Example Test Method Scripts

Description: This document contains a general code structure using SmartRDI syntax, followed by three complete scripts of different test methods. Each script demonstrates a specific test method within the SmartRDI framework, providing a clear structure and unique code blocks for each method. This document aims to provide an overall structure for different test methods and serve as a guide for embedding specific test method code blocks within the general SmartRDI structure.

SmartRDI: Code structure

Describes the general structure of programming with SmartRDI for test methods. Use this as a framework:

```
/************************* SmartRDI Code Structure *******************************************/
virtual void run()
{

RDI_INIT();

// Set the primary timing and levels
Primary.timing(TIMING_SPEC(1,1)); Primary.timing(1);
Primary.level(LEVEL_SPEC(1,1)); Primary.level(1);

  ON_FIRST_INVOCATION_BEGIN();

    CONNECT();

    RDI_BEGIN();

/************************** Start of Test Method *********************************/

      // Insert the code bloeck for SmartRDI Test Method here

/************************** End of Test Method *********************************/

    RDI_END();

  ON_FIRST_INVOCATION_END();

  SMC_ARM();

  return;
}

/******************************************************************************/
virtual void SMC_backgroundProcessing()
{
  // Log the test results
/************************** Start of logging Test Method *********************************/


     // Insert the code block for Logging SmartRDI Test Method Results


/************************** End of logging Test Method *********************************/


  return;
}
```

Example Script (1): Leakage Test

```
 /************************* SmartRDI Test Method *******************************************/
virtual void run()
{

// Prepare the pins for testing
string PinsHi  = "Pin1,Pin3,Pin5,Pin7,Pin9";
string PinsLo  = "Pin0,Pin2,Pin4,Pin6,Pin8";

RDI_INIT();

// Set the primary timing and levels
Primary.timing(TIMING_SPEC(1,1)); Primary.timing(1);
Primary.level(LEVEL_SPEC(1,1)); Primary.level(1);

  ON_FIRST_INVOCATION_BEGIN();

   CONNECT();

   RDI_BEGIN();

/************************** Start of Leakage Test Method *********************************/

     // Initialize variables and values
     double VDD33 = 3.3; // V
     double VSS = 0.0; // V

     // Set and test the device by executing a test pattern
     rdi.func("Functional_Leakage_Test").label("Leakage_Test_Pattern").execute();

     // Set the pins corresponding to conditions for testing
     rdi .dc().pin(PinsHi).vForce(VDD33).iRange(40 mA).execute();
     rdi .dc().pin(PinsLo).vForce(VSS).iRange(40 mA).execute();

     // Apply some settling time
     rdi.wait (10 us);
     // Measure the pin(s) to be tested
     rdi.dc("Parametric_Leakage_Test").pin("PinA").vForce(VDD33).iMeas().iRange(40 mA).execute();

     // Optional: Set the pins and relays ready for the next step
     rdi .hwRelay() .pin("trst_n") .setOn("AC") .setOff("PPMU") .execute();

     // Apply some settling time
     rdi.wait (10 us);

/************************** End of Leakage Test Method *********************************/


   RDI_END();

  ON_FIRST_INVOCATION_END();

  SMC_ARM();

  return;
}

/*****************************************************************************/
virtual void SMC_backgroundProcessing()
{
  // Log the test results

/************************** Start of logging Leakage Test Method *********************************/

  int functional_result = rdi.id("Functional_Leakage_Test").getPassFail();
  double parametric_result = rdi.id("Parametric_Leakage_Test").getValue("PinA");

  cout << " Site Number: " << SMC_SITE_NUMBER() << " Functional test result: " <<  functional_result << endl;
  cout << " Site Number: " << SMC_SITE_NUMBER() << " Parametric test result: " <<  parametric_result << endl;

/************************** End of logging Leakage Test Method *********************************/

  return;
}
```

Example Script (2): Functional Test

```
 /************************* SmartRDI Test Method *******************************************/
virtual void run()
{

RDI_INIT();

// Set the primary timing and levels
Primary.timing(TIMING_SPEC(1,1)); Primary.timing(1);
Primary.level(LEVEL_SPEC(1,1)); Primary.level(1);

  ON_FIRST_INVOCATION_BEGIN();

    CONNECT();

    RDI_BEGIN();

/************************** Start of Functional Test Method *********************************/

      // Apply some settling time
      rdi.wait (10 us);

      // Set and test the device by executing a test pattern
      rdi.func("Functional_Test").label("Functional_Test_Pattern").execute();

/************************** End of Functional Test Method *********************************/

    RDI_END();

  ON_FIRST_INVOCATION_END();

  SMC_ARM();

  return;
}

/*****************************************************************************/
virtual void SMC_backgroundProcessing()
{
  // Log the test results

/************************** Start of logging Functional Test Method *********************************/

  int functional_result = rdi.id("Functional_Test").getPassFail();

  cout << " Site Number: " << SMC_SITE_NUMBER() << " Functional test result: " <<  functional_result << endl;

/************************** End of logging Functional Test Method *********************************/

  return;
}
```

Example Script (3): DC Test

```
 /************************* SmartRDI Test Method *******************************************/

virtual void run()
{

RDI_INIT();

// Set the primary timing and levels
Primary.timing(TIMING_SPEC(1,1)); Primary.timing(1);
Primary.level(LEVEL_SPEC(1,1)); Primary.level(1);

  ON_FIRST_INVOCATION_BEGIN();

   CONNECT();

   RDI_BEGIN();

/************************** Start of DC Test FVMI Test Method *********************************/

     // Initialize variables and values
     double VDD33 = 3.3; // V

     // Set and test the device by executing a test pattern
     rdi.func("DC_Test").label("DC_Test_Pattern").execute();

     // Apply some settling time
     rdi.wait (10 us);

     // Measure the pin(s) to be tested
     rdi.dc("Parametric_Test").pin("PinA").vForce(VDD33).iMeas().iRange(40 mA).execute();

/************************** End of DC Test FVMI Test Method *********************************/

   RDI_END();

  ON_FIRST_INVOCATION_END();

  SMC_ARM();

  return;
}

/******************************************************************************/
virtual void SMC_backgroundProcessing()
{
  // Log the test results
/************************** Start of logging DC Test FVMI Test Method *********************************/

  int dc_result = rdi.id("DC_Test").getPassFail();
  double parametric_result = rdi.id("Parametric_Test").getValue("PinA");

  cout << " Site Number: " << SMC_SITE_NUMBER() << " DC test result: " <<  dc_result << endl;
  cout << " Site Number: " << SMC_SITE_NUMBER() << " Parametric test result: " <<  parametric_result << endl;

/************************** End of logging DC Test FVMI Test Method *********************************/

  return;
}
```