بنام خداوند جان و خرد

هم‌طراحی سخت‌افزار نرم‌افزار

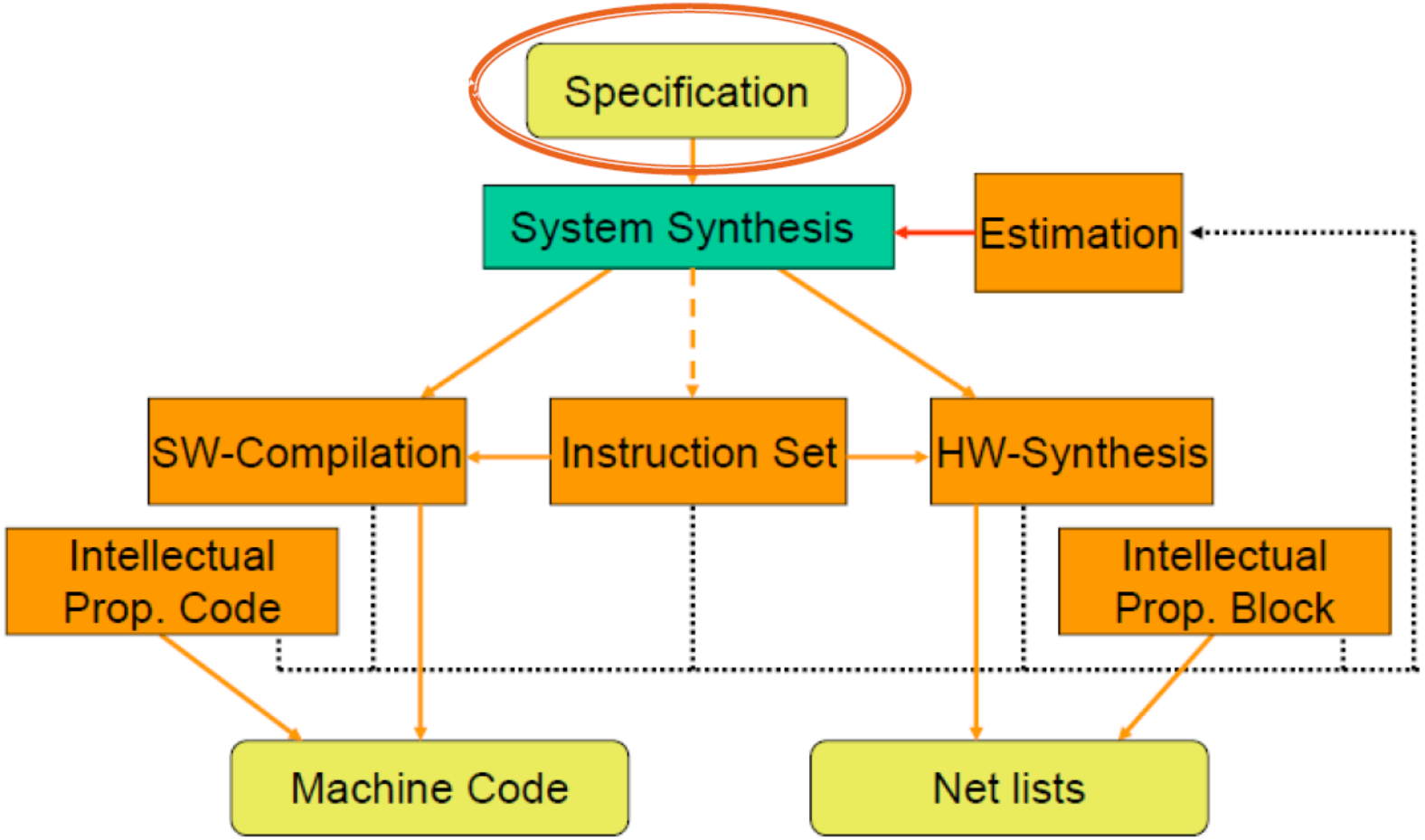
جلسه هشتم: توصیف سیستم-زبان SystemC

ارائه‌دهنده: آتنا عبدی
a_abdi@kntu.ac.ir

مباحث این بخش

- توصیف یک سیستم (System Specification)
- مدل‌های محاسباتی
- معماری‌ها
- زبان‌های توصیف
  - اشنایی با زبان توصیف سیستم SystemC

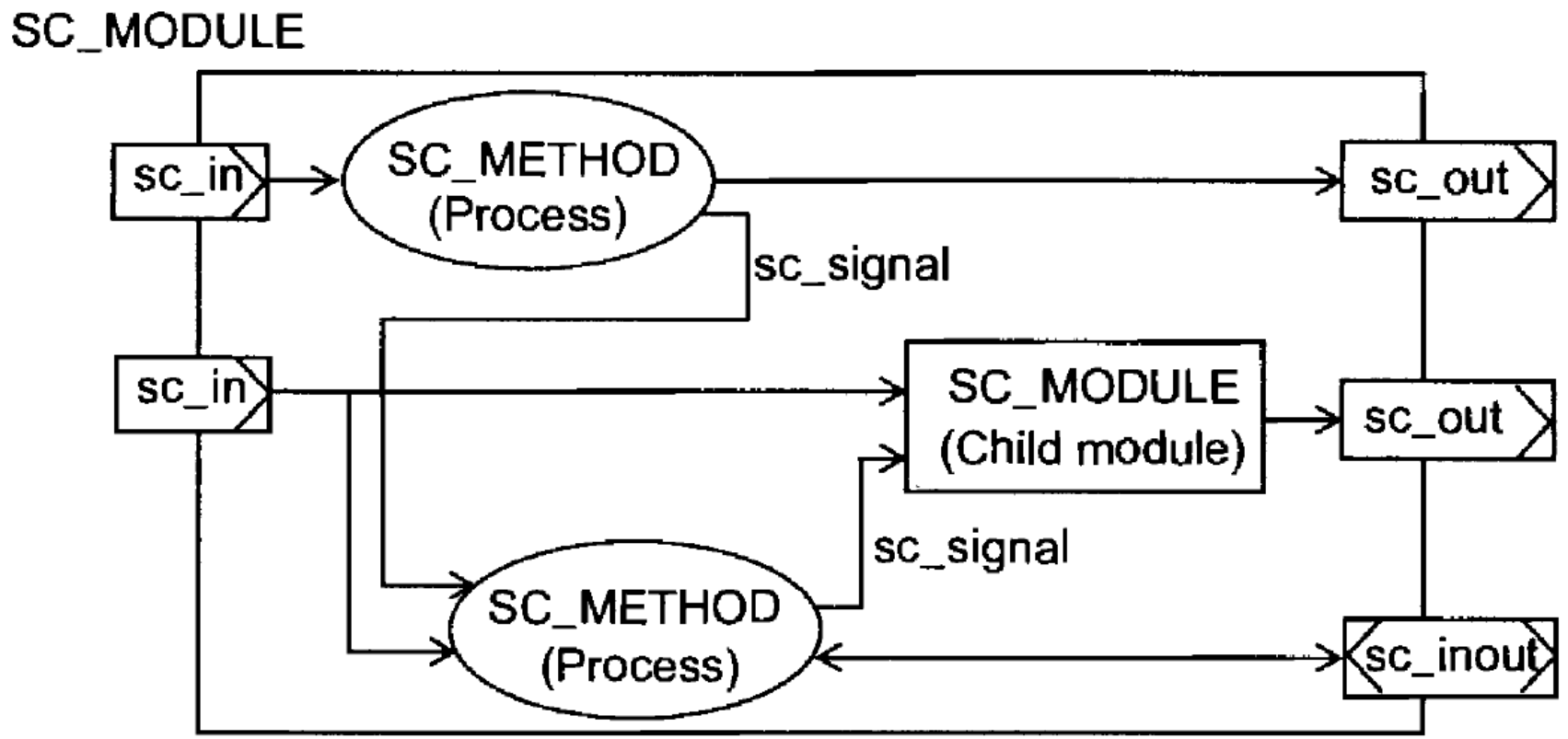
توصیف سیستم


Specification
System Synthesis
Estimation
SW-Compilation
Instruction Set
HW-Synthesis
Intellectual Prop. Code
Intellectual Prop. Block
Machine Code
Net lists

SystemC

- یکپارچگی توصیف سخت‌افزار و نرم‌افزار در سطح سیستم
  - نیاز به محیطی جهت توصیف هم‌زمان سخت‌افزار و نرم‌افزار
  - افزودن قابلیت‌های لازم در توصیف سخت‌افزار به زبان‌های برنامه‌نویسی
    - زمان‌بندی بین رخدادها، همروندی، نوع داده‌های خاص مانند Z
    - زبان SystemC: HDL + C++

ساختار ماژول و اجزای آن در SystemC



ساختار برنامه‌نویسی پایه در SystemC

```
SC_MODULE( <module_name> ) {
  // declaring port types
  sc_in<int> in;

  // definition of processes
  void func1() {
        // circuit functionality
   }
   void func2() {
        // functionality
   }


  SC_CTOR( <module_name> ) {
    // declaring processes
        SC_METHOD(func1);
        sensitive<<in;
  }
};
```

تعریف سیستم در قالب یک ماژول

تعریف کارکرد سیستم در قالب تابع (تابع C++)

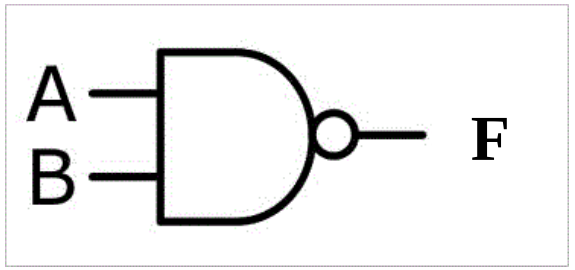
ایجاد کننده (constructor) ماژول

معرفی تابع به‌عنوان پروسه در ماژول و مشخص کردن لیست حساسیت

ساختار برنامه‌نویسی پایه در SystemC (مثال)

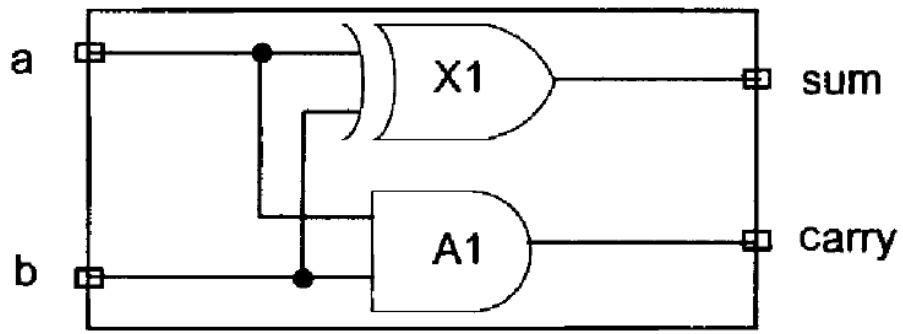
- پیاده‌سازی گیت NAND

```
#include "systemc.h"
SC_MODULE(nand2)
{
sc_in<bool> A, B;
sc_out<bool> F;
void do_nand2()
{
F.write( !(A.read() && B.read()) );
}
SC_CTOR(nand2)
{
SC_METHOD(do_nand2);
sensitive << A << B;
}
};
```



ساختار برنامه‌نویسی پایه در SystemC (مثال)

- پیاده‌سازی نیم‌جمع‌کننده




```
// File: half_adder.h
#include "systemc.h"

SC_MODULE (half_adder) {
  sc_in<bool> a, b;
  sc_out<bool> sum, carry;

  void prc_half_adder ();

  SC_CTOR (half_adder) {
    SC_METHOD (prc_half_adder);
    sensitive << a << b;
  }
};
```

```
// File: half_adder.cpp
#include "half_adder.h"

void half_adder::prc_half_adder () {
  sum = a ^ b;
  carry = a & b;
}
```

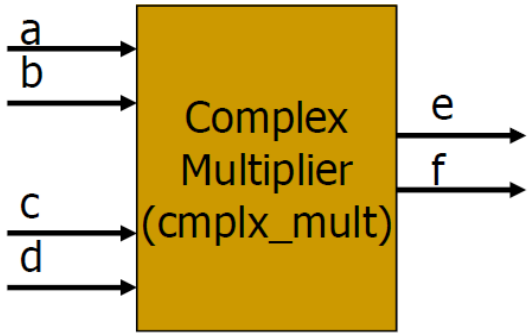
ساختار برنامه‌نویسی پایه در SystemC (مثال)

- ضرب دو عدد مختلط

$$(a+bi)*(c+di) = (ac-bd)+(ad+bc)i$$

```
SC_MODULE(cmplx_mult) {
   sc_in<int> a,b;
   sc_in<int> c,d;

   sc_out<int> e,f;

 void calc();

 SC_CTOR(cmplx_mult) {
   SC_METHOD(calc);
   sensitive<<a<<b<<c<<d;

 }

};
```

```
void cmplx_mult::calc()

{
   e = a*c-b*d;
   f = a*d+b*c;

}
```



ساختار برنامه‌نویسی پایه در SystemC

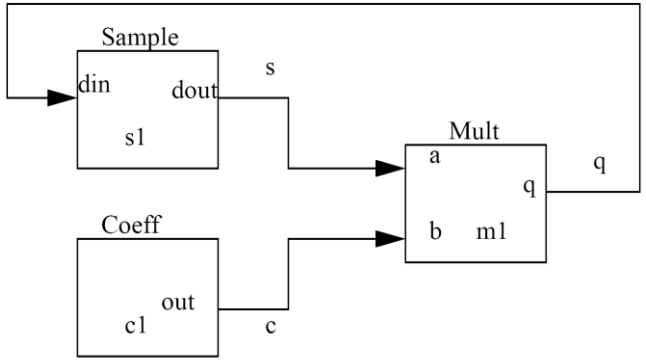
- توصیف سیستم با زیرماژول (سلسله‌مراتب)
- بحث انواع اتصال ماژول‌ها و فراخوانی و مقداردهی سیگنال‌ها

```
Instancename.portname (signalname);
Instancename->portname (signalname);
```

- اتصال اسمی (Named Connection)

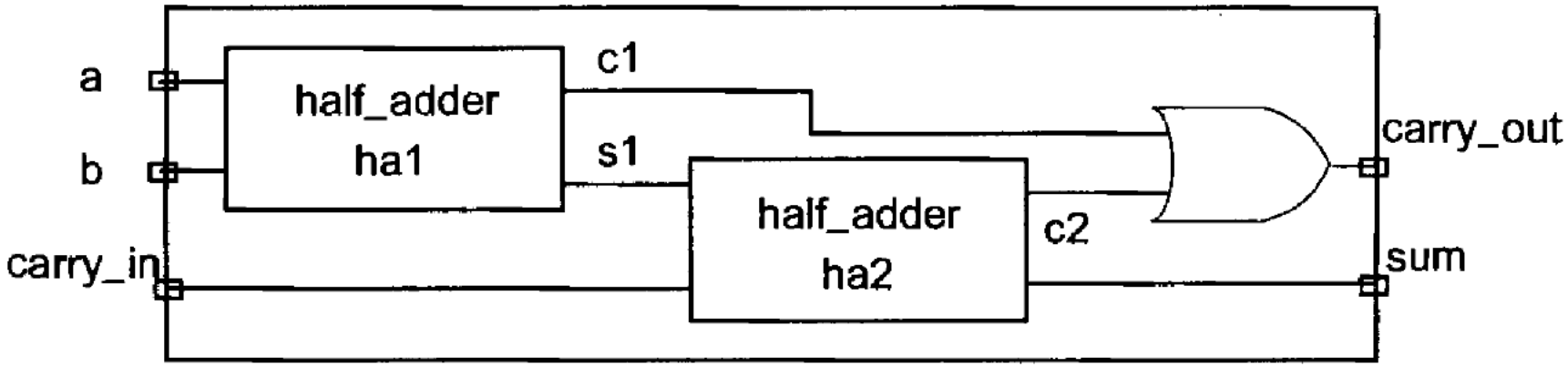
```
Instancename (sig1, sig2, …);
```

- اتصال مکانی (Positional Connection)



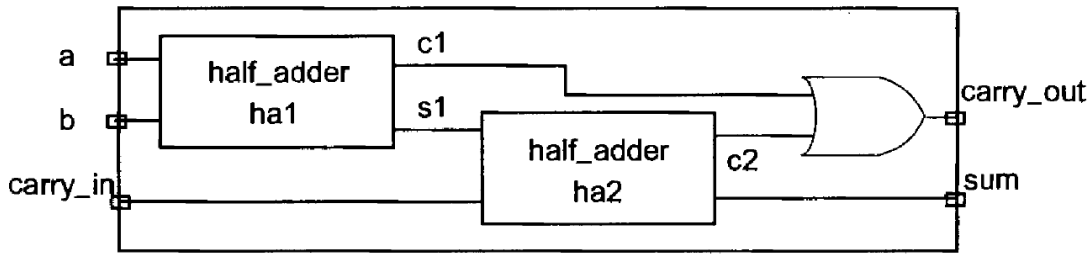
ساختار برنامه‌نویسی پایه در SystemC (مثال)

- توصیف سیستم با زیرماژول (سلسله‌مراتب)
- ساخت یک تمام‌جمع‌کننده با استفاده از دو نیم‌جمع‌کننده



ساختار برنامه‌نویسی پایه در SystemC (مثال)

```
// File: full_adder.h
#include "half_adder.h"

SC_MODULE (full_adder) {
  sc_in<bool> a, b, carry_in;
  sc_out<bool> sum, carry_out;

  sc_signal<bool> c1, s1, c2;
  void prc_or ();
  half_adder *ha1_ptr, *ha2_ptr;

  SC_CTOR (full_adder) {
    ha1_ptr = new half_adder ("ha1");
    // Named association:
    ha1_ptr->a (a);
    ha1_ptr->b (b);
    ha1_ptr->sum (s1);
```




```
    ha1_ptr->carry (c1);

    ha2_ptr = new half_adder ("ha2");
    // Positional association:
    (*ha2_ptr) (s1, carry_in, sum, c2);

    SC_METHOD (prc_or);
    sensitive << c1 << c2;
  }
};
```

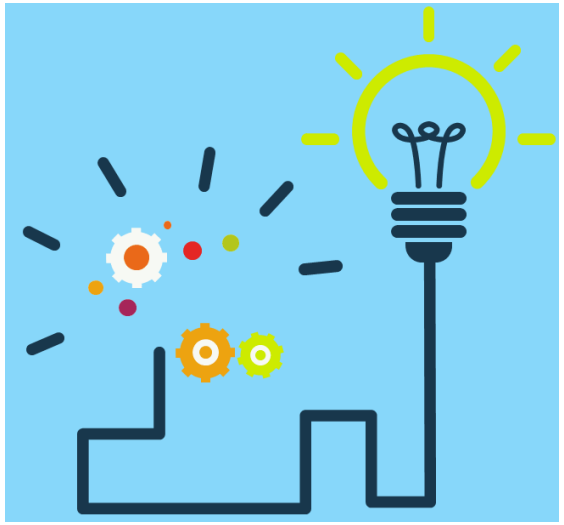
```
// File: full_adder.cpp
#include "full_adder.h"

void full_adder::prc_or () {
  carry_out = c1 | c2;
}
```

ساختار برنامه‌نویسی پایه در SystemC

- هر پروسه لیست حساسیت دارد:
  - حساسیت به تغییر مقادیر پورت‌ها یا سیگنال‌ها
    - `sensitive << port_1 << signal_1 << ... << port_n`
  - حساسیت به لبه بالارونده یک پورت یا سیگنال
    - `sensitive_pos << port_1 << signal_1;`
  - حساسیت به لبه پایین‌رونده یک پورت یا سیگنال
    - `sensitive_neg << port_1 << signal_1;`

مباحثی که این جلسه آموختیم

- توصیف سیستم و زبان
- آشنایی با مقدمات زبان SystemC
- ساختار برنامه‌نویسی و مدل برنامه
- ساده و سلسله‌مراتبی



مباحث جلسه آینده

- توصیف سیستم (زبان)
- آشنایی بیشتر با زبان SystemC
- انواع پروسه و انواع داده