

Rechnergesteuerte

Systeme 2

Übung 5

Dipl.-Ing. Henner Heck

Universität Duisburg-Essen, Lehrstuhl verteilte Systeme

Skalare Datentypen

Aufzählungen

```
type MVL is ( 'U' , '0' , '1' , 'Z' ) ;
```

```
type MICRO_OP is ( LOAD , STORE , ADD , SUB , MUL , DIV ) ;
```

```
subtype ARITH_OP is MICRO_OP range ADD to DIV ;
```

(**subtype**: Einschränkung des übergeordneten Typs)

Skalare Datentypen

Integer-Typen

```
type INDEX is range 0 to 15;
```

```
type WORD_LENGTH is range 31 downto 0;
```

```
subtype DATA_WORD is WORD_LENGTH range 15 downto  
0;
```

Skalare Datentypen

Fließkomma-Typen

```
type TTL_VOLTAGE is range -5.5 to -1.4;
```

```
type REAL_DATA is range 0.0 to 31.9;
```

```
subtype RD16 is REAL_DATA range 0.0 downto 15.9;
```

Skalare Datentypen

Physikalische-Typen

```
type CURRENT is range 0 to 1E9
```

```
units
```

```
    nA;          -- Basis-Einheit
```

```
    uA =1000 nA;
```

```
    mA =1000uA;
```

```
    Amp =1000 mA;
```

```
end units;
```

```
subtype FILTER_CURRENT is CURRENT range 10
```

```
uA to 5 mA;
```

Zusammengesetzte Datentypen

Ein zusammengesetzter Typ repräsentiert eine Sammlung von Werten.

In VHDL gibt es 2 solcher Typen:

- Arrays
- Records

Array(1)

Ein Array repräsentiert eine Sammlung von Werten, die alle vom gleichen Typ sind.

Beispiele für Typdeklarationen vom Typ Array:

```
type ADRESS_WORT is array (0 to 63) of BIT;
```

```
type DATENWORT is array (7 downto 0) of  
STD_LOGIC;
```

```
type SPEICHER is array (0 to 127) of  
DATENWORT;
```

```
type DMatrix is array (0 to 3, 0 to 7) of BIT;
```

Array(2)

Objektdeklarationen der deklarierten Typen:

```
signal ADRESSBUS:DATENWORT;
```

ADRESSBUS ist ein eindimensionales Array-Objekt, bestehend aus 64 Elementen vom Typ BIT.

```
variable SPEICHER_ADDR:SPEICHER;
```

SPEICHER_ADDR ist ein eindimensionales Array-Objekt, bestehend aus 128 Elementen. Jedes dieser Elemente ist wiederum ein Array-Objekt, bestehend aus 8 Elementen vom Typ `std_logic`.

```
variable DMAT:DMATRIX;
```

DMAT ist ein zweidimensionales Array-Objekt, bestehend aus 4x8 Elementen vom Typ BIT.

Array(3) - Adressierung von Elementen

Adressierung der Elemente eines Arrays:

```
Array_Name(Elementindex) { (weitere Indizes) }
```

Beispiele

ADRESSBUS (26) bezieht sich auf das 27. Element des Array-Objektes ADRESSBUS .

SPEICHER_ADDR (10) (5) bezieht sich auf den Wert (Typ `std_logic`) bei Index 5 des bei Index 10 in SPEICHER_ADDR befindlichen Datenobjektes vom Typ DATENWORT.

DMAT (2 , 5) bezieht sich das 6te Element in der dritten Zeile der Matrix DMAT.

Array(4)

Es gibt 2 vordefinierte eindimensionale Array-Typen,
String und BIT_VECTOR:

```
variable Meldung:STRING(1 to 17) :=  
    "Hello, VHDL world";
```

```
signal RX_BUS:BIT_VECTOR(0 to 5) :=  
    0"37";  
    -- (oktaler Wert 37)
```

Record(1)

Ein Record repräsentiert eine Sammlung von Werten, die vom gleichen oder von verschiedenen Typen sind.

Beispiel für eine Typdeklaration vom Typ Record:

```
type PIN_TYP is range 0 to 10;  
type MODUL is  
  record  
    GROESSE      : INTEGER range 20 to 200;  
    KRIT_VERZ    : TIME;  
    EINGANG      : PIN_TYP;  
    AUSGANG      : PIN_TYP;  
end record;
```

Record(2)

Objektdeklarationen des deklarierten Typen

```
variable MOD1 : MODULE ;
```

MOD1 ist ein Record-Objekt vom Typ MODUL.

Wertzuweisung an MOD1

- Aggregate (in Klammern gesetzte Parameterliste, deren Elemente durch Kommata getrennt sind)

```
MOD1 := ( 50 , 20 ns , 3 , 2 ) ;
```

- Direkte Zuweisung eines Wertes an ein Element des Records

```
MOD1.KRIT_VERZ := 20 ns ;
```

Es geht noch weiter !