Cheatography

Direct Programming Interface Cheat Sheet

by Bug Hunter (kamezian) via cheatography.com/126408/cs/24449/

Few things to remember

#include <svdpi.h> in the C	non context functions with output/inout instantiated more than once will crash
imported tasks always return a void value	context required for functions with output/input arguments
imported functions can retun a result or be a void function	
svLogic/svLogicVecVal are 4 states elements	
svLogicVecVal is 32 bits only	vectors >32 bits will require use of mda in C
SV int sized array will be passed seamlessly	Ex: int a[2][2] in SV is mapped with const int[2][2]
logic array mapped with const svLogicVecVal array	
unpacked struct will be redefined in the C like in the SV	

Pure Functions

pure functions can be removed or replaced by simulator to optimize objects, previous values computed for given argument values can be reused
pure functions have their result depending exclusively on their input arguments
ONLY non-void functions without output/inout can be pure
pure functions cannot perform fileoperations, read/write i/o,env variables, OS/program/process/shared memory objects, glocal/static variables

context methods

implicit scope for context methods	
SV methods from other scopes can be called after modifying current scope	
svSetScope to modify current scope	svGetScope to retrieve current scope
svGetNameFromScope	svGetScopeFromName

Open Array

dimension unspecified a[][]	Not a dynamic array!!
limited to single packed dimension	
C code access through query functions	svLeft, svRight, svLow,svHigh,svIncrement,svSize, svDimention
Access functions: svGetArrayPtr/ svSizeOfArray/ svGetArrElemPtr-{,1,2,3}	

Datatype Mapping

SV	C input	C output/inout	description
int	int	int*	int passed by value
reg/logic	svLogic	svLogic*	reg/logic passed by value
shortint	short int	short int*	shortint(16 bits) passed by value
longint	long int	long int*	longint (64bits) passed by value
real	double	double*	real passed by value
string	char*	char**	string passed by value

Complex Data type mapping

SV	C input	C output/inout	description
logic/reg[]	svLogicVecVal	svLogicVecVal*	logic vector passed by value
bit[]	const svBitVecVal*	svBitVecVal*	bit vector assed by value
open array(-import only)	const svOpenArrayHandle	svOpenArrayHandle	array passed by value
chandle	const void*	void*	allows C to allocate memory

Export SV Methods

Exported methods have only the routine name	No argument or return type!!
Map SV method name if it conflicts with exisiting C name :	in C use "extern" keyword
export "DPI-C" yoman = function hellosir;	extern void morning(int,int*)