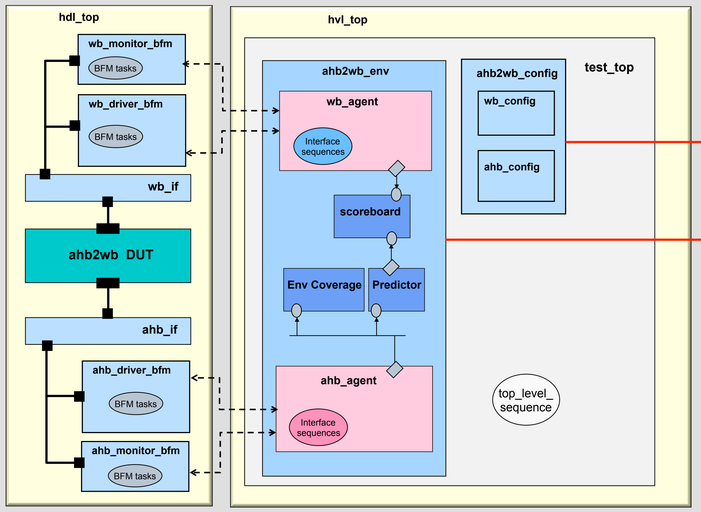
Slaying the UVM Reuse Dragon

Issues and Strategies for Achieving UVM Reuse

Mike Baird, Willamette HDL, mike@whdl.com

Bob Oden, Mentor Graphics, bob_oden@mentor.com

AHB to WB Block Level Test Bench



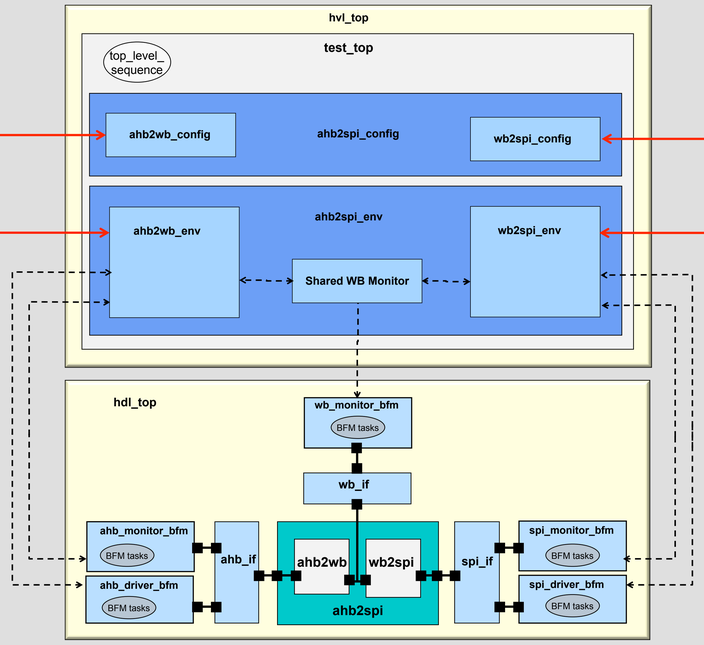

```
class test_top   // UVM test for block level simulations
virtual function void build_phase(uvm_phase phase);
   string interface_names[] =  {AHB_BFM, WB_BFM} ;
   super.build_phase(phase);
   configuration.initialize(BLOCK, "uvm_test_top.environment", interface_names );
endfunction
```

1- test class initializes top level configuration

```
class ahb2wb_configuration
function void initialize(uvmf_sim_level_t sim_level,
                         string environment_path,
                         string interface_names [],
                         uvm_reg_block register_model = null,
                         uvmf_active_passive_t interface_activity[] = null);

  if ( sim_level == BLOCK ) begin
    ahb_config.initialize( ACTIVE, {environment_path,".a_agent"}, interface_names[0]);
    wb_config.initialize ( ACTIVE, {environment_path,".b_agent"}, interface_names[1]);
  end else if ( sim_level == CHIP ) begin
    ahb_config.initialize( ACTIVE, {environment_path,".a_agent"}, interface_names[0]);
    wb_config.initialize ( PASSIVE,{environment_path,".b_agent"}, interface_names[1]);
  end else if ( sim_level == SYSTEM ) begin
    ahb_config.initialize( PASSIVE, {environment_path,".a_agent"}, interface_names[0]);
    wb_config.initialize ( PASSIVE, {environment_path,".b_agent"}, interface_names[1]);
  end else begin
    `uvm_fatal("CONFIGURATION", ". . .
  end
endfunction
```

AHB to SPI ChipLevel Test Bench




```
class test_top    // UVM test for chip level simulations
virtual function void build_phase(uvm_phase phase);
   string interface_names[] =  { AHB_BFM, WB_BFM, SPI_BFM } ;
   super.build_phase(phase);
   configuration.initialize(CHIP, "uvm_test_top.environment",interface_names);
endfunction
```

1 – test classs initializes top level configuration

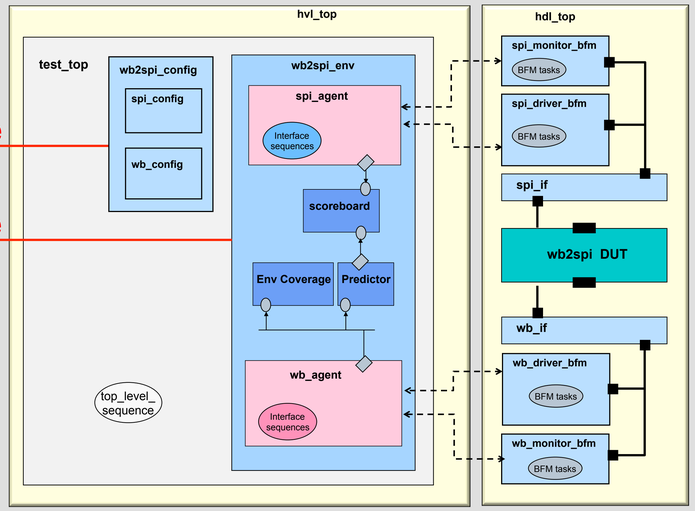
```
class ahb2spi_configuration
function void initialize(uvmf_sim_level_t sim_level,
                         string environment_path,
                         string interface_names [],
                         uvm_reg_block register_model = null,
                         uvmf_active_passive_t interface_activity[] = null);

  if (register_model == null) begin
    reg_model = ahb2spi_reg_block::type_id::create("reg_model");
    reg_model.build();
    enable_reg_adaptation = 1;
    wb2spi_config.enable_reg_prediction = 1;
  end

  ahb2wb_config.initialize(sim_level, {environment_path, ".ahb2wb_env"},
                           { interface_names[0], interface_names[1]} );
  wb2spi_config.initialize(sim_level, {environment_path, ".wb2spi_env"},
                           { interface_names[1], interface_names[2]},
                           reg_model.wb2spi);
endfunction
```

2 – chip level configuration Initializes block level configuration

WB to SPI Block Level Test Bench




```
class test_top   // UVM test for block level simulations
virtual function void build_phase(uvm_phase phase);
  string interface_names [] ={ WB_BFM, SPI_BFM } ;
  super.build_phase(phase);
  configuration.initialize(BLOCK, "uvm_test_top.environment", interface_names);
endfunction
```

1 - test class initializes top level configuration

```
class wb2spi_configuration
function void initialize(uvmf_sim_level_t sim_level,
                         string environment_path,
                         string interface_names[],
                         uvm_reg_block register_model = null,
                         uvmf_active_passive_t interface_activity[] = null
                         );
  if ( sim_level == BLOCK ) begin
    wb_config.initialize( ACTIVE, {environment_path,".wb_agent"}, interface_names[0]);
    spi_config.initialize( ACTIVE, {environment_path,".spi_agent"}, interface_names[1]);
  end else if ( sim_level == CHIP ) begin
    wb_config.initialize( PASSIVE, {environment_path,".wb_agent"}, interface_names[0]);
    spi_config.initialize( ACTIVE, {environment_path,".spi_agent"}, interface_names[1]);
  end else if ( sim_level == SYSTEM ) begin
    wb_config.initialize( PASSIVE, {environment_path,".wb_agent"}, interface_names[0]);
    spi_config.initialize( PASSIVE, {environment_path,".spi_agent"}, interface_names[1]);
  end else begin
    `uvm_fatal("CONFIGURATION", . . .
  end

  if (register_model == null) begin
    reg_model = wb2spi_reg_block::type_id::create("reg_model");
    reg_model.build();
    enable_reg_adaptation = 1;
    enable_reg_prediction = 1;
  end else begin
    $cast(reg_model,register_model);
    enable_reg_prediction = 1;
  end
endfunction
```

3 – chip level configuration Initializes block level configuration