SIDEBAND ENDPOINT SGCDC

IP’s instantiating sideband endpoint (SBEP) should be able to follow these steps to run SGCDC and generate the required output collaterals.

Special notes to consider:

- SGCDC is supported only from PICr26 version of EP. If you are using any of the earlier versions, please download PICr26 from IRR and run SGCDC stand alone on EP and use those output files.
- Later versions of EP might have RTL changes through HSD’s filed after SGCDC runs– This is just a placeholder note for future such releases)

Please run these steps for the IRR downloaded version of the EP only. If the tarball is modified (uniquified or any other changes), then these scripts/steps will not apply.

1. Uniquify the SBEP with the IP specific PUNI prefix name <**ip_prefix**>
   a. In SBEP_ROOT, source HDK and uniquify the endpoint

```
➢cd SBEP_ROOT
➢wash -n users fabric soc socenv dk1273 siphdk hdk10nm dk10nm hdk10nmproc coe73 coe73lay soc73 datools hdk22nm hdk22nmproc
➢source /p/hdk/rtl/hdk.rc -cfg sip -reentrant
➢setenv MODEL_ROOT $cwd
➢source scripts/uniquifyme <ip_prefix>
```

Ignore the “Error in PUNI” message. As long as the source/rtl files are all uniquified and the ace files are uniquified, you are good.

2. Convert the SBEP parameters to CSV:
   a. Create an **<ip_prefix>_sbendpoint.csv** file in the ***SBEP_ROOT/source/cfg/endpoint/csv*** area. There are already ~20 examples.
   b. In the CSV file, list the RTLTOP as **<ip_prefix>_sbendpoint** (or **<ip_prefix>_sbebase** whichever applicable)
   c. List all the parameters used to program the SBEP instance in your IP, with their respective values using the SBEP parametric names and **not your IP’s parametric names.**

Note the following in the CSV. Example below shows an instance of sbendpoint.

- RTLTOP name
- Parameter name in CSV is the EP parameter name (not how the IP calls it)
- Parameter value in CSV is value programmed by the IP

FROM	TO

Parameter CLAIM_DELAY = 0;
parameter MATCHED_INTERNAL_WIDTH = 1;
parameter SB_MAXPLDBIT = 7;
parameter SB_QUEUEDEPTH = 2;
parameter SB_CUP2PUT1CYC = 1;
parameter SB_LATCHQUEUES = 0;
 	
psf20_sbendpoint
#(
.CLAIM_DELAY (CLAIM_DELAY),
.MATCHED_INTERNAL_WIDTH (MATCHED_INTERNAL_WIDTH),
.MAXPLDBIT (SB_MAXPLDBIT),
.NPQUEUEDEPTH (32'(SB_QUEUEDEPTH)),
.PCQUEUEDEPTH (32'(SB_QUEUEDEPTH)),
.CUP2PUT1CYC (SB_CUP2PUT1CYC),
.LATCHQUEUES (SB_LATCHQUEUES),
.MAXPCTRGT (32'(SB_MAXPCTRGT)),
.MAXNPTRGT (32'(SB_MAXNPTRGT)),
.MAXMSTRADDR (SB_MAXMSTRADDR),
	RTLTOP,<ip_prefix>_sbendpoint
	PARAM,CLAIM_DELAY,0
	PARAM,MATCHED_INTERNAL_WIDTH ,1
	PARAM,MAXPLDBIT ,7
	PARAM,NPQUEUEDEPTH,2
	PARAM,PCQUEUEDEPTH,2
	PARAM,CUP2PUT1CYC,1
	PARAM,LATCHQUEUES,1

```
.MAXMSTRDATA                 (SB_MAXMSTRDATA),
.RELATIVE_PLACEMENT_EN       (RELATIVE_PLACEMENT_EN),
. . . . . . . .  . . . . . . . . . . . . . . . .
. . . . . . . .  . . . so on.. . . . . . . . . . . .
          )
      i_sbendpoint
        (
         .side_clk        (side_clk),
         .gated_side_clk  (gated_side_clk),
          . . . . . . .  . . . . . . . . . .
         )
```

3. Create the executables/helper scripts:
   This is also described in the SBEP integration guide (Section 6.7) when IP needs to run tools on SBEP.
   In any xterm, do the following (do not copy the commands, type them out instead).

   ➢ `cd $SBEP_ROOT/scripts/qa`
   ➢ `runFullConfig -endpoint`

   You should see a bunch of executables **<ip_prefix>_sbendpoint_run*** in **`SBEP_ROOT/tools`** directory.

4. Source HDK in a **fresh XTERM.** HDK steps are also listed in the SBEP_ROOT/cfg/README.hdk
   ➢ `cd SBEP_ROOT`
   ➢ `wash -n users fabric soc socenv dk1273 siphdk hdk10nm dk10nm hdk10nmproc coe73 coe73lay soc73 datools hdk22nm hdk22nmproc`
   ➢ `source /p/hdk/rtl/hdk.rc -cfg sip –reentrant`
   ➢ `setenv MODEL_ROOT $cwd`

5. Generate the CDC tests area: Use the gensgcdccollat script to copy and edit the cdc collateral and hdl files
   ➢ `perl $SBEP_ROOT/unsupported/scripts/gensgcdccollat <ip_prefix> <rtltop>`
   `e.g. perl $SBEP_ROOT/unsupported/scripts/gensgcdccollat psf20 sbendpoint`

6. Run simbuild and CDC

   ➢ `simbuild -dut sbe -1c -CUST` ***`<p1273 or p1274>`*** `-toolset mat1.4.10plus.p2 -1c- -ace xterm &`
   `e.g. simbuild -dut sbe -1c -CUST p1273 -toolset mat1.4.10plus.p2 -1c- -ace xterm &`

   In the xterm that pops up, run the process specific HDK executable for CDC
   ➢ `cd tools`
   ➢ **`<ip_prefix>_sbendpoint`**`_runSGCDC_HDKp1273`
   `e.g. psf20_sbendpoint_runSGCDC_HDKp1273`

7. The final abstract can be found here:
   IP_ROOT/target/sbe/<CUST>/<mat>/aceroot/results/cdc/tests/spyglasscdc_<ip_prefix_rtltop>/<ip_prefix_rtltop>/<ip_prefix>_iosf_sbc_ep_rtl_lib.<ip_prefix_rtltop>/cdc/cdc_verify_struct/spyglass_reports/abstract_view/<ip_prefix_rtltop>_......_cdc_abstract.sgdc

NON UNIQUIFIED EP:

If the endpoint is consumed as it is (without uniquification), SKIP 1 and 5, and in all the above/previous steps (2-6), instead of the “<ip_prefix>_<rtltop>”, just use the <rtltop>.

1. SKIP the uniquification
2. Convert the SBEP parameters to CSV: There is already a sbendpoint.csv and sbebase.csv in ***SBEP_ROOT/source/cfg/endpoint/csv,*** that can be edited to the customer IP’s parameters, following the same guidelines as uniquified (except, of course, the uniquified name)

3. Same as uniquified (except there will not be any <ip_prefix> executables. Instead sbendpoint_run* or sbebase_run* and other flavors will be generated.
4. Same as uniquified
5. SKIP
6. Same as uniquified, instead use “sbendpoint_runACECDC_HDKp1273 or sbebase_runACECDC_HDKp1273”