

هم طراحی سخت افزار نرم افزار

جلسه نهم: توصیف سیستم-زبان SystemC-۲

ارائه دهنده: آتنا عبدی

a_abdi@kntu.ac.ir

مباحث این بخش



- توصیف یک سیستم (System Specification)

- مدل‌های محاسباتی

- معماری‌ها

- زبان‌های توصیف

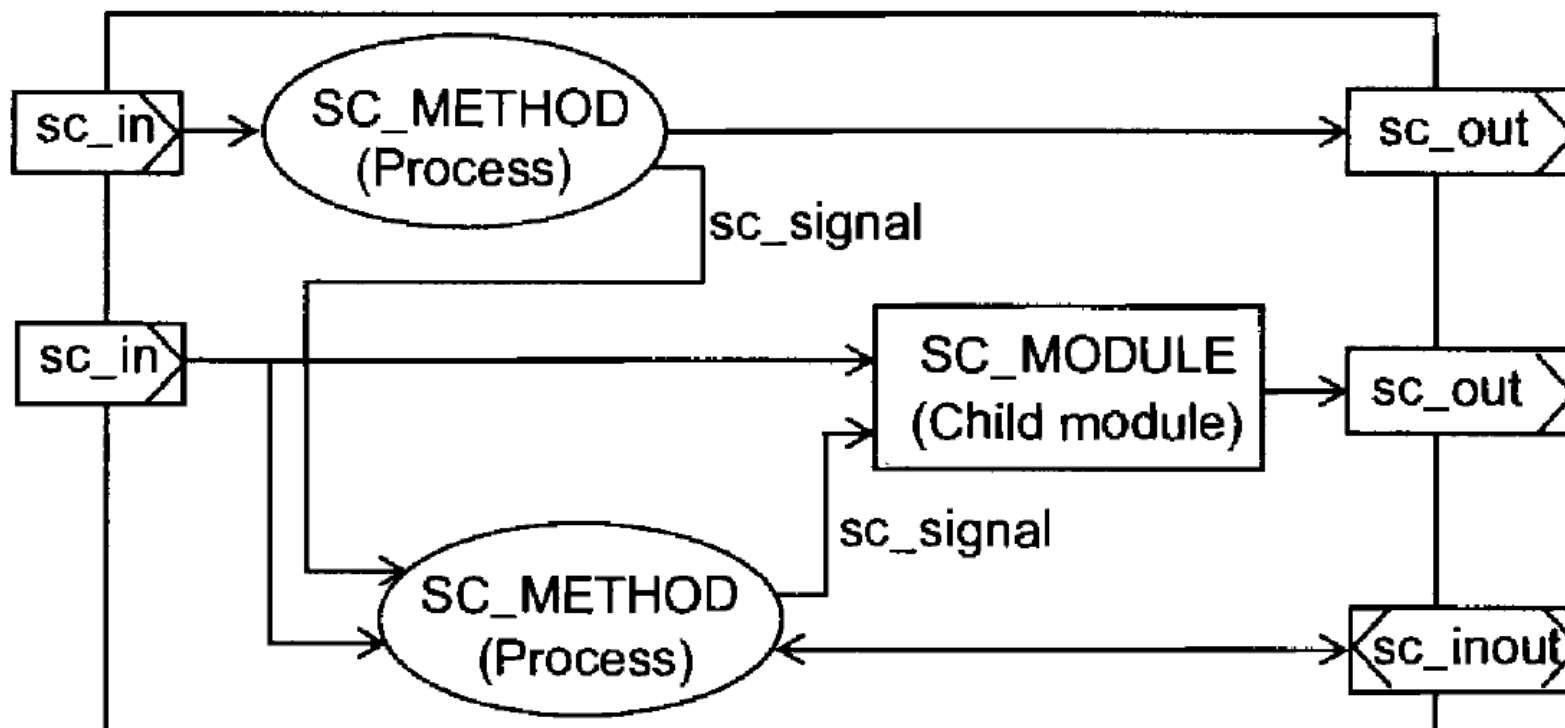
- آشنایی با زبان توصیف سیستم SystemC



ساختار ماژول و اجزای آن در SystemC



SC_MODULE



انواع داده در SystemC



• علاوه بر انواع داده موجود در C/C++ داده‌هایی نیز در کتابخانه SystemC تعریف شده‌اند:

• نوع تک‌بیتی

• نوع صحیح

• نوع بردار بیت (Bit-Vector)

انواع داده در SystemC



- نوع تک‌بیتی:

- `sc_bit`: منطق دو مقدار صفر و یک (`bool`)

- `sc_bit A; A = '1';`

- عملگرهای مجاز این نوع داده:

<code>&</code>	Bitwise AND	<code>&=</code>	Compound AND assignment
<code> </code>	Bitwise OR	<code> =</code>	Compound OR assignment
<code>^</code>	Bitwise XOR	<code>^=</code>	Compound XOR assignment
<code>~</code>	Bitwise NOT	<code>==</code>	Equality
<code>=</code>	Assignment	<code>!=</code>	Inequality

انواع داده در SystemC - تکبیتی



- `sc_logic`: منطق چهار مقدار (0,1,X,Z)

- `sc_logic A; A = 'Z';`

- `X=x` و `Z=z`

- عملگرهای مجاز این نوع داده:

- مشابه حالت `sc_bit`

- ترکیب انواع `bool`، `sc_bit` و `sc_logic` در عملیات ممکن است

- تبدیل‌ها انجام می‌شوند

انواع داده در SystemC (مثال)



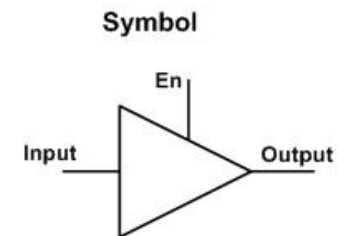
• پیاده‌سازی بافر سه‌حالت (Tri-state Buffer)

```
SC_MODULE(tristate_buf) {
    sc_in< sc_bit > input;
    sc_out< sc_logic > output;
    sc_in< sc_bit > enable;

    void process() {
        sc_bit in, en;
        sc_logic out;

        in=input; en=enable; // reading inputs to temporary variables
        if(en)
            out = in;
        else
            out = 'z';
        output = out; // writing a temporary variable to output
    }

    SC_CTOR(tristate_buf) {
        SC_METHOD(process);
        sensitive<<enable<<input;
    }
};
```



Truth Table

En	Input	Output
0	X	Hi-Z
1	0	0
1	1	1

انواع داده در SystemC-نوع صحیح



- اعداد صحیح با طول ثابت (Fixed):
 - `sc_int`: عدد صحیح با حداکثر سائز ۶۴ بیت و علامت‌دار (مکمل ۲)
 - `sc_uint`: عدد صحیح بدون علامت با حداکثر سائز ۶۴ بیت
 - `sc_int/sc_uint <25> x` (عدد ۲۵ بیتی صحیح x)
- اعداد صحیح با طول دلخواه (Arbitrary):
 - `sc_bigint/sc_bigunit`: عدد صحیح با/بی علامت با سائز بزرگتر از ۶۴ بیت
- انواع داده‌های اعداد صحیح با نوع متناظرشان در C++ قابل استفاده همزمان هستند

انواع داده در SystemC



• عملگرهای مجاز در نوع داده صحیح:

Bitwise	~	&		^	>>	<<
Arithmetic	+	-	*	/	%	
Assignment	+=	-=	*=	/=	%=	
		&=	=	^=	=	
Equality	==	!=				
Relational	<	<=	>	>=		
Auto-inc/dec	++	--				
Bit Select	[]					
Part Select	.range(,)					
Concatenation	(,)					

انواع داده در SystemC (مثال)



```
sc_logic mybit;  
sc_uint<8> myuint;  
mybit = myuint[7];  
  
sc_uint<4> myrange;  
myrange = myuint.range(5,2);  
  
sc_uint<12> my12int;  
my12int = (myuint, myrange);
```

```
sc_int<5> a;  
a = 13; // a gets 01101,  
a[4] = 0, a[3] = 1, ..., a[0] = 1  
bool b;  
b = a[4]; // b gets 0  
c = a.range(3, 1); // c gets 110 - interpreted as -2
```

انواع داده در SystemC-نوع آرایه



- آرایه بیتی (Bit Vector/ Logic Vector)

- `sc_bv`: آرایه‌ای از بیت‌ها در منطق دو مقداری / `sc_lv`: آرایه‌ای از بیت‌ها در منطق چهار مقداری

- مقداردهی: رشته‌های صفر و یک

- عدم امکان اعمال عملیات محاسباتی (تبدیل به `int` و بازگشت بعد از محاسبات با عملگر “=“)

- عملگرهای مجاز:


Bitwise	~	&		^	>>	<<
Assignment	=	&=	=	^=		
Equality	==	!=				
Bit Select	[]					
Part Select	.range(,)					
Concatenation	(,)					
Reduction	and_reduce() xor_reduce()	or_reduce()				

انواع داده در SystemC (مثال)



```
#include <systemc.h>
```

```
int sc_main (int argc, char* argv[]) {  
    sc_lv<8> data_bus (sc_logic ('z')); // All bits are Z  
    sc_lv<16> addr_bus ; // All bits are X  
    sc_logic parity ;  
    // Print Default value of data_bus  
    cout << "Value of data_bus : " << data_bus << endl;  
    // Assign value to sc_bv  
    data_bus = "00001011";  
    cout << "Value of data_bus : " << data_bus << endl;  
    // Use range operator  
    addr_bus.range(7,0) = data_bus;  
    cout << "Value of addr_bus : " << addr_bus << endl;  
    // Assign reverse to addr bus using range operator  
    addr_bus.range(0,7) = data_bus;  
    cout << "Value of addr_bus : " << addr_bus << endl;  
    // Use bit select to set the value  
    addr_bus[10] = "1";  
    cout << "Value of addr_bus : " << addr_bus << endl;  
    // Use reduction operator  
    parity = data_bus.xor_reduce();  
    cout << "Value of parity : " << parity << endl;  
  
    return 1;  
}
```



```
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
```

انواع Process از لحاظ اجرا



• Method (sc_method)

- یکبار صدا زده می‌شود و پس از فراخوانی امکان متوقف کردن آن وجود ندارد و تا پایان اجرا می‌شود
- مناسب برای طراحی بلوک‌های منطق ترکیبی مدار (محدود به این حالت نمی‌باشد)

• Thread (sc_thread)

- قابلیت متوقف شدن با دستور wait() را دارد و با فعال شدن لیست حساسیت، پروسه دوباره فعال می‌شود

• Clock Thread (sc_cthread)

- مشابه حالت قبل است و فقط حساس به لبه یک کلاک است و نیاز به تعریف لیست حساسیت نیست
- `sc_cthread (name,clock.pos());`
- متوقف شدن با `wait_until (<signal condition>)` و پس از پایان شرط بولی، پروسه دوباره فعال می‌شود

کوئیز کلاسی



• پیاده‌سازی گیت XOR با استفاده از NAND در SystemC ؟

• فرض کنید مدل پیاده‌سازی شده گیت NAND را داریم

• زمان حل: ۱۰ دقیقه

• ارسال عکس از طریق کوئرا (+۵ دقیقه)

