

# Hardware Modeling [VU] (191.011)

## – WS25 –

### Block Statements

Florian Huemer & Sebastian Wiedemann & Dylan Baumann

WS 2025/26

- Concurrent statement (like processes, instantiations, concurrent signal assignments, etc.)
- Blocks group concurrent statements
- Restrict scope of objects (e.g., signals) within an architecture
- Can be viewed as “inline module” or “module light” (combined module declaration and instantiation)
- Can be loosely compared to inner (nested) classes in e.g., Java

# Block Statement - Syntax

HWMod  
WS25

Blocks  
Introduction  
Syntax  
Example  
Block Header

## ■ Block syntax

```
BLOCK_LABEL : block[ ( guard_condition ) ] [ is ]  
    block_header  
    block_declarative_part  
begin  
    block_statement_part  
end block;
```

- Label is **not** optional!
- Optional guard condition (not covered in this course) 
- Optional block header: explicit block interface specification
- Declarative/statement part
  - can contain the same objects as in the respective parts of architectures → blocks can be nested
  - cannot be accessed from outside the block
  - can access objects from outer scope

# Block Statement - Example

HWMod  
WS25

Blocks  
Introduction  
Syntax  
Example  
Block Header

```
1 architecture arch of demo is
2   signal a, b, cin : std_ulogic;
3   signal cout, sum : std_ulogic;
4 begin
5   full_adder : block
6     signal x, y, z : std_ulogic;
7   begin
8     x <= a xor b;
9     y <= a and b;
10    sum <= cin xor x;
11    z <= cin and x;
12    cout <= y or z;
13  end block;
14
15  -- do something with a, b, etc.
16  some_logic: process(all)
17    [...]
18 end architecture;
```

## Note

The **process** `some_logic` **cannot** access the signals `x`, `y` and `z`.

# Block Header

HWMod  
WS25

Blocks  
Introduction  
Syntax  
Example  
Block Header

- Defines an explicit interface to a block

- **Block header syntax**

```
block_header ::=  
    [ generic ([...]); [ generic map ([...]); ] ]  
    [ port ([...]); [ port map ([...]); ] ]
```

- **Everything is optional**

- Port/generic map clause only valid if a port/generic clause is present
- If port/generic map clauses are omitted the respective port/generic clause must have default values
- Block header does not prevent the block from accessing objects from the outer scope

# Block Header - Example

HWMod  
WS25

Blocks  
Introduction  
Syntax  
Example  
Block Header

```
1 architecture arch2 of demo is
2   signal i1, i2, i3, o1, o2 : std_ulogic;
3 begin
4   full_adder : block
5     port (
6       a, b, cin : in std_ulogic;
7       sum, cout : out std_ulogic
8     );
9     port map (
10       a => i1, b => i2, cin => i3,
11       cout => o1, sum => o2
12     );
13     signal x, y, z : std_ulogic;
14 begin
15   x <= a xor b; y <= a and b;
16   z <= cin and x;
17   cout <= y or z; sum <= cin xor x;
18 end block;
19 [...]
20 end architecture;
```

## Note

Block is now self-contained. We no longer need to directly access signals from the architecture.

## Note

Trivial to move the block into a separate module (entity / architecture).

# Lecture Complete!