

bscan tile flow

1. tap tile

注意项：在顶层抽取BSDL时，IO 模块不能是black box□

1.1 script

```
#!/bin/sh
#\
exec tesseract -shell -log logfiles/$0.log -replace -dofile "$0" -arguments
${1+"$@"}

# Set the context to insert DFT into top-level design
set_context dft -rtl -design_id rtl2

# Set the location of the TSDB. Default is the current working directory.
set_tsdb_output_directory ../tsdb_outdir

# Open the TSDB of all the child cores

# Read the tesseract cell library
read_cell_library ../../library/standard_cells/tesseract/adk.tcelllib
read_cell_library ../../library/dummy_pad/dummy_pad.tcelllib

# Read hard_macros
read_verilog ../../library/plls/pll.v -blackbox -
exclude_from_file_dictionary

# Read the design view of chip_top
set_design_sources -format verilog -y ../../library/pad_cells -extension v
set_design_sources -format verilog -y ../../library/dummy_pad -extension v
read_verilog ../rtl/tap_t.v
read_verilog ../rtl/tap_t_rtl1_tesseract_tap_main.v
read_icl      ../rtl/tap_t_rtl1_tesseract_tap_main.icl
set_current_design tap_t

#set_design_level physical_block
set_design_level sub_block

# Define set_dft_specification_requirements to insert boundary scan at chip-
level
# Set memory_test on even if there is not memories at the top level so that
the memory
# clock DRCs are run outside the memory inserted blocks.
#set_dft_specification_requirements -boundary_scan on -memory_test on
set_dft_specification_requirements -logic_test on
```

```
# Toggle the enable to relock the PLL

# Specify the TAP pins using set_attribute_value
set_attribute_value tck -name function -value tck
set_attribute_value tdi -name function -value tdi
set_attribute_value tms -name function -value tms
set_attribute_value trst -name function -value trst
set_attribute_value tdo -name function -value tdo

# Specify all clocks so that the proper BSCAN cells gets inserted
automatically for them

check_design_rules

# Create and report a DFT Specification
# set spec [create_dft_specification -existing_ijtag_host_scan_in
tap/host_1_from_so -sri_sib_list occ -tile_ijtag_host_list {r1}]
set spec [create_dft_specification -existing_bscan_host_scan_in
tap/host_bscan_from_so -existing_ijtag_host_scan_in tap/host_1_from_so -
sri_sib_list occ -tile_ijtag_host_list {r1}]

report_config_data $spec

read_config_data -in_wrapper $spec/IjtagNetwork -from_string {
  HostScanInterface(bscan) {
    Interface {
      design_instance : tap;
      scan_interface : host_bscan;
    }
  }
}

read_config_data -in_wrapper $spec -from_string {
  EmbeddedBoundaryScan {
    ImplementationOptions {
      clocking : gated_tck;
      update_stage : flop;
      scan_path_retiming : flop;
    }
    HostBscanInterface(tap) {
      Interface {
        ijtag_host_interface : HostScanInterface(bscan);
      }
      EBScanPipeline(p1) {
      }
      SecondaryEBScanInterface(to) {
      }
      EBScanPipeline(p0) {
      }
    }
  }
}
```

```

    }
}
report_config_data $spec
    # ac_init_clock0_present: on;
    # ac_init_clock1_present: on;
    # ac_signal_present: on;
    # ac_mode_en_present: on;

# Segment the boundary scan to be used during logic test
# set_config_value $spec/BoundaryScan/max_segment_length_for_logictest 80

# Add auxiliary mux on the inputs and outputs used for SSN bus and bus_clock
# bus_in {GPIO3_0 GPIO3_1} bus_out {GPIO4_0 GPIO4_1} bus_clock {GPIO3_2}
# read_config_data -in ${spec}/BoundaryScan -from_string {
#   AuxiliaryInputOutputPorts {
#     auxiliary_input_ports   : GPIO3_0, GPIO3_1, GPIO3_2;
#     auxiliary_output_ports  : GPIO4_0, GPIO4_1 ;
#   }
# }

# report_config_data $spec

# Generate and insert the hardware
process_dft_specification

# Extract IJAG network and create ICL file for the design
extract_icl -create_ijtag_graybox on

# Create patterns(testbenches) to verify the inserted DFT logic
set spec [create_patterns_specification]
process_pattern_specification

# Point to the libraries and run simultion
set_simulation_library_sources -v ../../library/standard_cells/verilog/*.v \
    -y ../../library/plls \
    -y ../../library/memories \
    -extension v

run_testbench_simulations \
    -keep_simulation_data on \
    -simulator vcs \
    -compilation_options "-sverilog +define+debussy -kdb" \
    -simulator_options "-sverilog -debug_access+all -kdb" \
    -use_design_view_per_simulation on

exit

```

1.2 spec

```
DftSpecification(tap_t,rtl2) {
```

```
IjtagNetwork {
  HostScanInterface(ijtag) {
    Interface {
      design_instance : tap;
      scan_interface : host_ijtag_1;
    }
    Sib(sri) {
      Attributes {
        tessent_dft_function : scan_resource_instrument_host;
      }
      Sib(occ) {
      }
    }
    Sib(thc) {
      Attributes {
        tessent_dft_function : tile_host_collector;
      }
      Sib(th_r1) {
        to_scan_in_feedthrough : pipeline;
        SecondaryHostScanInterface(r1) {
        }
      }
    }
  }
  HostScanInterface(bscan) {
    Interface {
      design_instance : tap;
      scan_interface : host_bscan;
    }
  }
}
EmbeddedBoundaryScan {
  ImplementationOptions {
    clocking : gated_tck;
    update_stage : flop;
    scan_path_retiming : flop;
  }
  HostBScanInterface(tap) {
    Interface {
      ijtag_host_interface : HostScanInterface(bscan);
    }
    EBScanPipeline(p1) {
    }
    SecondaryEBScanInterface(to) {
    }
    EBScanPipeline(p0) {
    }
  }
}
}
```

1.3 icl

*注意CLAMP, HIGHZ命令时是mux到bypass register
*且必须要描述这两个指令，否则process pattern spec时会报错。

```
Module tap_t_rttl_tessent_tap_main {
  TCKPort      tck;
  ScanInPort   tdi;
  ScanOutPort  tdo { Source IRMux;
    Attribute forced_high_output_port_list = "tdo_en";
    Attribute forced_low_dft_signal_list = "tms_disable";
  }
  DataOutPort  tdo_en {
    Attribute associated_scan_port_list = "tdo";
    Attribute connection_rule_option = "allowed_no_destination";
    Attribute function_modifier = "tdo_enable_active_high";
  }
  TMSPort      tms {
    Attribute forced_low_dft_signal_list = "tms_disable";
  }
  TRSTPort     trst {
    Attribute connection_rule_option = "allowed_tied_high";
  }
  ToCaptureEnPort capture_dr_en;
  ToShiftEnPort  shift_dr_en;
  ToUpdateEnPort update_dr_en;
  ToResetPort    test_logic_reset { ActivePolarity 0; }
  ToSelectPort   host_1_to_sel { Source host_1_to_sel_int;
    Attribute connection_rule_option = "allowed_no_destination";
  }
  LogicSignal    host_1_to_sel_int { instruction == HOSTIJTAG_1; }
  ScanInPort     host_1_from_so {
    Attribute connection_rule_option = "allowed_no_source";
  }
  ScanInPort     host_bscan_from_so {
    Attribute connection_rule_option = "allowed_no_source";
    Attribute tessent_bscan_pipeline_stages = "0";
  }
  ToSelectPort   host_bscan_to_sel { Source bscan_select_int;
    Attribute connection_rule_option = "allowed_no_destination";
    Attribute tessent_bscan_function = "select";
  }
  LogicSignal    bscan_select_int {
    (instruction == EXTEST) ||
    (instruction == INTEST) ||
    (instruction == EXTEST_PULSE) ||
    (instruction == EXTEST_TRAIN) ||
    (instruction == SAMPLE) ||
    (instruction == PRELOAD) ;
  }
}
```

```
DataOutPort    force_disable { Source force_disable_int;
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute tesseract_bscan_function = "force_disable";
}
LogicSignal force_disable_int { instruction == HIGHZ; }
DataOutPort    select_jtag_input { Source select_jtag_input_int;
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute tesseract_bscan_function = "select_jtag_input";
}
LogicSignal select_jtag_input_int { instruction == INTEST; }
DataOutPort    select_jtag_output { Source select_jtag_output_int;
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute tesseract_bscan_function = "select_jtag_output";
}
LogicSignal select_jtag_output_int {
  (instruction == EXTEST) ||
  (instruction == EXTEST_PULSE) ||
  (instruction == EXTEST_TRAIN) ||
  (instruction == CLAMP) ||
  (instruction == HIGHZ) ;
}
DataOutPort    extest_pulse { Source ext_test_pulse_int;
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute tesseract_bscan_function = "extest_pulse";
}
LogicSignal ext_test_pulse_int { instruction == EXTEST_PULSE; }
DataOutPort    extest_train { Source ext_test_train_int;
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute tesseract_bscan_function = "extest_train";
}
LogicSignal ext_test_train_int { instruction == EXTEST_TRAIN; }
DataOutPort fsm_state[3:0]{
  Attribute connection_rule_option = "allowed_no_destination";
  Attribute function_modifier = "tap_fsm_state";
  RefEnum state_encoding;
}

Enum state_encoding {
  test_logic_reset = 4'b1111;
  run_test_idle    = 4'b1100;
  select_dr        = 4'b0111;
  capture_dr       = 4'b0110;
  shift_dr         = 4'b0010;
  exit1_dr         = 4'b0001;
  pause_dr         = 4'b0011;
  exit2_dr         = 4'b0000;
  update_dr        = 4'b0101;
  select_ir        = 4'b0100;
  capture_ir       = 4'b1110;
  shift_ir         = 4'b1010;
  exit1_ir         = 4'b1001;
```

```
    pause_ir          = 4'b1011;
    exit2_ir           = 4'b1000;
    update_ir          = 4'b1101;
}

ScanInterface tap_client {
    Port tdi;
    Port tdo;
    Port tms;
}

ScanInterface host_ijtag_1 {
    Port host_1_from_so;
    Port host_1_to_sel;
}

ScanInterface host_bscan {
    Port host_bscan_to_sel;
    Port host_bscan_from_so;
    Port capture_dr_en;
    Port shift_dr_en;
    Port update_dr_en;
    Port test_logic_reset;
    Attribute          tesseract_is_bscan_host = "true";
}

Instance fsm_0f_tap_t_rtl1_tesseract_tap_main_fsm {
    InputPort tck = tck;
    InputPort tms = tms;
    InputPort trst = trst;
}

ScanRegister instruction[3:0] {
    CaptureSource 4'b0001;
    ResetValue    4'b1000;
    ScanInSource  tdi;
    RefEnum       instruction_opcodes;
}

Enum instruction_opcodes {
    BYPASS          = 4'b1111;
    CLAMP           = 4'b0000;
    EXTEST          = 4'b0001;
    EXTEST_PULSE   = 4'b0010;
    EXTEST_TRAIN    = 4'b0011;
    INTEST          = 4'b0100;
    SAMPLE          = 4'b0101;
    PRELOAD         = 4'b0101;
    HIGHZ           = 4'b0110;
    HOSTIJTAG_1     = 4'b0111;
    IDCODE          = 4'b1000;
}

ScanRegister bypass {
    CaptureSource 1'b0;
}
```

```

    ScanInSource    tdi;
}
ScanRegister device_id[31:0] {
    CaptureSource  4'b0000,16'b000000000000000000,11'b000000000000,1'b1;
    ScanInSource    tdi;
}
ScanMux IRMux SelectedBy fsm.irSel {
    1'b0 : DRMux;
    1'b1 : instruction[0];
}
ScanMux DRMux SelectedBy instruction {
    4'b1111          : bypass;
    4'b0000          : bypass;
    4'b0001          : host_bscan_from_so;
    4'b0010          : host_bscan_from_so;
    4'b0011          : host_bscan_from_so;
    4'b0100          : host_bscan_from_so;
    4'b0101          : host_bscan_from_so;
    4'b0110          : bypass;
    4'b0111          : host_1_from_so;
    4'b1000          : device_id[0];
    'bX              : bypass;
}
Attribute          keep_active_during_scan_test = "true";
Attribute          abc = 1;
Attribute          tesseract_instruction_reg    = "instruction";
Attribute          tesseract_bypass_reg        = "bypass";
Attribute          tesseract_device_id_reg      = "device_id";
Attribute          tesseract_instrument_container =
"tap_t_rtl1_ijtag";
Attribute          tesseract_use_in_dft_specification = "false";
Attribute          tesseract_instrument_type      =
"mentor::ijtag_node";
Attribute          tesseract_instrument_subtype   = "tap_controller";
Attribute          tesseract_signature           =
"d7c6347c88a60873d82843c778bd6818";
}
Module tap_t_rtl1_tesseract_tap_main_fsm {
    TCKPort          tck;
    TMSPort          tms;
    TRSTPort         trst;
    ToIRSelectPort   irSel;
    ToResetPort      tlr;
}

```

2. io tile

2.1 script

```
#!/bin/sh
#\
exec tesseract -shell -log logfiles/$0.log -replace -dofile "$0" -arguments
${1+"$@"}

# Set the context to insert DFT into top-level design
set_context dft -rtl -design_id rtl1

# Set the location of the TSDB. Default is the current working directory.
set_tsdb_output_directory ../tsdb_outdir

# Open the TSDB of all the child cores

# Read the tesseract cell library
read_cell_library ../../library/standard_cells/tesseract/adk.tcelllib
read_cell_library ../../library/dummy_pad/dummy_pad.tcelllib

# Read hard_macros
read_verilog ../../library/plls/pll.v -blackbox -
exclude_from_file_dictionary

# Read the design view of chip_top
set_design_sources -format verilog -y ../../library/pad_cells -extension v
set_design_sources -format verilog -y ../../library/dummy_pad -extension v
read_verilog ../rtl/io_a_t.v
set_current_design io_a_t

set_design_level physical_block
#set_design_level sub_block

# Define set_dft_specification_requirements to insert boundary scan at chip-
level
# Set memory_test on even if there is not memories at the top level so that
the memory
# clock DRCs are run outside the memory inserted blocks.
#set_dft_specification_requirements -boundary_scan on -memory_test on
set_dft_specification_requirements -boundary_scan on

set_boundary_scan_port_options -pad_io_ports {BP_A0 BP_A1 BP_A2 BP_A3}

# Toggle the enable to relock the PLL

# Specify the TAP pins using set_attribute_value
# set_attribute_value tck -name function -value tck
# set_attribute_value tdi -name function -value tdi
# set_attribute_value tms -name function -value tms
# set_attribute_value trst -name function -value trst
# set_attribute_value tdo -name function -value tdo
```

```
# Specify all clocks so that the proper BSCAN cells gets inserted
automatically for them

check_design_rules

# Create and report a DFT Specification
# set spec [create_dft_specification -existing_ijtag_host_scan_in
tap/host_1_from_so -sri_sib_list occ -tile_ijtag_host_list {r1}]
set spec [create_dft_specification]

report_config_data $spec

read_config_data -in $spec/EmbeddedBoundaryScan -from_string {
  ImplementationOptions {
    clocking          : gated_tck;
    update_stage      : flop;
    scan_path_retiming : flop;
  }

  InternalBScanCells {
    InternalBScanCell(pipe0) {
      insert_before_port : BP_A0 ;
      safe_value          : 0;
      type                 : observation;
    }
    InternalBScanCell(pipe1) {
      insert_after_port  : BP_A3 ;
      safe_value          : 0;
      type                 : observation;
    }
  }
}

report_config_data $spec

# Segment the boundary scan to be used during logic test
# set_config_value $spec/BoundaryScan/max_segment_length_for_logictest 80

# Add auxiliary mux on the inputs and outputs used for SSN bus and bus_clock
# bus_in {GPIO3_0 GPIO3_1} bus_out {GPIO4_0 GPIO4_1} bus_clock {GPIO3_2}
# read_config_data -in ${spec}/BoundaryScan -from_string {
#   AuxiliaryInputOutputPorts {
#     auxiliary_input_ports  : GPIO3_0, GPIO3_1, GPIO3_2;
#     auxiliary_output_ports : GPIO4_0, GPIO4_1 ;
#   }
# }

# report_config_data $spec

# Generate and insert the hardware
```

```

process_dft_specification

# Extract IJAG network and create ICL file for the design
extract_icl -create_ijtag_graybox on

# Create patterns(testbenches) to verify the inserted DFT logic
set spec [create_patterns_specification]
process_pattern_specification

# Point to the libraries and run simultion
set_simulation_library_sources -v ../../library/standard_cells/verilog/*.v \
                               -y ../../library/plls \
                               -y ../../library/memories \
                               -extension v

run_testbench_simulations \
  -keep_simulation_data on \
  -simulator vcs \
  -compilation_options "-sverilog +define+debussy -kdb" \
  -simulator_options "-sverilog -debug_access+all -kdb"

exit

```

2.2 spec

```

DftSpecification(io_a_t,rtl1) {
  EmbeddedBoundaryScan {
    pad_io_ports : BP_A0, BP_A1, BP_A2, BP_A3;
    ImplementationOptions {
      clocking : gated_tck;
      update_stage : flop;
      scan_path_retiming : flop;
    }
  }
  InternalBScanCells {
    InternalBScanCell(pipe0) {
      insert_before_port : BP_A0;
      safe_value : 0;
      type : observation;
    }
    InternalBScanCell(pipe1) {
      insert_after_port : BP_A3;
      safe_value : 0;
      type : observation;
    }
  }
}

```

3. chip

详细说明参考tessent文档BSDL Extraction部分

- BSDL不能抽取在CHIP LEVEL创建的Boundary Scan cells[]工具会报错
- CHIP LEVEL代码需要把all block的BSCAN给连好
- block需要提供tcd_bscan文件
- BondingConfig只能存在于block level, 顶层不支持再划分bypass[]

3.1 rtl

```
module chip_top(  
    // Port Declarations  
    input  wire    BP_A00,  
    input  wire    BP_A01,  
    input  wire    BP_A02,  
    input  wire    BP_A03,  
  
    input  wire    BP_A10,  
    input  wire    BP_A11,  
    input  wire    BP_A12,  
    input  wire    BP_A13,  
  
    input  wire    TDI,  
    input  wire    TRST,  
    input  wire    TMS,  
    input  wire    TCK,  
    output wire    TDO  
);  
  
tap_t tap_t(  
    .tdi    (TDI_core),  
    .tms    (TMS_core),  
    .tck    (TCK_core),  
    .trst   (TRST_core),  
    .tdo    (TDO_core),  
    .tdo_en (TDO_core_en),  
  
    //input  
    .to_bscan_from_scan_out    (to_bscan_from_scan_out  
), // input  
  
    //output  
    .to_bscan_to_select        (to_bscan_to_select  
), // output  
    .to_bscan_to_clock        (to_bscan_to_clock
```

```
), // output
  .to_bscan_to_capture_en      (to_bscan_to_capture_en
), // output
  .to_bscan_to_shift_en       (to_bscan_to_shift_en
), // output
  .to_bscan_to_update_en      (to_bscan_to_update_en
), // output
  .to_bscan_to_scan_in        (to_bscan_to_scan_in
), // output
  .to_bscan_to_force_disable  (to_bscan_to_force_disable
), // output
  .to_bscan_to_select_jtag_input (to_bscan_to_select_jtag_input
), // output
  .to_bscan_to_select_jtag_output (to_bscan_to_select_jtag_output
) // output
  // .to_bscan_to_ac_init_clk0   (to_bscan_to_ac_init_clk0
), // output
  // .to_bscan_to_ac_init_clk1   (to_bscan_to_ac_init_clk1
), // output
  // .to_bscan_to_ac_signal       (to_bscan_to_ac_signal
), // output
  // .to_bscan_to_ac_mode_en      (to_bscan_to_ac_mode_en
) // output
);
```

```
io_a_t io_a_t0(
  .BP_A0      (BP_A00
), // input
  .BP_A1      (BP_A01
), // input
  .BP_A2      (BP_A02
), // input
  .BP_A3      (BP_A03
), // input
  .bscan_select      (to_bscan_to_select
), // input
  .bscan_force_disable (to_bscan_to_force_disable
), // input
  .bscan_select_jtag_input (to_bscan_to_select_jtag_input
), // input
  .bscan_select_jtag_output (to_bscan_to_select_jtag_output
), // input
  .bscan_clock      (to_bscan_to_clock
), // input
  .bscan_capture_en (to_bscan_to_capture_en
), // input
  .bscan_shift_en   (to_bscan_to_shift_en
), // input
  .bscan_update_en  (to_bscan_to_update_en
), // input
```

```
    .bscan_scan_in          (to_bscan_to_scan_in
), // input

    .bscan_scan_out        (bscan_scan_out_t0 ) // output
);
io_a_t io_a_t1(
    .BP_A0                  (BP_A10
), // input
    .BP_A1                  (BP_A11
), // input
    .BP_A2                  (BP_A12
), // input
    .BP_A3                  (BP_A13
), // input
    .bscan_select          (to_bscan_to_select
), // input
    .bscan_force_disable   (to_bscan_to_force_disable
), // input
    .bscan_select_jtag_input (to_bscan_to_select_jtag_input
), // input
    .bscan_select_jtag_output (to_bscan_to_select_jtag_output
), // input
    .bscan_clock           (to_bscan_to_clock
), // input
    .bscan_capture_en      (to_bscan_to_capture_en
), // input
    .bscan_shift_en        (to_bscan_to_shift_en
), // input
    .bscan_update_en       (to_bscan_to_update_en
), // input
    .bscan_scan_in         (bscan_scan_out_t0
), // input

    .bscan_scan_out        (to_bscan_from_scan_out ) // output
);

ipad TDI_inst (
    .PAD (TDI),
    .C(TDI_core)
);

ipad TCK_inst (
    .PAD (TCK),
    .C(TCK_core)
);

ipad TMS_inst (
    .PAD (TMS),
    .C(TMS_core)
);
```

```
ipad TRST_inst (  
  .PAD (TRST),  
  .C(TRST_core)  
);  
  
opad TDO_inst (  
  .PAD (TDO),  
  .I(TDO_core),  
  .OEN(TDO_core_en)  
);  
  
endmodule
```

3.2 script

```
#!/bin/sh  
#\nexec tesseract -shell -log logfiles/$0.log -replace -dofile "$0" -arguments  
${1+"$@"}  
  
# Set the context to insert DFT into top-level design  
set_context dft -rtl -design_id rtl1  
  
# Set the location of the TSDB. Default is the current working directory.  
set_tsdb_output_directory ../tsdb_outdir  
  
# Open the TSDB of all the child cores  
open_tsdb ../../tap_t/tsdb_outdir  
open_tsdb ../../io_a_t/tsdb_outdir  
  
# Read the tesseract cell library  
read_cell_library ../../library/standard_cells/tesseract/adk.tcelllib  
read_cell_library ../../library/dummy_pad/dummy_pad.tcelllib  
  
# Read hard_macros  
read_verilog ../../library/plls/pll.v -blackbox -  
exclude_from_file_dictionary  
  
# Read the design view of chip_top  
set_design_sources -format verilog -y ../../library/pad_cells -extension v  
set_design_sources -format verilog -y ../../library/dummy_pad -extension v  
read_verilog ../rtl/chip_top.v  
  
#read_design tap_t -design_id rtl1 -view interface -verbose  
read_design tap_t -design_id rtl2 -verbose  
read_design io_a_t -design_id rtl1 -verbose  
set_current_design chip_top
```

```
set_design_level chip
```

```
# Define set_dft_specification_requirements to insert boundary scan at chip-level
```

```
# Set memory_test on even if there is not memories at the top level so that the memory
```

```
# clock DRCs are run outside the memory inserted blocks.
```

```
#set_dft_specification_requirements -boundary_scan on -memory_test on
```

```
#set_dft_specification_requirements -bsdl_extraction on
```

```
# Toggle the enable to relock the PLL
```

```
# Specify the TAP pins using set_attribute_value
```

```
set_attribute_value TCK -name function -value tck
```

```
set_attribute_value TDI -name function -value tdi
```

```
set_attribute_value TMS -name function -value tms
```

```
set_attribute_value TRST -name function -value trst
```

```
set_attribute_value TDO -name function -value tdo
```

```
# Specify all clocks so that the proper BSCAN cells gets inserted automatically for them
```

```
# add_clocks PLL_1/pll_clock_0 -reference REF_CLK -freq_multiplier 16
```

```
# add_clock REF_CLK -period 48ns
```

```
# add_clock INCLK -period 10ns
```

```
check_design_rules
```

```
set_system_mode setup
```

```
set_dft_specification_requirements -bsdl_extraction on
```

```
check_design_rules
```

```
extract_icl -create_ijtag_graybox on
```

```
set_context patterns -ijtag -rtl
```

```
set spec [create_patterns_specification -bscan_sim_views ijtag_graybox]
```

```
report_config_data $spec
```

```
process_patterns_specification
```

```
set_simulation_library_sources \  
-Y ../../library/pad_cells/ \  
-Y ../../library/dummy_pad/ \  
-extensions {v} \  
-V ../../library/standard_cells/verilog/adk.v
```

```
run_testbench_simulations \  
-keep_simulation_data on \  
-simulator vcs \  
-compilation_options "-sverilog +define+debussy -kdb" \  
-simulator_options "-sverilog -debug_access+all -kdb"
```

4. tcd_bscan

集成第三方IP(比如serdes phy) BSCAN

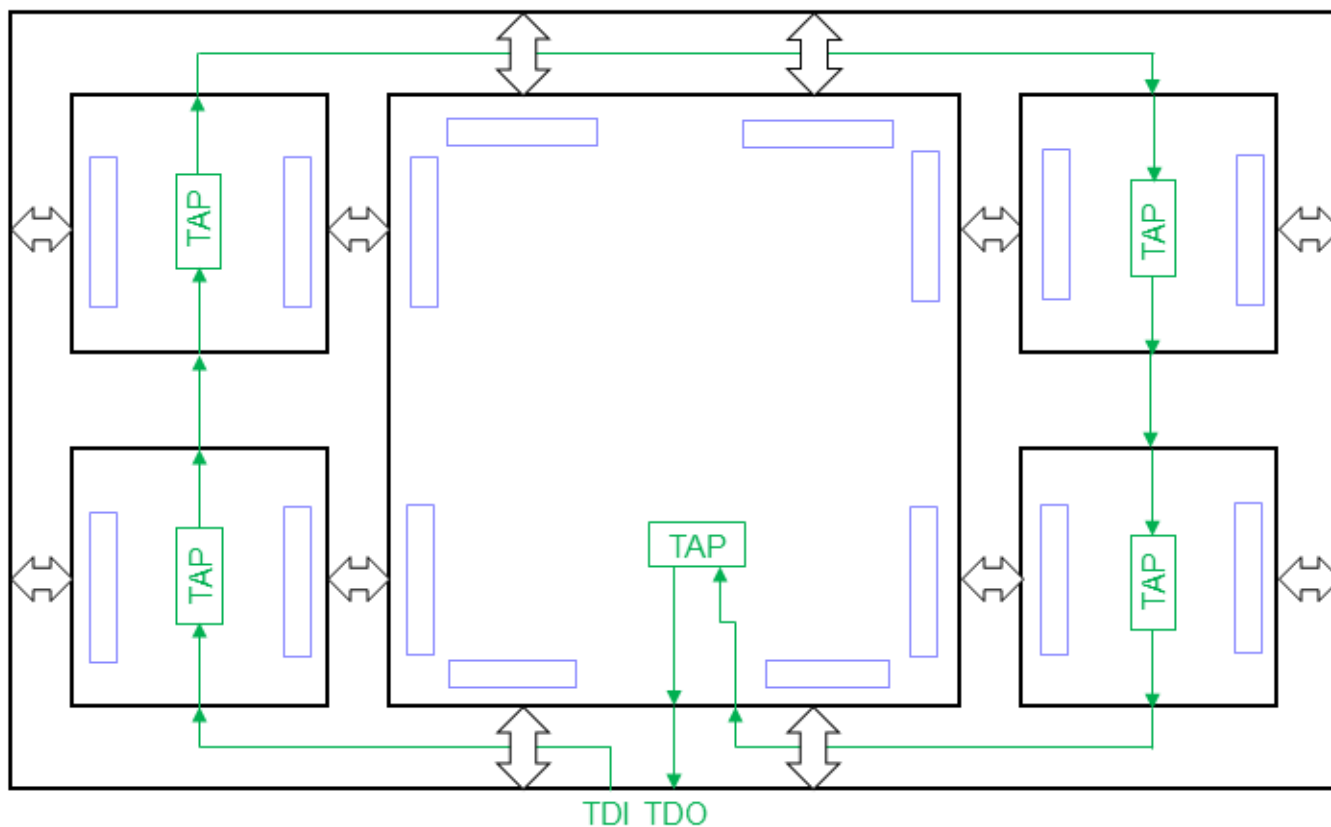
```
# 读第三方.v文件  
read_verilog xxx.v  
  
# 读第三方tcd_bscan描述  
read_core_descriptions xxx.tcd_bscan
```

工具就能自动识别了。

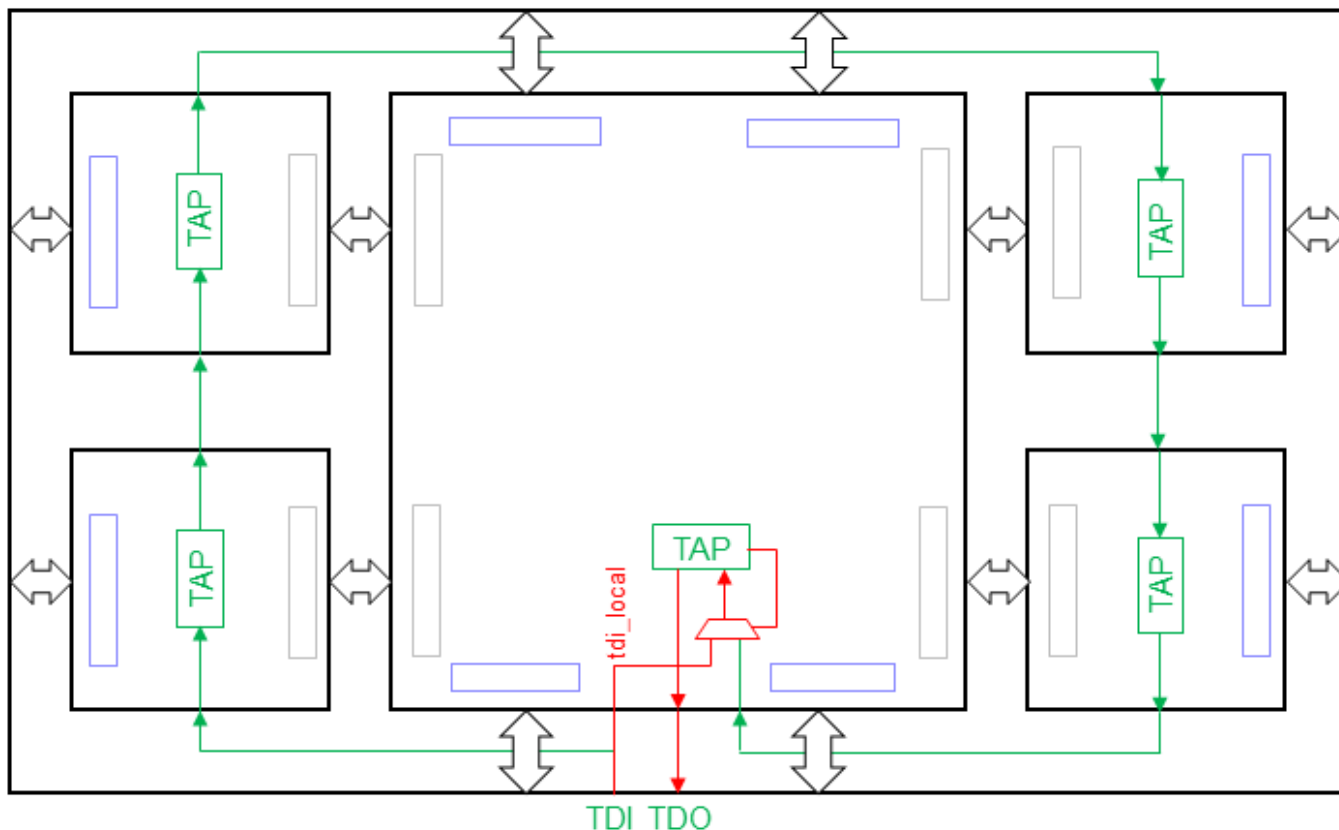
注：有的第三方只提供了IP的BSDL文件，需要根据BSDL文件，手动的写出对应的tcd_bscan文件给工具。

5. 多DIE互联

传统菊花链连接方式



DEVICE_ID BYPASS



6. bsdI抽取

详细说明参考essent文档BSDI Extraction部分

- BSDI不能抽取在CHIP LEVEL创建的Boundary Scan cells[]工具会报错
- CHIP LEVEL代码需要把all block的BSCAN给连好
- block需要提供tcd_bscan文件
- BondingConfig只能存在于block level, 顶层不支持再划分bypass[]

顶层不能单独存在PORT[]TAG PORT除外), 存在的话不支持BSDI抽取flow, 这种情况下可以直接使用正向的BondingConfig来生成对应的BSDI文件。

6.1 BSDI文件举例

```
-- BSDI Version 2001

entity chip_top is
  generic (PHYSICAL_PIN_MAP : string := "DEFAULT_PACKAGE_NAME");

  port (
    -- Port List
    BP_A00      : in  bit;
    BP_A01      : in  bit;
    BP_A02      : in  bit;
    BP_A03      : in  bit;
  );
end entity;
```

```
BP_A10      : in  bit;
BP_A11      : in  bit;
BP_A12      : in  bit;
BP_A13      : in  bit;
TDI         : in  bit;
TRST        : in  bit;
TMS         : in  bit;
TCK         : in  bit;
TDO         : out bit);
```

```
use STD_1149_1_2001.all;
```

```
attribute COMPONENT_CONFORMANCE of chip_top: entity is
"STD_1149_1_2001";
```

```
--Pin mappings
```

```
attribute PIN_MAP of chip_top: entity is PHYSICAL_PIN_MAP;
```

```
constant DEFAULT_PACKAGE_NAME: PIN_MAP_STRING :=
```

```
"BP_A00      : I1      , " &
"BP_A01      : I2      , " &
"BP_A02      : I3      , " &
"BP_A03      : I4      , " &
"BP_A10      : I5      , " &
"BP_A11      : I6      , " &
"BP_A12      : I7      , " &
"BP_A13      : I8      , " &
"TDI         : I9      , " &
"TRST        : I10     , " &
"TMS         : I11     , " &
"TCK         : I12     , " &
"TD0         : 01      " ;
```

```
attribute TAP_SCAN_RESET of TRST      : signal is true;
attribute TAP_SCAN_IN   of TDI        : signal is true;
attribute TAP_SCAN_MODE of TMS        : signal is true;
attribute TAP_SCAN_OUT  of TDO        : signal is true;
attribute TAP_SCAN_CLOCK of TCK       : signal is (1.00e+07, BOTH);
```

```
attribute INSTRUCTION_LENGTH of chip_top: entity is 4;
```

```
attribute INSTRUCTION_OPCODE of chip_top: entity is
```

```
"IDCODE      (1000)," &
"BYPASS      (1111)," &
"EXTEST      (0001)," &
"EXTEST_PULSE (0010)," &
"EXTEST_TRAIN (0011)," &
"SAMPLE      (0101)," &
"PRELOAD     (0101)," &
"HIGHZ       (0110)," &
```

```

"CLAMP      (0000) " ;

attribute INSTRUCTION_CAPTURE of chip_top: entity is "0001";

attribute IDCODE_REGISTER of chip_top: entity is
  "0000"          & -- version
  "0000000000000000" & -- part number
  "000000000000"   & -- manufacturer's identity
  "1";             -- required by 1149.1

attribute REGISTER_ACCESS of chip_top: entity is
  "DEVICE_ID ( IDCODE )," &
  "BOUNDARY  ( EXTEST, EXTEST_PULSE, EXTEST_TRAIN, SAMPLE, PRELOAD ),"
&
  "BYPASS    ( BYPASS, HIGHZ, CLAMP ) " ;

--Boundary scan definition
attribute BOUNDARY_LENGTH of chip_top: entity is 14;

attribute BOUNDARY_REGISTER of chip_top: entity is
  -- num      cell      port      function      safe [ccell
disval rslt]
  " 13      (bc_0      , *      , internal      , X
),"&
  " 12      (BC_0      , *      , internal      , 0
),"&
  " 11      (BC_2      , BP_A00   , input        , X
),"&
  " 10      (BC_2      , BP_A01   , input        , X
),"&
  " 9       (BC_2      , BP_A02   , input        , X
),"&
  " 8       (BC_2      , BP_A03   , input        , X
),"&
  " 7       (BC_0      , *      , internal      , 0
),"&
  " 6       (BC_0      , *      , internal      , 0
),"&
  " 5       (BC_2      , BP_A10   , input        , X
),"&
  " 4       (BC_2      , BP_A11   , input        , X
),"&
  " 3       (BC_2      , BP_A12   , input        , X
),"&
  " 2       (BC_2      , BP_A13   , input        , X
),"&
  " 1       (BC_0      , *      , internal      , 0
),"&
  " 0       (bc_0      , *      , internal      , X
) ";

```

```
end chip_top;
```

6.2 第三方TAP抽BSDL

不知道为啥又可以了，在example上作测试，添加了自己的IR指令和TDR register，把signature部分给删除了，工具也能够正常跑下去（2023.1 2023.4）。

在chip层抽BSDL时，需要能够识别到tap controller，它是由ICL里面的属性决定的。类似如下的ICL描述：

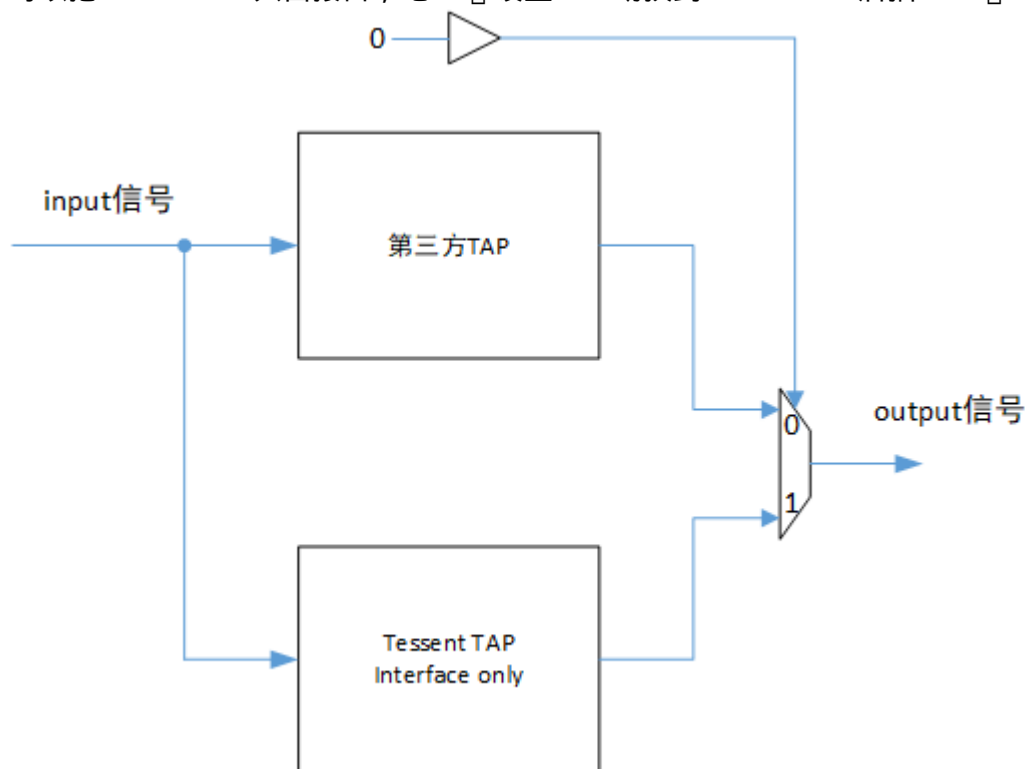
```
Attribute          tessent_instrument_type          =
"mentor::ijtag_node";
Attribute          tessent_instrument_subtype      = "tap_controller";
Attribute          tessent_signature              =
"7f9f11188d689c2876eccd7c615da355";
```

signature会保护module name, port name, ScanRegister name等,signature不对会报如下错误.

```
// Error: The TDI 'TDI' could not be traced to a TAP controller. (BSCAN1-1)
// Error: There was 1 BSCAN1 violation (Boundary Scan extraction TAP
controller identification).
// Error: Rules checking unsuccessful, cannot exit SETUP mode.
```

所以导致第三方的TAP目前是不支持直接抽取BSDL的，抽取BSDL时需要用tessent的TAP

但可以有一个变通方案, 在RTL例化的时候也加一个tessent的TAP
可以把tessent TAP只留接口，吃ICL，设置case切换到tessent TAP后抽BSDL



6.3 带TcdBscan分段抽取BSDL

无需再配置BondingConfig spec, 只需要case相应seg段的enable信号就行。

```
set_dft_specification_requirements -bsdl_extraction on

add_primary_inputs [get_pins cphy_t0/bs_bypass]
add_input_constraints [get_pins cphy_t0/bs_bypass] -C0; # 工具自动根据此值的状态,
来抽取成对应的BSDL文件。

set_attribute_value [get_pins cphy_t0/BS_BYPASS_PORT] -name
unused_tcd_bscan_external_port -value yes; # 设置对应假PORT不出现在BSDL文件中

check_design_rules
```

6.4 多DIE抽取BSDL

tessent多DIE合封, 抽一个整体的BSDL需要设置一个main DIE, 它得有额外的TDI_LOCAL信号, 用于在BYPASS/IDCODE模块兼容IEEE JTAG标准。

当PACKAGE PIN与多个DIE线与时, 不会输出该PIN的BSDL相当于不测。

抽取多DIE BSDL

```
read_verilog multi_die.v
set_current_design multi_die
set_design_level chip

extract_icl
check_design_rules

set_system_mode setup
set_dft_specification_requirements -bsdl_extraction on
read_bsd die.bsd
check_design_rules
```

7. BondingConfiguration

可能某inst不用或powerdown, 有多种封装形态, 此时需要配置BondingConfigurations生成对应的BSDL文件。

TcdBscanSegment指定inst中具体segment段选择, 比如可以让某些bscan cell不上bscan chain, 直接bypass掉, 也可以全部上bscan chain.

工具会自动生成指定bonding configuration的BSDL文件。

```
DftSpecification(module_name,id) {
  EmbeddedBoundaryScan {
```

```

BondingConfigurations {
  BondingConfiguration(name) {
    enable_signal      : port_pin_or_net_name ;
    bypassed_logical_groups : logical_group_name, ... ;
    active_logical_groups  : logical_group_name, ... ;
    TcdBscanSegment(instance_name) {
      segment_selection : segment_selection_name;
    }
  }
}

```

```

DftSpecification(module_name,id) {
  BoundaryScan {
    BondingConfigurations {
      BondingConfiguration(name) {
        enable_signal      : pin_or_net_name ;
        part_number_code   : binary ; // default: 16'h0
        version_code       : binary ; // default: 4'h0
        unused_ports       : port_name_pattern, ... ;
        bypassed_logical_groups : logical_group_name, ... ;
        active_logical_groups  : logical_group_name, ... ;
        TcdBscanSegment(instance_name) {
          segment_selection : segment_selection_name;
        }
      }
    }
  }
}

```

疑问，如果只是抽BSDL也可以这样用吗？

8. bypass segment

如果芯片想支持BSCAN分段bypass就必须设置>1个Logic group, 这样在Bonding config中就可以配置哪些logic group是bypass或者是active

为了支持多段bypass 最好中间插入dummy port, 只是为了方便加logic group 可以把一段功能PORT给bypass掉。

集成TcdBscanSegment时，segment select只能选择其它一种来生成对应BSDL文件

所以为了适应多segment bypass建议每个模块就是一个segment

比如每个PHY BSCAN是一个tcdbscan segment, 这样就可实现多个PHY同时bypass的BSDL配置。

如果顶层IO没有分segment的话，只是集成模块的tcdbscan可以不加enable_signal
tile flow都推荐segment在模块内部进行划分，顶层只负责把所有的tcdbscan给串起来，这样便于bypass各个模块的bscan，利于BSDL自动生成。

```

BondingConfigurations {
  BondingConfiguration(name) {
    TcdBscanSegment(phy1) {
      segment_selection : bypass;
    }
    TcdBscanSegment(phy2) {
      segment_selection : bypass;
    }
  }
}

```

8.1 模块带1个bypass segment结构

推荐, 一个模块只分一个segment

```

# pin order:
DUMMY_PORT    // 加假PORT方便把PORT0 PORT1 BYPASS掉
FUNC_PORT0
FUNC_PORT1

```

```

graph LR
  bscan_scan_in --> DUMMY_PORT\n_BC_CELL
  DUMMY_PORT\n_BC_CELL --> MUX
  DUMMY_PORT\n_BC_CELL --> FUNC\n_BC_CELL0
  FUNC\n_BC_CELL0 --> FUNC\n_BC_CELL1
  FUNC\n_BC_CELL1 --> MUX
  MUX --> bscan_scan_out

```

8.2 模块带2个bypass segment结构

不推荐

如果不加DUMMY PORT, 无法实现功能PORT 分段BYPASS解耦。

```

# pin order:
DUMMY_PORT    // 加假PORT方便把PORT0 PORT1 BYPASS掉
FUNC_PORT0
FUNC_PORT1
DUMMY_PORT2   // 加假PORT方便把PORT2 PORT3 BYPASS掉
FUNC_PORT2
FUNC_PORT3

```

```

graph LR
  bscan_scan_in --> DUMMY_PORT\n_BC_CELL
  DUMMY_PORT\n_BC_CELL --> MUX
  DUMMY_PORT\n_BC_CELL --> FUNC\n_BC_CELL0
  FUNC\n_BC_CELL0 --> FUNC\n_BC_CELL1
  FUNC\n_BC_CELL1 --> MUX
  MUX --> DUMMY_PORT2\n_BC_CELL
  DUMMY_PORT2\n_BC_CELL --> MUX2
  DUMMY_PORT2\n_BC_CELL --> FUNC\n_BC_CELL2
  FUNC\n_BC_CELL2 --> FUNC\n_BC_CELL3
  FUNC\n_BC_CELL3 --> MUX2
  MUX2 --> bscan_scan_out

```