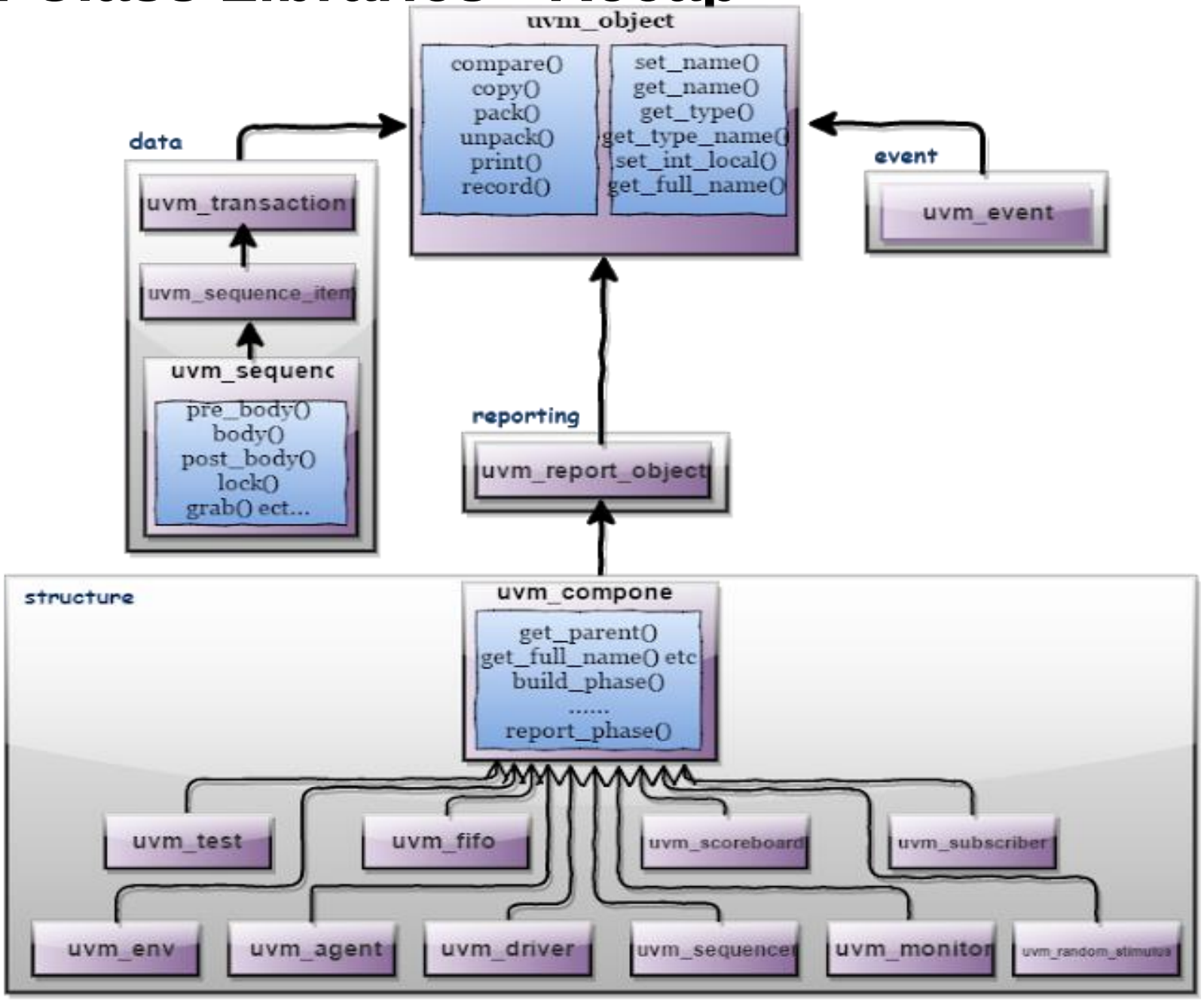
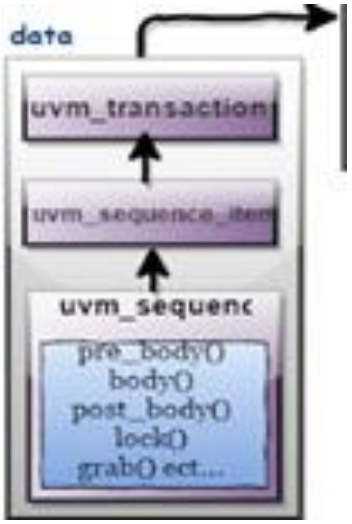
UVM Class Libraries - Recap



UVM sequence_item

Sequence_item:

- Generate the stimulus
- Randomized



Data fields represent the following information:

- Control information – a type of transfer, transfer size, etc
- Payload Information – data content of the transfer
- Configuration Information – mode of operation, error behaviour, etc
- Analysis Information – fields used to capture information from DUT, ex: read data, response, etc

UVM sequence_item

- sequence_item extended using uvm_sequence_item,
  - Inherits from the uvm_object via the uvm_transaction class
  - Therefore uvm_sequence_item is of an object type
  - Few virtual methods in uvm_object are copy, clone, compare, print, transaction, and recording
  - Utility Macros and Field Macros of uvm_object are also used

UVM sequence_item example

```
import uvm_pkg::*;
class mem_seq_item extends uvm_sequence_item;
  //Control Information
  rand bit [3:0] addr;
  rand bit       wr_en;
  rand bit       rd_en;

  //Payload Information
  rand bit [7:0] wdata;

  //Analysis Information
       bit [7:0] rdata;

  //Utility and Field macros,
  `uvm_object_utils_begin(mem_seq_item
    `uvm_field_int(addr,UVM_ALL_ON)
    `uvm_field_int(wr_en,UVM_ALL_ON)
    `uvm_field_int(rd_en,UVM_ALL_ON)
    `uvm_field_int(wdata,UVM_ALL_ON)
  `uvm_object_utils_end

  //Constructor
  function new(string name = "mem_seq_item");
    super.new(name);
  endfunction
  //constaint, to generate any one among write and read
  constraint wr_rd_c { wr_en != rd_en; };

endclass
```

```
`include "mem_seq_item.sv"
module seq_item_tb;

  //instance
  mem_seq_item seq_item;

  initial begin
    //create method
    seq_item = mem_seq_item::type_id::create();

    //randomizing the seq_item
    seq_item.randomize();

    //printing the seq_item
    seq_item.print();
  end
endmodule
```

UVM sequence_item example

```
----------------------------------------
Name           Type           Size  Value
----------------------------------------
mem_seq_item   mem_seq_item   -     @334
  addr         integral       4     'h9
  wr_en        integral       1     'h1
  rd_en        integral       1     'h0
  wdata        integral       8     'h6c
----------------------------------------
```

UVM sequence_item example

```
class instruction extends uvm_sequence_item;
  typedef enum {ADD,PUSH_A,PUSH_B,SUB,MUL,DIV,POP_C} inst_t;
  rand inst_t inst;

  `uvm_object_utils_begin(instruction)
  `uvm_field_enum(inst_t,inst, UVM_ALL_ON)
  `uvm_object_utils_end

  function new (string name = "instruction");
    super.new(name);
  endfunction

endclass

module seq_item_tb;

  instruction inst;

  initial begin
    repeat(6) begin
      inst = instruction::type_id::create();
      inst.randomize();
      inst.print();
    end
  end
endmodule
```

UVM sequence_item example

```
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @336
  inst        inst_t        32    ADD
----------------------------------------
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @340
  inst        inst_t        32    PUSH_B
----------------------------------------
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @344
  inst        inst_t        32    POP_C
----------------------------------------
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @348
  inst        inst_t        32    PUSH_B
----------------------------------------
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @352
  inst        inst_t        32    PUSH_B
----------------------------------------
----------------------------------------
Name          Type          Size  Value
----------------------------------------
instruction   instruction   -     @356
  inst        inst_t        32    PUSH_B
----------------------------------------
```

UVM Sequence Item methods

- create() : allocates a new object of the same type as this object and returns it via a base uvm_object handle.
- print() : deep-prints this object's properties in a format and manner governed by the given printer argument

UVM sequence_item copy()

- Makes this object a copy of the specified object

```
`include "mem_seq_item.sv"
module seq_item_tb;

  mem_seq_item seq_item_0;
  mem_seq_item seq_item_1;

initial begin
  //create method
  seq_item_0 = mem_seq_item::type_id::create("seq_item_0");
  seq_item_1 = mem_seq_item::type_id::create("seq_item_1");

  seq_item_0.randomize(); //randomizing the seq_item
  seq_item_0.print();     //printing the seq_item_0

 //copy method
 seq_item_1.copy(seq_item_0); //copy seq_item_0 to seq_item_1
 seq_item_1.print();          //printing the seq_item_1

  end
endmodule
```

UVM sequence_item copy()

- Makes this object a copy of the specified object

```
---------------------------------------
Name         Type           Size  Value
---------------------------------------
seq_item_0   mem_seq_item   -     @336
  addr       integral       4     'h9
  wr_en      integral       1     'h1
  rd_en      integral       1     'h0
  wdata      integral       8     'h6c
---------------------------------------
---------------------------------------
Name         Type           Size  Value
---------------------------------------
seq_item_1   mem_seq_item   -     @340
  addr       integral       4     'h9
  wr_en      integral       1     'h1
  rd_en      integral       1     'h0
  wdata      integral       8     'h6c
---------------------------------------
```

UVM sequence_item clone()

- Creates and returns an exact copy of this object = create() + copy()

- `uvm_info
- `uvm_info(ID,MSG,VERBOSITY)
- *ID* is given as the message tag and *MSG* is given as the message text. The file and line are also sent to the uvm_report_info call.

```
module seq_item_tb;

  //instance
  mem_seq_item seq_item_0;
  mem_seq_item seq_item_1;
 initial begin
   //create method
   seq_item_0 = mem_seq_item::type_id::create("seq_item_0");
   seq_item_0.randomize(); //randomizing the seq_item
   seq_item_0.print();     //printing the seq_item_0
  //clone method
  $cast(seq_item_1,seq_item_0.clone()); //create seq_item_1
  //and copy seq_item_0 to seq_item_1
  //changing the seq_item_1 values will not reflect on
  //seq_item_0 values.
  seq_item_1.addr  = 8;
  seq_item_1.wdata = 'h56;
  `uvm_info("","Printing seq_item_0", UVM_LOW)
  seq_item_0.print();          //printing the seq_item_0
  `uvm_info("","Printing seq_item_1", UVM_LOW)
  seq_item_1.print();          //printing the seq_item_1
 end
endmodule
```

UVM sequence_item clone()

- Creates and returns an exact copy of this object = create() + copy()

```
----------------------------------
Name        Type          Size  Value
----------------------------------
seq_item_0  mem_seq_item  -     @334
  addr      integral      4     'h9
  wr_en     integral      1     'h1
  rd_en     integral      1     'h0
  wdata     integral      8     'h6c
----------------------------------
UVM_INFO testbench.sv(47) @ 0: reporter [] Printing seq_item_0
----------------------------------
Name        Type          Size  Value
----------------------------------
seq_item_0  mem_seq_item  -     @334
  addr      integral      4     'h9
  wr_en     integral      1     'h1
  rd_en     integral      1     'h0
  wdata     integral      8     'h6c
UVM_INFO testbench.sv(49) @ 0: reporter [] Printing seq_item_1
----------------------------------
Name        Type          Size  Value
----------------------------------
seq_item_0  mem_seq_item  -     @338
  addr      integral      4     'h8
  wr_en     integral      1     'h1
  rd_en     integral      1     'h0
  wdata     integral      8     'h56
----------------------------------
```

UVM sequence_item compare()

Deep compares members of this data object with those of the object provided in the RHS argument, returning 1 on a match, 0 otherwise.

```
module seq_item_tb;
  //instance
  mem_seq_item seq_item_0;
  mem_seq_item seq_item_1;
  initial begin
    //create method
    seq_item_0 = mem_seq_item::type_id::create("seq_item_0");
    seq_item_1 = mem_seq_item::type_id::create("seq_item_1");
    //---------------Mismatch Case------------------------------
    seq_item_0.randomize(); //randomizing the seq_item_0
    seq_item_1.randomize(); //randomizing the seq_item_1

    seq_item_0.print();     //printing the seq_item_0
    seq_item_1.print();     //printing the seq_item_1

    //compare method
    if(seq_item_0.compare(seq_item_1))
      `uvm_info("","seq_item_0 matching with seq_item_1", UVM_LOW)
    else
      `uvm_error("","seq_item_0 is not matching with seq_item_1")
    //---------------Matching Case------------------------------
    seq_item_1.copy(seq_item_0); //copy seq_item_0 to seq_item_1
    //compare method
    if(seq_item_0.compare(seq_item_1))
      `uvm_info("","seq_item_0 matching with seq_item_1", UVM_LOW)
    else
      `uvm_error("","seq_item_0 is not matching with seq_item_1")
  end
endmodule
```

UVM sequence_item compare()

```
----------------------------------------
Name         Type           Size  Value
----------------------------------------
seq_item_0   mem_seq_item   -     @334
  addr       integral       4     'h9
  wr_en      integral       1     'h1
  rd_en      integral       1     'h0
  wdata      integral       8     'h6c
----------------------------------------
----------------------------------------
Name         Type           Size  Value
----------------------------------------
seq_item_1   mem_seq_item   -     @338
  addr       integral       4     'h7
  wr_en      integral       1     'h1
  rd_en      integral       1     'h0
  wdata      integral       8     'h14
----------------------------------------
UVM_ERROR testbench.sv(55) @ 0: reporter [] seq_item_0 is not matching with seq_item_1
UVM_INFO testbench.sv(61) @ 0: reporter [] seq_item_0 matching with seq_item_1
```

UVM sequence_item pack()

- The pack methods bit-wise concatenate this object's properties into an array of bits, bytes, or ints.
- The unpack methods extract property values from an array of bits, bytes, or ints.

UVM sequence_item pack()

```
class mem_seq_item extends uvm_sequence_item;

  rand bit [3:0] addr;
  rand bit [7:0] wdata;

  `uvm_object_utils_begin(mem_seq_item)
  `uvm_field_int(addr,UVM_DEFAULT)
  `uvm_field_int(wdata,UVM_DEFAULT)
  `uvm_object_utils_end

  function new(string name = "mem_seq_item");
    super.new(name);
  endfunction

endclass
```

UVM sequence_item pack()

```
module seq_item_tb;

  //instance
  mem_seq_item seq_item_0;
  mem_seq_item seq_item_1;
  bit bit_packed_data[];

  initial begin
    //create method
    seq_item_0 = mem_seq_item::type_id::create("seq_item_0");
    seq_item_1 = mem_seq_item::type_id::create("seq_item_1");

    //---------------------- PACK  -----------------------------
    seq_item_0.randomize(); //randomizing the seq_item_0
    seq_item_0.print();     //printing the seq_item_0

    seq_item_0.pack(bit_packed_data);    //pack method
    foreach(bit_packed_data[i])
      `uvm_info("PACK",$sformatf("bit_packed_data[%0d] = %b",i,bit_packed_data[i]),
UVM_LOW)

  end
endmodule
```

UVM sequence_item pack()

```
----------------------------------------
Name         Type           Size  Value
----------------------------------------
seq_item_0   mem_seq_item   -     @334
  addr       integral       4     'h9
  wdata      integral       8     'h6c
----------------------------------------
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[0] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[1] = 0
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[2] = 0
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[3] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[4] = 0
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[5] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[6] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[7] = 0
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[8] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[9] = 1
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[10] = 0
UVM_INFO testbench.sv(36) @ 0: reporter [PACK] bit_packed_data[11] = 0
```