Avery 验证 PCIe VIP 使用总结【1】

VIP

关键词	笔记
apci_device	该类相当于pcie的设备，一般例化rc、ep0、ep1，相当于PCIe的Root Complex，ep0、ep1相当于PCIe Endpoint。在VIP中一般例化中间变量bfm、rc_app_bfm；其中rc_app_bfm可以看作rc，bfm通常为测试用例的待测设计，在vip中大部分场景都是rc。
apci_device_mgr	该类存放枚举的所有结果，例化名mgrs[$]是队列形式；rc调用方法collect_devices(-1, mgrs)来获取枚举信息，其中第一个参数是枚举的层级，VIP 中的-1基本都表示枚举所有的层级。 主要的变量和方法： ▪ dev_type ▪ bus_num ▪ finfs：apci_func_info finfs[$] ▪ sprint: 打印信息 ▪ wait_configured(time)：等待 device 启动。一般时间设置1e9，即 $1x10^9$ fs即1μs。
apci_log	apci_log test_log = apci_pkg_test::test_log; 会在log中以uvm_info、uvm_error等形式打印出来，自己写用例时直接引用，无需更改。 ❗ VIP uvm 用例集中class mycb extends apci_callbacks;会调用test_log.info, 需要修改成bfm.log.info，否则会报错。
apci_seq_util	一般例化名seq，VIP uvm 测试集中调用方法creat_seq_util(p_sequencer, seq, bfm, rc_app_bfm)，主要目的是seq创建空间。在我们的SoC TB中，无法直接调用该方法，所以我们直接例化空间就行：改为seq = new（bfm， rc_app_bfm）；

pre_bfm_started	我们可以自己写这个task，顾名思义这个task是在BFM开始之前就要将参数传进去。以SoC TB中举例，m_env.rcs[0].cfg_info.ARI_sup = 1;m_env.eps[0].append_callback(cb0)。 🦌 具体的configure参数可以在html中apci_cfg_info中找到所有参数。但是VIP没有写这些参数的默认值，用到的参数就全部配一遍吧。 🍺 注意我们SoC TB 流程中，都要基于DPI_C来开始和枚举，在pcie_225_top_dpi_example_test.c中，用了apci_set()方法进行了初始配置，比如就取消了ARI forward。这里apci_set()引用的VIP function为apci_device.set()。
apci_device.set()	Ask BFM to start an action or change the BFM's parameters. DPI_C --> **apci_set()**
apci_device.get()	Read BFM state and status variables. DPI_C --> **apci_get()**
apci_device.port_get()	Get status information from a port. DPI_C --> **apci_port_get()**
apci_device.post_transaction()	Post a transaction for BFM to send
apci_device.post_tlp()	Post a TLP to a port
apci_device.append_callback()	Add a callback object at the end of the existing callback queue.
apci_device.prepend_callback()	Add a callback object at the beginning of the existing callback queue.

apci_device.unregister_callback()	Remove the callback object.
apci_tlp	这个类包含了PCIe TLP 所有信息，使用apci_device.post_tlp()来调用。 我们一般用到以下几个变量和方法： ▪ Kind：TLP类型，例子 APCI_TLP_cfgrd ▪ U：union of header dwords in big-endian（没看懂） ▪ rx_cpls：Cpl/CplD ▪ req: request TLP, for completion. ▪ is_write(): 1 is write, 0 is read. ▪ sprint(): srting print **使用方法：** 1. tlp.randomize() with { 2. kind == **; 3. u.cfg.bdf.bus == **; 4. u.cfg.bdf.dev == **; 5. u.cfg.bdf.func != **; 6. } **检查返回TLP包：** ◦ tlp.rx_cpls[0].u.cpl.cpl_status[2:0] ▪ 000b (SC) ▪ 001b (UR) ▪ 010b (CRS) ▪ 100b (CA) ◦ tlp.rx_cpls[0].u.cpl.cpl_id[15:0] ▪ [15:8] Completer Bus # ▪ [7:3] Completer Dev # ▪ [2:0] Completer Function # ◦ tlp.rx_cpls[0].u.cpl.req_id[15:0] ◦ tlp.rx_cpls[0].u.cpl.tag[7:0] 通过上述变量，我们可以在测试用例中或者monitor中添加针对CPL包的检查。
apci_transcation	这个类是模型应用层的数据，使用apci_device.post_transaction()来调用。不需要TLP信息。每一个Transaction包含一个或者多个TLPs数据包。

	我们一般用到以下几个变量和方法： ▪ Kind：例子 APCI_TRANS_mem ▪ port_id ▪ func_id ▪ bdf ▪ addr ▪ payload: · k = %urandom; · repeat(tr.length); · tr.payload.push_back(k++); ▪ length: APCI_TRANS_cfg的length只能为1，而且只能rc来产生发送。 ▪ is_write: 1 is write, 0 is read. ▪ err_code: tr.err_code != apci_transcation::OK {OK, ERROR, RESET, TIMEOUT, RETRY, ABORTED} ▪ tlps: apci_tlp tlps[$], cpl 可以通过 tlp.rx_cpls[$] 来调用。 ▪ last_rx_cpl: 从 Non-Posted 事务返回的最后一笔 Cpl/CplD 数据包。 ▪ msg_code: 只适用在 Msg 事务中。 ▪ user_ctrl: 具体参考 html apci_transcation:user_ctrl，示例：配置 tr.user_ctrl.retry_tlp = 1；用于产生CPL(CRS)状态。 ▪ sprint(): srting print，打印tr信息，参考VIP用例。
set_severity_by_id	用于取消断言检查。 例如 ep0.log.set_severity_by_id(APCI2_3_1n1, AVY_EXPECT) APCI2_3_1n1代码对应的断言检查，可以在下述路径Excel表格中找到，并且可以对应代码看到相关PCIe协议原文。 /apciexactor-2.3e/doc/apcie_checklistx.xls

结语

HTML 中能查到的信息有限，可以在下述路径中，追一下部分未加密的源码。

/apciexactor-2.3e/src.IEEE/