CCU VC FAQs

IP Rev. 0.7

Rajitha Ravindran

1 FAQs

Q: Do i need multiple instances of ccu-vc in my env? So one instance per clk?
A: NO. You should be able to program multiple slices to drive different clks that you need in your environment.
There needs to be just one instance per ip. You just need one slice per clk

Q: Does this bfm drive resets?
A: NO. This bfm is focused on clks/gating/handshake etc. We dont have reset driving ability added to it currently

Q: Is this the same as CRG bfm? IF not Which one should i use?
A: NO. This is a Chassis spec based bfm , which makes sure the ip clks are Chassis Spec compliant.
Please make sure you use this bfm for all clks that need to be chassis compliant (All ip clks have that
requirement). If you are generating some internal clks for checkers/generators etc and choose to generate it using
crg or some other bfm then that is fine.

Q:Does this bfm support gating/ungating?
A:YES. That is supported, please see userguide for examples and more details

Q: I was trying to reuse your ccu_seq in order to configure CCU Unit and I am hitting a problem with num_slices variable not populated before randomization is called.
A: Usually validators use the set cfg call and then randomize, You wouldn't hit this issue if you do that

Q: CCU vc specifies hdl files inside its hdl? Is that allowed?
A: Yes it does, If you refer to the block architecture explained in the userguide it actually uses 2 other sub agents to function. It is allowed to point to sub hdl files in your top hdl file. Please add the ccu_vc hdl search path in your acerc.

Q: What interface does this agent drive CCU-DCG or CCU-IP?
A: CCU interface provides clkreq/ack clk that follow IOSF req-ack handshake rules. So it is modelling the DCG downstream interface.

Q: I have a CRG in my env can I still use CCU VC?
A: Yes we have taken care of the uniquification in release r121015. So you should be able to use both agents in your env if needed

Q: The clocks are all active by default ? I need clocks low in my env
A: Please use the cmd - CLK_GATE to gate all your clks from your test anytime.

Q: Can I program delay from clkreq -> 0 to clkack ->0 ?
A: According to spec , as soon as IP deasserts its clkreq the DCG block should serve that request and make clkack 0 immediately. You can program the number of clks the clk stays active after the clkack -> 0. You can always program the clk gated/ungated irrespective of clkreq/ack by using the commands CLK_GATE or CLK_UNGATE from the test, in case you want to model error conditions for negative testing.

Q: The Configuration Object is not updated when configuration is change through sequence. It will be great if it can be done
A: I have made this update so the config object gets updated everytime there is a test command executed on the ccu vc. You can choose to print it from the test if needed. Also

there is a print statement given from the ccu_seq. So everytime you issue a command from the test you should see something like this get printed in the the transcript window. This can also serve debug purpose.

```
-------------------------------------------------------------------
Name                    Type              Size              Value
-------------------------------------------------------------------
ccu_xaction              ccu_vc_pkg::ccu_xa+ -          ccu_xaction@240
  slice_num              integral           32                 'd2
  clk_src              integral           32                 'd7
  cmd                  ccu_types::cmd_e    4             CLKACK_DLY
  div_ratio            ccu_types::div_rat+ 5                  DIV_6
  half_div_ratio         integral           32                 'h1
  clkack_delay           integral           32                 'd8
  begin_time             time             64             420094000
  end_time               time             64             410094000
  depth                int              32                  'd2
  parent sequence (name) string              7                ccu_seq
  parent sequence (full+ string             54   ovm_test_top.i_ccu_+
  sequencer              string             46   ovm_test_top.i_ccu_+
-------------------------------------------------------------------
```

Apart from this there is a initial configuration message also printed. It is just a print of the config object which should look somewhat like this -

```
-------------------------------------------------------------------
Name                    Type              Size              Value
-------------------------------------------------------------------
i_ccu_vc_cfg             ccu_vc_pkg::ccu_vc+ -          i_ccu_vc_cfg@14
  is_active            ovm_active_passive+ 1             OVM_ACTIVE
  num_slices             integral           32                 'd4
  i_ccu_crg_cfg          ccu_crg_cfg         -        i_ccu_crg_cfg@15
    gate_all_clocks      integral            1                 'h0
    assert_all_reset     integral            1                 'h0
    deassert_all_reset   integral            1                 'h0
    ti_name              string             29   ccu_vc_tb.ccu_ti.i_+
    is_active          ovm_active_passive+ 1             OVM_ACTIVE
  i_ccu_ob_cfg           ccu_ob_cfg          -          ccu_ob_cfg@16
    is_active          ovm_active_passive+ 1             OVM_ACTIVE
  slices               sa(object)         4                    -
    [0]                ccu_vc_pkg::slice_+ -        slice_cfg[0]@33
      slice_num          integral           32                 'd0
      slice_name         string             4                PSF0
      clk_src          integral           32                 'd0
      clk_status         ccu_types::clk_gat+ 1              CLK_GATED
      divide_ratio       ccu_types::div_rat+ 5                  DIV_4
      half_divide_ratio  integral           32                 'h1
      clkack_delay       integral           32                 'd8
      usync_enabled      integral            1                 'h0
    [1]                ccu_vc_pkg::slice_+ -        slice_cfg[1]@34
      slice_num          integral           32                 'd1
      slice_name         string             10           PSF0_PORT0
      clk_src          integral           32                'd10
      clk_status         ccu_types::clk_gat+ 1            CLK_UNGATED
      divide_ratio       ccu_types::div_rat+ 5                  DIV_2
      half_divide_ratio  integral           32                 'h1
      clkack_delay       integral           32                'd24
      usync_enabled      integral            1                 'h1
    [2]                ccu_vc_pkg::slice_+ -         slice_cfg[2]@35
      slice_num          integral            32                'd2
      slice_name         string              10          PSF0_PORT1
      clk_src            integral            32                'd5
      clk_status         ccu_types::clk_gat+ 1            CLK_UNGATED
      divide_ratio       ccu_types::div_rat+ 5                  DIV_8
      half_divide_ratio  integral            32                'h1
      clkack_delay       integral            32                'd8
      usync_enabled      integral             1                'h0
    [3]                ccu_vc_pkg::slice_+ -         slice_cfg[3]@36
      slice_num          integral            32                'd3
      slice_name         string              10          PSF0_PORT2
      clk_src            integral            32                'd3
      clk_status         ccu_types::clk_gat+ 1            CLK_UNGATED
      divide_ratio       ccu_types::div_rat+ 5                 DIV_16
      half_divide_ratio  integral            32                'h1
      clkack_delay       integral            32                'd8
      usync_enabled      integral             1                'h0
  clk_sources            sa(object)          16                 -
    [0]                ccu_vc_pkg::clksrc+ -        clksrc_cfg[0]@17
      clk_num            integral            32                'd0
      clk_name           string               5              CLK_0
      period             real                64                31
```