Part1 :

ALSU_top

```
import uvm_pkg::*;
`include "uvm_macros.svh"
import ALSU_test_pkg::*;

module ALSU_top();
    bit clk;
    initial begin
        clk = 0 ;
        forever begin
            #1 clk = ~clk ;
        end
    end

    ALSU_if inst_if (clk) ;
    ALSU inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
                   inst_if.serial_in, inst_if.red_op_A,
                       inst_if.red_op_B, inst_if.opcode,
                           inst_if.bypass_A, inst_if.bypass_B,
                               inst_if.clk, inst_if.rst, inst_if.direction,
                                   inst_if.leds, inst_if.out);

    initial begin
        uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
        run_test ("ALSU_test") ;
    end
endmodule
```

ALSU_test

```
package ALSU_test_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_env_pkg::*;
    // import ALSU_config_pkg::*;
    // ALSU_config ALSU_cfg;

    class ALSU_test extends uvm_test;
        `uvm_component_utils(ALSU_test)
        ALSU_env env ;

        function new(string name = "ALSU_test" , uvm_component parent = null);
            super.new(name , parent) ;
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase(phase) ;
            env = ALSU_env::type_id::create("env",this);
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase (phase);
            phase.raise_objection (this) ;
            #100;
            `uvm_info ("run_phase" , "“Inside the ALSU test”" , UVM_MEDIUM)
            phase.drop_objection (this) ;
        endtask
    endclass
endpackage
```

ALSU_env

```
package ALSU_env_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"

    class ALSU_env extends uvm_env;
        `uvm_component_utils(ALSU_env)
        function new(string name = "ALSU_env" , uvm_component parent = null);
            super.new(name , parent);
        endfunction
    endclass
endpackage
```

ALSU_if

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

Source file

```
ALSU_if.sv
ALSU.v
ALSU_env.sv
ALSU_test.sv
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] âInside the ALSU testâ
# 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
```

Part2 :

ALSU_top

```
import uvm_pkg::*;
`include "uvm_macros.svh"
import ALSU_test_pkg::*;

module ALSU_top();
    bit clk;
    initial begin
        clk = 0 ;
        forever begin
            #1 clk = ~clk ;
        end
    end

    ALSU_if inst_if (clk) ;
    ALSU inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
                   inst_if.serial_in, inst_if.red_op_A,
                       inst_if.red_op_B, inst_if.opcode,
                           inst_if.bypass_A, inst_if.bypass_B,
                               inst_if.clk, inst_if.rst, inst_if.direction,
                                   inst_if.leds, inst_if.out);

    initial begin
        uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
        run_test ("ALSU_test") ;
    end
endmodule
```

ALSU_test

```
package ALSU_test_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_env_pkg::*;
    // import ALSU_config_pkg::*;
    // ALSU_config ALSU_cfg;

    class ALSU_test extends uvm_test;
        `uvm_component_utils(ALSU_test)
        ALSU_env env ;
        virtual ALSU_if alsu_test_vif ;

        function new(string name = "ALSU_test" , uvm_component parent = null);
            super.new(name , parent) ;
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase(phase) ;
            env = ALSU_env::type_id::create("env",this);

            if (!uvm_config_db #(virtual ALSU_if)::get(this,"","ALSU_if",alsu_test_vif))
                `uvm_fatal("build_phase" , "TEST - unable to get the virtual intrface");

            uvm_config_db #(virtual ALSU_if)::set(this,"*","TEST",alsu_test_vif);
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase (phase);
            phase.raise_objection (this) ;
            #100;
            `uvm_info ("run_phase" , "“Inside the ALSU test”" , UVM_MEDIUM)
            phase.drop_objection (this) ;
        endtask
    endclass
endpackage
```

ALSU_env

```
package ALSU_env_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_driver_pkg::*;

    class ALSU_env extends uvm_env;
        `uvm_component_utils(ALSU_env)
        ALSU_driver driver ;
        function new(string name = "ALSU_env" , uvm_component parent = null);
            super.new(name , parent);
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase (phase);
            driver = ALSU_driver::type_id::create("driver",this) ;
        endfunction
    endclass
endpackage
```

ALSU_if

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

ALSU_driver

```
package ALSU_driver_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"

    class ALSU_driver extends uvm_driver;
        `uvm_component_utils(ALSU_driver)
        virtual ALSU_if alsu_driver_vif ;
        function new(string name = "ALSU_driver" , uvm_component parent = null);
            super.new(name , parent);
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase(phase) ;
            uvm_config_db #(virtual ALSU_if)::get(this , "" , "TEST" , alsu_driver_vif) ;
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase(phase) ;
            alsu_driver_vif.rst = 1 ;
            forever begin
                @(negedge alsu_driver_vif.clk) ;
                alsu_driver_vif.red_op_A  = $random ;
                alsu_driver_vif.red_op_B  = $random ;
                alsu_driver_vif.bypass_A  = $random ;
                alsu_driver_vif.bypass_B  = $random ;
                alsu_driver_vif.direction = $random ;
                alsu_driver_vif.serial_in = $random ;
                alsu_driver_vif.cin       = $random ;
                alsu_driver_vif.opcode    = $random ;
                alsu_driver_vif.A         = $random ;
                alsu_driver_vif.B         = $random ;
            end
        endtask
    endclass
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 testá
# 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

```
import uvm_pkg::*;
`include "uvm_macros.svh"
import ALSU_test_pkg::*;

module ALSU_top();
    bit clk;
    initial begin
        clk = 0 ;
        forever begin
            #1 clk = ~clk ;
        end
    end

    ALSU_if inst_if (clk) ;
    ALSU inst_DUT (inst_if.A, inst_if.B, inst_if.cin,
                   inst_if.serial_in, inst_if.red_op_A,
                       inst_if.red_op_B, inst_if.opcode,
                           inst_if.bypass_A, inst_if.bypass_B,
                               inst_if.clk, inst_if.rst, inst_if.direction,
                                   inst_if.leds, inst_if.out);

    initial begin
        uvm_config_db #(virtual ALSU_if)::set(null,"uvm_test_top","ALSU_if",inst_if) ;
        run_test ("ALSU_test") ;
    end
endmodule
```

ALSU_test

```
package ALSU_test_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_env_pkg::*;
    import ALSU_config_pkg::*;

    class ALSU_test extends uvm_test;
        `uvm_component_utils(ALSU_test)
        ALSU_env env ;
        ALSU_config_obj alsu_config_obj_test;

        function new(string name = "ALSU_test" , uvm_component parent = null);
            super.new(name , parent) ;
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase(phase) ;
            env = ALSU_env::type_id::create("env",this);
            alsu_config_obj_test = ALSU_config_obj::type_id::create("alsu_config_obj_test");

            if (!uvm_config_db #(virtual ALSU_if)::get(this,"","ALSU_if",alsu_config_obj_test.alsu_config_vif))
                `uvm_fatal("build_phase" , "TEST - unable to get the virtual intrface");

            uvm_config_db #(ALSU_config_obj)::set(this,"*","TEST",alsu_config_obj_test);
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase (phase);
            phase.raise_objection (this) ;
            #100;
            `uvm_info ("run_phase" , "“Inside the ALSU test”" , UVM_MEDIUM)
            phase.drop_objection (this) ;
        endtask
    endclass
endpackage
```

ALSU_env

```
package ALSU_env_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_driver_pkg::*;

    class ALSU_env extends uvm_env;
        `uvm_component_utils(ALSU_env)
        ALSU_driver driver ;
        function new(string name = "ALSU_env" , uvm_component parent = null);
            super.new(name , parent);
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase (phase);
            driver = ALSU_driver::type_id::create("driver",this) ;
        endfunction
    endclass
endpackage
```

ALSU_driver

```
package ALSU_driver_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    import ALSU_config_pkg::*;

    class ALSU_driver extends uvm_driver;
        `uvm_component_utils(ALSU_driver)
        virtual ALSU_if alsu_driver_vif ;
        ALSU_config_obj alsu_config_obj_driver;

        function new(string name = "ALSU_driver" , uvm_component parent = null);
            super.new(name , parent);
        endfunction

        function void build_phase (uvm_phase phase);
            super.build_phase(phase) ;
            uvm_config_db #(ALSU_config_obj)::get(this , "" , "TEST" , alsu_config_obj_driver) ;
        endfunction

        function void connect_phase (uvm_phase phase);
            super.connect_phase(phase) ;
            alsu_driver_vif = alsu_config_obj_driver.alsu_config_vif ;
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase(phase) ;
            alsu_driver_vif.rst = 1 ;
            forever begin
                @(negedge alsu_driver_vif.clk) ;
                alsu_driver_vif.red_op_A = $random ;
                alsu_driver_vif.red_op_B = $random ;
                alsu_driver_vif.bypass_A = $random ;
                alsu_driver_vif.bypass_B = $random ;
                alsu_driver_vif.direction = $random ;
                alsu_driver_vif.serial_in = $random ;
                alsu_driver_vif.cin = $random ;
                alsu_driver_vif.opcode = $random ;
                alsu_driver_vif.A = $random ;
                alsu_driver_vif.B = $random ;
            end
        endtask
    endclass
endpackage
```

ALSU_configure

```
package ALSU_config_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"

    class ALSU_config_obj extends uvm_object;
        `uvm_object_utils(ALSU_config_obj)
        virtual ALSU_if alsu_config_vif ;
        function new(string name = "ALSU_config_obj");
            super.new(name) ;
        endfunction
    endclass
endpackage
```

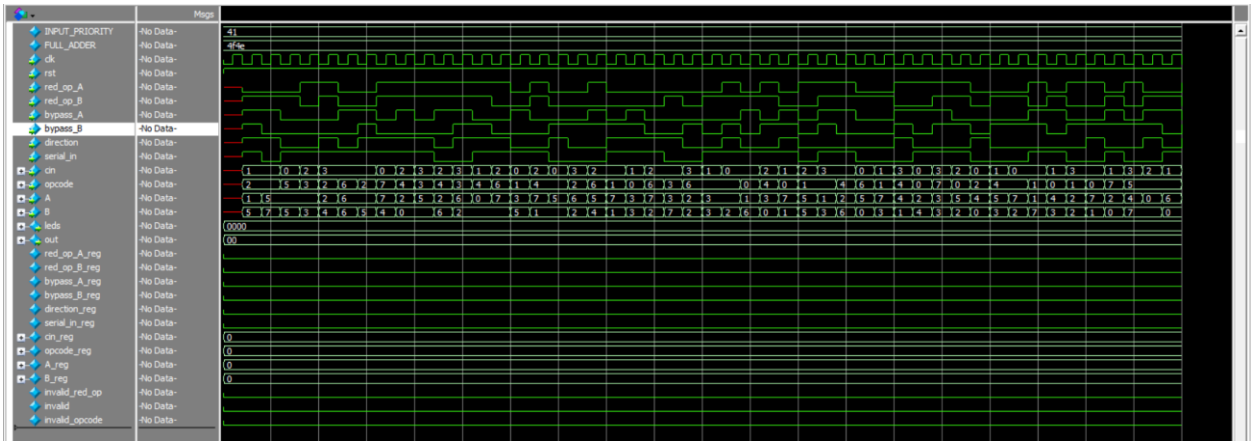
ALSU_if

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

Source file

```
ALSU_if.sv
ALSU.v
ALSU_config.sv
ALSU_driver.sv
ALSU_env.sv
ALSU_test.sv
ALSU_top.sv
```

Waveform



Transcript

```
# ----------------------------------------------------------------
# UVM-1.1d
# (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 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] åInside the ALSU testå
# 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
```

Do file

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