SystemC DataTypes
Part III

Feb-9-2014

SystemVerilog Migration

Arbitrary Width Bit Type

sc_bit is one bit wide, SystemC provides sc_bv or SystemC Bit Vector for multi bit vector declaration. The width of the vector is specified in integer . The rightmost index of the vector is 0 and is also the least significant bit. A width of M sets the vector size and direction to be M-1 down to 0 with the M^{th} bit being the most significant bit.

If assignement to bit vector variable is less size (width), then rest of the bits are extened with 0. If they are more, then they are truncated.

The sc_bv type introduces some new operators and methods that perform bit reduction, range selection. Bit reductions and range selection can not be performed on port and signals.

Operator	Description	Usage
&	Bitwise AND	expr1 & expr2
|	Bitwise OR	expr1 | expr2
^	Bitwise XOR	expr1 ^ expr2
~	Bitwise NOT	~expr
&=	AND assignment	variable &= expr
|=	OR assignment	variable |= expr
^=	XOR assignment	variable ^= expr
=	Equality	expr1 = expr2
!=	Inequality	expr1 != expr2
[]	Bit selection	variable [index]
(,)	Concatenation	(expr1, expr2, expr3)

Methods in Arbitrary Width Bit Type

Methods	Description	Usage
range()	Range selection	variable.range(indexl, index2)
and_reduce()	Reduction AND	variable.and_reduce()

or_reduce()	Reduction OR	variable.or_reduce()
xor_reduce()	Reduction XOR	variable.xor_reduce()

Example : Arbitrary Width Bit Type

```
1  #include <systemc.h>
2
3  int sc_main (int argc, char* argv[]) {
4    sc_bv<8>  data_bus  ;
5    sc_bv<16> addr_bus  ;
6    sc_bit    parity    ;
7    // Assign value to sc_bv
8    data_bus = "00001011";
9    cout <<"Value of data_bus : " << data_bus << endl;
10   // Use range operator
11   addr_bus.range(7,0) = data_bus;
12   cout <<"Value of addr_bus : " << addr_bus << endl;
13   // Assign reverse to addr bus using range operator
14   addr_bus.range(0,7) = data_bus;
15   cout <<"Value of addr_bus : " << addr_bus << endl;
16   // Use bit select to set the value
17   addr_bus[10] = "1";
18   cout <<"Value of addr_bus : " << addr_bus << endl;
19   // Use reduction operator
20   parity  = data_bus.xor_reduce();
21   cout <<"Value of parity   : " << parity << endl;
22
23   return 1;
24 }
```

You could download file arb_width_bit_type.cpp here

Simulation : Arbitrary Width Bit Type

```
        SystemC 2.0.1 --- Oct  6 2006 19:17:37
    Copyright (c) 1996-2002 by all Contributors
                ALL RIGHTS RESERVED
Value of data_bus : 00001011
Value of addr_bus : 0000000000001011
Value of addr_bus : 0000000011010000
Value of addr_bus : 0000010011010000
Value of parity   : 1
```

Arbitrary Width Logic Type

sc_logic is one bit wide, SystemC provides sc_lv or SystemC Bit Vector for multi logic vector declaration. The width of the vector is specified in integer . The rightmost index of the vector is 0 and is also the least significant bit. A width of M sets the vector size and direction to be M-1 down to 0 with the M^{th} bit being the most significant bit.

If assignement to logic vector variable is less size (width), then rest of the bits are extened with 0. If they are more, then they are truncated.

The sc_lv type introduces some new operators and methods that perform bit reduction, range selection. Bit reductions and range selection can not be performed on port and signals.

No arthemetic operations are allowed on sc_lv types, instead they need to be performed on sc_int or sc_uint and then assigned to sc_lv.

Example : Arbitrary Width Logic Type

```
1  #include <systemc.h>
2
3  int sc_main (int argc, char* argv[]) {
4    sc_lv<8>  data_bus  (sc_logic ('z')); // All bits are Z
5    sc_lv<16> addr_bus  ; // All bits are X
6    sc_logic  parity    ;
7    // Print Default value of data_bus
8    cout <<"Value of data_bus : " << data_bus << endl;
9    // Assign value to sc_bv
10   data_bus = "00001011";
11   cout <<"Value of data_bus : " << data_bus << endl;
12   // Use range operator
13   addr_bus.range(7,0) = data_bus;
14   cout <<"Value of addr_bus : " << addr_bus << endl;
15   // Assign reverse to addr bus using range operator
16   addr_bus.range(0,7) = data_bus;
17   cout <<"Value of addr_bus : " << addr_bus << endl;
18   // Use bit select to set the value
19   addr_bus[10] = "1";
20   cout <<"Value of addr_bus : " << addr_bus << endl;
21   // Use reduction operator
22   parity  = data_bus.xor_reduce();
23   cout <<"Value of parity   : " << parity << endl;
24
25   return 1;
26 }
```

You could download file arb_width_logic_type.cpp here

Simulation : Arbitrary Width Logic Type

```
        SystemC 2.0.1 --- Oct  6 2006 19:17:37
    Copyright (c) 1996-2002 by all Contributors
                ALL RIGHTS RESERVED
Value of data_bus : ZZZZZZZZ
Value of data_bus : 00001011
Value of addr_bus : XXXXXXXX00001011
Value of addr_bus : XXXXXXXX11010000
```

```
Value of addr_bus : XXXXX1XX11010000
Value of parity   : 1
```

Do you have any Comment? mail me at:deepak@asic-world.com