



زبان های توصیف سخت افزار و مدارها

دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

امیر خورسندی

زمستان ۱۴۰۱

زبان توصیف سخت افزار Verilog

تاریخچه

- زبان توصیف سخت افزار
- ارائه شده در سال ۱۹۸۳
- استاندارد IEEE 1364
- قواعد دستوری مشابه زبان C
- اجرای موازی و همروند دستورات

مقایسه HDL با Programming Language

- خروجی PL یک برنامه کامپیوتری است.
- برنامه کامپیوتری مجموعه ای از دستورات متوالی است که برای پردازنده مشخص می کند به ترتیب و در طول زمان چه کارهایی را انجام دهد.
- خروجی HDL یک مدل برای مدار سخت افزاری دیجیتال است.
- این مدار شامل بلوک های مختلفی است که به طور همزمان کارهای متفاوتی را انجام می دهند.

سطوح مختلف توصیف سخت افزار

- ترانزیستورها

- دروازه های منطقی

- RTL

- توصیف رفتاری (الگوریتمی)

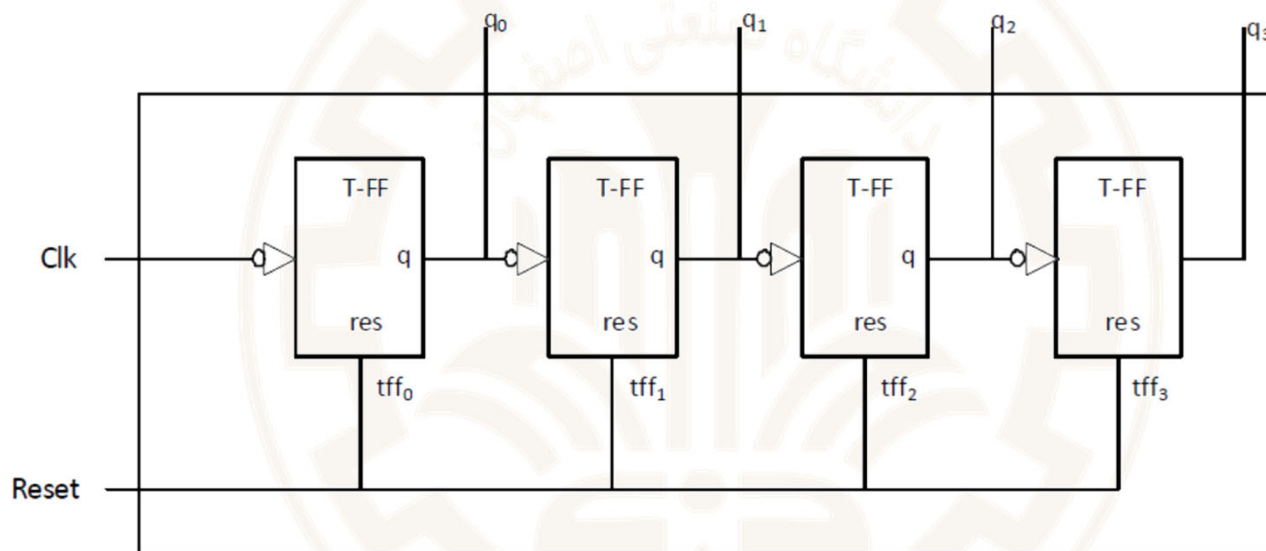


روش های مختلف طراحی

- طراحی سلسله مراتبی
- طراحی از بالا به پایین
- طراحی از پایین به بالا
- در عمل طراحی با استفاده از روش ترکیبی و با شروع از هر دو جهت انجام می شود.

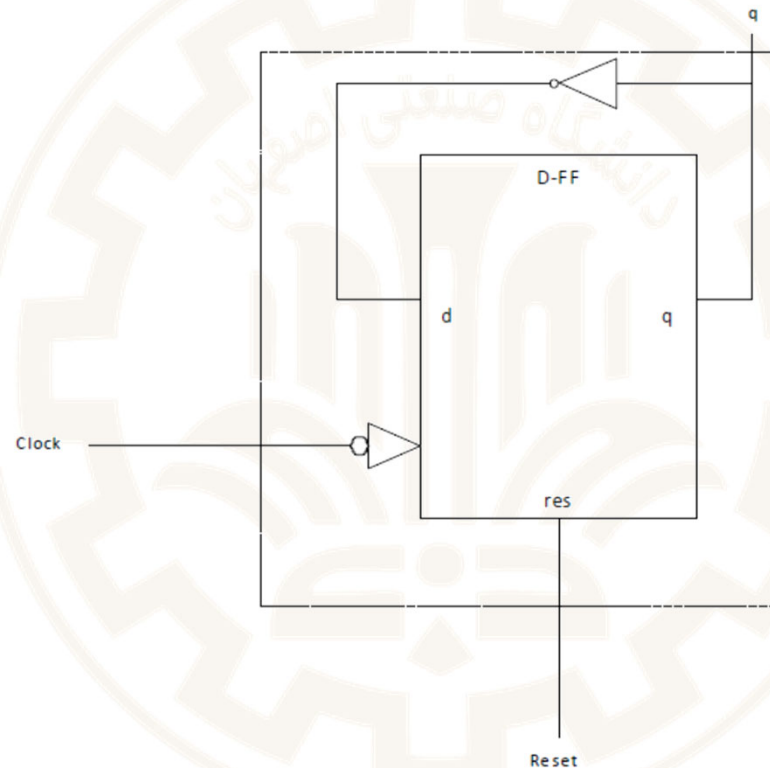
مثال

• شمارنده ۴ بیتی



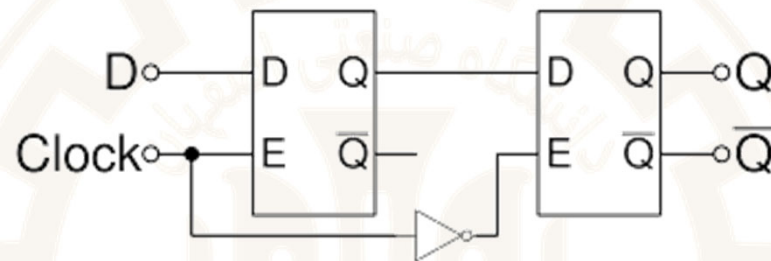
مثال (ادامه)

- فلیپ فلاپ نوع T

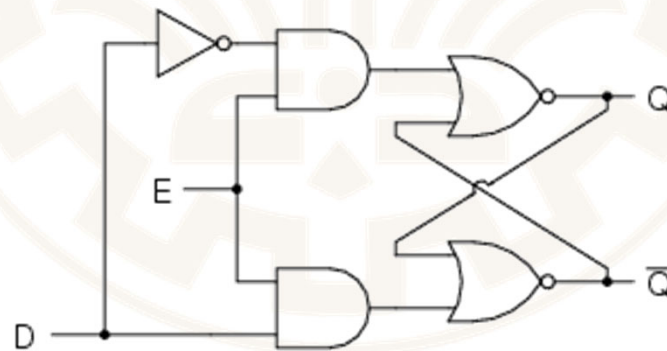


مثال (ادامه)

- فلیپ فلاپ نوع D

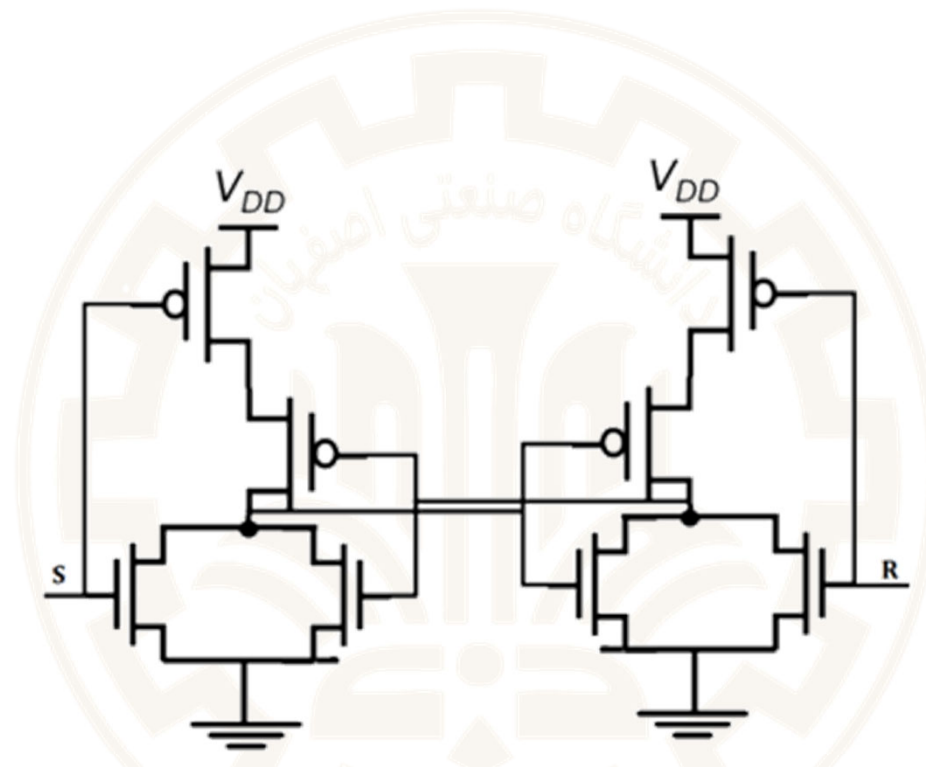


- نگه دار نوع D



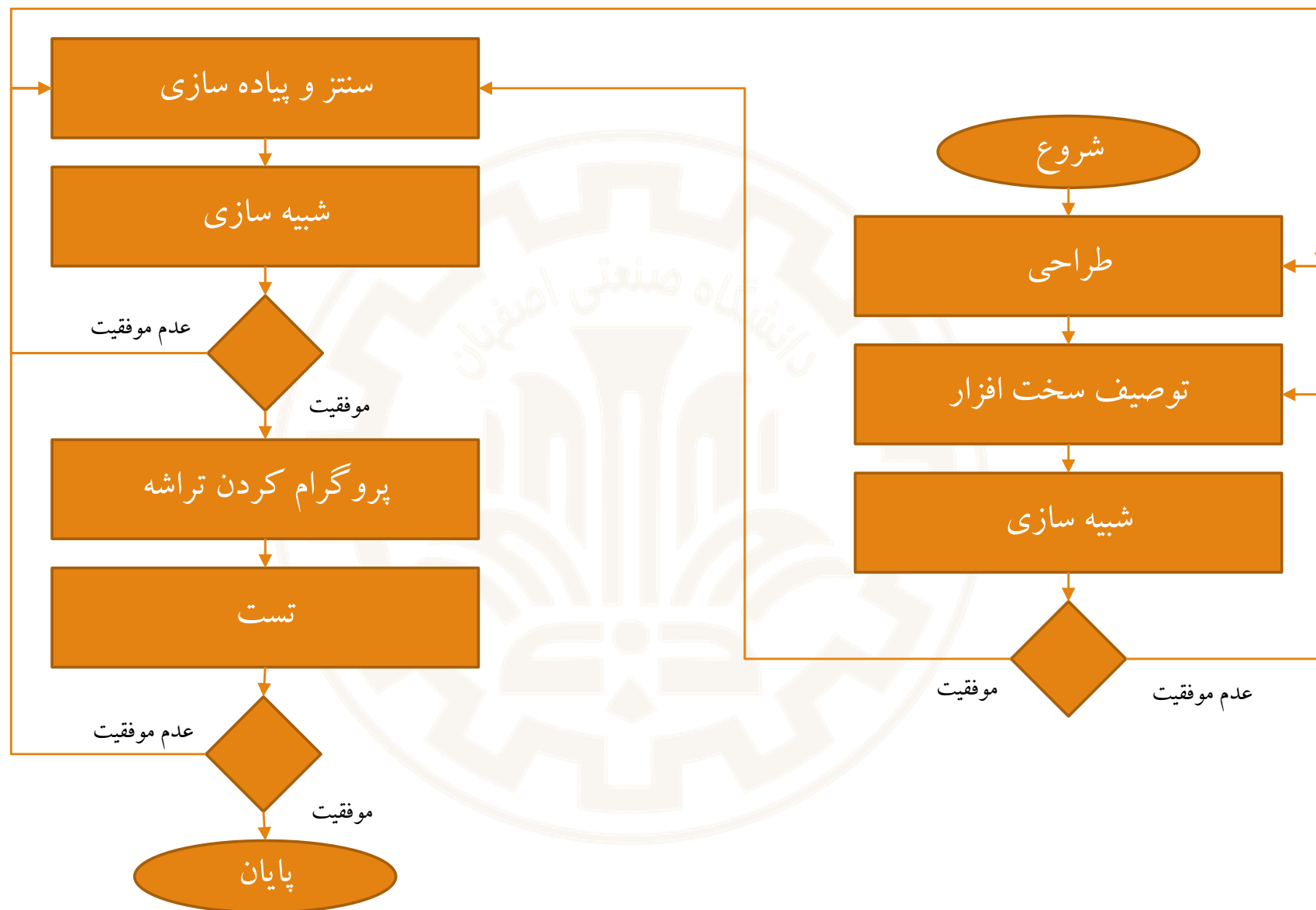
مثال (ادامه)

- نگه دار نوع SR



توصیف در سطح ترانزیستور برای طراحی با FPGA کاربردی ندارد.

روندنمای طراحی سخت افزار با FPGA



مقادیر سیگنال

| مقدار | توضیحات |
|-------|-----------|
| 0 | صفر منطقی |
| 1 | یک منطقی |
| z | شناور |
| x | نامشخص |

سطوح توان سیگنال

| سطح | توضیحات |
|--------|-------------|
| Supply | تغذیه کننده |
| Strong | تغذیه کننده |
| Pull | تغذیه کننده |
| Large | ذخیره کننده |
| Weak | ذخیره کننده |
| Medium | ذخیره کننده |
| Small | ذخیره کننده |
| Highz | مقاومت زیاد |

انواع متغیر

- سیم: دقیقاً مانند یک سیم عمل می کند که همواره باید به آن یک مقدار تغذیه شود. در غیر این صورت مقدار آن Z خواهد بود.

wire a;

- متغیر حافظه دار: مقدار آن همواره برابر مقداری است که در آخرین انتساب به آن داده شده است. مقدار پیش فرض آن ها X است.

reg b;

- متغیر reg لزوماً معادل سخت افزار یک ثبات نیست.

انواع متغیر (ادامه)

• بردار:

```
wire [7:0] c;  
reg [0:31] d;
```

• امکان دسترسی انتخابی وجود دارد:

```
c[7]  
d[5:8]
```



انواع متغیر (ادامه)

- اعداد صحیح

```
integer cnt;
```

- اعداد حقیقی

```
real pi;
```

- آرایه

```
integer x [0:7];  
reg [0:3] port_id [9:0];
```


انواع متغیر (ادامه)

• زمان

```
time sim_time;
```



روش های مقداردهی

• اعداد:

`<size>`<base><num>`

`4`b11?0`

`12`haxc`

`-16`d2_439`

- علامت سوال معادل Z است.
- در مبنای ۱۶ هر رقم X و یا Z معادل ۴ بیت هستند.
- در مقداردهی اگر بیت پرارزش 0، X و یا Z باشد به تمام بیت های بالا دست سرایت می کند اما اگر یک باشد مابقی بیت ها صفر خواهند شد.

قواعد نامگذاری

- ترکیب حروف، اعداد، _ و \$
- حساس به حروف کوچک و بزرگ
- حرف اول نمی تواند عدد و یا \$ باشد.
- استفاده از کلمات کلیدی مجاز نیست.
- شناسه مرکب:

New-Name

- معنادار بودن نامگذاری

کلمات کلیدی

| | | | | |
|--------------|--------------|-------------|-----------|----------|
| always | endmodule | large | reg | tranif0 |
| and | endprimitive | macromodule | release | tranif1 |
| assign | endspecify | nand | repeat | tri |
| attribute | endtable | negedge | rnmos | tri0 |
| begin | endtask | nmos | rpmos | tri1 |
| buf | event | nor | rtran | triand |
| bufif0 | for | not | rtranif0 | trior |
| bufif1 | force | notif0 | rtranif1 | triereg |
| case | forever | notif1 | scalared | unsigned |
| casex | fork | or | signed | vectored |
| casez | function | output | small | wait |
| cmos | highz0 | parameter | specify | wand |
| deassign | highz1 | pmos | specparam | weak0 |
| default | if | posedge | strength | weak1 |
| defparam | ifnone | primitive | strong0 | while |
| disable | initial | pull0 | strong1 | wire |
| edge | inout | pull1 | supply0 | wor |
| else | input | pulldown | supply1 | xnor |
| end | integer | pullup | table | xor |
| endattribute | join | rcmos | task | |
| endcase | medium | real | time | |
| endfunction | module | realtime | tran | |

عناصر پایه

| Primitive | Description | Outputs | Inputs |
|-----------|---|--------------|---------------------|
| and | and | output | in1, ..., inN |
| nand | nand | output | in1, ..., inN |
| or | or | output | in1, ..., inN |
| nor | nor | output | in1, ..., inN |
| xor | xor | output | in1, ..., inN |
| xnor | xnor | output | in1, ..., inN |
| buf | buffer input | output | input |
| bufif0 | buffer input if 0, otherwise hi-z | output | input, control |
| bufif1 | buffer input if 1, otherwise hi-z | output | input, control |
| not | not input | output | input |
| notif0 | not input if 0, otherwise hi-z | output | input, control |
| notif1 | not input if 1, otherwise hi-z | output | input, control |
| pulldown | Pulldown to supply 0 | output | none |
| pullup | Pullup to supply 1 | output | none |
| nmos | Unidirectional NMOS transistor | output | input, gate |
| pmos | Unidirectional PMOS transistor | output | input, gate |
| cmos | Unidirectional CMOS transistor | output | input, ngate, pgate |
| nmos | Unidirectional NMOS transistor (resistive) | output | input, gate |
| rpmos | Unidirectional PMOS transistor (resistive) | output | input, gate |
| rcmos | Unidirectional CMOS transistor (resistive) | output | input, ngate, pgate |
| tran | Bi-directional pass gate | inout, inout | none |
| tranif0 | Bi-directional pass (if 0) gate | inout, inout | control |
| tranif1 | Bi-directional pass (if 1) gate | inout, inout | control |
| rtran | Bi-directional pass gate (resistive) | inout, inout | none |
| rtranif0 | Bi-directional pass (if 0) gate (resistive) | inout, inout | control |
| rtranif1 | Bi-directional pass (if 1) gate (resistive) | inout, inout | control |

عملگرها

• تک عملوندی: $!$, \sim , $\&$, $|$, $^$

• دو عملوندی: $+$, $-$, $*$, $/$, $\%$, $\&$, $|$, $^$, \sim , $\&\&$, $||$, $<<$, $>>$, $<$, $>$, $<=$, $>=$, $!=$, $==$, $===$, $!==$

• سه عملوندی: $x?y:z$

• الحاق: $\{x, y\}$

• تکرار: $\{i\{x\}\}$

ماژول

- هر ماژول توصيف كننده يك بلوك سخت افزاري با عملکرد و اتصالات مشخص است.

```
module <name> [(<port_list>)];
```

```
...
```

```
    [<ports and variables>]
```

```
...
```

```
    [< implementation>]
```

```
...
```

```
endmodule
```

```
module SR_Latch(Q, Qbar, Sbar, Rbar);  
    output Q, Qbar;  
    input Sbar, Rbar;  
    nand n1(Q, Sbar, Qbar);  
    nand n2(Qbar, Rbar, Q);  
endmodule
```


فراخوانی ماژول

- پس از تعریف یک ماژول باید متناسب با بلوک های سخت افزاری موجود در طرح، این ماژول فراخوانی گردد.

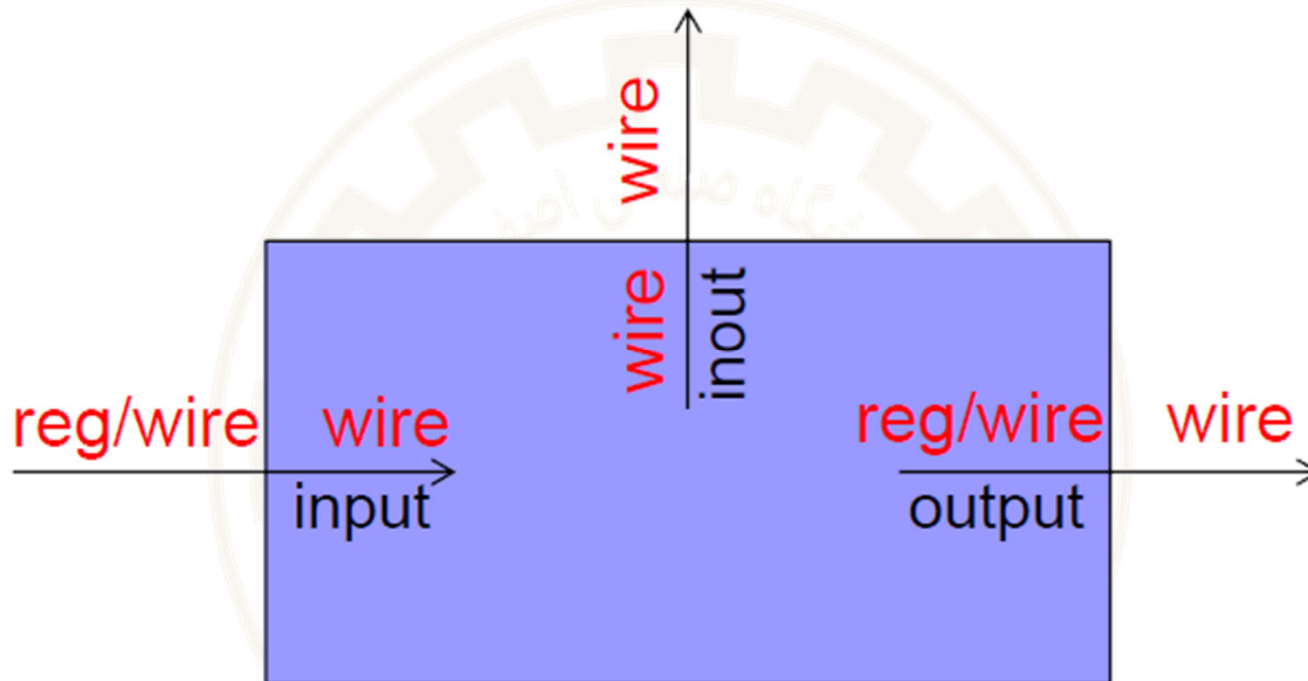
`<module_name> <instance_name>[(port_mapping)];`

- به این کار Instantiation گویند.
- به ازای هر فراخوانی یک Instance مستقل ایجاد می گردد.
- ماژول ها در سلسله مراتب طراحی می توانند به صورت تو در تو درون ماژول های دیگر فراخوانی شوند.
- تعریف ماژول نمی تواند درون یک ماژول دیگر قرار گیرد.

پورت

- برای ارتباط دادن ماژول با ماژول های سطح بالاتر در سلسله مراتب طرح به کار می رود.
- انواع:
 - input
 - output
 - inout
- وجود پورت در ماژول اختیاری است به گونه ای که مثلاً ماژول سطح بالا در شبیه سازی پورت ندارد.

قواعد اتصال پورت ها



نحوه اتصال پورت ها

- بر اساس ترتیب

```
SR_Latch L1(L1_Q, L1_Qbar, L1_Sbar, L1_Rbar)
```

- بر اساس نام

```
SR_Latch L2(.Sbar(L2_Sbar),  
             .Rbar(L2_Rbar),  
             .Q(L2_Q),  
             .Qbar(L2_Qbar));
```

- نمی توان به صورت ترکیبی از هر دو روش پورت ها را متصل نمود.
- در روش دوم ممکن است تعدادی از پورت ها متصل نشوند.