Partage d'outils Opensource

PeakRDL

https://peakrdl-regblock-vhdl.readthedocs.io/en/latest/

Amaury HERVO

01/06/2026

PeakRDL

Fonctionnalités principales:

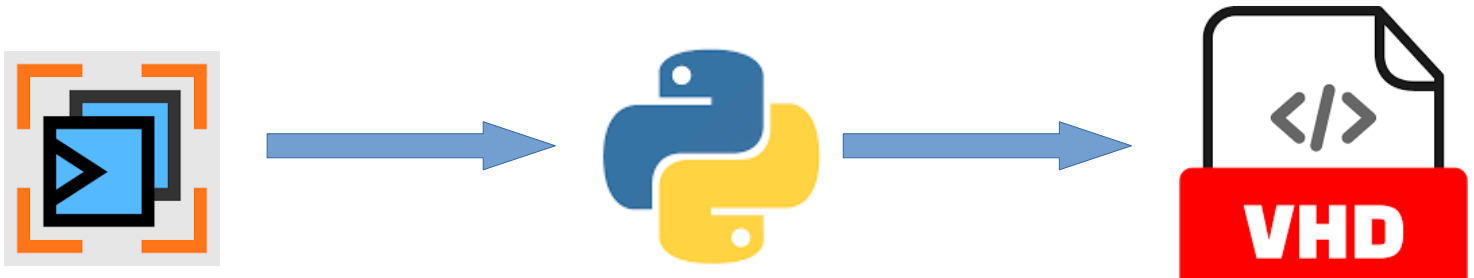
- Générer du code RTL
  - → VHDL/System Verilog
  - → Registres de contrôles et de status
  - → Synthétisable
  - → Compatibilité protocole d'interfaçage : AXI4, Avalon,…

PeakRDL

Fonctionnalités principales:

- Générer du code RTL
  - → VHDL/System Verilog
  - → Registres de contrôles et de status
  - → Synthétisable
  - → Compatibilité protocole AXI4, Avallon,…

- Outil Python
  - → open-source
  - → SystemRDL (*Register Description Language*)



SystemRDL **PeakRDL**

Exemple :

Description RDL

```
addrmap top_regmap {
    default regwidth = 32;
    default accesswidth = 32;

   reg {
        name = "LED managment";
        field {
            sw = rw;
            hw = r;
            desc = "enable/disable front-end LED";
        } led [4] = 4'h0;
    } led_reg @0x10;
};
```

SystemRDL

Exemple :

Description RDL

```
addrmap top_regmap {
    default regwidth = 32;
    default accesswidth = 32;

   reg {
        name = "LED managment";
        field {
            sw = rw;
            hw = r;
            desc = "enable/disable front-end LED";
        } led [4] = 4'h0;
    } led_reg @0x10;
};
```

```
$ peakrdl regblock-vhdl top_regmap.rdl --cpuif avalon-mm-flat -o . --addr-width 32
```

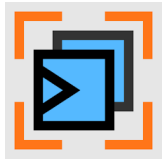
SystemRDL

PeakRDL

Exemple :

Description RDL

```
addrmap top_regmap {
    default regwidth = 32;
    default accesswidth = 32;

   reg {
        name = "LED managment";
        field {
            sw = rw;
            hw = r;
            desc = "enable/disable front-end LED";
        } led [4] = 4'h0;
    } led_reg @0x10;
};
```



SystemRDL

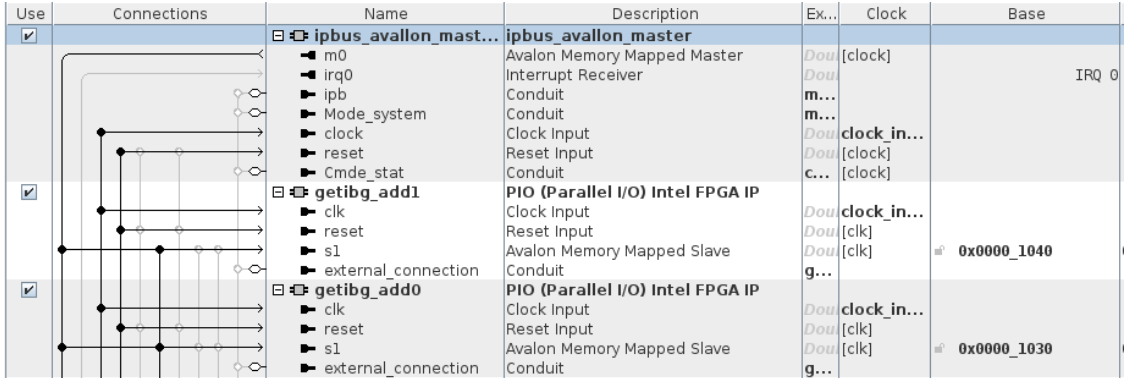
```
$ peakrdl regblock-vhdl top_regmap.rdl --cpuif avalon-mm-flat -o . --addr-width 32
```

PeakRDL

```
-- Generated by PeakRDL-regblock-vhdl - A free and open-source VHDL generator
--  https://github.com/SystemRDL/PeakRDL-regblock-vhdl
entity top_regmap is
    port (
        clk : in std_logic;
        rst : in std_logic;

        avalon_read : in std_logic;
        avalon_write : in std_logic;
        avalon_waitrequest : out std_logic;
        avalon_address : in std_logic_vector(29 downto 0);
        avalon_writedata : in std_logic_vector(31 downto 0);
        avalon_byteenable : in std_logic_vector(3 downto 0);
        avalon_readdatavalid : out std_logic;
        avalon_writeresponsevalid : out std_logic;
        avalon_readdata : out std_logic_vector(31 downto 0);
        avalon_response : out std_logic_vector(1 downto 0);

        hwif_out : out top_regmap_out_t
    );
end entity top_regmap;

architecture rtl of top_regmap is*

    process(clk) begin
        if false then -- async reset
            field_storage.led_reg.led.value <= 4x"0";
        elsif rising_edge(clk) then
            if rst then -- sync reset
                field_storage.led_reg.led.value <= 4x"0";
            else
                if field_combo.led_reg.led.load_next then
                    field_storage.led_reg.led.value <= field_combo.led_reg.led.next_q;
                end if;
            end if;
        end if;
    end process;
    hwif_out.led_reg.led.value <= field_storage.led_reg.led.value;

end architecture rtl;
```

Evolution de pratique:

AVANT



Pour chaque nouveaux registres :

→ Ajout de PIO sur le bus avalon

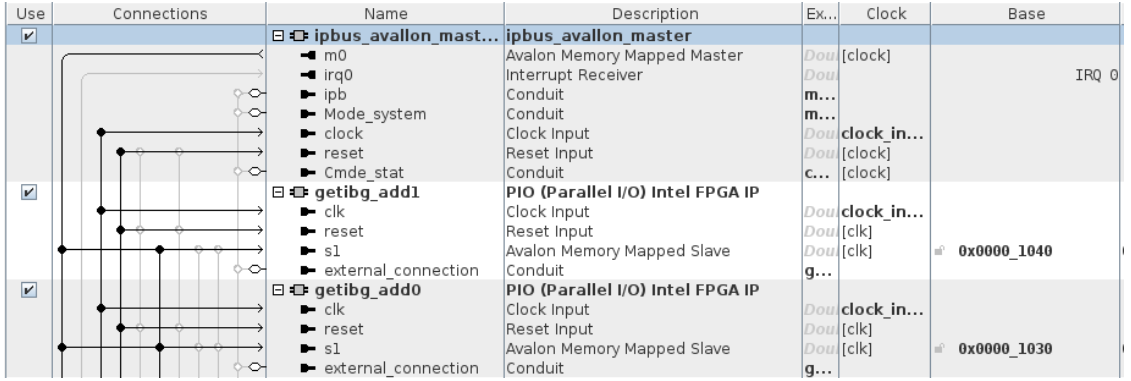
Inconvénients :

→ Maintenance

→ Cartographie de registres

Evolution de pratique:

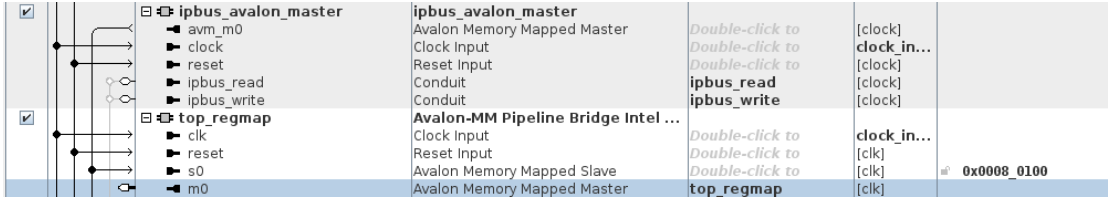
AVANT



Pour chaque nouveaux registres :
→ Ajout de PIO sur le bus avalon

Inconvénients :
→ Maintenance
→ Cartographie de registres

APRES



Pour chaque nouveaux registres :
→ Ajout dans le fichier RDL

Avantages :
→ Transparent dans le développement
→ Suivi des registres centralisées

Conclusions :

- Gain de temps de développement (toolchain automatique)
- Standardisation de description de registres
- Facilite la maintenance et le travail collaboratif
- Autres capacités : Export Doc (HTML), export ip-xact (XML)