc/0x100
[1377614.579919] [<0000001f1f4b7248>] rcu_dump_cpu_stacks+0xc0/0x100
[1377614.579924] [<0000001f1ecdd10c>] rcu_sched_clock_irq+0x75c/0x980
[1377614.579926] [<0000001f1eceb26c>] update_process_times+0x3c/0x80
[1377614.579931] [<0000001f1ecfcfea>] tick_sched_handle.isra.0+0x4a/0x70
[1377614.579932] [<0000001f1ecfd28e>] tick_sched_timer+0x5e/0xc0
[1377614.579933] [<0000001f1ecec294>] __hrtimer_run_queues+0x114/0x2f0
[1377614.579935] [<0000001f1ececfdc>] hrtimer_interrupt+0x12c/0x2a0
[1377614.579938] [<0000001f1ebecb6a>] do_IRQ+0xaa/0xb0
[1377614.579942] [<0000001f1f4c6d08>] ext_int_handler+0x130/0x134
[1377614.579945] [<0000001f1ec0af10>] ptep_zap_key+0x40/0x60
```



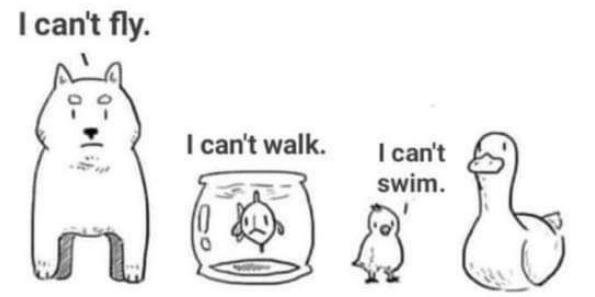
The scale of the project (CPU + SoC)

- Many skills involved
 - Hardware design (to get the best performances)
 - CPU design / FPU / JTAG debug
 - Memory interconnect / memory coherency
 - Peripheral design / USB / Ethernet / ...
 - Hardware verification (to not miss a bug)
 - software modeling / Lock-step / ...
 - Hardware debugging (to not waste weeks on bugs)
 - Assembly (objdump -S -d vmlinux nightmare.asm)
 - Wave / execution traces
 - Good guess / patience
 - Hardware backends (FPGA / ASIC)
 - Baremetal / linux drivers
- One baby steps at the time

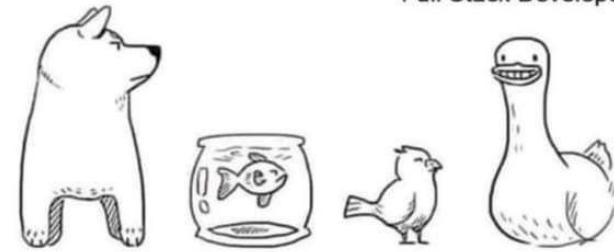


The human side of things

- People tends to specialise (too much / large field)
 - Few people have the “full picture”
 - Suboptimal results / buried performance degradation
- Very hard to steer away from the common path
 - Graduation → Stress to find a job → Employment → Closed source
 - The need of stability → Employment → Closed source



Full Stack Developer



Some other time sinks

- Community related
 - Issue tracker / False positive issues (scary / stressful)
 - Pull-requests (Mental load, risks, debug, ossification)
 - Student spam / thesis / over ambitious emails

Free and Open-source hardware and the industry

- It depends the companies (startup / small / medium / large) (FPGA / ASIC)
- Incentives may be misaligned
 - Companies core business having too much proximity
 - Companies looking for exclusive differentiation
 - Companies looking to file patents on what ever they can
 - Companies trading cash for ownership
 - Companies trying to lock people in their ecosystem (Nios-V Microblaze-V ...)
 - Companies “owning” employee / non-compete clause / NDA
 - Companies freezing tools versions

Free and Open-source hardware and the industry

- Other disconnects
 - Fear of legal uncertainty
 - Quality requirements
 - Providing paid support
 - The tragedy of the commons

Reaching critical mass

- CVA6 (In-order CPU, ASIC)
- VexRiscv (In-order CPU, FPGA)
- VexiiRiscv (~Cortex M0 up to ~Cortex A53 CPU, WIP)
- NaxRiscv (Out-of-order CPU, may be “assimilated” in VexiiRiscv in the future)
- Litex (SoC generation framework)
- ...

Questions ?