dft spec

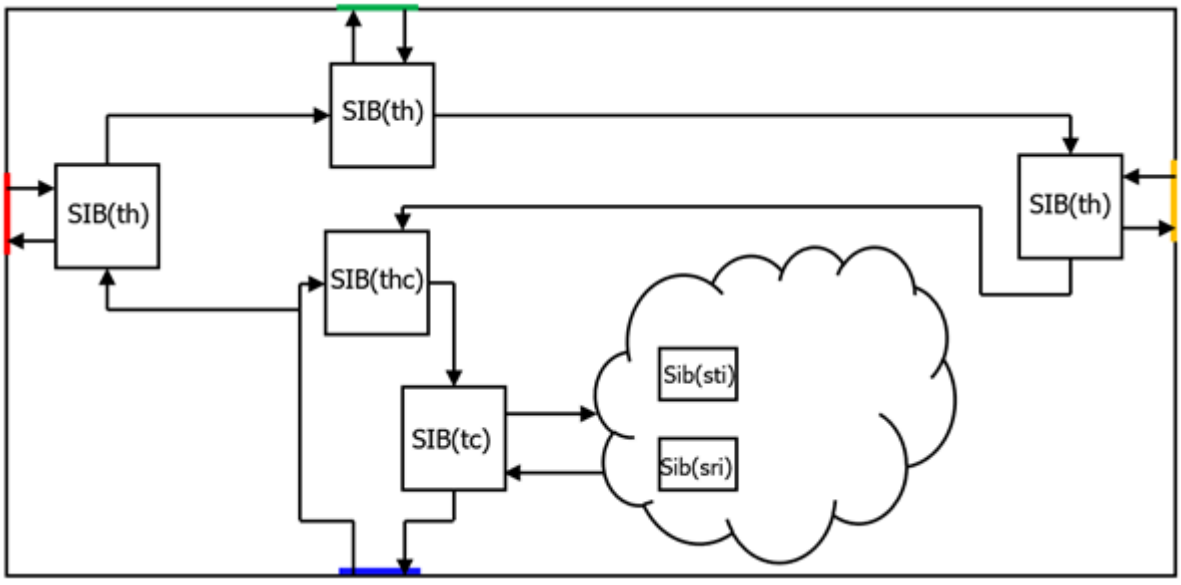
1. create_dft_specification

http://vmcc.vicp.net:9090/tessent_v2023.1_doc/htmldocs/mgchelp.htm#context=tshell_ref&id=142

```
create_dft_specification [-existing_ijtag_host_scan_in
host_scan_in_design_pin_spec]
[-existing_primary_tap_scan_out primary_tap_client_scan_out_design_pin_spec]
[-existing_bscan_host_scan_in bscan_host_scan_in_design_pin_spec]
[-tile_ijtag_host_list tile_ijtag_hosts]
[-stap_host_list stap_nodes]
[-active_high_compliance_enables enable_port_name ...]
[-active_low_compliance_enables enable_port_name ...]
[-sri_sib_list sri_sib_list]
[-sti_sib_list sti_sib_list]
[-replace]
```

1.1 example 4

example 4




```
set_dft_specification_requirements -design_type tile
check_design_rules
set spec [create_dft_specification -tile_ijtag_host_list {left top right}]
report_config_data $spec
```

```
DftSpecification(tile_core,gate) {
  IjtagNetwork {
    HostScanInterface(ijtag) {
```

```
      Sib(tc) {
        Attributes {
          tessent_dft_function : tile_client_sib;
        }
        to_scan_in_feedthrough : pipeline;
        so_retiming : off;
        Sib(sti) {
          [...]
        }
        Sib(sri) {
          [...]
        }
      }
      Sib(thc) {
        Attributes {
          tessent_dft_function : tile_host_collector;
        }
        Sib(th_right) {
          to_scan_in_feedthrough : pipeline;
          SecondaryHostScanInterface(right) {
          }
        }
        Sib(th_top) {
          to_scan_in_feedthrough : pipeline;
          SecondaryHostScanInterface(top) {
          }
        }
        Sib(th_left) {
          to_scan_in_feedthrough : pipeline;
          SecondaryHostScanInterface(left) {
          }
        }
      }
    }
  }
}
```

1.2 tap_t

```
set spec [create_dft_specification -existing_ijtag_host_scan_in
tap/host_1_from_so \
                                   -sri_sib_list occ -tile_ijtag_host_list
{r1}]
report_config_data $spec
```

```
DftSpecification(tap_t,rtl1) {
  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) {
          }
        }
      }
    }
  }
}
```

1.3 chip_top

command

```
set spec [create_dft_specification -existing_ijtag_host_scan_in
tap_t/r1_ijtag_from_so \
                                   -existing_bscan_host_scan_in
tap_t/host_bscan_from_so]
report_config_data $spec
```

spec

```
DftSpecification(chip_top,rtl1) {
  IjtagNetwork {
    HostScanInterface(ijtag) {
      Interface {
        design_instance : tap_t;
        scan_interface : r1;
      }
      Sib(sri) {
        Attributes {
          tessent_dft_function : scan_resource_instrument_host;
        }
        Sib(pb1) {
          DesignInstance(GPS_1) {
            scan_interface : ijtag;
```

```
        }
      }
      Sib(pb2) {
        DesignInstance(GPS_2) {
          scan_interface : ijtag;
        }
      }
      Sib(pb3) {
        DesignInstance(PROCESSOR_1) {
          scan_interface : ijtag;
        }
      }
      Sib(sri_local) {
        Tdr(sri_tdr1) {
          DataInPorts {
            connection(0) : tap_t/select_jtag_output;
            connection(1) : tap_t/select_jtag_input;
            connection(2) : tap_t/force_disable;
            connection(3) : tap_t/extest_train;
            connection(4) : tap_t/extest_pulse;
          }
          reset_value : 5'b00000;
        }
      }
      Sib(sri_ctrl) {
        Tdr(sri_ctrl) {
          Attributes {
            tessent_dft_function : scan_resource_instrument_dft_control;
          }
        }
      }
    }
  }
  HostScanInterface(tap) {
    Interface {
      tck : TCK;
      trst : TRST;
      tms : TMS;
      tdi : TDI;
      tdo : TDO;
    }
  }
  HostScanInterface(bscan) {
    Interface {
      design_instance : tap_t;
      scan_interface : H0;
    }
  }
}
BoundaryScan {
  ijtag_host_interface : HostScanInterface(bscan);
```

```
    BoundaryScanCellOptions {
      REF_CLK : clock;
      INCLK : clock;
    }
  }
}
```

2. get_config_elements

```
get_config_elements [name_patterns] [-hierarchical] [-in_wrappers
wrapper_object_spec]
[-partition partition] [-count] [-type type] [-filter filter]
[-regexp] [-nocase] [-silent]
```

假设spec如下：

```
tmp(1) {
  ABC (Def,2) {
    prop1 : 1;
    Prop2 : 2;
  }
  abc(def,1) {
  }
  abcdef {
  }

}
```

Case 1

This matches the wrappers with leaf name “abc” inside wrapper tmp(1) independent of whether it has an id. The matching is case-insensitive because only the id matching is case-sensitive.

get_config_elements abc -in tmp(1)

{/tmp(1)/ABC(Def,2) /tmp(1)/abc(def,1)}

Case 2

This matches the wrappers with the first id equal to “Def”. Notice that it does not match abc(def,1) as ids are matched considering casing unless the -nocase option is used.

get_config_elements *(Def,*) -in tmp(1)

{/tmp(1)/ABC(Def,2)} get_config_elements *(Def,*) -in tmp(1) -nocase

{/tmp(1)/ABC(Def,2) /tmp(1)/abc(def,1)}

Case 3

This matches elements below the top-level wrapper starting with a*.

get_config_element */a*

{/tmp(1)/ABC(Def,2) /tmp(1)/abc(def,1) /tmp(1)/abcdef}}

Case 4

This matches elements below the top-level wrapper starting with a* and having two ids.

get_config_element */a*(*,*)

{/tmp(1)/ABC(Def,2) /tmp(1)/abc(def,1)}}

Case 5

This matches elements starting with ‘p’ anywhere inside tmp(1).

get_config_element p* -in tmp(1) -hierarchical

{/tmp(1)/ABC(DeF,2)/prop1 /tmp(1)/ABC(DeF,2)/Prop2}

3. get_name_list

```
foreach pat [get_name_list [get_config_element Patterns -hierarchical]] {
  report_config_data $pat;
}
```