**Module: UVM-2**

**Register Abstraction Layer**

**RAL**

Assignment 02

* **Configure Method:**

There are actually 9 arguments passed to a uvm\_reg\_field’s configure method, each one of the argument and its working is briefly explained below:

1. **uvm\_reg parent:** Specifies the parent register to which this field belongs.
2. **int unsigned size:** Defines the size of the field in bits.
3. **int unsigned lsb\_pos:** Specifies the position of the least significant bit (LSB) of the field within the register. For example, if lsb\_pos = 0, the field starts at the least significant bit of the register.
4. **string access:** Defines the field's access type. Common values include:
   * + "RW": Read-Write
     + "RO": Read-Only
     + "WO": Write-Only
     + "W1": Write-One (write a 1 to set, writing 0 has no effect).
5. **bit volatile:** Indicates whether the field is volatile.
6. **uvm\_reg\_data\_t reset:** This value determines the initial state of the field after a reset. It’s useful for modeling reset behavior in registers.
7. **bit has\_reset:** Indicates whether the field has an explicitly defined reset value.
8. **bit is\_rand:** Specifies whether the field is randomized during constrained random testing.
9. **bit individually\_accessible:** Indicates whether the field can be accessed independently of other fields in the register.

* **Modeling Memory:**

UVM allows us to model the DUT memory. Such memory is constructed using a class extended from the **uvm\_mem** class as following:

| class mem\_mod extends uvm\_mem;  `uvm\_object\_utils(mem\_mod)    // Constructor  function new( string name = "" );  super.new(.name(name), // Name of the Memory  .size(256), // Depth of the Memory  .n\_bits(32), // Width of Memory  .access("RW") // Access Policy  );  endfunction : new  endclass |
| --- |

* **Register Classes:**

| **Control Register** |
| --- |
| class cntrl extends uvm\_reg;  // Factory Registration  `uvm\_object\_utils(cntrl)  rand uvm\_reg\_field control;  // Constructor  function new ( string name = "cntrl");  super.new( name, 4, UVM\_NO\_COVERAGE )  endfunction : new  virtual function void build();  control = uvm\_reg\_field::type\_id::create("control");  control.configure( this, 4, 0, "RW", 0, 0, 1, 1, 0 );  endfunction : build  endclass : cntrl |

| **Register 1** |
| --- |
| class reg1 extends uvm\_reg;  // Factory Registration  `uvm\_object\_utils(reg1)  rand uvm\_reg\_field Register1;  // Constructor  function new ( string name = "reg1");  super.new( name, 32, UVM\_NO\_COVERAGE )  endfunction : new  virtual function void build();  control = uvm\_reg\_field::type\_id::create("Register1");  control.configure( this, 32, 0, "RW", 0, 0, 1, 1, 0 );  endfunction : build  endclass : reg1 |

| **Register 2** |
| --- |
| class reg2 extends uvm\_reg;  // Factory Registration  `uvm\_object\_utils(reg2)  rand uvm\_reg\_field Register2;  // Constructor  function new ( string name = "reg2");  super.new( name, 32, UVM\_NO\_COVERAGE )  endfunction : new  virtual function void build();  control = uvm\_reg\_field::type\_id::create("Register2");  control.configure( this, 32, 0, "RW", 0, 0, 1, 1, 0 );  endfunction : build  endclass : reg2 |

| **Register 3** |
| --- |
| class reg3 extends uvm\_reg;  // Factory Registration  `uvm\_object\_utils(reg3)  rand uvm\_reg\_field Register3;  // Constructor  function new ( string name = "reg3");  super.new( name, 32, UVM\_NO\_COVERAGE )  endfunction : new  virtual function void build();  control = uvm\_reg\_field::type\_id::create("Register3");  control.configure( this, 32, 0, "RW", 0, 0, 1, 1, 0 );  endfunction : build  endclass : reg3 |

| **Register 4** |
| --- |
| class reg4 extends uvm\_reg;  // Factory Registration  `uvm\_object\_utils(reg4)  rand uvm\_reg\_field Register4;  // Constructor  function new ( string name = "reg4");  super.new( name, 32, UVM\_NO\_COVERAGE )  endfunction : new  virtual function void build();  control = uvm\_reg\_field::type\_id::create("Register4");  control.configure( this, 32, 0, "RW", 0, 0, 1, 1, 0 );  endfunction : build  endclass : reg4 |