Module: SV 2
Section: Direct Programming Interface **Task:** DPI

Direct Programming Interface - EDA Link

Task

➢ C function:

```
// Author: Noman Rafiq
// Dated: Oct 21, 2024

#include <svdpi.h>

void alu_reference_model(const svLogicVecVal* A,
                         const svLogicVecVal* B,
                        const svLogicVecVal* ALU_Sel,
                         svLogicVecVal* ALU_Out,
                         svLogic* Carry_Out) {

  // Type-Cast values to 8-bit vectors
  unsigned char a = (A->aval) & 0xFF;
  unsigned char b = (B->aval) & 0xFF;
  unsigned char select = (ALU_Sel->aval) & 0xF;      // Use only last 4-bits
  unsigned char result;

//      printf("a = %0d, b = %0d, select = %0d, A->aval = %0d, B->aval = %0d", a, b,
select, A->aval, B->aval);

  switch(select) {

    // ADD CASE
    case 0:
    result     = a + b;
    *Carry_Out = (result < a)? 1:0;

    //     printf("A = %0d, B = %0d, ALU_Out(ADD) = %0d\n", A, B, ALU_Out);

    //     printf("C Side :: Select = %0d\n", select);
    break;

    // SUB CASE
    case 1:
    result     = a - b;
    *Carry_Out = (result > a)? 1:0;
    break;

    // MULT CASE
    case 2:
    result             = a * b;
    *Carry_Out = 0;
    break;
```

```
    // DIV CASE
    case 3:
    result             = (b != 0) ? a / b : 0;
    *Carry_Out = 0;
    break;

    // Left-Shift CASE
    case 4:
    result             = a << 1;
    *Carry_Out = (a & 0x80)? 1:0;
    break;

    // Right-Shift CASE
    case 5:
    result             = a >> 1;
    *Carry_Out = (a & 0x01)? 1:0;
    break;

    // Rotate-Left CASE
    case 6:
    result             = (a << 1) | (a >> 7);
    *Carry_Out = 0;
    break;

    // Rotate-Right CASE
    case 7:
    result             = (a >> 1) | (a << 7);
    *Carry_Out = 0;
    break;

    // AND CASE
    case 8:
    result             = a & b;
    *Carry_Out = 0;
    break;

    // OR CASE
    case 9:
    result             = a | b;
    *Carry_Out = 0;
    break;

    // XOR CASE
    case 10:
    result             = a ^ b;
    *Carry_Out = 0;
        break;

    // NOR CASE
    case 11:
    result             = ~(a | b);
    *Carry_Out = 0;
    break;
```

```
    // NAND CASE
    case 12:
    result             = ~(a & b);
    *Carry_Out = 0;
    break;

    // XNOR CASE
    case 13:
    result             = ~(a ^ b);
    *Carry_Out = 0;
    break;

    // Greater CASE
    case 14:
    result             = (a > b) ? 1:0;
    *Carry_Out = 0;
    break;

    // Equal CASE
    case 15:
    result             = (a == b) ? 1:0;
    *Carry_Out = 0;
    break;

    default:
    result             = a + b;
    *Carry_Out = (result < a)? 1:0;
    break;
  }

  // Assign values to Actual variable
  ALU_Out->aval = result;
  ALU_Out->bval = 0;

//   printf("result = %0d", result);

}
```

➢ Updated Testbench:

```
`timescale 1ns / 1ps

module tb_alu;

//Import Function for ALU
  import "DPI-C" context function void alu_reference_model(input logic [7:0] A, B,
                                                            input logic [3:0] ALU_Sel,
                                                            output logic [7:0] ALU_Out,
                                                            output logic CarryOut
                                                           );

//Inputs
 logic [7:0] A,B;
```

```
 logic [3:0] ALU_Sel;

//Outputs
 logic [7:0] ALU_Out;
 logic  CarryOut;

 //Expected Outputs variables
 logic [7:0] exp_result;
 logic exp_carry;

 // Verilog code for ALU
 integer i;
 alu test_unit(
            A,B,  // ALU 8-bit Inputs
            ALU_Sel,// ALU Selection
            ALU_Out, // ALU 8-bit Output
            CarryOut // Carry Out Flag
     );
    initial begin
    // hold reset state for 100 ns.
      A = 8'h0;
      B = 8'h0;
      ALU_Sel = 4'h0;
      #100;

      A = 8'hFA;
      B = 8'hF0;
      ALU_Sel = 4'h0;

      for (i=0;i<=15;i=i+1)
      begin
       #1;

       //Call ALU function here and store the result in expected output variables
        alu_reference_model(A, B, ALU_Sel, exp_result, exp_carry);

        if (exp_carry != CarryOut && exp_result != ALU_Out)
          $display("test Fail: Expected=%d, Actual=%d", exp_result, ALU_Out);
      else
        $display("test Pass: Expected=%d, Actual=%d", exp_result, ALU_Out);
       #10;
       ALU_Sel = ALU_Sel + 4'h1;
      end;

      A = 8'hF6;
      B = 8'h0A;

    end
endmodule
```

➢ **Output Screenshot:**

```
g++  -o ../simv      -rdynamic  -Wl,-rpath='$ORIGIN'/simv.daidir -Wl,-rpath=./simv.daidir -Wl,-rpath=/apps/vcsmx/vcs/U-2023.03-SP2/linux64/lib -L/apps/vcsmx/
../simv up to date
CPU time: .367 seconds to compile + .295 seconds to elab + .395 seconds to link
Chronologic VCS simulator copyright 1991-2023
Contains Synopsys proprietary information.
Compiler version U-2023.03-SP2_Full64; Runtime version U-2023.03-SP2_Full64;  Oct 22 09:12 2024
test Pass: Expected=234, Actual=234
test Pass: Expected= 10, Actual= 10
test Pass: Expected= 96, Actual= 96
test Pass: Expected=  1, Actual=  1
test Pass: Expected=244, Actual=244
test Pass: Expected=125, Actual=125
test Pass: Expected=245, Actual=245
test Pass: Expected=125, Actual=125
test Pass: Expected=240, Actual=240
test Pass: Expected=250, Actual=250
test Pass: Expected= 10, Actual= 10
test Pass: Expected=  5, Actual=  5
test Pass: Expected= 15, Actual= 15
test Pass: Expected=245, Actual=245
test Pass: Expected=  1, Actual=  1
test Pass: Expected=  0, Actual=  0
          V C S   S i m u l a t i o n   R e p o r t
Time: 276000 ps
CPU Time:      0.360 seconds;       Data structure size:   0.0Mb
Tue Oct 22 09:12:09 2024
Done
```