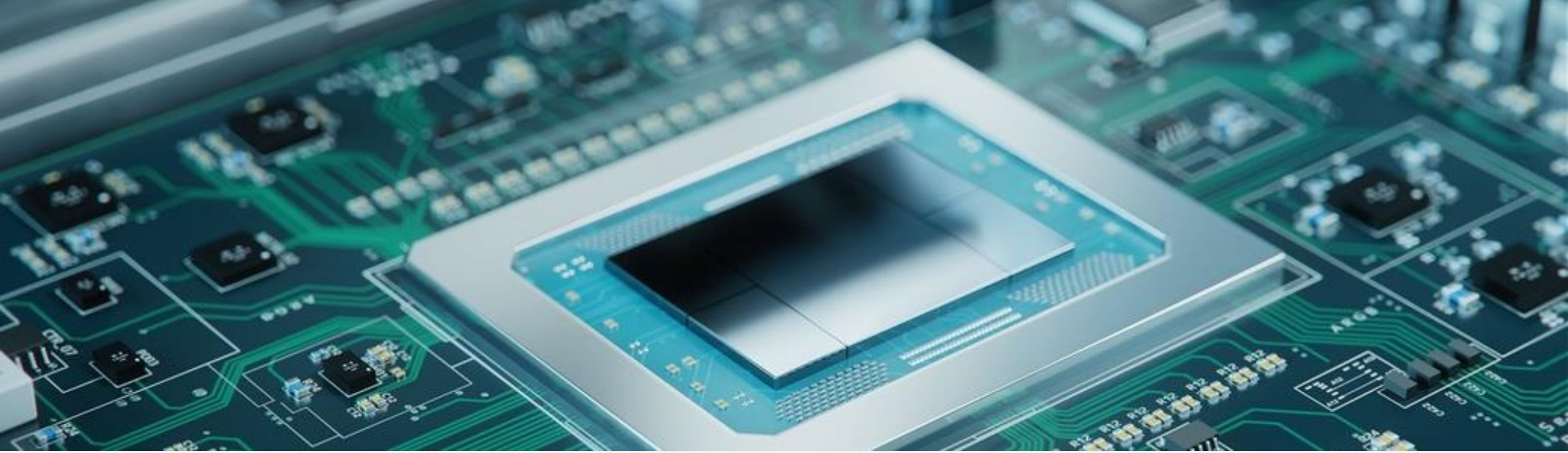

Data Types and Procedural Statements

Built-In Unpacked Arrays

Bob Oden

UVM Field Specialist – Siemens EDA

SIEMENS

Dynamic Arrays

Unpacked array whose size can be changed during runtime

Declaration

```
integer my_dynamic_array[];
```

Declaration with initialization

```
integer my_dynamic_array[]=new[5];
```

Construction

```
my_dynamic_array=new[5];
```

Construction and copy

```
my_dynamic_array = new[7] (copied_dynamic_array);
```


Dynamic Array Methods

New

```
integer my_dynamic_array[]=new[5];
my_dynamic_array[]=new[7];
integer my_dynamic_array[]=new[9](copied_array);
```

Size

- Returns current size, zero if not constructed

```
my_dynamic_array.size();
```

Delete

- Delete all entries

```
my_dynamic_array.delete();
```


SIEMENS

Associative Arrays

Dynamic array indexed by unique keys of any standard type

Syntax

```
data_type array_id [index_type];
```

Index types

- Integral: `byte my_assoc_array [ int ];`
- Wildcard: `byte my_assoc_array [ * ];`
- String: `byte my_assoc_array [ string ];`
- Class: `byte my_assoc_array [ my_class ];`
- User-defined type: `byte my_assoc_array [ my_struct_t ];`

SIEMENS

Associative Array Methods

Num and size

- Returns number of entries

```
my_assoc_array.num();
my_assoc_array.size();
```

Exists

- Returns 1 if element exists in array

```
my_assoc_array.exists(idx);
```

Delete

- Single entry or whole array

```
my_assoc_array.delete(idx); my_assoc_array.delete();
```


Associative Array Traversing Methods

```
int map[ string ];
string s;
```

First

- Assigns the first (smallest) index in the associative array into the argument

```
map.first(s)
```

Last

- Assigns the last (largest) index in the associative array into the argument

```
map.last(s)
```

Next

- Assigns the next index after the provided index into the argument

```
map.next(s)
```

Prev

- Assigns the previous index before the provided index into the argument

```
map.prev(s)
```


Queues

Variable sized-ordered collection of elements

Access to all elements using numeric index

Methods for accessing both ends of the array

Useful for arrays, FIFO, stack, etc.

Syntax

```
data_type array_id [$];
```

Examples

```
int my_ints [$];
string array_id [$] = {“ECE745”, “ECE748”};
bit array_id [$:63];  // Queue with maximum size 63
```


SIEMENS

Queue Methods

Size

- Prototype

```
function int size();
```

- Example

```
my_queue.size();
```

Insert

- Prototype

```
function void insert(input integer index, input element_t item);
```

- Example

```
my_queue.insert(4, my_element);
```

Delete

- Prototype

```
function void delete( [input integer index]);
```

- Example

```
my_queue.delete();
```


SIEMENS

Queue Methods – Front

Push front

- Prototype

```
function void push_front(input element_t item);
```

- Example

```
my_queue.push_front(my_element);
```

Pop front

- Prototype

```
function element_t pop_front();
```

- Example

```
my_element = my_queue.pop_front();
```

Queue Methods – Back

Push back

- Prototype

```
function void push_back(input element_t item);
```

- Example

```
my_queue.push_back(my_element);
```

Pop back

- Prototype

```
function element_t pop_back();
```

- Example

```
my_element = my_queue.pop_back();
```