

Faire du hard avec du soft

@Pa0x73cal (Pascal)

Sylvain et Auguste

@_JCLL_



ENSTA
BRETAGNE

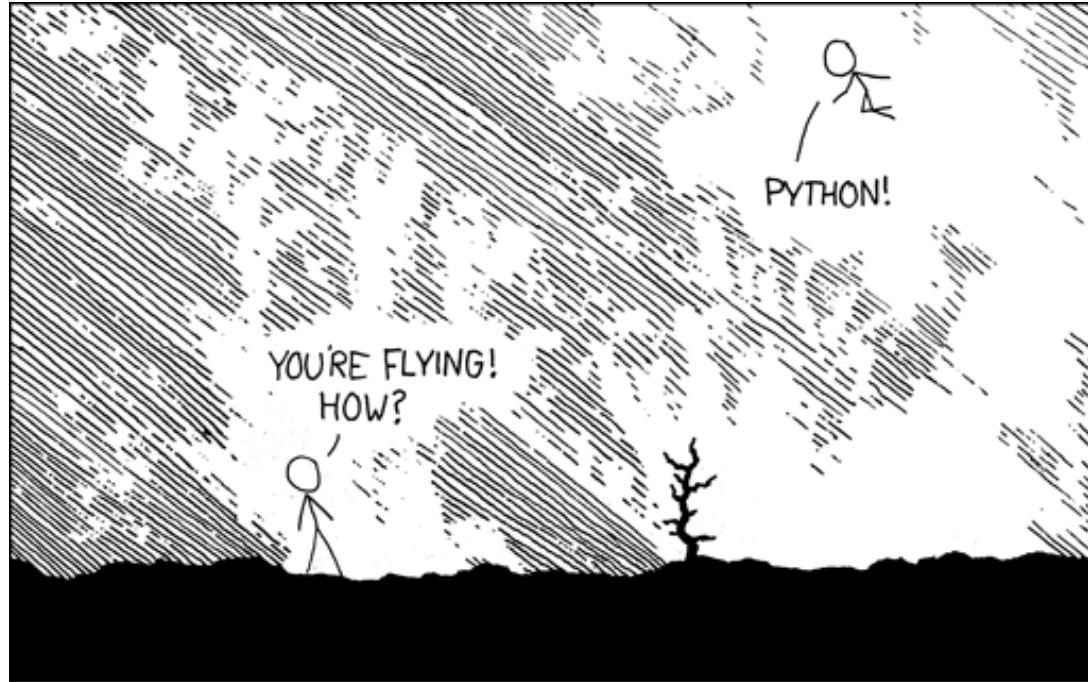
```
class PicoRV32(CPU):
    name          = "picorv32"
    data_width    = 32
    endianness   = "little"
    gcc_triple   = ("riscv64-unknown-elf", "riscv32-unkn
    linker_output_format = "elf32-littleriscv"
    io_regions   = {0x80000000: 0x80000000} # origin, len

@property
def gcc_flags(self):
    flags = "-mno-save-restore "
    flags += GCC_FLAGS[self.variant]
    flags += "-D_picorv32_"
    return flags

@property
def reserved_interrupts(self):
    return {
        "timer":                 0,
        "ebreak_ecall_illegal": 1,
        "bus_error":             2
    }

def __init__(self, platform, variant="standard"):
    assert variant in CPU_VARIANTS, "Unsupported variant %s"
    self.platform = platform
    self.variant = variant
    self.reset = Signal()
    self.idbus = idbus = wishbone.Interface()
    self.buses = [idbus]
    self.interrupt = Signal(32)
```

Utiliser du Python pour faire du matériel ? 😊



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST
`print "Hello, world!"`

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?
COME JOIN US!
PROGRAMMING IS FUN AGAIN!
IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?

I JUST TYPED
`import antigravity`
THAT'S IT?
... I ALSO SAMPLED
EVERYTHING IN THE
MEDICINE CABINET
FOR COMPARISON.
BUT I THINK THIS
IS THE PYTHON.

Utiliser du Python pour faire du matériel ?

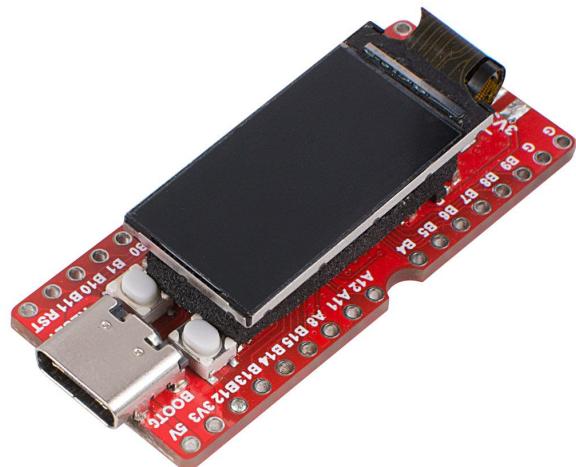
Avant ⇒ 😠

Utiliser du Python pour faire du matériel ?

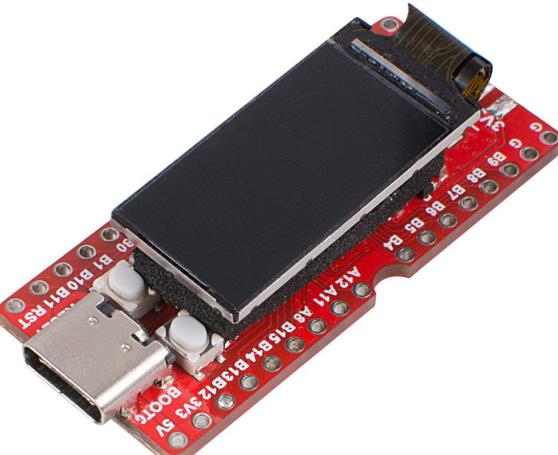
Avant ⇒ 

Maintenant ⇒ 

Plateformes visées



Plateformes visées



- De quoi faire tourner un Linux embarqué

TESLA MODEL S MODEL 3 MODEL X MODEL Y **THALES** EN ▾

Careers

Software Engineer, Embedded Linux Platforms

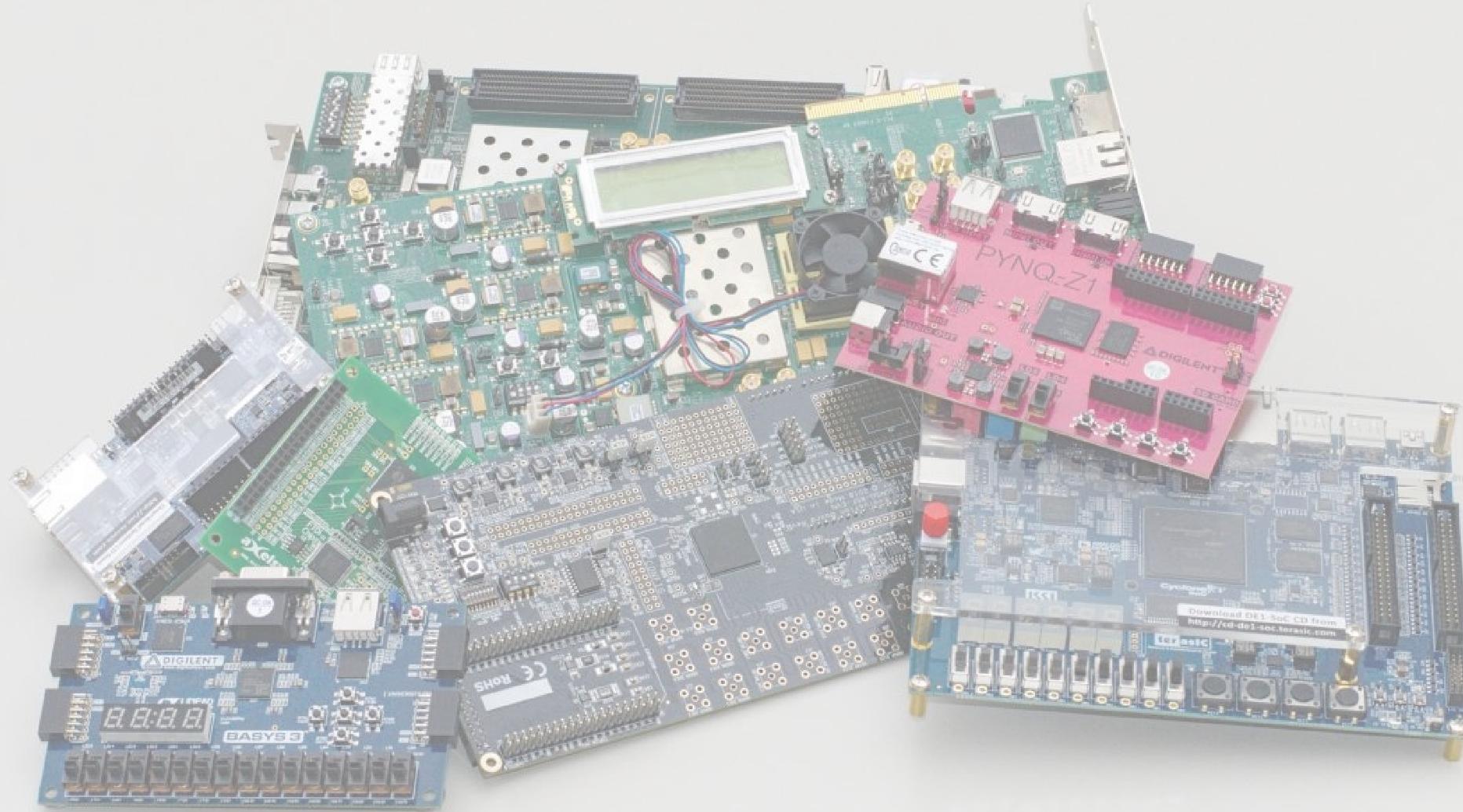
Job Category Engineering & Information Technology

**C EMBEDDED DEVELOPER-
SPACE BUSINESS**

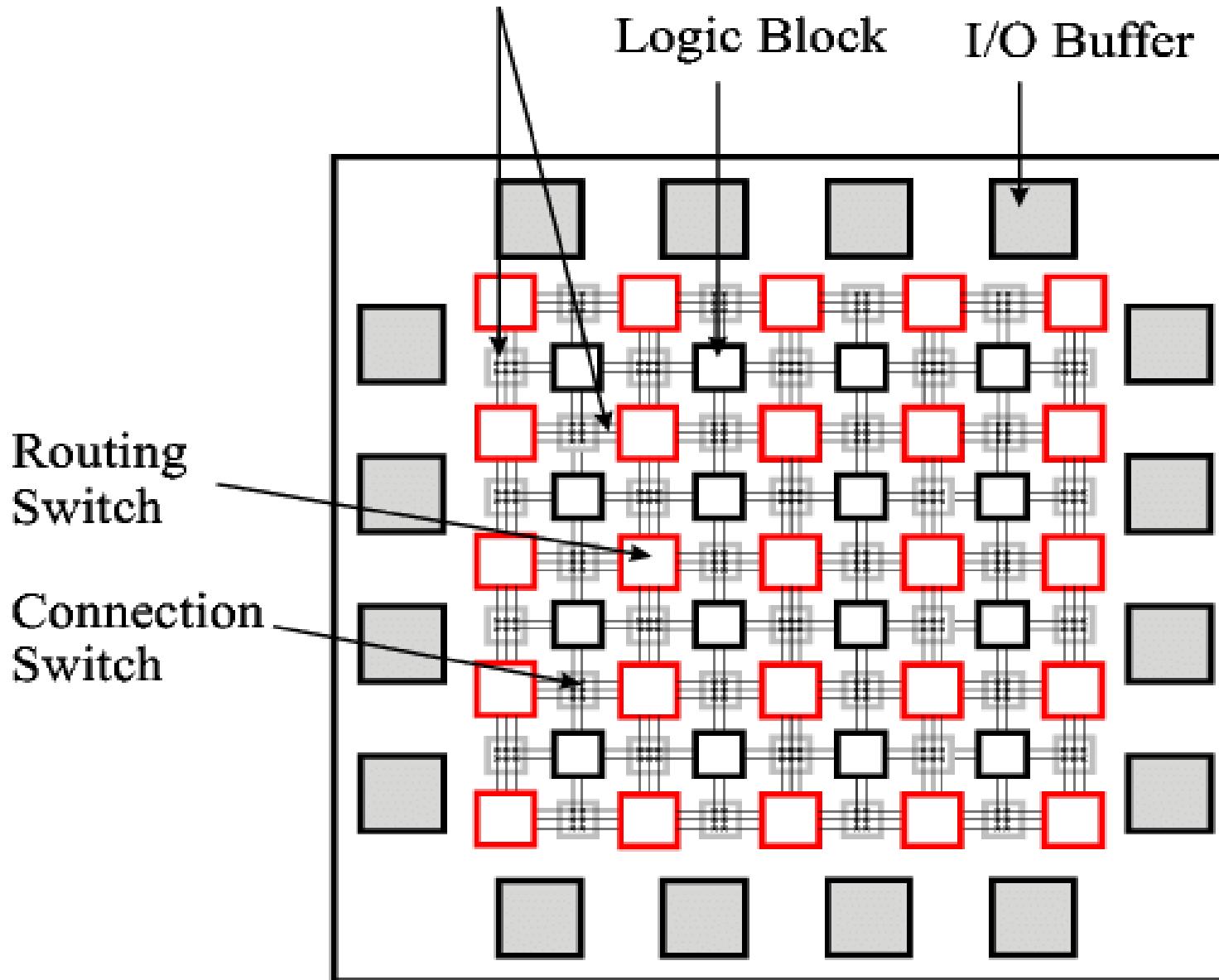
SEARCH AGAIN

Job ID: R0084096

FPGA, c'est quoi ?



Interconnection Resources



Programmation du matériel

~~Programmation~~ Description du matériel

Deux langages principaux : VHDL et Verilog

(HDL = Hardware Description Language)

Compter sur 1 octet en VHDL

```
entity mon_compteur is
port(clk : in std_logic;
      cout : out std_logic_vector (7 downto 0));
end mon_compteur;

architecture rtl of mon_compteur is
signal count :std_logic_vector (7 downto 0);
begin
  process (clk) begin
    if (rising_edge(clk)) then
      count <= count + 1;
    end if;
  end process;
  cout <= count;
end architecture;
```

Compter sur 1 octet en VHDL

```
entity mon_compteur is -- Les entrées/sorties de ma "fonction"
port(clk : in std_logic;
      cout : out std_logic_vector (7 downto 0));
end mon_compteur;

architecture rtl of mon_compteur is
signal count :std_logic_vector (7 downto 0);
begin
  process (clk) begin
    if (rising_edge(clk)) then
      count <= count + 1;
    end if;
  end process;
  cout <= count;
end architecture;
```

Compter sur 1 octet en VHDL

```
entity mon_compteur is -- Les entrées/sorties de ma "fonction"
port(clk : in std_logic;
     cout : out std_logic_vector (7 downto 0));
end mon_compteur;

architecture rtl of mon_compteur is -- Le coeur de la fonction
signal count :std_logic_vector (7 downto 0);
begin
  process (clk) begin
    if (rising_edge(clk)) then
      count <= count + 1;
    end if;
  end process;
  cout <= count;
end architecture;
```

Type any word...



TOP DEFINITION



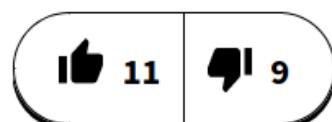
VHDL

A [programming language](#) designed by electrical engineers instead of people who actually know how to design a programming language. Features an incredibly confusing and awkward [syntax](#) and weird dependency issues where the user must [import](#) the same library multiple times in the same file.

Hey, what's this [contrived](#) mess of code supposed to be? You wrote this much to represent a simple [flip-flop](#)? Oh, you must be programming a [circuit](#) in VHDL!

#disaster #verilog #circuit #programming language #electrical engineering #computer engineering

by [majorb](#) February 07, 2011





TOP DEFINITION



VHDL

A ~~pro~~ print ('Et pourquoi pas du Python ?')

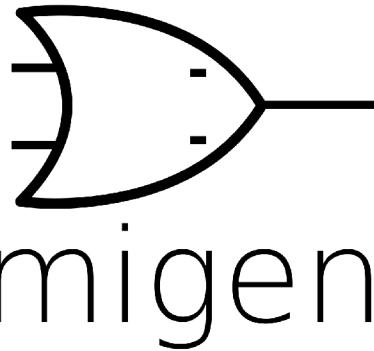
language. Features an incredibly confusing and awkward [syntax](#) and weird dependency issues where the user must [import](#) the same library multiple times in the same file.

Hey, what's this [contrived](#) mess of code supposed to be? You wrote this much to represent a simple [flip-flop](#)? Oh, you must be programming a [circuit](#) in VHDL!

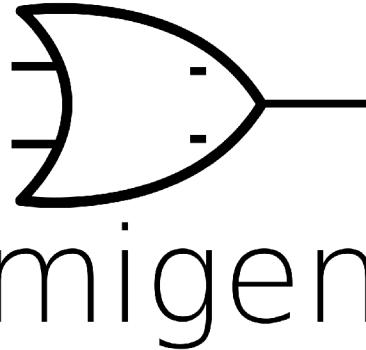
#disaster #verilog #circuit #programming language #electrical engineering #computer engineering

by [majorb](#) February 07, 2011





```
# Compteur sur 10 bits
from migen import *
class MonCompteur(Module):
    def __init__(self):
        self.counter = counter = Signal(10)
        self.sync += counter.eq(counter + 1)
```

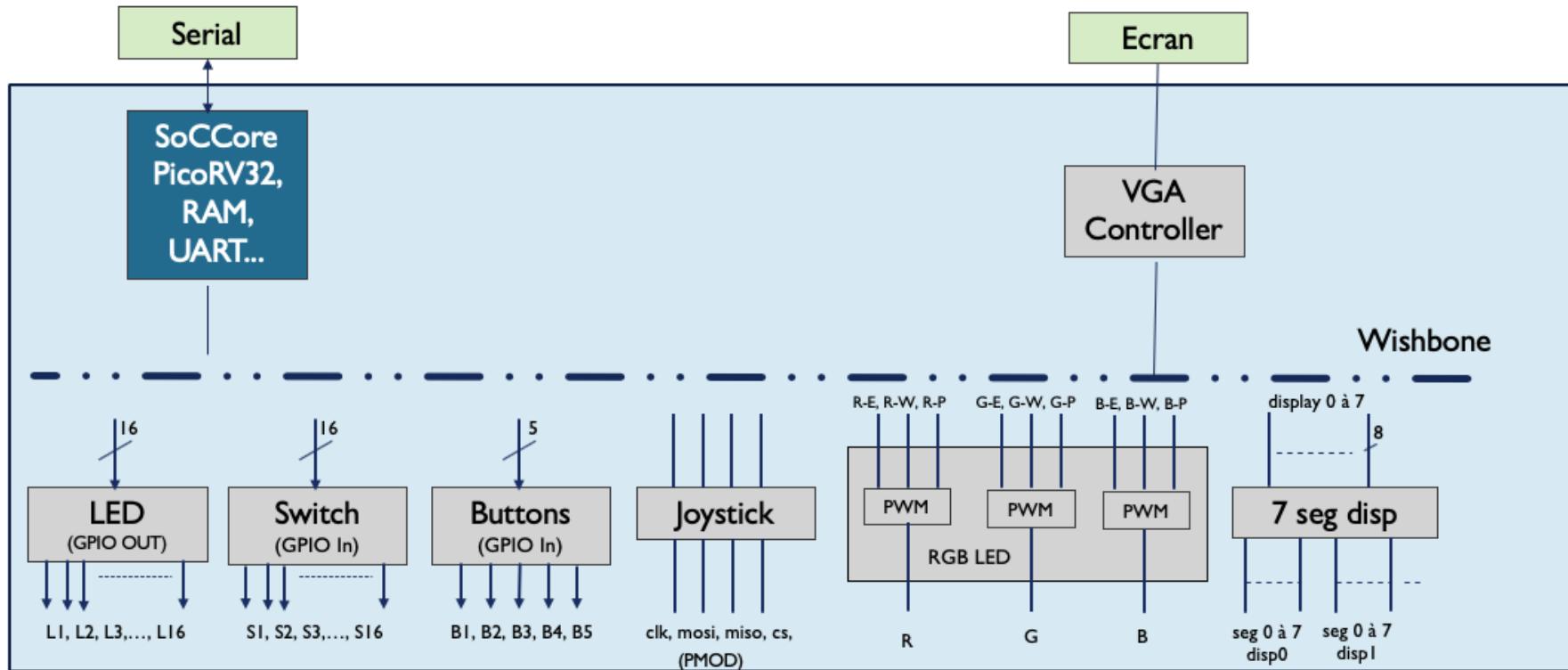


```
# Compteur sur 10 bits
from migen import *
class MonCompteur(Module):
    def __init__(self):
        self.counter = counter = Signal(10)
        self.sync += counter.eq(counter + 1)
```

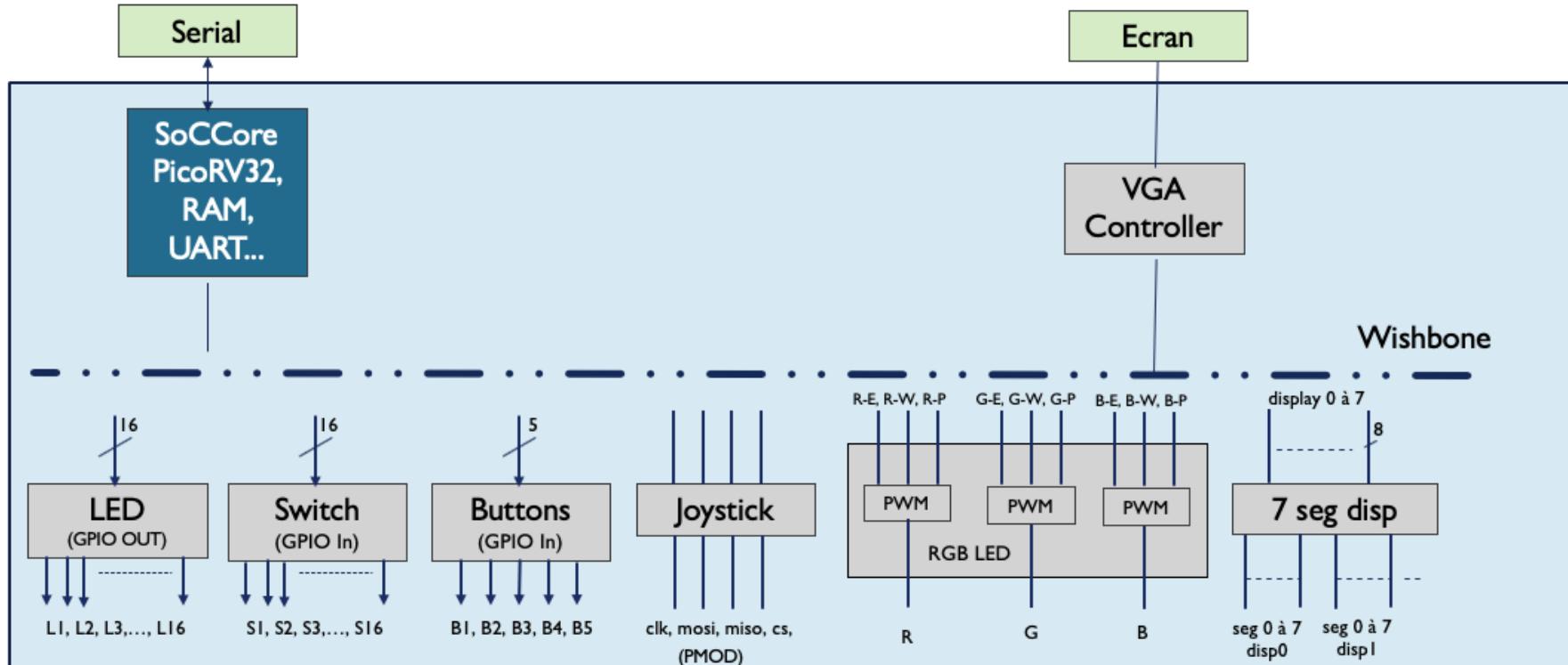
- Langage basé sur du Python
- Plein de fonctions basiques dispos
- Orienté objet, surcharge d'opérateurs, paramètres de fonctions...

⇒ Pas "naturel" en hard mais fichrement pratique !

Un processeur, ça ne marche pas tout seul...



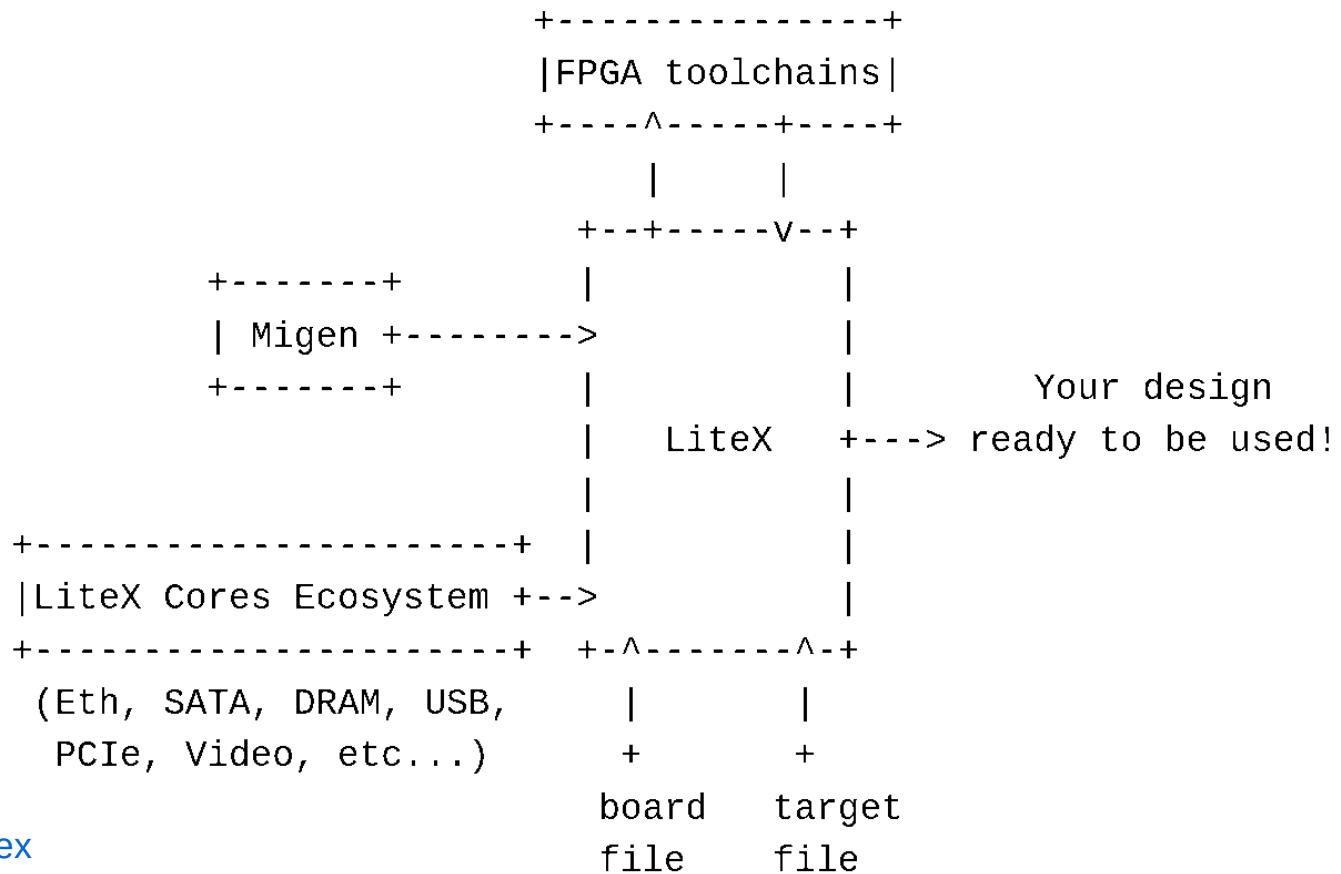
Un processeur, ça ne marche pas tout seul...



- Modèle de processeur en Python ?
- Comment connecter tout ça ?



Build your hardware, easily!



```
class BaseSoC(SoCCore):
    def __init__(self):
        platform = nexys4ddr.Platform()

        # Mon processeur instancié en Python !
        SoCCore.__init__(self, platform,
                          cpu_type="picorv32", # D'autres CPU dispos
                          clk_freq=100e6, integrated_rom_size=0x8000,
                          integrated_main_ram_size=16*1024)
        # On ajoute des LEDs
        SoCCore.add_csr(self, "leds")
        user_leds = Cat(*[platform.request("led", i) for i in range(16)])
        self.submodules.leds = gpio.GPIOOut(user_leds)
    [...]
    # On construit notre système
    if __name__ == "__main__":
        builder = Builder(BaseSoC())
        builder.build()
```

Et l'open-source matériel dans tout ça ? 😕

Et l'open-source matériel dans tout ça ? 😕

- Hégémonie de 2 fabricants (dont Intel)
- Dépendance compilo/puce... 😠

Et l'open-source matériel dans tout ça ? 😕

- Hégémonie de 2 fabricants (dont Intel)
- Dépendance compilo/puce... 😠
- Tout est "closed-source" 😠 😠 😠



Architecture RISC-V 😎



- Standard ouvert, supporté par Linux, cross-compilateur possible, etc.
- Chacun peut faire son implémentation ! <https://github.com/riscv/riscv-cores-list>

The screenshot shows the 01net homepage with a red header. The header features the 01net logo, a search bar, and links for 'Codes Promo', 'Services', 'Forum', and 'Newsletters'. Below the header, there's a navigation bar with categories: VIDÉOS, ACTUALITÉS, TESTS, ASTUCES, TELECHARGER.COM, JEUX VIDÉO, and BONS PLANS. The main content area displays a news article with the following details:

RISC V déménage en Suisse pour échapper à la guerre commerciale sino-américaine

The article is categorized under 'Actualités' and 'Technos'. A sidebar on the left contains a back arrow icon.

Architecture RISC-V 😎

- Standard ouvert, supporté par Linux, cross-compilateur possible, etc.
- Chacun peut faire son implémentation ! <https://github.com/riscv/riscv-cores-list>
- ARM a les jetons 😢 <https://frama.link/arm-basics>



Miguel de Icaza
@migueldeicaza

Follow

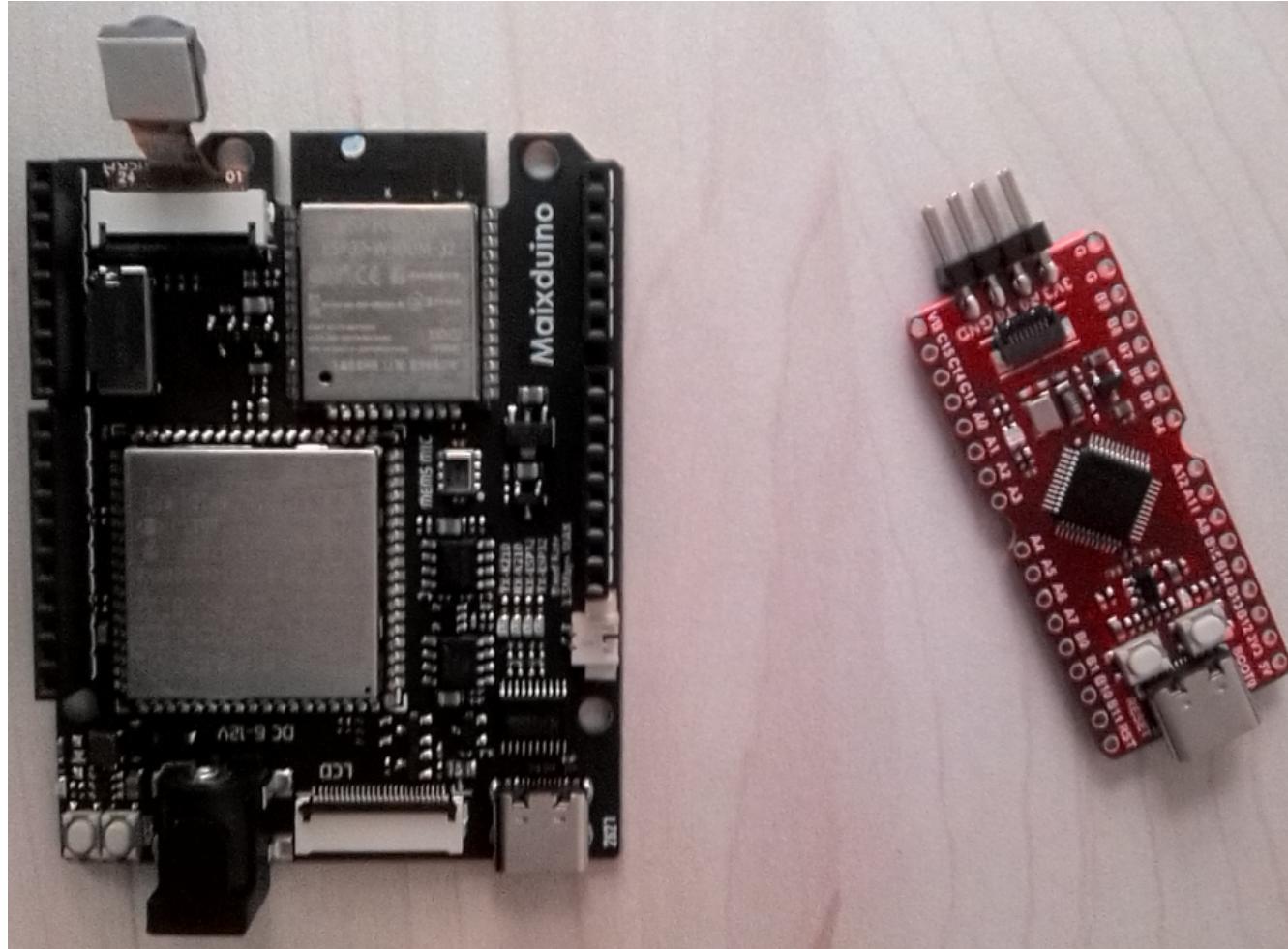
ARM's negative campaign against RISC-V can only backfire. Also, their points are kind of weak, this was attempted before against open source, and all it achieved was eggs on people's faces



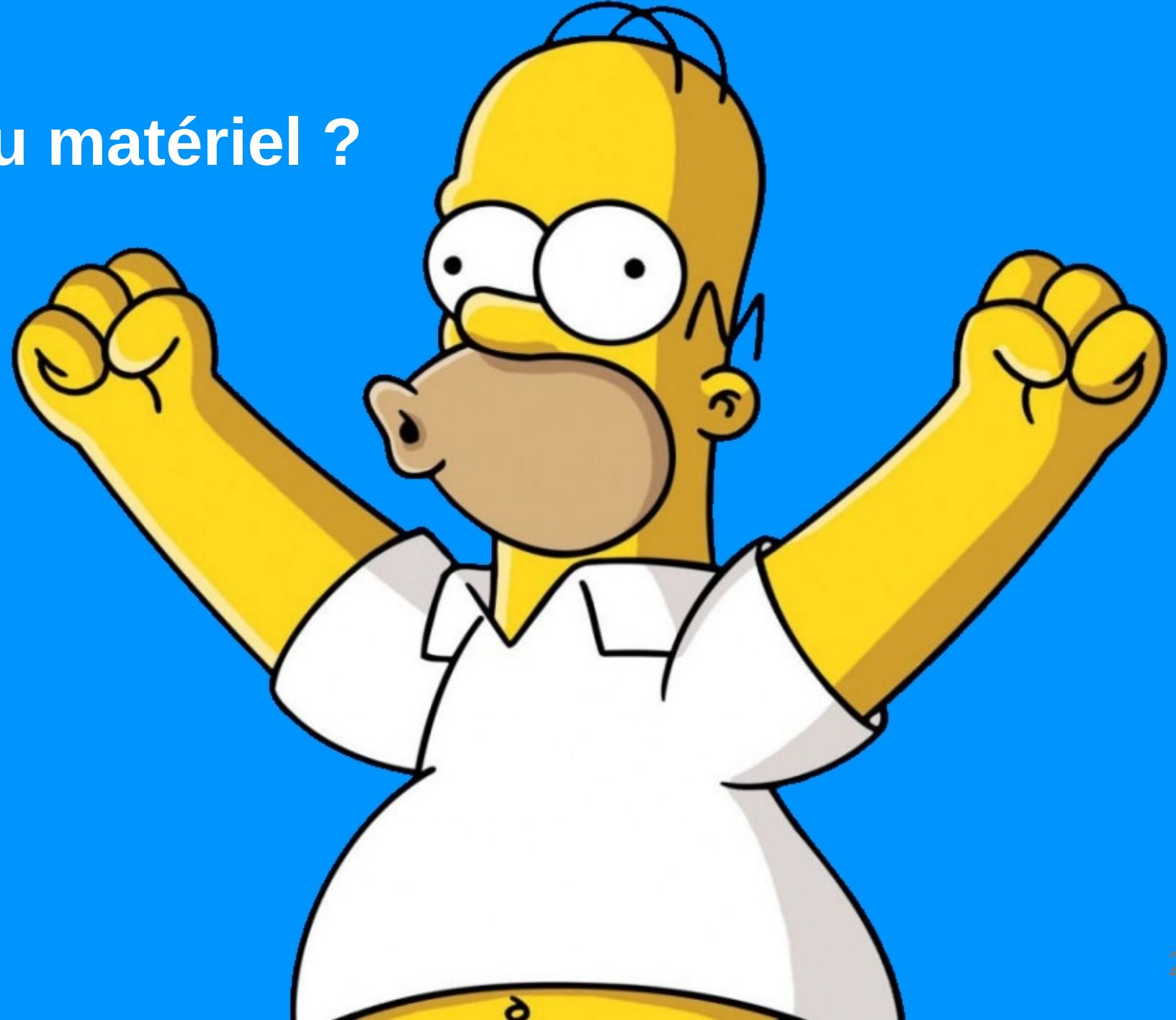
RISC Architecture: Understanding the Facts
Five Things to Consider before Designing a System-on-Chip
riscv-basics.com

Architecture RISC-V 😎

- Standard ouvert, supporté par Linux, cross-compilateur possible, etc.
- Chacun peut faire son implémentation ! <https://github.com/riscv/riscv-cores-list>
- ARM a les jetons 😢 <https://frama.link/arm-basics>

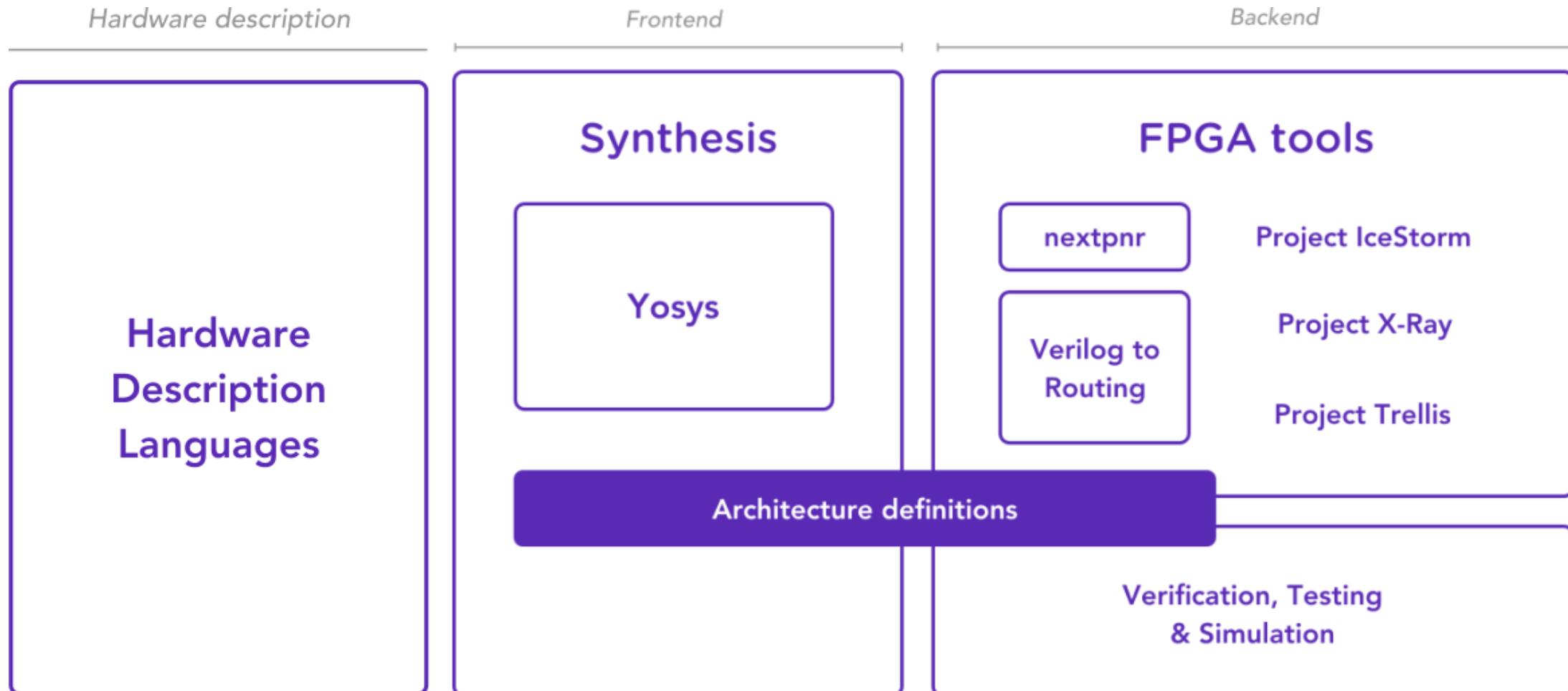


Vers un GCC du matériel ?



Symbiflow by SymbioticEDA (entre autres !)

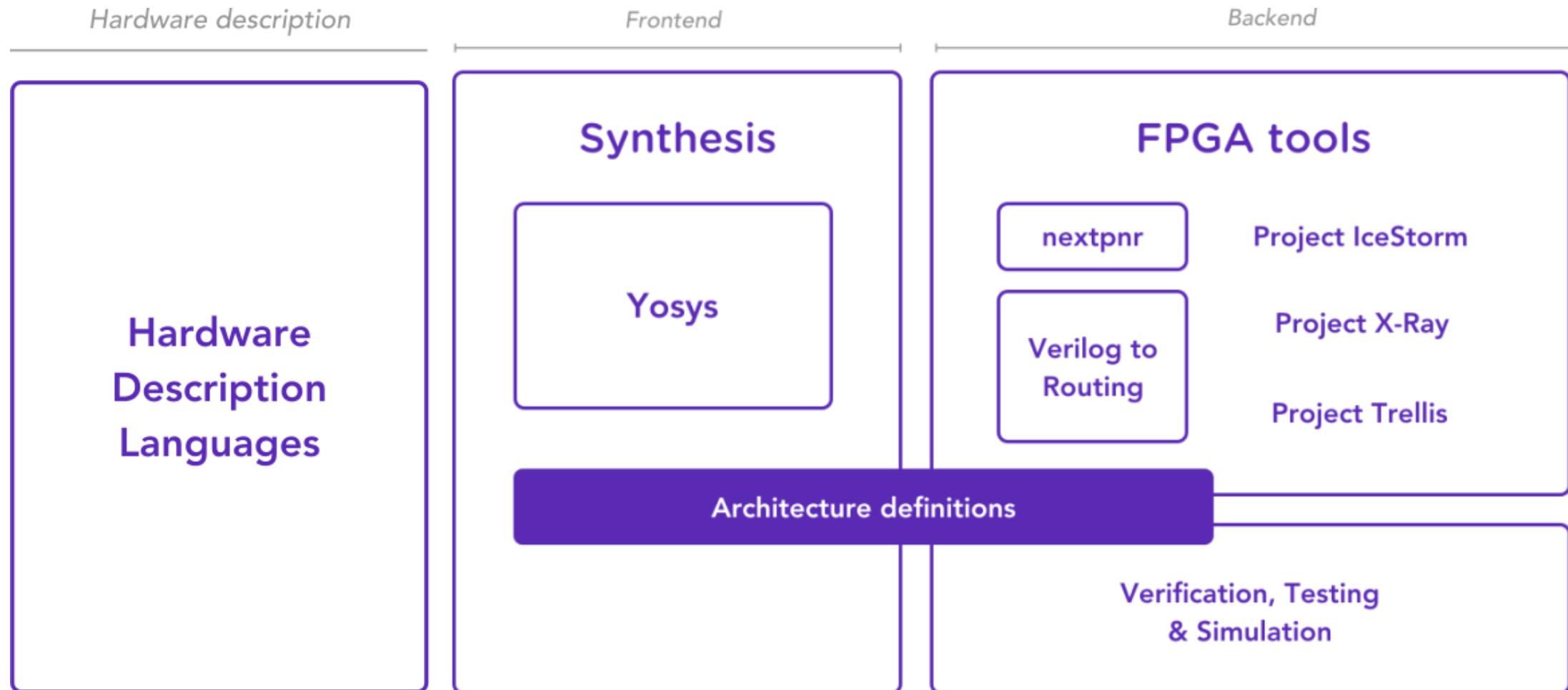
Symbiflow ⇒ 100% open-source



Symbiflow by SymbioticEDA (entre autres !)

Symbiflow ⇒ 100% open-source

Circuits FPGA closed-source



Symbiflow by SymbioticEDA (entre autres !)

⇒ Symbiflow ⇒ **100% open-source**

⇒ Circuits FPGA *closed-source* ⇒ **Pas de problème, on reverse !** 😊

The screenshot shows the GitHub repository page for Symbiflow/prjxray. The top navigation bar includes links for Watch (52), Star (356), Fork (51), and the repository name. Below the navigation is a header with tabs for Code, Issues (80), Pull requests (16), Actions, Wiki, Security, and Insights. The main content area starts with a heading: "Documenting the Xilinx 7-series bit-stream format. <https://symbiflow.github.io/prjxray-db/>". Below this are several buttons for tags: fpga, xilinx, xilinx-fpga, artix, artix7, kintex7, bitstream, tools, toolchain, fuzzer, and vivado. A summary bar displays metrics: 2,985 commits, 7 branches, 0 packages, 4 releases, 30 contributors, and an ISC license. At the bottom, there are buttons for Branch: master, New pull request, Create new file, Upload files, Find file, and Clone or download. The main feed lists recent commits:

- litghost Merge pull request #1238 from litghost/remap_some_timing ... - Latest commit 84b1457 17 hours ago
- .github Avoid race in 001 fuzzer. - 2 days ago
- database database make: clean also part-dependent directories - 14 days ago
- docs Change theme to Sphinx Material Design - 5 months ago

RISC-V

+ Symbiflow

+ Migen/LiteX

(+ un p'tit Linux)

⇒ Du hard avec du soft, 100% open-source



```
----- Boot -----
```

```
Booting from Python-based CPU...
```

```
----- Liftoff! -----
```

```
Faire du hard avec du soft - DevFest Bout du Monde Feb 28 2020
```

```
Available commands: end
```

```
PROMPT>end
```

```
That's the end, folks!
```

```
Slides: https://github.com/pcotret/presentations
```

Pour plus d'infos :

- RISC-V sur linuxfr.org, Migen dans [GNU/Linux Mag](#)
- Merki: [@mithro](#) (thx Tim for some slide ideas!), Sylvain/Auguste/Jean-Christophe
- Refs : [@enjoy_digital](#) (**made in BZH**), [@oe1cxw](#), [@pdp7](#), plein d'autres encore...