

Part1 :

ALSU_top

```
1 import uvm_pkg::*;
2 `include "uvm_macros.svh"
3 import ALSU_test_pkg::*;
4
5 module ALSU_top();
6     bit clk;
7     initial begin
8         clk = 0 ;
9         forever begin
10             #1 clk = ~clk ;
11         end
12     end
13
14     ALSU_if inst_if (clk) ;
15     ALSU inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
16                 inst_if.serial_in, inst_if.red_op_A,
17                 inst_if.red_op_B, inst_if.opcode,
18                 inst_if.bypass_A, inst_if.bypass_B,
19                 inst_if.clk, inst_if.rst, inst_if.direction,
20                 inst_if.leds, inst_if.out);
21
22     initial begin
23         uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
24         run_test ("ALSU_test") ;
25     end
26 endmodule
```

ALSU_test

```
1 package ALSU_test_pkg;
2     import uvm_pkg::*;
3     `include "uvm_macros.svh"
4     import ALSU_env_pkg::*;
5     // import ALSU_config_pkg::*;
6     // ALSU_config ALSU_cfg;
7
8     class ALSU_test extends uvm_test;
9         `uvm_component_utils(ALSU_test)
10         ALSU_env env ;
11
12         function new(string name = "ALSU_test" , uvm_component parent = null);
13             super.new(name , parent) ;
14         endfunction
15
16         function void build_phase (uvm_phase phase);
17             super.build_phase(phase) ;
18             env = ALSU_env::type_id::create("env",this);
19         endfunction
20
21         task run_phase (uvm_phase phase);
22             super.run_phase (phase);
23             phase.raise_objection (this) ;
24             #100;
25             `uvm_info ("run_phase" , ""Inside the ALSU test"" , UVM_MEDIUM)
26             phase.drop_objection (this) ;
27         endtask
28     endclass
29 endpackage
```

ALSU_env

```
1 package ALSU_env_pkg;
2   import uvm_pkg::*;
3   `include "uvm_macros.svh"
4
5   class ALSU_env extends uvm_env;
6     `uvm_component_utils(ALSU_env)
7     function new(string name = "ALSU_env" , uvm_component parent = null);
8       super.new(name , parent);
9     endfunction
10  endclass
11 endpackage
```

ALSU_if

```
1 interface ALSU_if (clk);
2   input clk ;
3   logic rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
4   logic signed [1:0] cin ;
5   logic [2:0] opcode;
6   logic signed [2:0] A, B;
7   logic [15:0] leds;
8   logic signed [5:0] out;
9 endinterface //ALSU_if
```

Source file

```
1 ALSU_if.sv
2 ALSU.v
3 ALSU_env.sv
4 ALSU_test.sv
5 ALSU_top.sv
```

Transcript

```
***** IMPORTANT RELEASE NOTES *****
#
# You are using a version of the UVM library that has been compiled
# with `UVM_NO_DEPRECATED` undefined.
# See http://www.eda.org/svdb/view.php?id=3313 for more details.
#
# You are using a version of the UVM library that has been compiled
# with `UVM_OBJECT_MUST_HAVE_CONSTRUCTOR` undefined.
# See http://www.eda.org/svdb/view.php?id=3770 for more details.
#
# (Specify +UVM_NO_RELNOTES to turn off this notice)
#
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(277) @ 0: reporter [Questa UVM] QUESTA_UVM-1.2.3
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(278) @ 0: reporter [Questa UVM]  questa_uvm::init(all)
# UVM_INFO @ 0: reporter [RNTST] Running test ALSU_test...
# UVM_INFO ALSU_test.sv(25) @ 100: uvm_test_top [run_phase] Inside the ALSU test
# UVM_INFO verilog_src/uvm-1.1d/src/base/uvm_object.svh(1267) @ 100: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
#
--- UVM Report Summary ---
#
** Report counts by severity
# UVM_INFO : 5
# UVM_WARNING : 0
# UVM_ERROR : 0
# UVM_FATAL : 0
#
** Report counts by id
# [Questa UVM] 2
# [RNTST] 1
# [TEST_DONE] 1
# [run_phase] 1
#
** Note: $finish : C:/questasim64_2021.1/win64/.../verilog_src/uvm-1.1d/src/base/uvm_root.svh(430)
# Time: 100 ns Iteration: 54 Instance: /ALSU_top
#
1
```

Part2 :

ALSU_top

```
1  import uvm_pkg::*;
2  `include "uvm_macros.svh"
3  import ALSU_test_pkg::*;
4
5  module ALSU_top();
6      bit clk;
7      initial begin
8          clk = 0 ;
9          forever begin
10             #1 clk = ~clk ;
11         end
12     end
13
14     ALSU_if inst_if (clk) ;
15     ALSU inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
16                 inst_if.serial_in, inst_if.red_op_A,
17                 inst_if.red_op_B, inst_if.opcode,
18                 inst_if.bypass_A, inst_if.bypass_B,
19                 inst_if.clk, inst_if.rst, inst_if.direction,
20                 inst_if.leds, inst_if.out);
21
22     initial begin
23         uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
24         run_test ("ALSU_test") ;
25     end
26 endmodule
```

ALSU_test

```
1 package ALSU_test_pkg;
2   import uvm_pkg::*;
3   `include "uvm_macros.svh"
4   import ALSU_env_pkg::*;
5   // import ALSU_config_pkg::*;
6   // ALSU_config ALSU_cfg;
7
8   class ALSU_test extends uvm_test;
9     `uvm_component_utils(ALSU_test)
10    ALSU_env env ;
11    virtual ALSU_if alsu_test_vif ;
12
13    function new(string name = "ALSU_test" , uvm_component parent = null);
14      super.new(name , parent) ;
15    endfunction
16
17    function void build_phase (uvm_phase phase);
18      super.build_phase(phase) ;
19      env = ALSU_env::type_id::create("env",this);
20
21      if (!uvm_config_db #(virtual ALSU_if)::get(this,"", "ALSU_if",alsu_test_vif))
22        `uvm_fatal("build_phase" , "TEST - unable to get the virtual intrface");
23
24      uvm_config_db #(virtual ALSU_if)::set(this,"*", "TEST",alsu_test_vif);
25    endfunction
26
27    task run_phase (uvm_phase phase);
28      super.run_phase (phase);
29      phase.raise_objection (this) ;
30      #100;
31      `uvm_info ("run_phase" , "Inside the ALSU test" , UVM_MEDIUM)
32      phase.drop_objection (this) ;
33    endtask
34  endclass
35 endpackage
```

ALSU_env

```
1 package ALSU_env_pkg;
2   import uvm_pkg::*;
3   `include "uvm_macros.svh"
4   import ALSU_driver_pkg::*;
5
6   class ALSU_env extends uvm_env;
7     `uvm_component_utils(ALSU_env)
8     ALSU_driver driver ;
9     function new(string name = "ALSU_env" , uvm_component parent = null);
10       super.new(name , parent);
11     endfunction
12
13     function void build_phase (uvm_phase phase);
14       super.build_phase (phase);
15       driver = ALSU_driver::type_id::create("driver",this) ;
16     endfunction
17  endclass
18 endpackage
```

ALSU_if

```
1 interface ALSU_if (clk);
2     input clk ;
3     logic rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
4     logic signed [1:0] cin ;
5     logic [2:0] opcode;
6     logic signed [2:0] A, B;
7     logic [15:0] leds;
8     logic signed [5:0] out;
9 endinterface //ALSU_if
```

ALSU_driver

```
1 package ALSU_driver_pkg;
2     import uvm_pkg::*;
3     `include "uvm_macros.svh"
4
5     class ALSU_driver extends uvm_driver;
6         `uvm_component_utils(ALSU_driver)
7         virtual ALSU_if alsu_driver_vif ;
8         function new(string name = "ALSU_driver" , uvm_component parent = null);
9             super.new(name , parent);
10        endfunction
11
12        function void build_phase (uvm_phase phase);
13            super.build_phase(phase) ;
14            uvm_config_db #(virtual ALSU_if)::get(this , "" , "TEST" , alsu_driver_vif) ;
15        endfunction
16
17        task run_phase (uvm_phase phase);
18            super.run_phase(phase) ;
19            alsu_driver_vif.rst = 1 ;
20            forever begin
21                @(negedge alsu_driver_vif.clk) ;
22                alsu_driver_vif.red_op_A = $random ;
23                alsu_driver_vif.red_op_B = $random ;
24                alsu_driver_vif.bypass_A = $random ;
25                alsu_driver_vif.bypass_B = $random ;
26                alsu_driver_vif.direction = $random ;
27                alsu_driver_vif.serial_in = $random ;
28                alsu_driver_vif.cin = $random ;
29                alsu_driver_vif.opcode = $random ;
30                alsu_driver_vif.A = $random ;
31                alsu_driver_vif.B = $random ;
32            end
33        endtask
34    endclass
35 endpackage
```

Source file



Transcript

```
***** IMPORTANT RELEASE NOTES *****
# You are using a version of the UVM library that has been compiled
# with 'UVM_NO_DEPRECATED' undefined.
# See http://www.eda.org/svdb/view.php?id=3313 for more details.
#
# You are using a version of the UVM library that has been compiled
# with 'UVM_OBJECT_MUST_HAVE_CONSTRUCTOR' undefined.
# See http://www.eda.org/svdb/view.php?id=3770 for more details.
#
# (Specify +UVM_NO_RELNOTES to turn off this notice)
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(277) @ 0: reporter [Questa UVM] QUESTA_UVM-1.2.3
# UVM_INFO verilog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(278) @ 0: reporter [Questa UVM] questa_uvm::init(all)
# UVM_INFO @ 0: reporter [RNTST] Running test ALSU_test...
# UVM_INFO ALSU_test.sv(31) @ 100: uvm_test_top [run_phase] &inside the ALSU test
# UVM_INFO verilog_src/uvm-1.1d/src/base/uvm_objection.svh(1267) @ 100: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
# --- UVM Report Summary ---
# ** Report counts by severity
# UVM_INFO : 5
# UVM_WARNING : 0
# UVM_ERROR : 0
# UVM_FATAL : 0
# ** Report counts by id
# [Questa UVM] 2
# [RNTST] 1
# [TEST_DONE] 1
# [run_phase] 1
# ** Note: $finish : C:/questasim64_2021.1/win64/.../verilog_src/uvm-1.1d/src/base/uvm_root.svh(430)
# Time: 100 ns Iteration: 54 Instance: /ALSU_top
# 1
# Break in Task uvm_pkg/uvm_root::run_test at C:/questasim64_2021.1/win64/.../verilog_src/uvm-1.1d/src/base/uvm_root.svh line 430
```

Part 3 :

ALSU_top

```
1 import uvm_pkg::*;
2 `include "uvm_macros.svh"
3 import ALSU_test_pkg::*;
4
5 module ALSU_top();
6   bit clk;
7   initial begin
8     clk = 0 ;
9     forever begin
10       #1 clk = ~clk ;
11     end
12   end
13
14   ALSU_if inst_if (clk) ;
15   ALSU_inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
16     inst_if.serial_in, inst_if.red_op_A,
17     inst_if.red_op_B, inst_if.opcode,
18     inst_if.bypass_A, inst_if.bypass_B,
19     inst_if.clk, inst_if.rst, inst_if.direction,
20     inst_if.leds, inst_if.out);
21
22   initial begin
23     uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
24     run_test ("ALSU_test") ;
25   end
26 endmodule
```

ALSU_test

```
1 package ALSU_test_pkg;
2   import uvm_pkg::*;
3   `include "uvm_macros.svh"
4   import ALSU_env_pkg::*;
5   import ALSU_config_pkg::*;
6
7   class ALSU_test extends uvm_test;
8     `uvm_component_utils(ALSU_test)
9     ALSU_env env ;
10    ALSU_config_obj alsu_config_obj_test;
11
12    function new(string name = "ALSU_test" , uvm_component parent = null);
13      super.new(name , parent) ;
14    endfunction
15
16    function void build_phase (uvm_phase phase);
17      super.build_phase(phase) ;
18      env = ALSU_env::type_id::create("env",this);
19      alsu_config_obj_test = ALSU_config_obj::type_id::create("alsu_config_obj_test");
20
21      if (!uvm_config_db #(virtual ALSU_if)::get(this,"", "ALSU_if",alsu_config_obj_test.alsu_config_vif))
22        `uvm_fatal("build_phase" , "TEST - unable to get the virtual intrface");
23
24      uvm_config_db #(ALSU_config_obj)::set(this,"*", "TEST",alsu_config_obj_test);
25    endfunction
26
27    task run_phase (uvm_phase phase);
28      super.run_phase (phase);
29      phase.raise_objection (this) ;
30      #100;
31      `uvm_info ("run_phase" , "Inside the ALSU test" , UVM_MEDIUM)
32      phase.drop_objection (this) ;
33    endtask
34  endclass
35 endpackage
```

ALSU_env

```
1 package ALSU_env_pkg;
2   import uvm_pkg::*;
3   `include "uvm_macros.svh"
4   import ALSU_driver_pkg::*;
5
6   class ALSU_env extends uvm_env;
7     `uvm_component_utils(ALSU_env)
8     ALSU_driver driver ;
9     function new(string name = "ALSU_env" , uvm_component parent = null);
10       super.new(name , parent);
11     endfunction
12
13     function void build_phase (uvm_phase phase);
14       super.build_phase (phase);
15       driver = ALSU_driver::type_id::create("driver",this) ;
16     endfunction
17   endclass
18 endpackage
```

ALSU_driver

```
1 package ALSU_driver_pkg;
2     import uvm_pkg::*;
3     `include "uvm_macros.svh"
4     import ALSU_config_pkg::*;
5
6     class ALSU_driver extends uvm_driver;
7         `uvm_component_utils(ALSU_driver)
8         virtual ALSU_if alsu_driver_vif ;
9         ALSU_config_obj alsu_config_obj_driver;
10
11         function new(string name = "ALSU_driver" , uvm_component parent = null);
12             super.new(name , parent);
13         endfunction
14
15         function void build_phase (uvm_phase phase);
16             super.build_phase(phase) ;
17             uvm_config_db #(ALSU_config_obj)::get(this , "" , "TEST" , alsu_config_obj_driver) ;
18         endfunction
19
20         function void connect_phase (uvm_phase phase);
21             super.connect_phase(phase) ;
22             alsu_driver_vif = alsu_config_obj_driver.alsu_config_vif ;
23         endfunction
24
25         task run_phase (uvm_phase phase);
26             super.run_phase(phase) ;
27             alsu_driver_vif.rst = 1 ;
28             forever begin
29                 @(negedge alsu_driver_vif.clk) ;
30                 alsu_driver_vif.red_op_A = $random ;
31                 alsu_driver_vif.red_op_B = $random ;
32                 alsu_driver_vif.bypass_A = $random ;
33                 alsu_driver_vif.bypass_B = $random ;
34                 alsu_driver_vif.direction = $random ;
35                 alsu_driver_vif.serial_in = $random ;
36                 alsu_driver_vif.cin = $random ;
37                 alsu_driver_vif.opcode = $random ;
38                 alsu_driver_vif.A = $random ;
39                 alsu_driver_vif.B = $random ;
40             end
41         endtask
42     endclass
43 endpackage
```


ALSU_configure

```
1 package ALSU_config_pkg;
2     import uvm_pkg::*;
3     `include "uvm_macros.svh"
4
5     class ALSU_config_obj extends uvm_object;
6         `uvm_object_utils(ALSU_config_obj)
7         virtual ALSU_if alsu_config_vif ;
8         function new(string name = "ALSU_config_obj");
9             super.new(name) ;
10        endfunction
11    endclass
12 endpackage
```

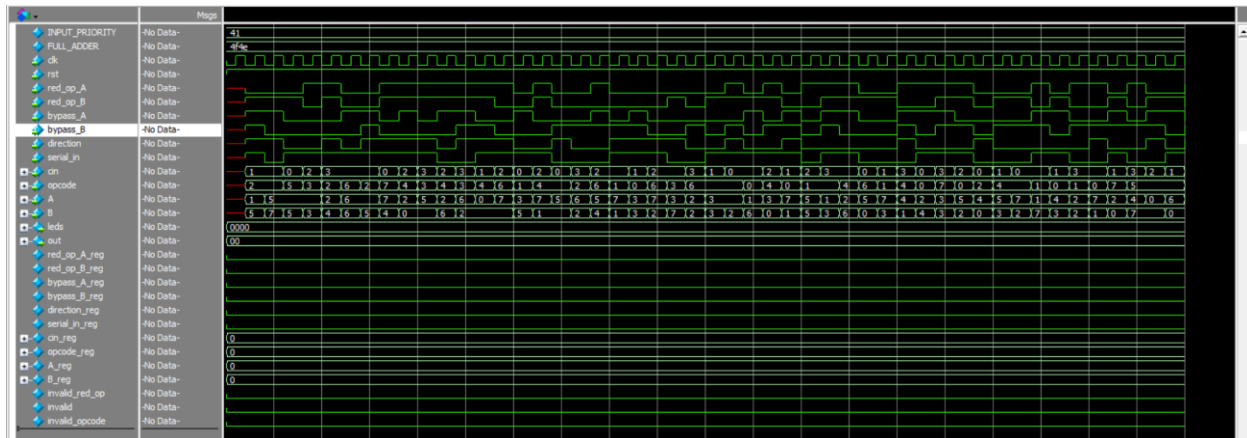
ALSU_if

```
1 interface ALSU_if (clk);
2     input clk ;
3     logic rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
4     logic signed [1:0] cin ;
5     logic [2:0] opcode;
6     logic signed [2:0] A, B;
7     logic [15:0] leds;
8     logic signed [5:0] out;
9 endinterface //ALSU_if
```

Source file

```
1 ALSU_if.sv
2 ALSU.v
3 ALSU_config.sv
4 ALSU_driver.sv
5 ALSU_env.sv
6 ALSU_test.sv
7 ALSU_top.sv
```

Waveform



Transcript

```
#
# UVM-1.1.d
# (C) 2007-2013 Mentor Graphics Corporation
# (C) 2007-2013 Cadence Design Systems, Inc.
# (C) 2006-2013 Synopsys, Inc.
# (C) 2011-2013 Cypress Semiconductor Corp.
#
# ***** IMPORTANT RELEASE NOTES *****
#
# You are using a version of the UVM library that has been compiled
# with 'UVM_NO_DEPRECATED undefined.
# See http://www.eda.org/svdb/view.php?id=3313 for more details.
#
# You are using a version of the UVM library that has been compiled
# with 'UVM_OBJECT_MUST_HAVE_CONSTRUCTOR undefined.
# See http://www.eda.org/svdb/view.php?id=3770 for more details.
#
# (Specify +UVM_NO_RELNOTES to turn off this notice)
#
# UVM_INFO verillog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(277) @ 0: reporter [Questa UVM] QUESTA_UVM-1.2.3
# UVM_INFO verillog_src/questa_uvm_pkg-1.2/src/questa_uvm_pkg.sv(278) @ 0: reporter [Questa UVM] questa_uvm::init(all)
# UVM_INFO @ 0: reporter [RNTST] Running test ALSU_test...
# UVM_INFO ALSU_test.sv(31) @ 100: uvm_test_top [run_phase] &Inside the ALSU test&
# UVM_INFO verillog_src/uvm-1.1.d/src/base/uvm_objection.svh(1267) @ 100: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
#
# --- UVM Report Summary ---
#
# ** Report counts by severity
# UVM_INFO : 5
# UVM_WARNING : 0
# UVM_ERROR : 0
# UVM_FATAL : 0
#
# ** Report counts by id
# [Questa UVM] 2
# [RNTST] 1
# [TEST_DONE] 1
# [run_phase] 1
#
# ** Note: $finish : C:/questasim64_2021.1/win64/./verilog_src/uvm-1.1.d/src/base/uvm_root.svh(430)
# Time: 100 ns Iteration: 54 Instance: /ALSU_top
```

Do file

```
1 vlib work
2 vlog -f src_files.list
3 vsim -voptargs=+acc work.ALSU_top -classdebug -uvmcontrol=all
4 run -all
```