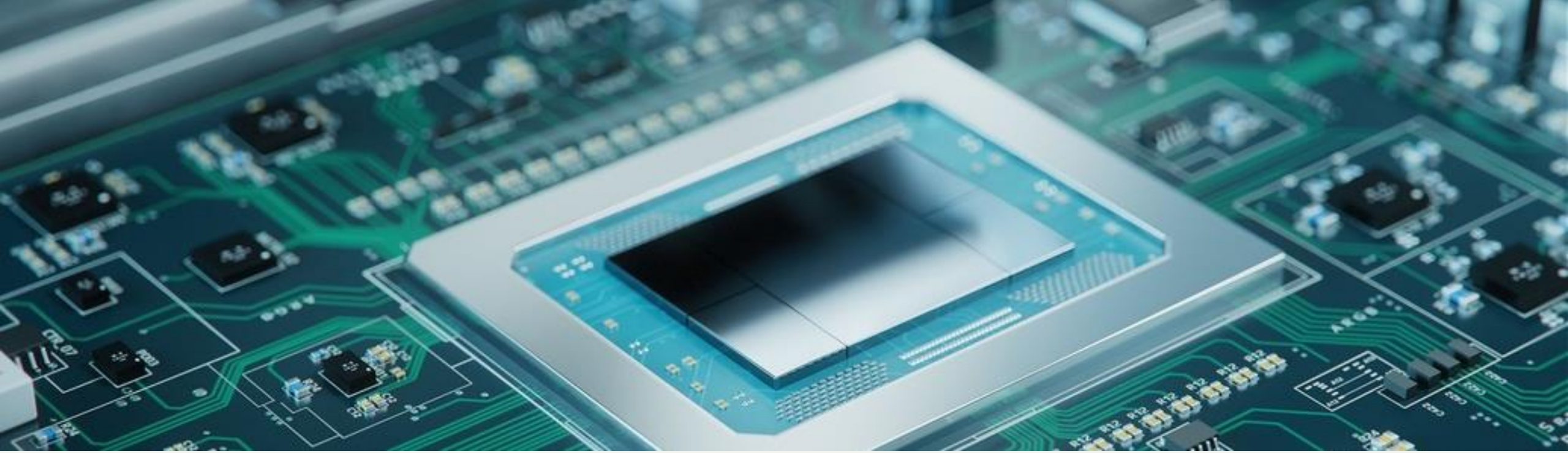

Data Types and Procedural Statements

SystemVerilog Data Types

Bob Oden

UVM Field Specialist – Siemens EDA

SIEMENS

Default data types

Type	Sign	state	Default	Storage	Size	Notes
shortint	signed	2	0	yes	16-bit	
int	signed	2	0	yes	32-bit	
longint	signed	2	0	yes	64-bit	
byte	signed	2	0	yes	8-bit	
bit	unsigned	2	0	yes	1-bit scalar	Vector formed by packing >1 bit
logic	unsigned	4	x	yes	1-bit scalar	Vector formed by packing >1 bit
reg	unsigned	4	x	yes	1-bit scalar	Vector formed by packing >1 bit
integer	signed	4	x	yes	32-bit	
time	unsigned	4	x	yes	64-bit	
wire	unsigned	4	z	no	1-bit scalar	Requires continuous assignment
tri	unsigned	4	z	no	1-bit scalar	Requires continuous assignment
string	n/a	n/a	null	yes	n/a	
chandle	n/a	n/a	null	yes	plat-dep	Handle used with DPI
real	signed	n/a	0.0	yes	64-bit	
shortreal	signed	n/a	0.0	yes	32-bit	
event	n/a	n/a	n/a	n/a		Handle to a synchronization object

Variable Declaration Examples – Integral/Real

```
shortint      my_shortint;
shortint      my_shortint_with_inital_value = 25;
int unsigned  my_unsigned_int;
int unsigned  my_int_unsigned;
longint       my_int;
byte          my_byte;
integer       my_integer;
time          my_time;
real          my_real;
shortreal     my_shortreal;
```


SIEMENS

Variable Declaration Examples – Scalar

```
bit                 my_bit;
bit                 my_bit_with_inital_value = 1'b1;
bit         [4:0]   my_5_bit_vector;
bit signed  [4:0]   my_5_bit_signed_vector;
bit                 my_bit;
logic               my_logic;
reg                 my_reg;
wire                my_wire;
tri                 my_tri;
```

Variable Declaration Examples – Handle

```
string  my_string_handle;
chandle my_chandle_handle;
event   my_event_handle;
```

Enumerations

Declares a set of integral named constants

- Abstractly declare strongly typed variables
- Type can be defined by user
- Defaults to int type

Examples

```
enum {RED,GREEN,BLUE} pixel_color;  // Defaults to int type
                                      // RED=0,GREEN=1,BLUE=2

typedef enum bit [1:0] {RED,GREEN,BLUE} pixel_color_e;

typedef enum bit [1:0] {RED=2'b01,GREEN=2'b10,BLUE=2'b11} pixel_color_e;

pixel_color_e pixel_color;
```

User-Defined Types – typedef

A user-defined name to an existing data type

- Increases code readability
- Simplifies code maintenance
- Reduces errors

Examples:

```
typedef byte my_byte_t;

typedef bit [6:0] my_custom_bit_vector_t;
my_custom_bit_vector_t my_bit_vector;
```


SIEMENS

Type Casting – Static

Expression to be cast

- Provided within parentheses
- Prefixed with casting type and apostrophe
- Assignment-compatible expression returns value casted to type

Examples:

```
int'(my_variable)

signed'(my_bit_vector)

unsigned'(byte)
```

Type Casting – Dynamic

Assign values to variables that might not be type compatible

function int $cast(singular dest_var, singular source_expression)

- Used with return value
- Success returns 1
- Failure returns 0 and does not modify dest_var

task $cast(singular dest_var, singular source_expression)

- Used without return value
- Failure results in runtime error