

# Langage de description de matériel VHDL

- Introduction
- Concepts de base
- Conception VHDL

andre.stauffer@epfl.ch

## Introduction: vue d'ensemble

VHDL signifie VHSIC (Very High Speed Integrated Circuit) Hardware Description Language. Le langage de description de matériel VHDL est:

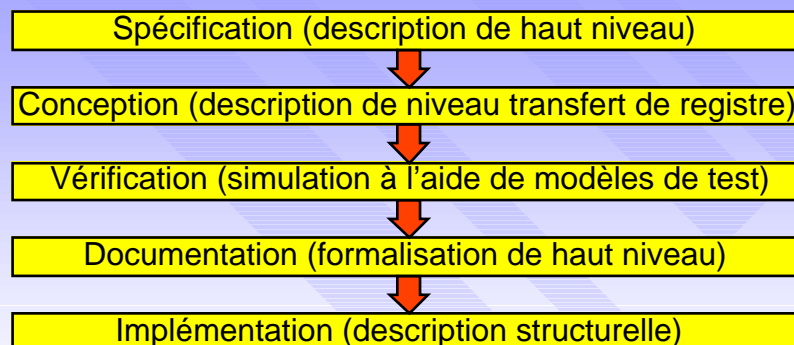
- un langage moderne, lisible, puissant et général mais complexe
- une norme de l'IEEE (Institute of Electrical and Electronics Engineers)
- un standard reconnu par tous les vendeurs d'outils CAO et par le département de la défense américaine (DOD)
- un langage commercialement inévitable et technologiquement incontournable pour la description des systèmes matériels

## Introduction: applications

Le langage de description de matériel VHDL sert à faire:

- de la spécification (description et simulation) de systèmes
- de la documentation
- de la preuve formelle
- de la conception (description de niveau transfert de registre et synthèse) de systèmes

## Introduction: méthodologie



## Concepts de base: types

**type (type): ensemble des valeurs prises par un objet et regroupées en quatre classes:**

- **type scalaires (scalar): entier (integer),  
réel (real),  
énuméré (enumerated),  
physique (physical)**
- **types composites (composite): tableau (array),  
enregistrement (record)**
- **type accès (access): pointeur (pointer) permettant d'accéder à  
des objets d'un type donné**
- **type fichier (file): séquence de valeurs d'un type donné**

## Concepts de base: types

**type physique (physical type): caractérisation d'un objet par son unité de base (base unit), l'intervalle de ses valeurs et ses éventuelles sous-unités**

```
type time is range -(2**63-1) to (2**63+1)

-- 64 bits nécessaire pour exprimer l'heure
-- en femtosecondes

units
  fs;                               -- unité de base
  ps = 1000 fs;
  ns = 1000 ps;
  us = 1000 ns;
  ms = 1000 us;
  sec = 1000 ms;
  min = 60 sec;
  hr = 60 min;
end units;
```

## Concepts de base: types

**type énuméré (enumerated type):** caractérisation d'un objet par la liste complète de ses valeurs

```
type std_logic is ( 'U', -- état non initialisé
                   'X', -- état logique indéfini fort
                   '0', -- état logique 0 fort
                   '1', -- état logique 1 fort
                   'Z', -- état à haute impédance
                   'W', -- état logique indéfini faible
                   'L', -- état logique 0 faible
                   'H', -- état logique 1 faible
                   '-' , -- état logique indifférent );
```

**type tableau (array type):** type composite consistant en un groupe d'objets dont le type est identique et la position indexée

```
type std_logic_vector(n downto 0) is array (n downto 0)
                                     of std_logic;
```

## Concepts de base: objets

**objet (object):** information manipulée par le langage et répartie en quatre classes:

- **constante (constant):** objet de valeur fixe
- **variable (variable):** objet de valeur modifiable par affectation (:=)
- **signal (signal):** objet spécifique communiqué entre les parties (processus) d'un modèle et dont l'affectation (<=) ne modifie pas la valeur présente mais les valeurs futures prévues
- **fichier (file):** objet servant de container pour le stockage de valeurs en mémoire de masse

## Unités de conception

**unité de conception (design unit):** unité de compilation VHDL correspondant à une spécification d'entité, une architecture, une spécification de paquetage, un corps de paquetage ou une configuration, stockée dans une bibliothèque

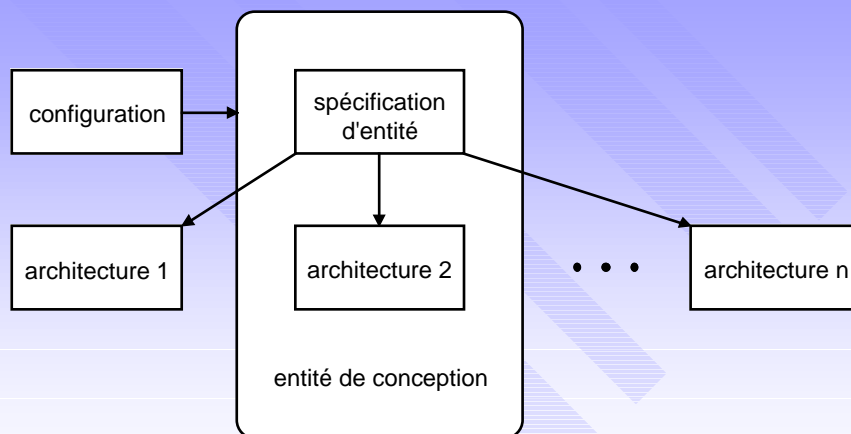
**entité de conception (design entity):** modèle VHDL formé d'un couple spécification d'entité / architecture

**spécification d'entité (entity declaration):** unité de conception correspondant à la vue externe du modèle et montrant ses connexions avec le monde extérieur

**architecture (architecture):** unité de conception correspondant à la vue interne du modèle et servant à décrire son comportement ou sa structure

**configuration (configuration):** unité de conception servant à attribuer une architecture particulière à une spécification d'entité

## Unités de conception



## Concepts de base: styles

Le langage VHDL permet la description d'un système selon trois styles de description:

- **comportemental (behavioral)**: description du fonctionnement du modèle qui ne précise rien de son éventuelle réalisation
- **flot de données (data flow)**: description basée sur les équations booléenne qui ne considère que les flots de données entrants et sortants du système
- **structurel (structural)**: description de la réalisation du système à l'aide de composants interconnectés

## Concepts de base: opérateurs

<b>logiques:</b>	<b>and , or , nand , nor , xor , not</b>
<b>relationnels:</b>	<b>= , /= , &lt; , &lt;= , &gt; , &gt;=</b>
<b>arithmétiques:</b>	<b>+ , - , * , / , ** , abs , mod , rem</b>
<b>concaténation:</b>	<b>&amp;</b>

## Conception VHDL

Le langage de description VHDL a été développé pour la documentation et la spécification des circuits

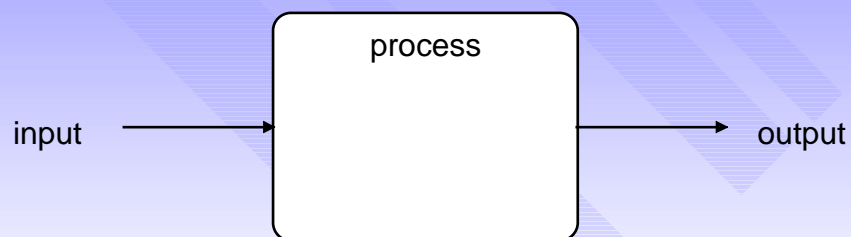
Toutes les descriptions VHDL ne sont par conséquent pas synthétisables

Nous appelons RTL (Register Transfer Level) le niveau de description pour la synthèse caractérisé par la réduction des possibilités du langage à un sous-ensemble dans lequel:

- les expressions temporelles sont ignorées
- certaines restrictions s'appliquent à l'écriture des processus
- seul un nombre limité d'objets et de types est admis
- la description est orientée vers les styles synchrones en explicitant les signaux d'horloge

## Conception VHDL

Le schéma-bloc d'un circuit combinatoire correspond à un processus



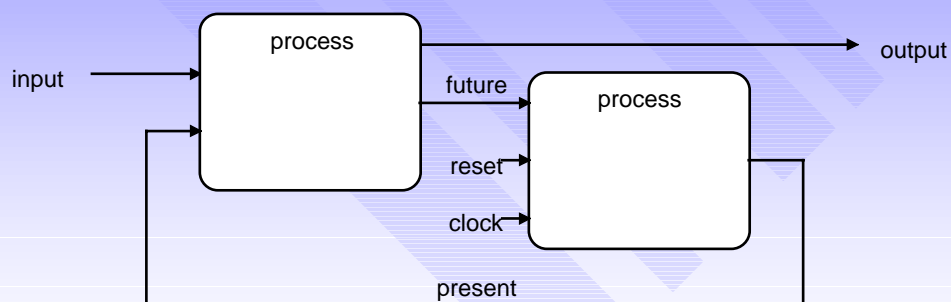
# Conception VHDL

Le fichier source d'un circuit combinatoire se compose d'une spécification d'entité et d'une architecture à un processus

```
library ieee;  
use ieee.std_logic_1164.all;  
  
entity combinational is  
  port (input :in std_logic_vector(1 to n);  
        output :out std_logic_vector(1 to r));  
end combinational;  
  
architecture rtl of combinational is  
begin  
  process (input)  
  begin  
    :  
  end process;  
end rtl;
```

# Conception VHDL

Le schéma-bloc d'un circuit séquentiel synchrone correspond à deux processus



# Conception VHDL

Le fichier source d'un circuit séquentiel synchrone se compose d'une spécification d'entité et d'une architecture à deux processus

```
library ieee;
use ieee.std_logic_1164.all;

entity sequential is
  port (input  : in  std_logic_vector(1 to n);
        reset  : in  std_logic;
        clock  : in  std_logic;
        output : out std_logic_vector(1 to r));
end sequential;
```

```
architecture rtl of sequential is
  type state is (zero, one, two, ...);
  signal present, future : state;
begin

  process (clock, reset)
  begin
    if reset='1' then
      present <= zero;
    elsif (clock'event and clock='1') then
      present <= future;
    end if;
  end process;

  process (input, present)
  begin
    case present is
      when zero =>
        :
        future <= one;
      when one  =>
        :
    end case;
  end process;

end rtl;
```