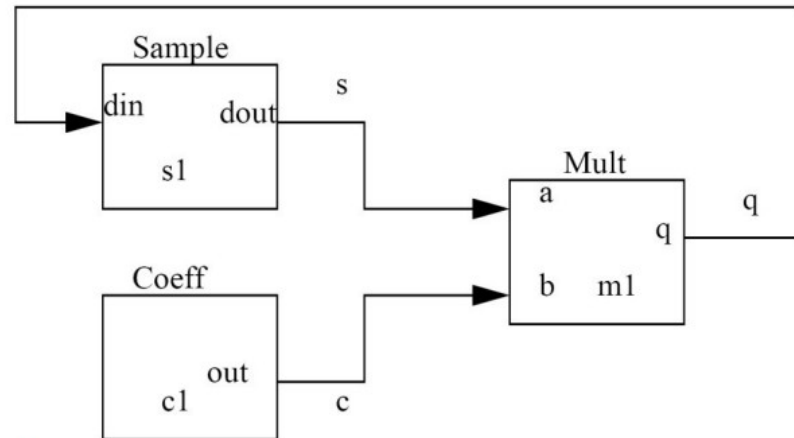

```
1 // mux2to1.h file
2 SC_MODULE( Mux2To1 ) {
3     sc_in< sc_uint<8> > in1;
4     sc_in< sc_uint<8> > in2;
5     sc_in< bool > selection;
6     sc_out< sc_uint<8> > out;
7
8     void muxImplement( void );
9     SC_CTOR( Mux2To1 ) {
10         SC_METHOD( muxImplement );
11         sensitive << selection;
12         sensitive << in1;
13         sensitive << in2;
14     }
15 }
```

```
1 // mux2to1.cpp file
2 void Mux2To1::muxImplement( void ) {
3     sc_uint<8> out_tmp;
4
5     if( selection.read() ) {
6         out_tmp = in2.read();
7     } else {
8         out_tmp = in1.read();
9     }
10    out.write( out_tmp );
11 }
```

Named and Positional Connections

```
SC_MODULE(filter) {  
    // Sub-modules: "components"  
    sample *s1;  
    coeff  *c1;  
    mult   *m1;  
    sc_signal<sc_uint <32> > q,s,c;  
    // Constructor : "architecture"  
    SC_CTOR(filter) {  
        //Sub-modules instantiation/mapping  
        s1 = new sample ("s1");  
        s1->din(q); // named mapping  
        s1->dout(s);  
        c1 = new coeff("c1");  
        c1->out(c); // named mapping  
        m1 = new mult ("m1");  
        (*m1)(s, c, q)//positional mapping  
    }  
}
```



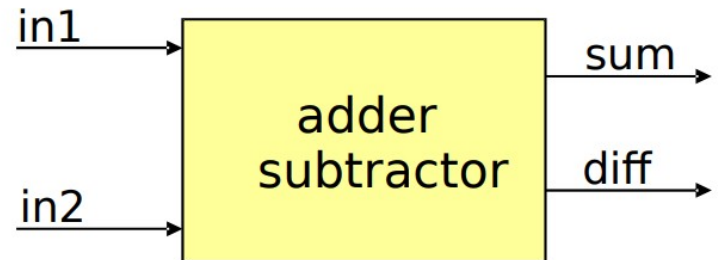
```

SC_MODULE(countsub)
{
    sc_in<double> in1;
    sc_in<double> in2;
    sc_out<double> sum;
    sc_out<double> diff;

    void addsub();

    // Constructor:
    SC_CTOR(countsub)
    {
        // declare addsub as SC_METHOD
        SC_METHOD(addsub);
        sensitive << in1;
        sensitive << in2;
    }
};

```



```

// addsub method
void countsub::addsub()
{
    double a;
    double b;
    a = in1.read();
    b = in2.read();
    sum.write(a+b);
    diff.write(a-b);
};

```

Usage of SystemC types

```
sc_bit y; sc_bv<8> x;  
y = x[6];
```

```
sc_bv<16> x; sc_bv<8> y;
```

```
y = x.range(0,7);
```

A diagram illustrating the range extraction operation. It features a long horizontal rectangle representing a 16-bit bus. A trapezoidal shape, wider on the left and tapering to the right, points from the left edge of the rectangle to the right edge of the code line 'y = x.range(0,7);'. The rectangle has a small notch on its left side where the trapezoid connects, and a small protrusion on its right side.

```
sc_bv<64> databus; sc_logic result;  
result = databus.or_reduce();
```

```
sc_lv<32> bus2;  
cout << "bus = " << bus2.to_string();
```

✦ 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 }
```

✦ 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 : 00000000000001011
Value of addr_bus : 0000000011010000
Value of addr_bus : 0000010011010000
Value of parity : 1