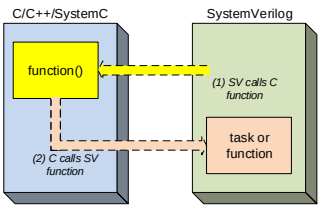
# DPI: C calls SV routine

2020

Ando Ki, Ph.D.
adki@future-ds.com

## Table of contents





# Syntax export method

export “DPI-C” [*c_identifier*=] [function | task ] <*dpi_function_prototype*>;

- export
- DPI or DPI-C
- c_identifier
- function
- task
- dpi_function_prototype



# C calls SV function with arguments (1/3)

```
#include <stdio.h>
#include "dpiheader.h"

int myCFunc ( int A, int B ) {
    printf("Hello world from C! %d %d₩n", A, B);
    int C = mySVFunc(A+10, B+20);
    return C;
}
```

‘xelb’ generates this with ‘-dpiheader’ option

# C calls SV function with arguments (2/3)

```
module top;
    export "DPI-C"      function     mySVFunc;
    import "DPI-C" pure function int myCFunc(input int A, input int B);

    initial begin
        $display("%m %d", myCFunc(10, 20));
        $finish;
    end

    function int mySVFunc(input int A, input int B);
    begin
        $display("Hello from SystemVerilog! %d %d", A, B);
        return A+B;
    end
    endfunction
endmodule
```



# C calls SV function with arguments (3/3)

```
SHELL=/bin/sh

DIR_C       = ../c
DIR_VERILOG = ../verilog
LIB_DPI     = mydpi

all:
        xelab -dpiheader dpiheader.h -svlog ${DIR_VERILOG}/file.sv
        xsc -compile ${DIR_C}/function.c
        xsc -shared -o ${LIB_DPI}.so
        xelab -svlog ${DIR_VERILOG}/file.sv -sv_lib ${LIB_DPI}
        xsim top -runall
```