IST(in-system test)

是用于车规级等安全性要求比较高的芯片场景。

主要是IST可以支持产生JTAG信号，用于在线控制logic bist & memory bist等。

其中如果为了加速memory bist, 可以使用BAP的DirectAccess 来控制（这块与IST产生的JTAG接口没有关系）。

Figure 1. Initial Design for Automotive Test Case

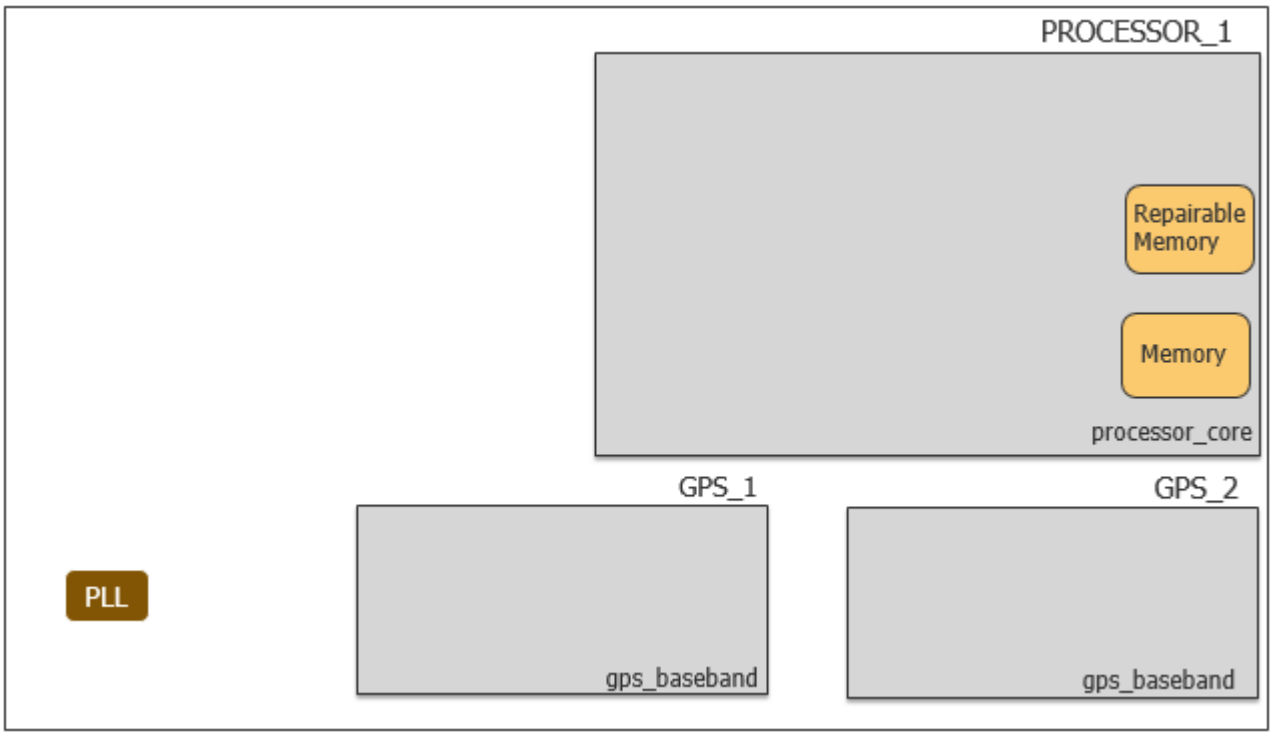


Figure 2. DFT Integration at the Core Level for Automotive

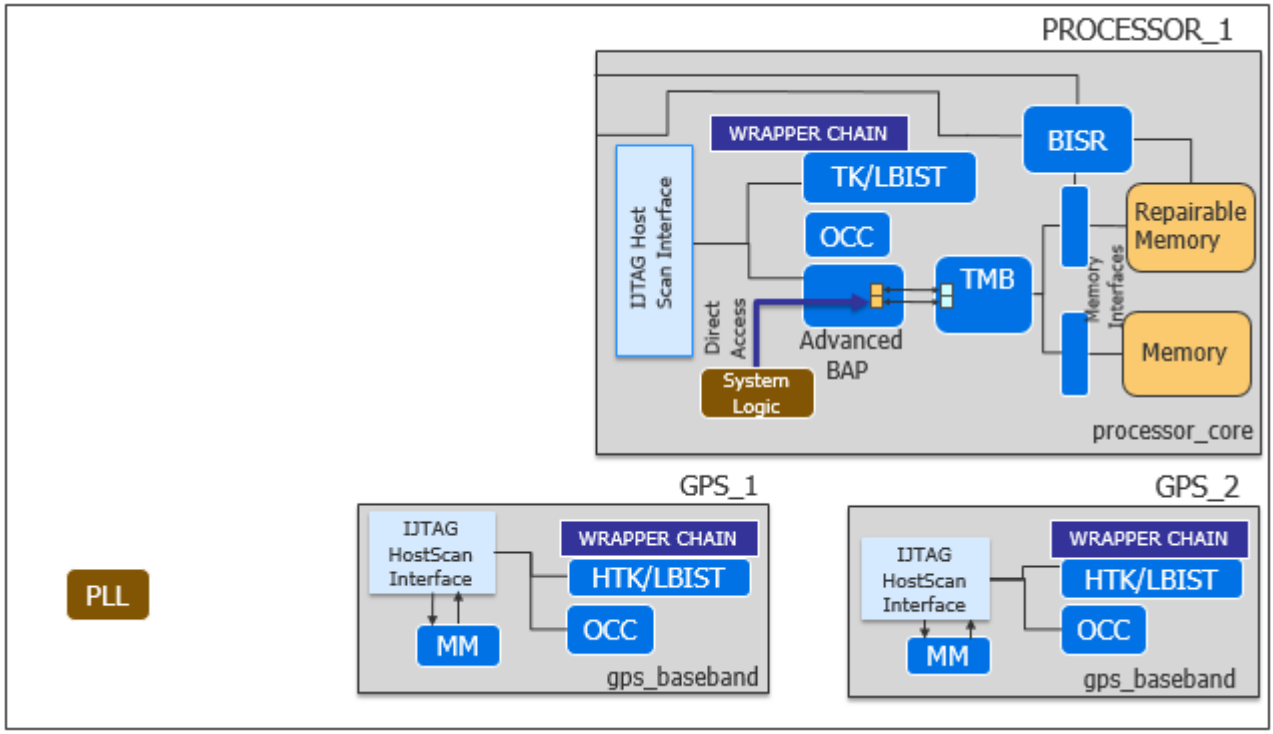
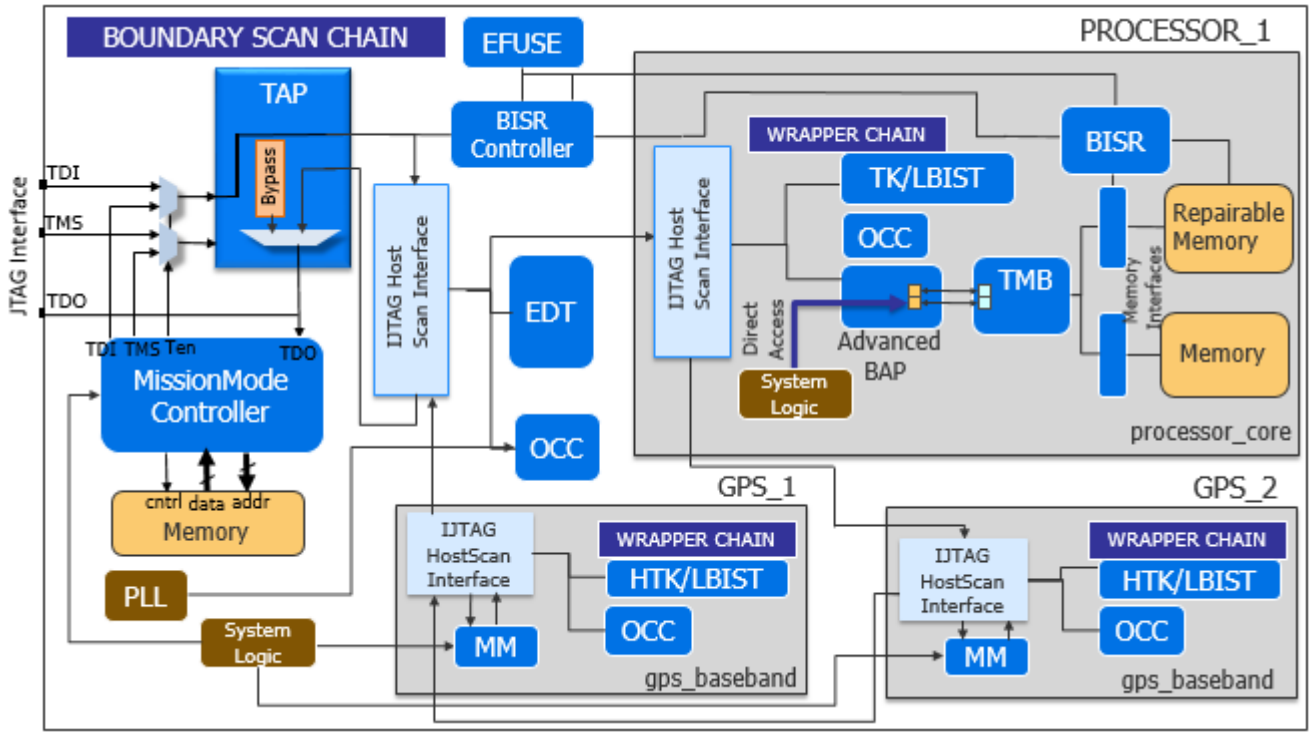


Figure 3. DFT Integration at the Top Level for Automotive



1. dft spec

注：目前IST只能通过把TRSTB拉低来实现JTAG复位，其无法发出连续把TMS拉高5个tck cycle这个是硬件代码限制。

IST数据接口有两种：

- cpu interface -- 数据直接由FW控制，方便直接写入到FW程序中

- dma interface -- 速度快，数据直接从memory内存中搬移

而且可以两种接口的IST都可以一起加进来，这个超级方便，不用二选一，两者都要。

```
InSystemTest {
   Controller(cpu) {
#指定为current design, 添加一个CPU接口的IST
      DesignInstance (.){}
    //host_interface: HostScanInterface(sri);
      data_width : 32 ;
      protocol : cpu_interface;
      ControllerChain {
        present : on ;
        clock: tck;
        segment_per_instrument: on ;
      }
      Connections {
        reset : ist_if/reset;
        CpuInterface {
          Generic {
            clock   : ist_if/clock    ;
            data_in : ist_if/data_in  ;
            data_out: ist_if/data_out ;
            write_en: ist_if/write_en ;
            enable  : ist_if/enable   ;
          }
        }
      }
   }

#指定instance的host scan interface, 添加一个DMA接口的IST
   Controller(dma) {
      DesignInstance(chip_top_rtl1_tessent_tap_main_inst) {
          client_interface : tap_client;
      }
    //host_interface: HostScanInterface(sri);
      data_width : 32 ;
      protocol : cpu_interface;
      ControllerChain {
        present : on ;
        clock: tck;
        segment_per_instrument: on ;
      }
      Connections {
        reset : ist_if/reset;
        CpuInterface {
          Generic {
            clock   : ist_if/clock    ;
            data_in : ist_if/data_in  ;
            data_out: ist_if/data_out ;
            write_en: ist_if/write_en ;
```

```
            enable  : ist_if/enable   ;
          }
        }
      }
    }
}
```

2. pattern spec

```
read_config_data -in $spec -last -from_string {
    InSystemTest {
   #这个可以指定具体的IST□当多个IST存在时，不会有pattern冲突
        Controller(icl_instance_name) {
            TestProgram(0) {
                pattern : pattern_wrapper_name ;
                finish_with_ireset : on | off ;
            }
    //可以写多个测试program，  cpu接口时，工具只跑最后一个tesetprogram的仿真。
            TestProgram(1) {
                pattern : pattern_wrapper_name ;
                finish_with_ireset : on | off ;
            }
        }
    }
}
```