بنام خداوند جان و خرد

هم‌طراحی سخت‌افزار نرم‌افزار

جلسه بیست‌وهشتم: واسط‌های ارتباطی

ارائه‌دهنده: آتنا عبدی
a_abdi@kntu.ac.ir

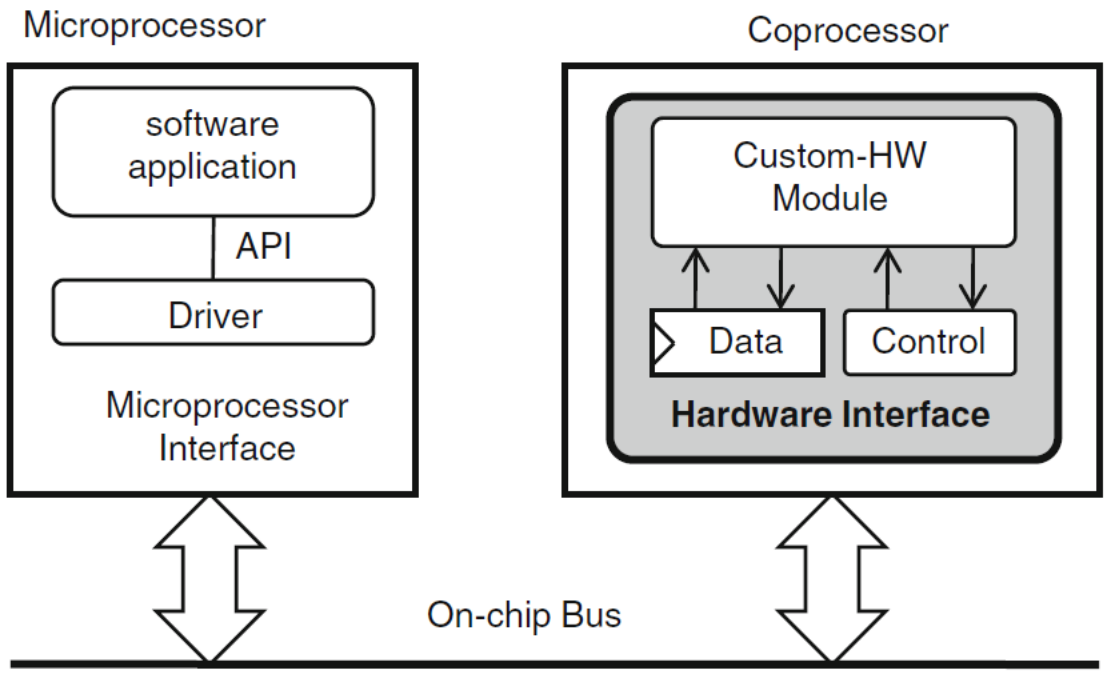
مباحث این جلسه

- واسط‌های برقراری ارتباط
  - تعریف روش‌های دسترسی
  - پیاده‌سازی در SystemC

آتنا عبدی
دانشکده مهندسی کامپیوتر-دانشگاه صنعتی خواجه نصیرالدین طوسی

واسط‌های ارتباطی

- مجرایی است که تجهیزات مستقل از طریق آن ارتباط برقرار می‌کنند
- به‌عنوان مثال، این واسط است که اطلاعات لازم را روی باس گذاشته و برمی‌دارد

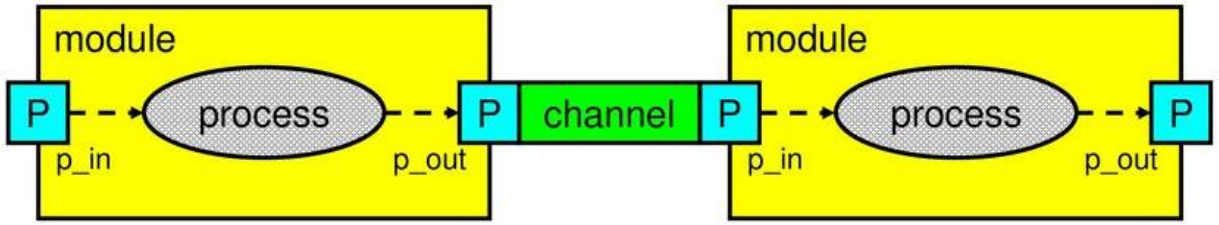


واسط‌های ارتباطی

- روش‌های متنوعی در این حیطه مطرح می‌شود
  - Memory-mapped I/F
    - تخصیص یک ثبات برای برقراری ارتباط بین سخت‌افزار و نرم‌افزار
  - Mailbox
    - اضافه کردن مکانیزم دست‌دهی به روش قبل
  - صف FIFO
  - و ....

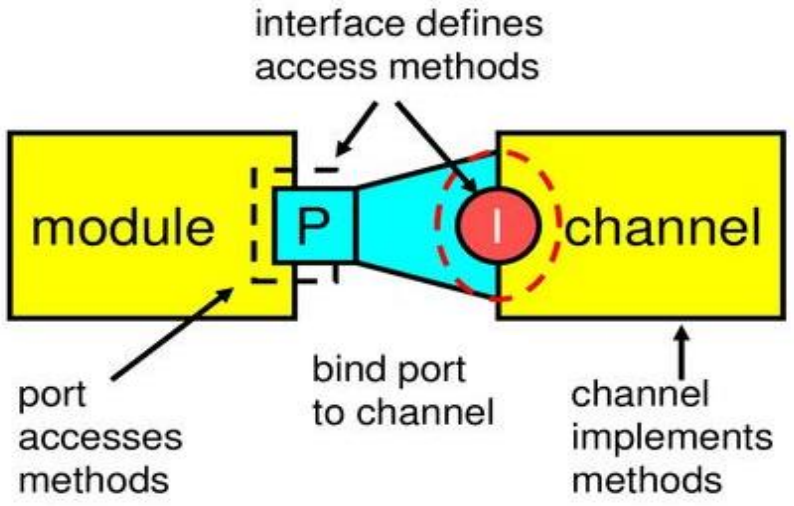
پیاده‌سازی واسط‌های ارتباطی

- قبلا دیدیم که ارتباط بین پروسه‌های همروند سیستم توسط Event
  - در این حالت نیاز به برنامه‌ریزی دقیق برای اشتراک داده و از دست نرفتن اطلاعات است
  - در مثال باس و دست‌دهی، از دست رفتن یک رخداد در حین برنامه‌نویسی ارتباط را مشکل‌دار می‌کند
  - به‌منظور تسهیل فرایند ارتباطی و گسترش دادن آن به ماژول‌ها
    - مکانیزم channels در SystemC برای اختصارسازی جزئیات ارتباط



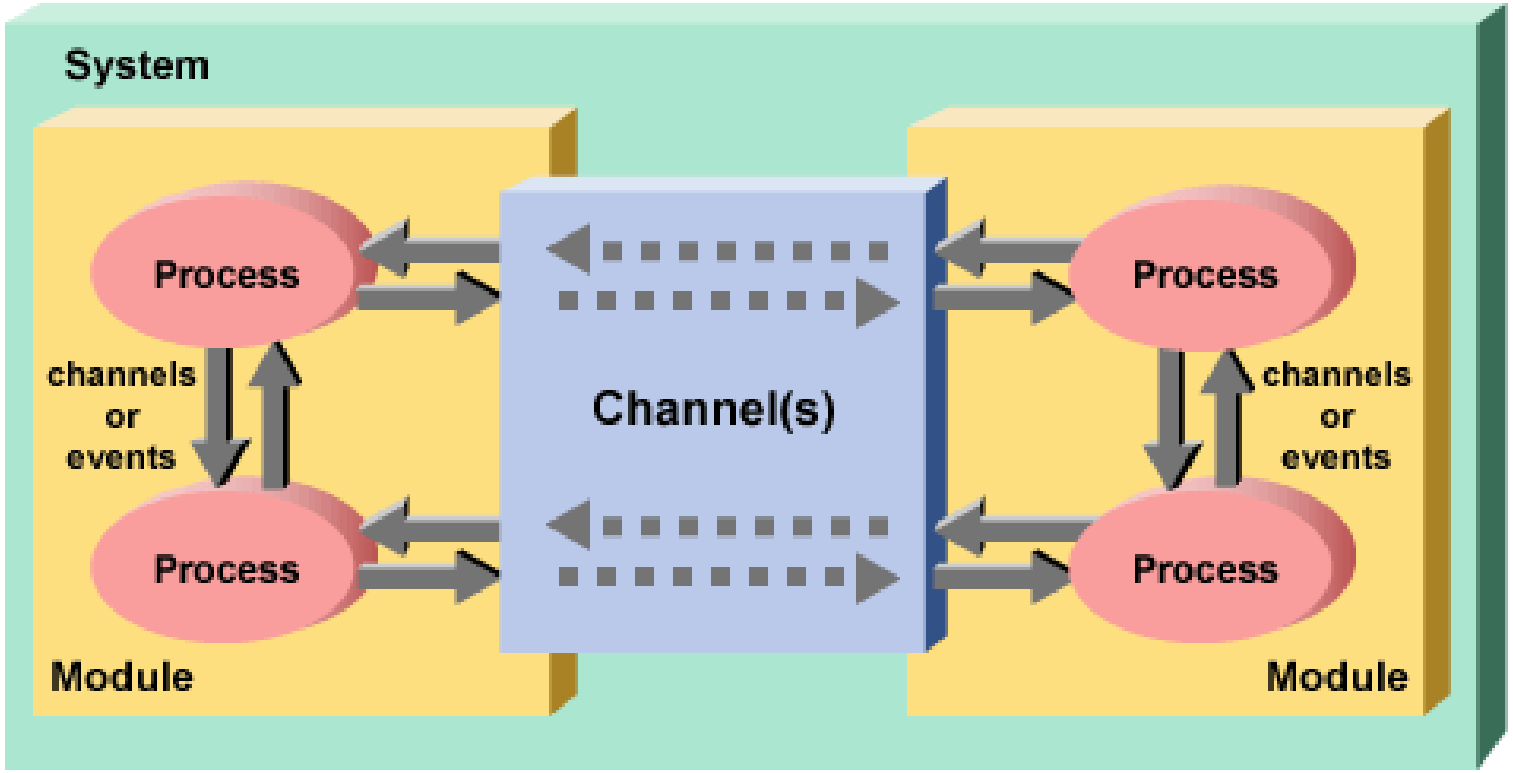
پیاده‌سازی واسط‌های ارتباطی

- در برقراری ارتباط، سه مفهوم داریم:
  - **واسط:** متدهای دسترسی را تعریف می‌کند
  - **کانال:** متدهای یک یا چند واسط را پیاده‌سازی می‌کند
  - **پورت:** دسترسی ماژول به متدهای واسط که توسط کانال پیاده‌سازی شده را میسر می‌کند



پیاده‌سازی واسط‌های ارتباطی

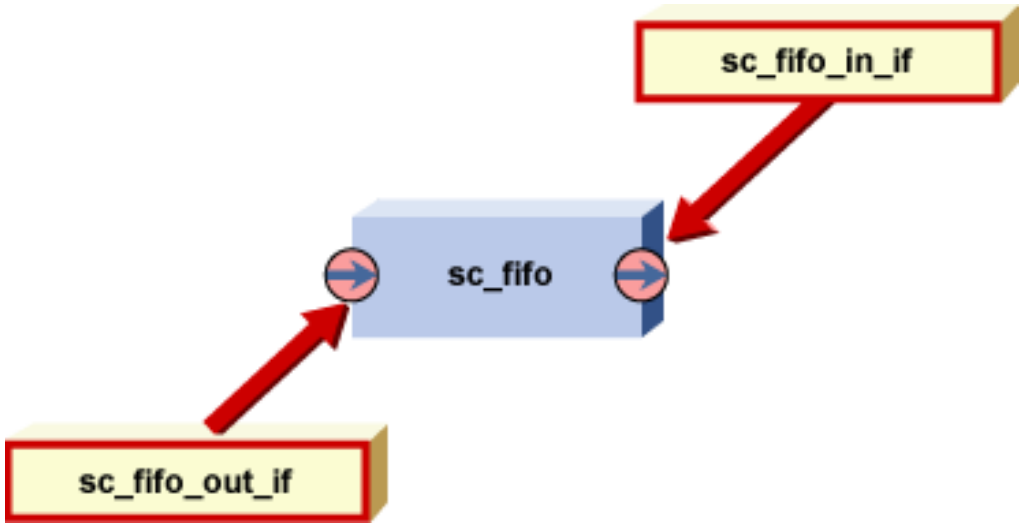
- امکان دسترسی کانال بطور مستقیم (داخل ماژول) و دسترسی از طریق پورت (بین ماژول‌ها)



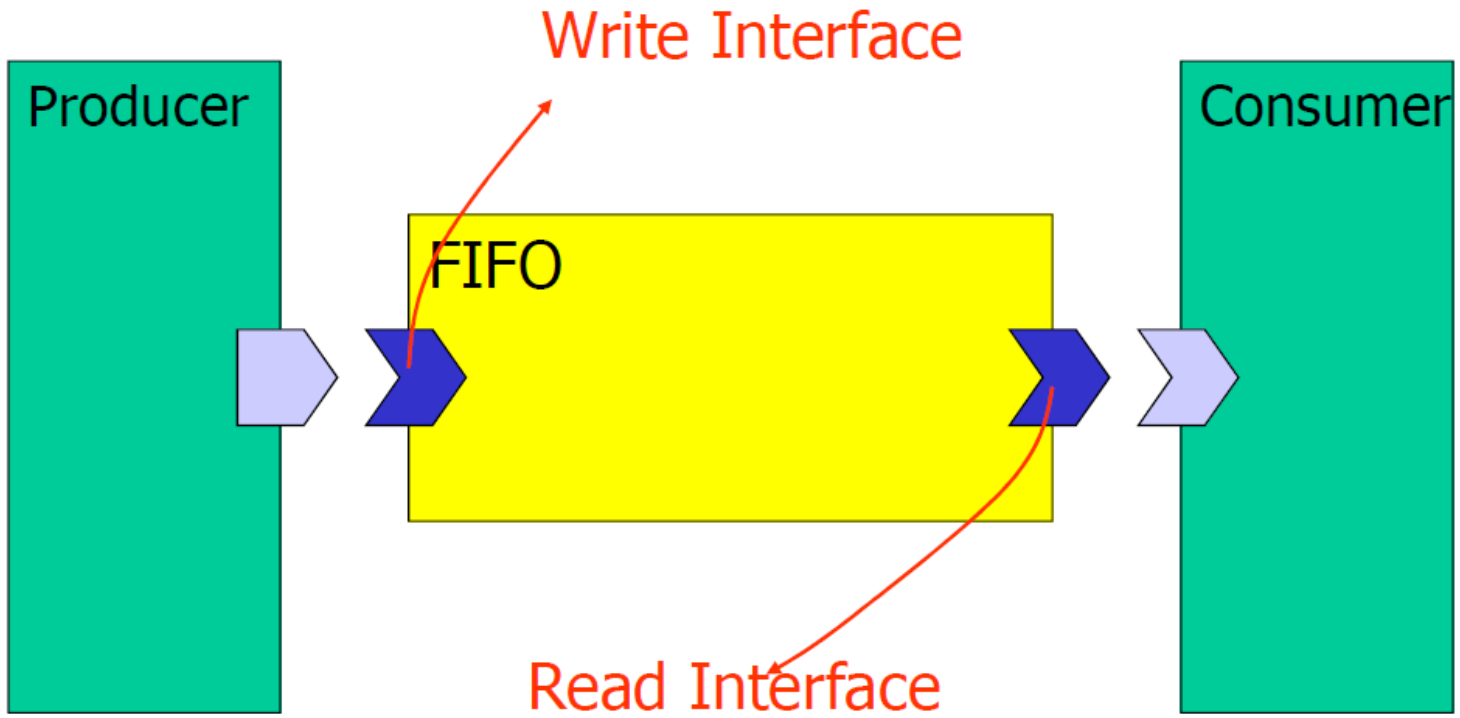
پیاده‌سازی کانال در SystemC

- sc_fifo

  - رایج‌ترین ساختار داده برای برقراری ارتباط و مدیریت جریان داده
  - شکل‌دهی ساختار صف انتقال داده و بافرینگ با عمق ۱۶
  - پیاده‌سازی در قالب دو واسط:
    - **sc_fifo_in_if**: خواندن از صف
    - **sc_fifo_out_if**: نوشتن در صف
  - برقراری ارتباط بین یک پورت ورودی و یک پورت خروجی



پیاده‌سازی کانال FIFO در SystemC



پیاده‌سازی کانال در SystemC

- تعریف sc_fifo

```
SC_MODULE (module_name) {
// channels
sc_fifo<int> d; // type int, depth of 16
sc_fifo<char> e ; // type char,depth of 16

// rest of module
} ;
```

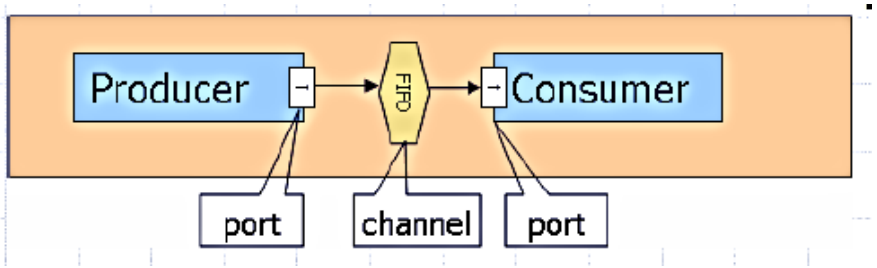
پیاده‌سازی کانال در SystemC

- متدهای read و write
  - قابل انجام به دو صورت blocking و non-blocking
    - read(var) و nb_read(var)
    - write(var) و nb_write(var)

```
// declarations
// fifo channel of depth 16 inside a module
sc_fifo<int> fifo_a;
int a;
a = fifo_a.read();
fifo_a.write(a);
if(fifo_a.num_available() > 3)
a = fifo_a.read();
```

مثال

- پیاده‌سازی سیستم producer-consumer
  - یک سیستم تولیدکننده و یک سیستم مصرف‌کننده داریم و سنکرون‌سازی بین آن‌ها مدنظر است
  - دو ماژول در پیاده‌سازی داریم:
    - Producer
    - Consumer
  - برقراری ارتباط از طریق کانال FIFO بین دو ماژول
    - متدهای هر ماژول؟



مثال-Producer/Consumer

- ماژول تولیدکننده

```
SC_MODULE(producer){
sc_in_clk iclk;
sc_fifo_out<int> data_out;
int data_value;
SC_CTOR(producer){
SC_CTHREAD(x_producer,iclk.pos());
data_value=0;}
void x_producer(){
while (true){
wait(2);
data_value=rand()%100;
data_out.write(data_value);
cout<<"At time "<<sc_time_stamp()<<"produces
data"<<data_value<<endl;
}}};
```

مثال-Producer/Consumer

- ماژول مصرف‌کننده

```
#include …
SC_MODULE(consumer)
{ sc_in_clk iclk;
sc_fifo_in<int> data_in;
int data_value;
SC_CTOR(consumer){
SC_CTHREAD(x_consumer,iclk.pos());
data_value=0;}
void x_consumer() {
while (true){
wait(3);
data_value=data_in.read();
cout<<"At time"<<sc_time_stamp()<<" consume
data"<<data_value<<endl;}}};
```

Producer/Consumer-مثال

- تابع main

```
int sc_main(int argc, char *argv[]){
// clock description (clk)
sc_fifo<int> x_fifo;
producer x_producer("x_producer");
consumer x_consumer("x_consumer");
x_producer.iclk(clk);
x_consumer.iclk(clk);
x_producer.data_out(x_fifo);
x_consumer.data_in(x_fifo);
sc_start(200,SC_NS);
return 0;}
```

```
At time 20 nsproduces data41
At time30 ns consume data41
At time 40 nsproduces data67
At time 60 nsproduces data34
At time60 ns consume data67
At time 80 nsproduces data0
At time90 ns consume data34
At time 100 nsproduces data69
At time 120 nsproduces data24
At time120 ns consume data0
At time 140 nsproduces data78
At time150 ns consume data69
At time 160 nsproduces data58
At time 180 nsproduces data62
At time180 ns consume data24
```

مباحثی که این جلسه آموختیم

- مفهوم واسط‌های ارتباطی در هم‌طراحی سخت‌افزار و نرم‌افزار
  - نقش واسط و نمونه‌های رایج
- پیاده‌سازی کانال‌های ارتباطی بین ماژول
  - sc_fifo در SystemC