

# هم طراحی سخت افزار نرم افزار

جلسه هفتم: توصیف سیستم-زبان

ارائه دهنده: آتنا عبدی

[a\\_abdi@kntu.ac.ir](mailto:a_abdi@kntu.ac.ir)

# مباحث این بخش



- توصیف یک سیستم (System Specification)

- مدل‌های محاسباتی

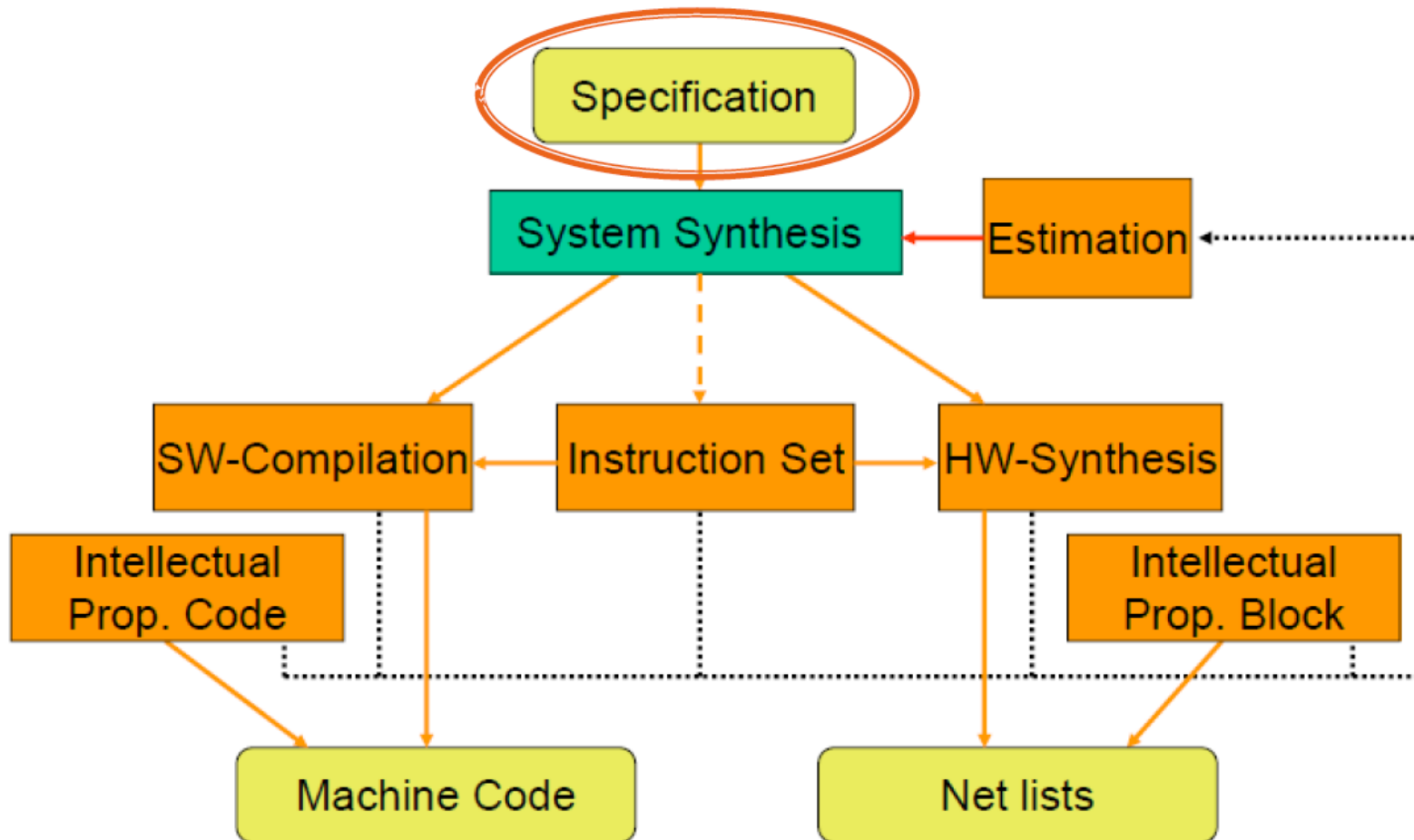
- معماری‌ها

- زبان‌های توصیف

- آشنایی با زبان توصیف سیستم SystemC



# توصیف سیستم



# توصیف سیستم



- در ابتدای فرایند طراحی لازم است سیستم، براساس الزامات آن توصیف شود
- فرایند توصیف سیستم توسط مدل‌ها، معماری‌ها و زبان‌ها انجام می‌گیرد
- **مدل:** دید مفهومی از رفتار و عملکرد سیستم (ساختار داده و کنترل)
- **معماری:** پیاده‌سازی کلی مدل با مشخص کردن نوع اجزای موردنیاز، تعداد آن‌ها، اتصالات و ....
- **زبان:** نگاشت مدل محاسباتی به معماری سیستم در سطح
  - سخت‌افزار: Verilog ، VHDL
  - نرم‌افزار: C ، C++ ، Java
  - سیستم: SystemC ، SDL



- هدف از زبان، نگاشت مدل محاسباتی به معماری می باشد
  - توصیف کارکرد سیستم در سطوح مختلف تجرید
  - ارزیابی و اصلاح کارکرد سیستم در حین توصیف
- زبان توصیف مناسب می بایست تمامی ویژگی های مدل را در نظر بگیرد
  - نگاشت متناظر بین ویژگی های مدل و اجزای زبان
  - همروندی، گذار بین حالات، سلسله مراتب، مدیریت شرایط خاص، زمان بندی، ارتباطات و ...



- زبان‌های توصیف سخت‌افزار:

- Verilog ، VHDL

- زبان‌های برنامه‌نویسی نرم‌افزار:

- Java ، C++ ، C

- زبان‌های توصیف سیستم:

- SLDL ، SDL ، **SystemC**

- زبان‌های ارزیابی

- OpenVERA ، PSL

# زبان‌های توصیف سیستم

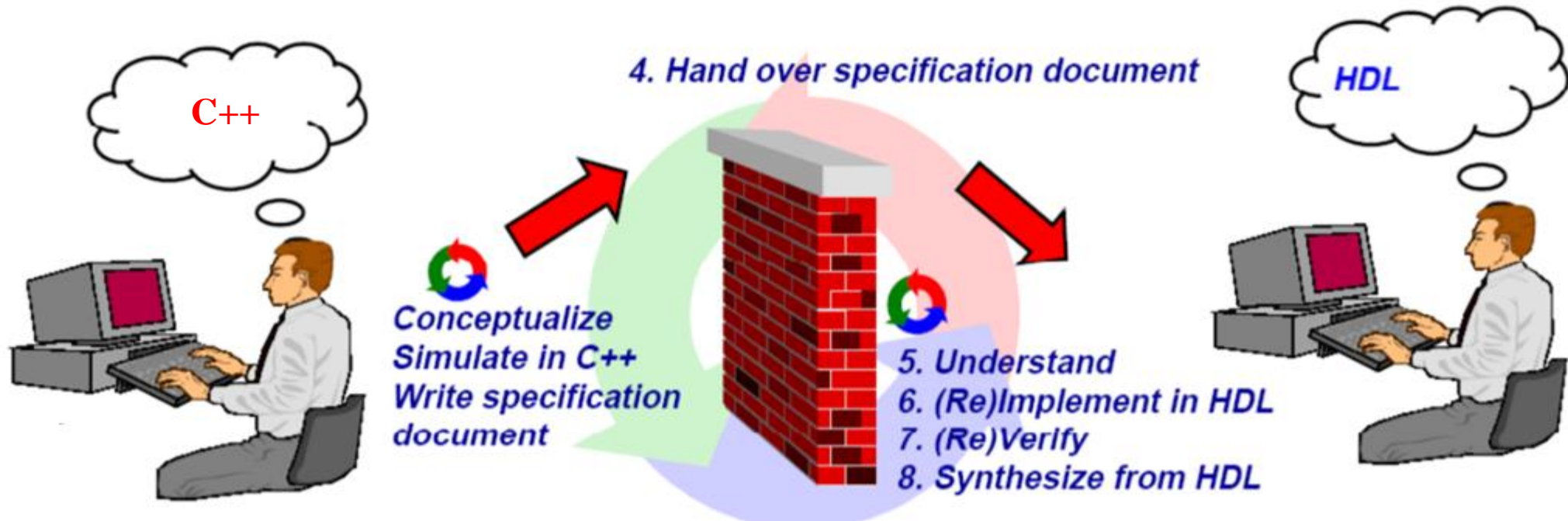


- مدلسازی سخت‌افزار و نرم‌افزار در طراحی سطح سیستم
- قابلیت پیاده‌سازی
- الگوریتم‌های نرم‌افزاری
- معماری‌های سخت‌افزاری
- زمان‌بندی سخت‌افزاری، هم‌روندی، رفتار واکنشی و ...
- واسط‌های طراحی سطح سیستم

# زبان‌های توصیف سیستم



- تفکیک توصیف سخت‌افزار و نرم‌افزار

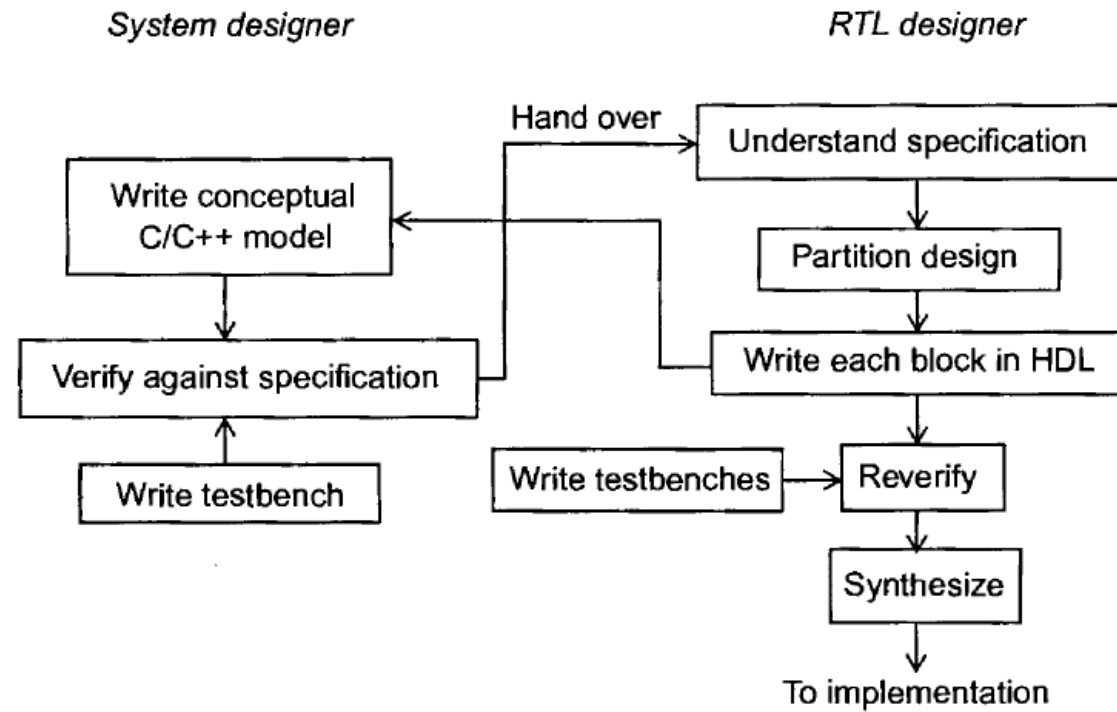




# زبان‌های توصیف سیستم



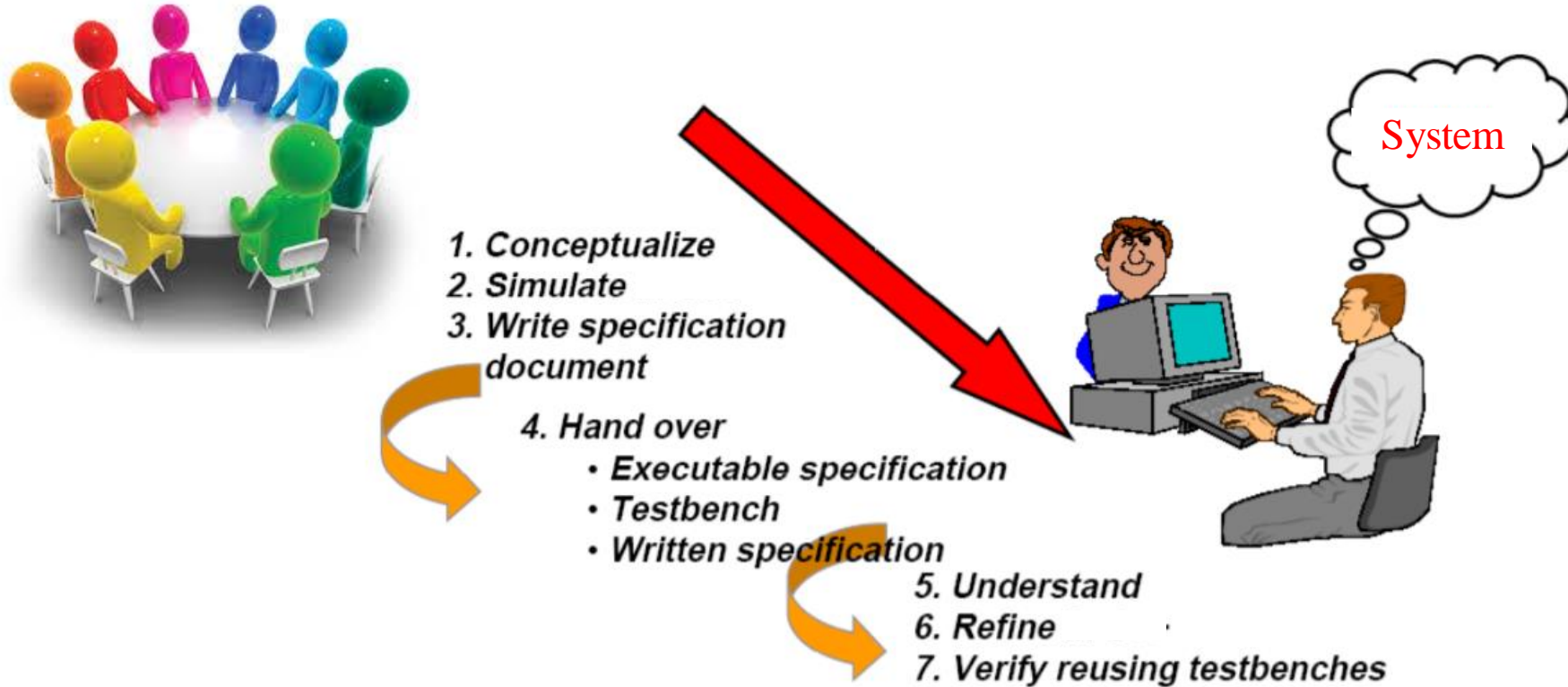
- تفکیک توصیف سخت‌افزار و نرم‌افزار



# زبان‌های توصیف سیستم



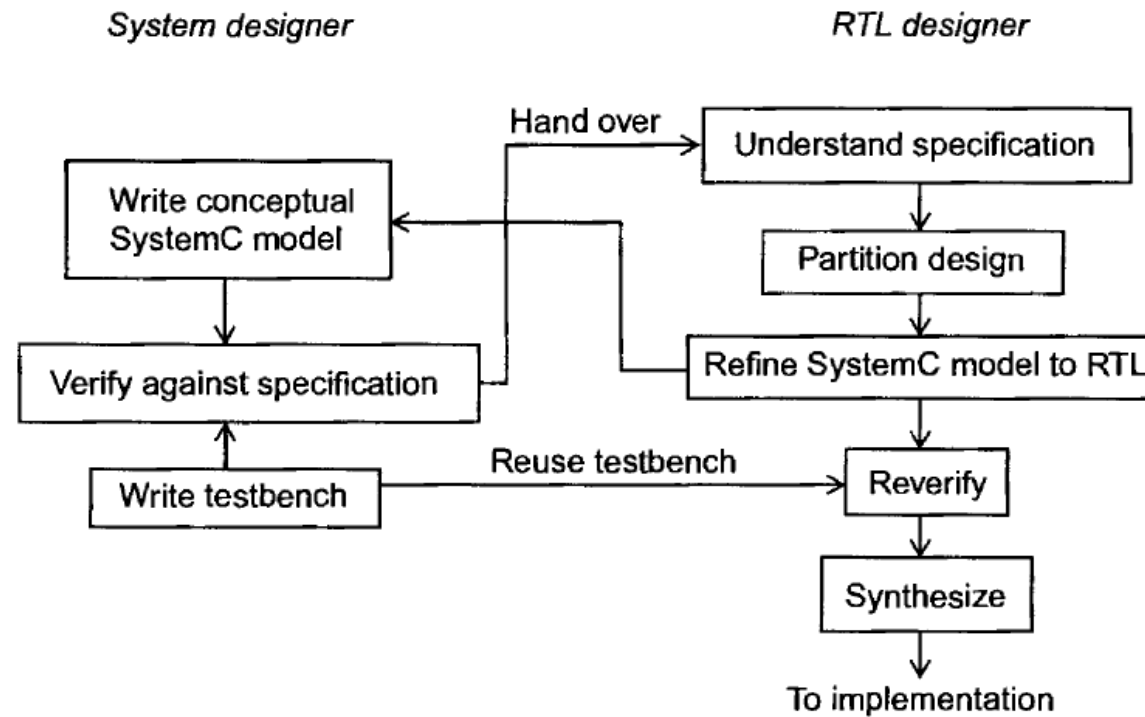
- یکپارچگی توصیف سخت‌افزار و نرم‌افزار در سطح سیستم



# زبان‌های توصیف سیستم



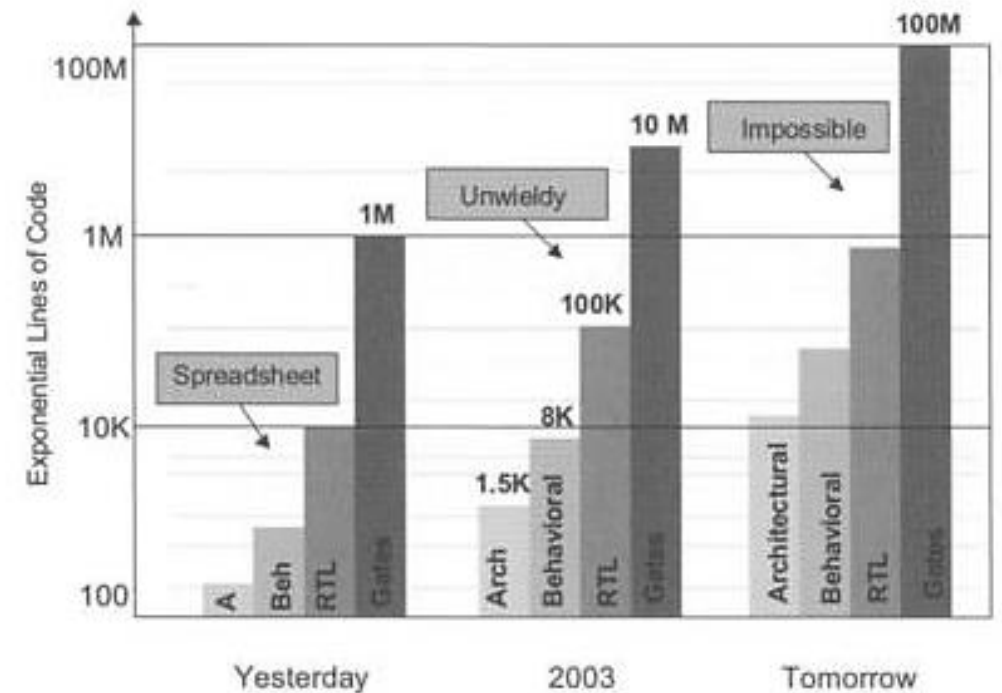
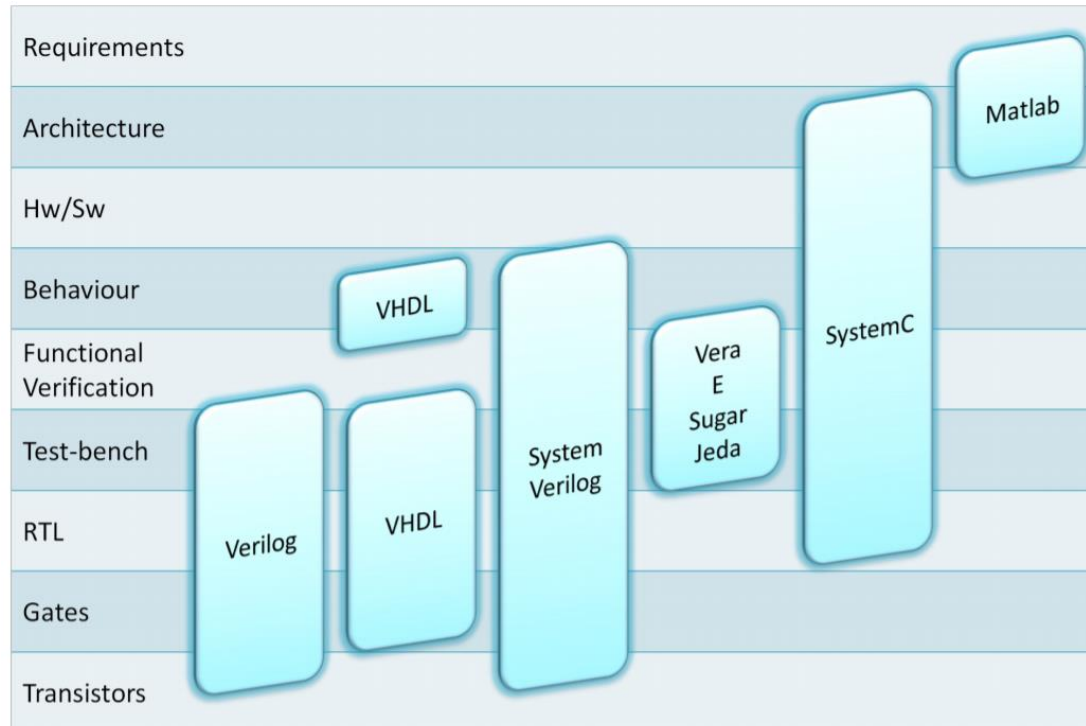
- یکپارچگی توصیف سخت‌افزار و نرم‌افزار در سطح سیستم



# توصیف سیستم در سطوح مختلف تجرید



- قابلیت‌های مختلف توصیف در سطوح مختلف تجرید
- با پیچیده شدن کاربردها، روال حرکت به سمت سطوح بالا



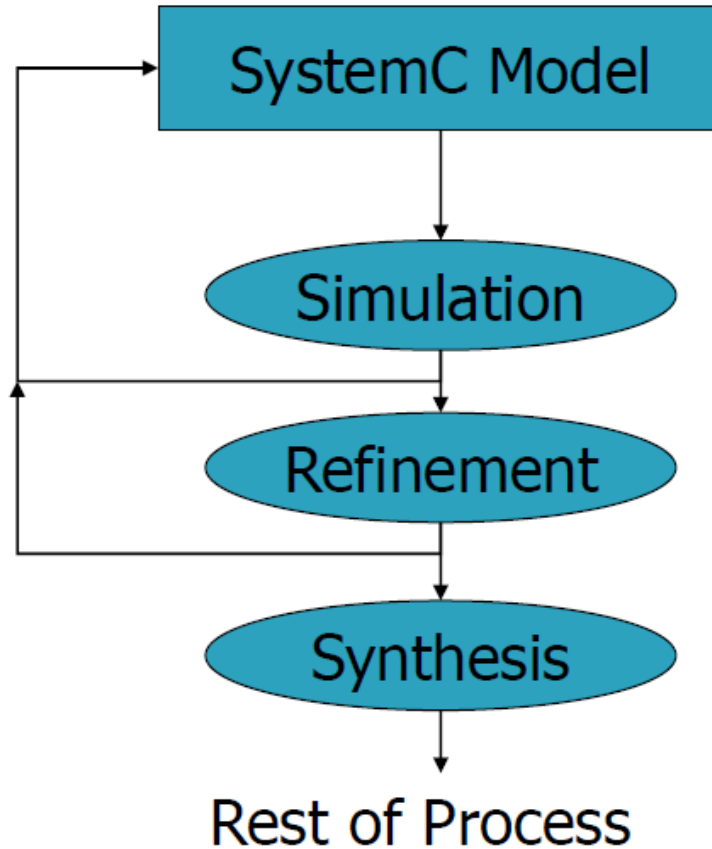
# SystemC



- 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: C++ + HDL
- یک class library در زبان C



# روال طراحی در SystemC



- طراحی یکپارچه سخت افزار و نرم افزار

- نیاز به یک زبان برنامه نویسی

- پیاده سازی سطح بالای مختصر که حجم کمتر و مدلسازی ساده تر دارد

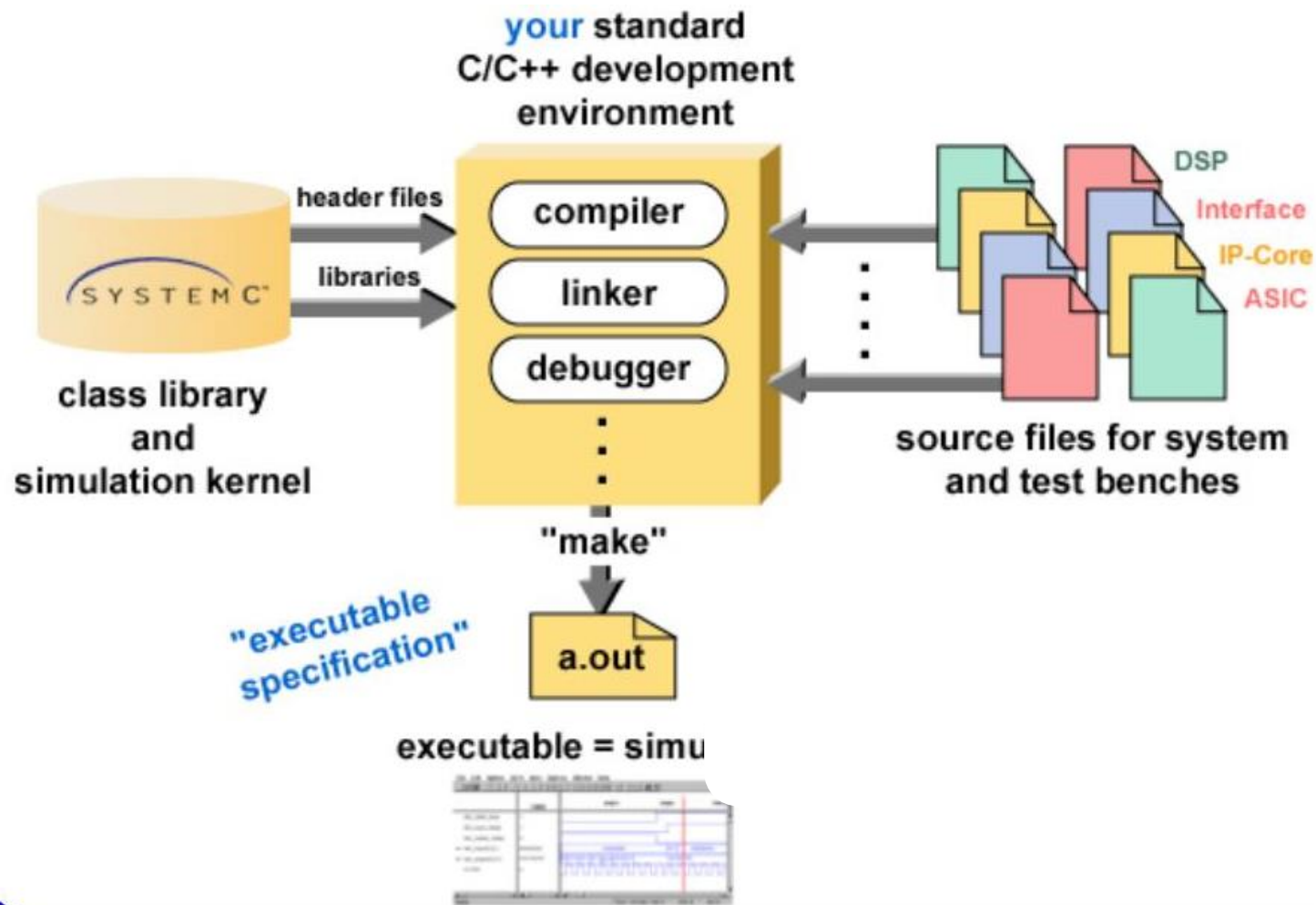
- توصیف در مراحل کوچک، بهبود داده می شود

- افزودن تدریجی قابلیت های مورد نیاز مانند هم رندی، زمان بندی و ...

- پیاده سازی ساده تغییرات

- تست تدریجی و کشف دقیق تر اشکالات

# محیط توسعه SystemC



# قابلیت‌های SystemC



- Modules: ساختار هر سیستم توسط یک یا چند ماژول توصیف می‌شود
- Processes: داخل ماژول تعریف شده و کارکرد را توصیف می‌کند
- Ports: هر ماژول پورت‌هایی برای اتصال به سایر ماژول‌ها دارد (پورت‌های یک‌طرفه/دوطرفه)
- Signals: انتقال داده برای اتصال ماژول‌ها و پروسه‌ها
- پشتیبانی از انواع زیاد داده‌ها
- منطق دو مقداره، منطق چهارمقداره، تمامی داده‌های موجود در C++



# قابلیت‌های SystemC



- Clock: امکان تعریف سیگنال کلاک
- مدلسازی در چندین سطح تجرید: مدل‌های کارکردی سطح بالا تا توصیف‌های جزئی
- پروتکل‌های ارتباطی متنوع: توصیف واسط‌ها و پروتکل‌ها در سطوح مختلف تجرید
- امکان Debug برخط
- ارائه شکل موج سیگنال‌ها و ...

# مدل برنامه‌نویسی در SystemC



- زبان SystemC همان C++ است
- مرور بر syntax کلی زبان C++
- تمامی عبارات C++ در اینجا مجاز است ( Cout, Cin و ... )
- تمامی کامپایلرهای C++ برای SystemC هم قابل استفاده است
  - MS VC++
  - GCC
  - دانلود از
- <https://www.accellera.org/downloads/standards/systemc>

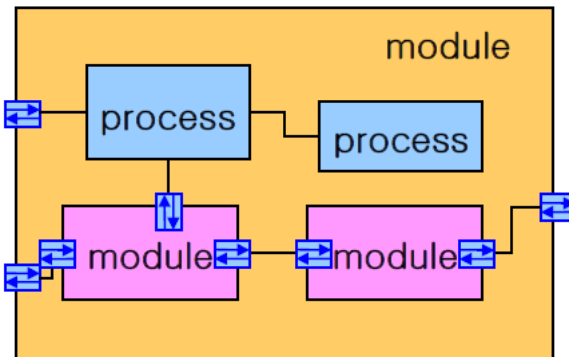
# مدل برنامه‌نویسی در SystemC



## Module •

- مفهومی مشابه module در Verilog دارد که در واقع یک کلاس C++ است
- داده‌های داخلی و الگوریتم‌ها داخل این واحد قرار داده می‌شوند
- بکارگیری:

SC\_MODULE(module\_name){ •



# مدل برنامه‌نویسی در SystemC



## • Process

- تعریف تابع در ماژول
- معادل Function در C++
- پروسه‌ها به صورت هم‌روند اجرا می‌شوند
- کارکرد موردنظر (functionality) در این بخش تعریف می‌شود
- دو نوع method و thread دارد
- هر پروسه لیستی از رخدادها دارد که به آن‌ها حساس است (Sensitivity List)
- با تغییر در این رخدادها، تابع پروسه اجرا می‌شود

# مدل برنامه‌نویسی در SystemC



- Ports: برقراری ارتباط ماژول با محیط

- به سه صورت تعریف می‌شوند:

- IN, OUT, INOUT

- مشابه Verilog در ابتدای ماژول تعریف می‌شود

```
#include "systemc.h"
```

```
SC_MODULE (fifo) {
```

```
    sc_in<bool> load;
```

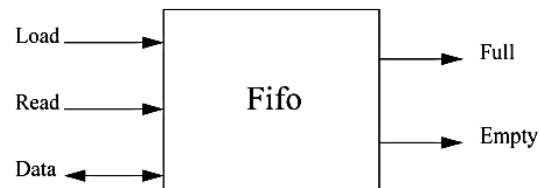
```
    sc_in<bool> read;
```

```
    sc_inout<int> data;
```

```
    sc_out<bool> full;
```

```
    sc_out<bool> empty;
```

```
}
```



- `sc_in <type> port_name`

- `sc_out <type> port_name`

- `sc_inout <type> port_name`

# مدل برنامه‌نویسی در SystemC



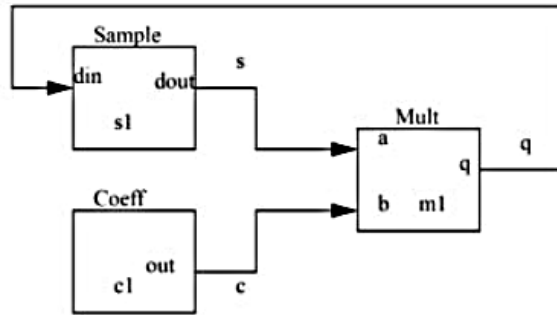
## • Signals

- اتصالات محلی بین اجزای داخلی سیستم جهت انتقال داده
- تخصیص در نمونه‌سازی از ماژول‌ها (module instantiation)
- توصیف به صورت

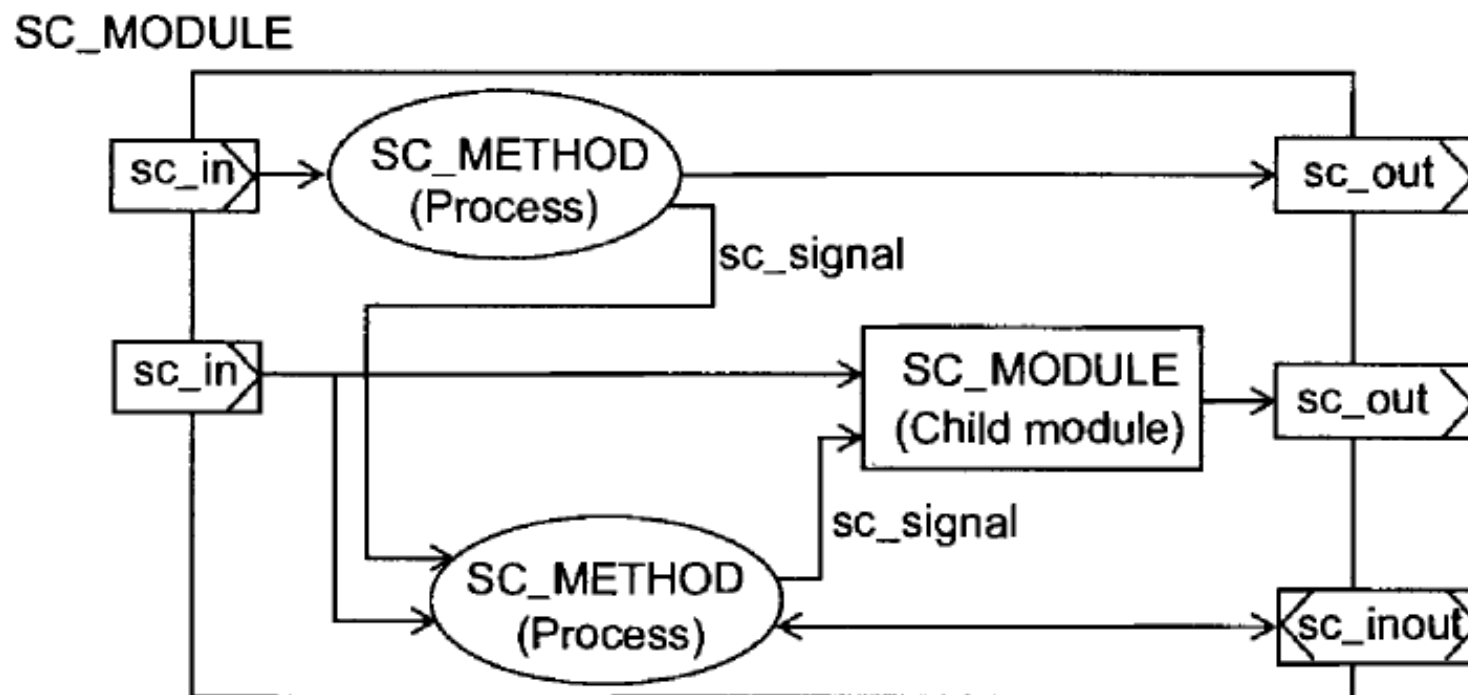
• `sc_signal <type> signal_name`

## • Data Type

- تمامی انواع موجود در C/C++، تمامی انواع موردنیاز در توصیف سیستم، امکان تعریف موارد موردنیاز



# ساختار ماژول و اجزای آن در SystemC



# مباحثی که این جلسه آموختیم



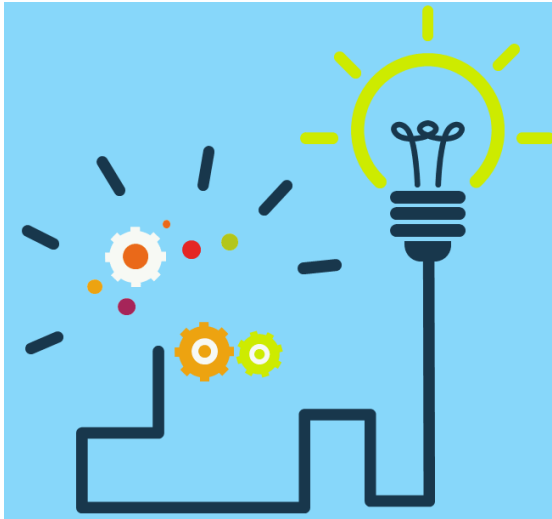
- توصیف سیستم

- زبان

- اهمیت استفاده از زبان در فرایند توصیف

- توصیف مجزا و یکپارچه سیستم

- زبان SystemC بعنوان زبان یکپارچه سطح سیستم





# مباحث جلسه آینده



- گام اول فرایند طراحی

- توصیف سیستم

- آشنایی بیشتر با زبان SystemC و توصیف مدل در این زبان

