

Hardware Modeling [VU] (191.011)

– WS25 –

Numeric Standard Package

Florian Huemer & Sebastian Wiedemann & Dylan Baumann

WS 2025/26

Integer Arithmetic in Hardware

HWMod
WS25

numeric_std
Motivation
IEEE Package
Types
Operators
Conversion

- When possible use ranged `integer` for synthesizable code
- However: `integer` type has some limitations
 - Only 32 / 64 bits guaranteed by standard (2008 / 2019)
 - Restricted to Boolean 0 and 1 vs. 9-valued logic
 - Behavior on over-/underflow undefined $\Rightarrow$ wrap-around **or** runtime error 

$\Rightarrow$ IEEE `numeric_std` package 

- Requires import from the `ieee` library

```
library ieee;  
use ieee.numeric_std.all;
```

The IEEE numeric_std Package

331

HWMod
WS25

numeric_std

Motivation

IEEE Package

Types

Operators

Conversion

- `unsigned` and `signed` with respective operator definitions
- Wrap-around on over-/underflow behavior

```
[...]: 2147483647
```

```
[...]: 2147483647
```

```
[...]: error: overflow detected
```

```
[...]: -2147483648
```

The unsigned and signed Types

HWMod
WS25

numeric_std

Motivation

IEEE Package

Types

Operators

Conversion

- Resolved array types of `std_ulogic`
 - Can be interpreted as `std_logic_vector` for arithmetic
 - Represent unsigned / two's complement binary integers
 - Bit string literal initialization possible
 - Elements are nine-valued $\Rightarrow$ useful for debugging
- Wrap around on over- / underflow $\Rightarrow$ behavior of “real” hardware
- Examples

Arithmetic Operators

HWMod
WS25

numeric_std

Motivation

IEEE Package

Types

Operators

Conversion

- Most common operators are defined and implemented 
 - Arithmetic: `+`, `-`, `*`, `/`, `rem`, `mod`, `abs`
 - Relational: `>`, `<`, `<=`, `>=`, `=`, `/=`
 - Logical: Same as for `std_ulogic_vector`
 - Shift / rotate: `sll`, `srl`, `rol`, `ror`, `sla`, `sra`
 - `resize` function
- Useful overloads for integer / natural operand

Arithmetic Operators (cont'd)

HWMod
WS25

numeric_std

Motivation

IEEE Package

Types

Operators

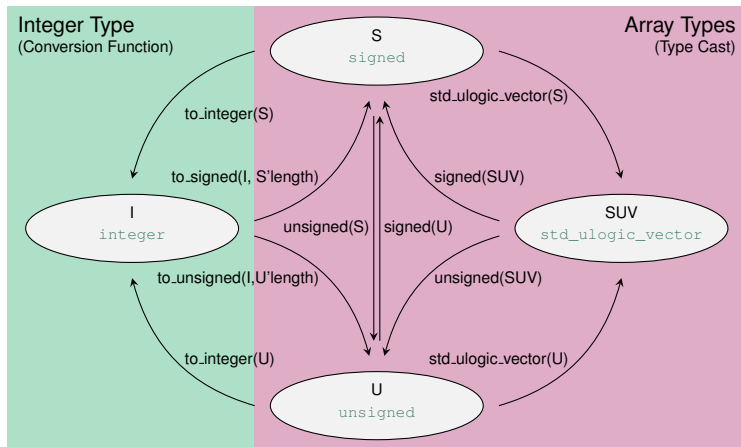
Conversion

```
1 process is
2
3
4 begin
5
6
7
8
9
10
11
12
13
14
15
16     wait;
17 end process;
```

Type Conversion

HWMod
WS25

Conversion function from / to `integer`, type casts between array types



Lecture Complete!