



بسمه تعالی

آموزش زبان VHDL

شبیه سازی و سنتز

واحد درسی مدارات منطقی دیجیتال و آزمایشگاه

دانشگاه فردوسی مشهد- گروه مهندسی برق

اردیبهشت ماه 1387

فهرست

■ مقدمه

■ توضیح FPGA و ساختمان آن

■ مراحل طراحی بوسیله FPGA

■ زبان برنامه نویسی VHDL

مقدمه

□ سیستم های دیجیتال امروزی:

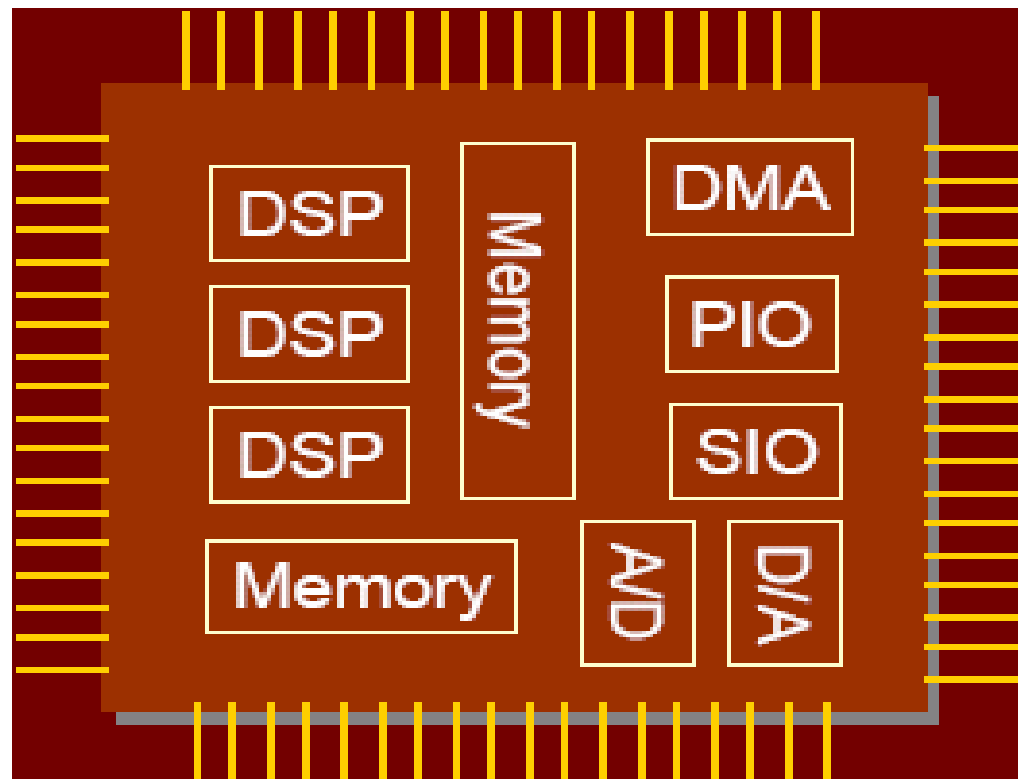
ASIC ■

FPGA ■

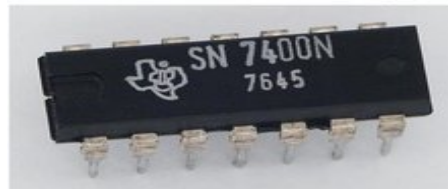
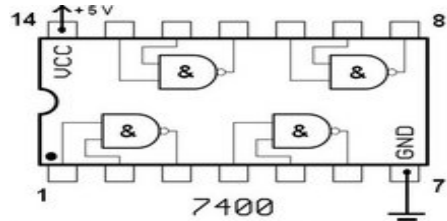
Microcontroller ■

■ ترکیبی از قطعات بالا

[System On chip Design]



[ASIC]



- طراحی شده برای کاربرد خاص
- ساختمان غیر قابل تغییر

[FPGA]



Field Programmable gate arrays

آرایه ای از گیت های آماده

قابل برنامه ریزی برای کاربرد دلخواه

[Microcontroller]



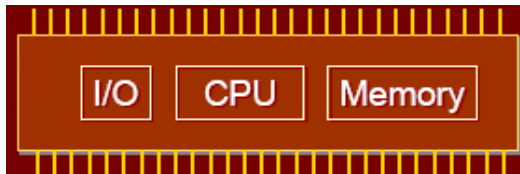
تراشه همه منظوره

شامل

1. CPU

2. I/O

3. Memory



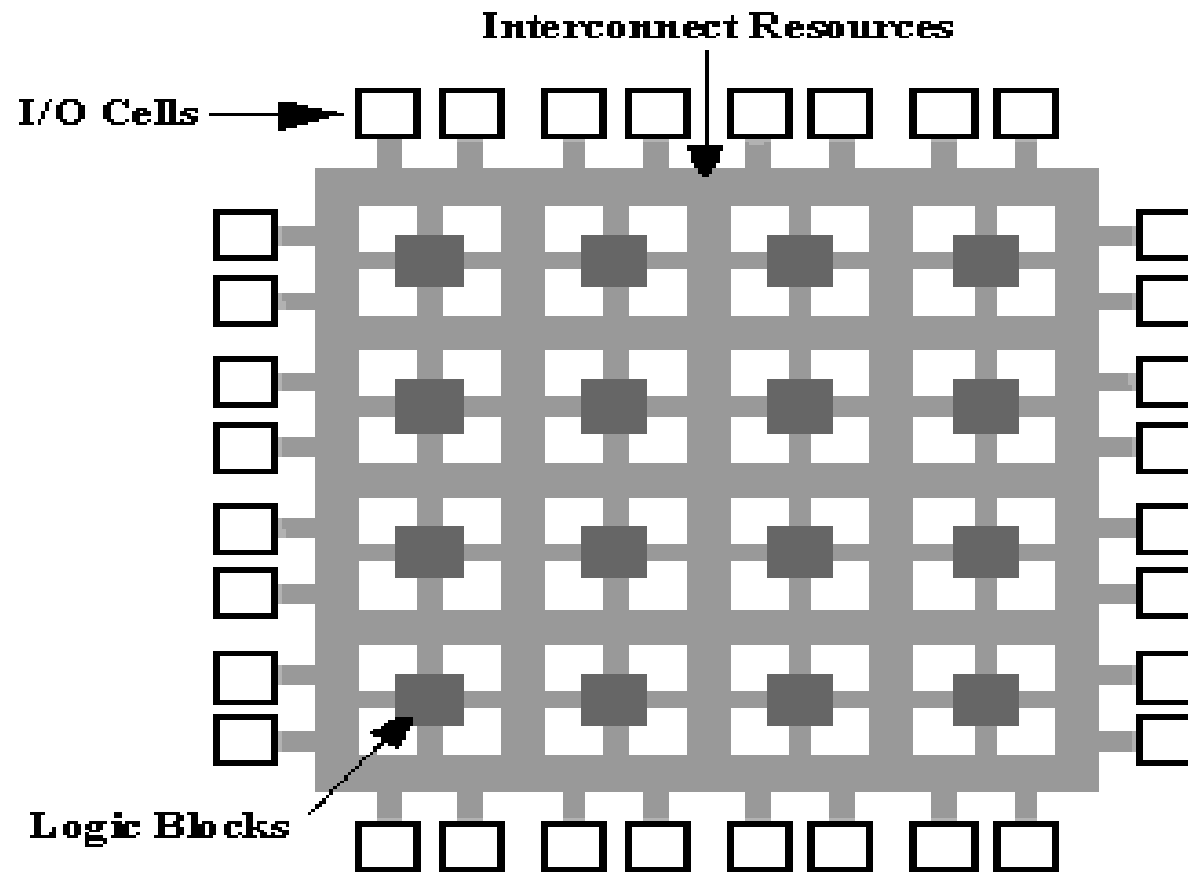
مقایسه

	ASIC	FPGA	μP
انعطاف	Low	High	Highest
سرعت	Highest	High	Low
قیمت	Highest	High	Low
توان	Low	High	Highest
سهولت دسترسی	Low	High	Highest

[FPGA]



ساختمان FPGA



مراحل طراحی بوسیله FPGA

- تعریف پروژه
- طراحی بلوک دیاگرام و ماشین حالت
- کد نویسی طراحی
- شبیه سازی رفتاری
- سنتز مدار منطقی
- شبیه سازی مدار منطقی
- پیاده سازی و تست بر روی Evaluation Board

زبان برنامه نویسی VHDL <= FPGA

■ تعریف VHDL:

VHSIC Hardware Description Language

زبان توصیف سخت افزار مدارات مجتمع بسیار سرعت بالا

فهرست مطالب

- ساختار کد در زبان VHDL
- توصیف تاخیر
- مفهوم سیگنال در VHDL
- معرفی دستور Port MAP
- آشنایی با محیط نرم افزار ActiveHDL
- ایجاد TestBench رفتاری

[

ساختار یک طراحی در VHDL

]

شروع با یک مثال

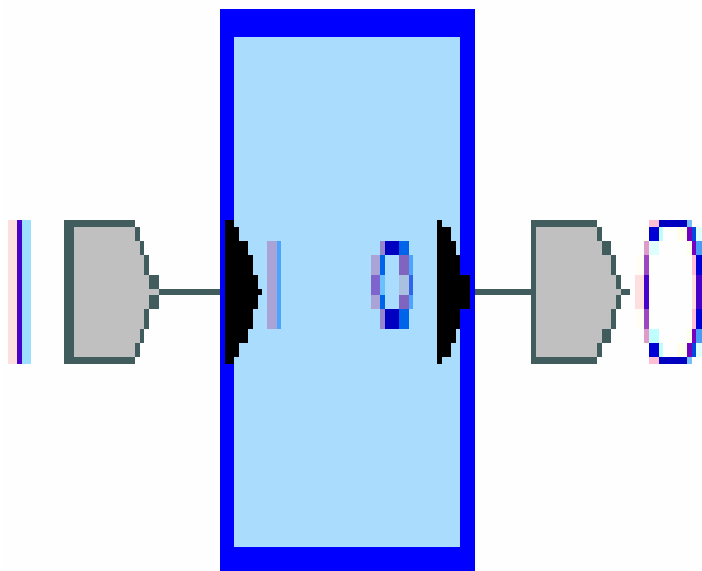
توصیف گیت NOT

]

[

سوالات زیر را از خود بپرسید:

1. نام بلوک؟
2. ورودی و خروجی؟
3. معماری داخلی؟



پاسخ به سوالات => توصیف سخت افزار

1. نام بلوک:

Inv: دلخواه

2. ورودی و خروجی:

یک ورودی تک بیتی: Input

یک خروجی تک بیتی: Output

2. معماری داخل:

$\text{Not (Input)} \Rightarrow \text{Output}$

نتیجه گیری


یک برنامه VHDL دو قسمت دارد:

1. Entity:

نام ، ورودی – خروجی

2. Architecture:

معماری داخل بلوک



گیت NOT
کد VHDL

نام فایل: Inv.vhd

[NOT Gate (Entity)]

```
entity Inv is
  port(
    i : in STD_LOGIC;
    o : out STD_LOGIC
  );
end Inv;
```

[NAND Gate (Architecture)]

```
architecture arch_inv of Inv is  
begin  
  O <= not i;  
end arch_inv;
```

مثال: گیت NAND

نام فایل: Nand2.vhd

[NAND Gate (Entity)]

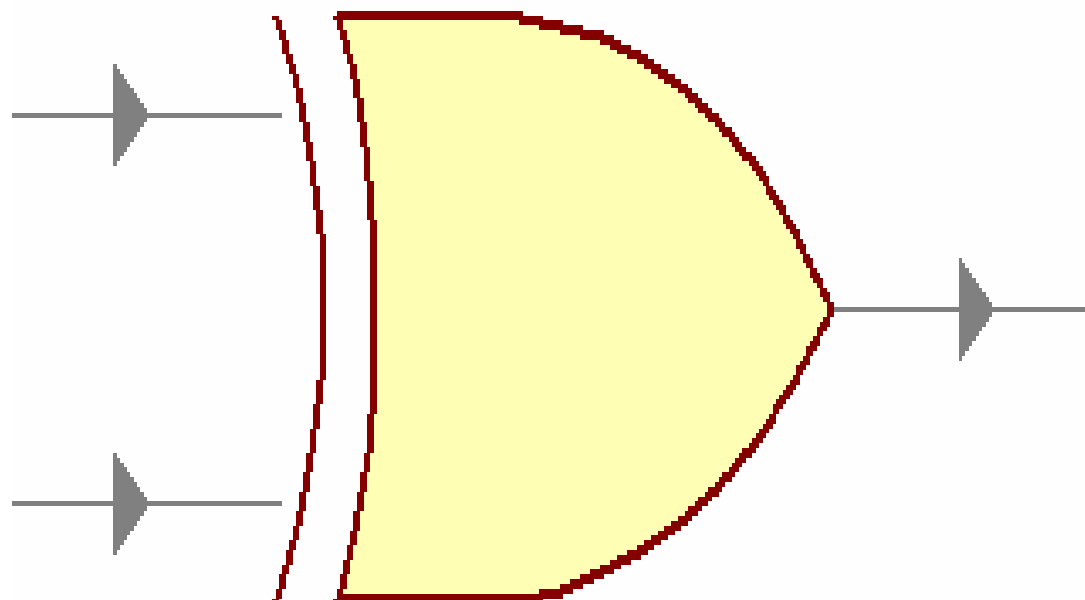
```
entity nand2 is
    port(
        i1: in STD_LOGIC;
        i2: in STD_LOGIC;
        o : out STD_LOGIC
    );
end nand2 ;
```

[NAND Gate (Architecture)]

```
architecture arch_nand2 of Inv is  
begin  
  O <= i1 nand i2;  
end arch_nand2 ;
```


مثال: گیت XOR

نام فایل: Xor2.vhd



[NAND Gate (Entity)]

```
entity Xor2 is
  port(
    i1 : in STD_LOGIC;
    i2 : in STD_LOGIC;
    o  :out STD_LOGIC

  );
end Xor2;
```

[NAND Gate (Architecture)]

```
architecture arch_ Xor2 of Inv is  
begin  
  O <= i1 xor i2;  
end arch_ Xor2;
```

[توصیف تاخیر در VHDL]

تاخیر در گیت AND

نام فایل: and2_delay.vhd

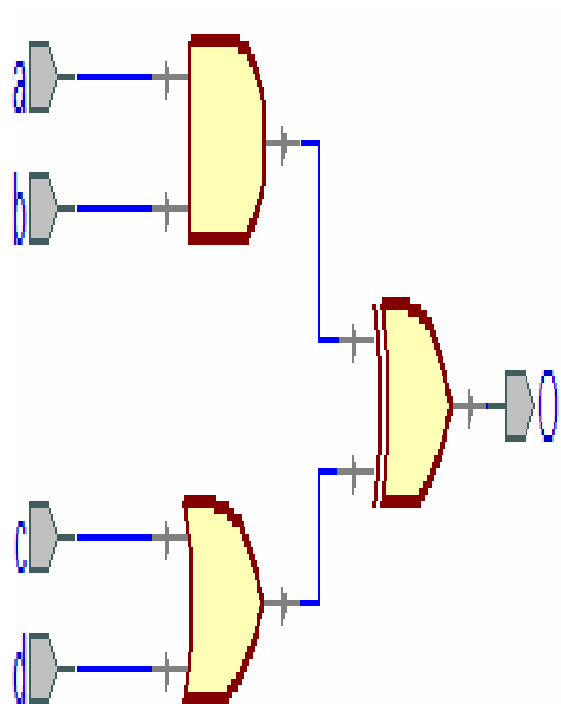
$O \leq i1 \text{ and } i2 \text{ after}$

مفهوم سیگنال

1. انتقال داده دینامیک
2. اتصال قطعات

1. انتقال داده دینامیک

مثال: توصیف مدار روبرو



راه های نوشتن کد

راه اول: توصیف مستقیم

نام فایل: Example3.vhd

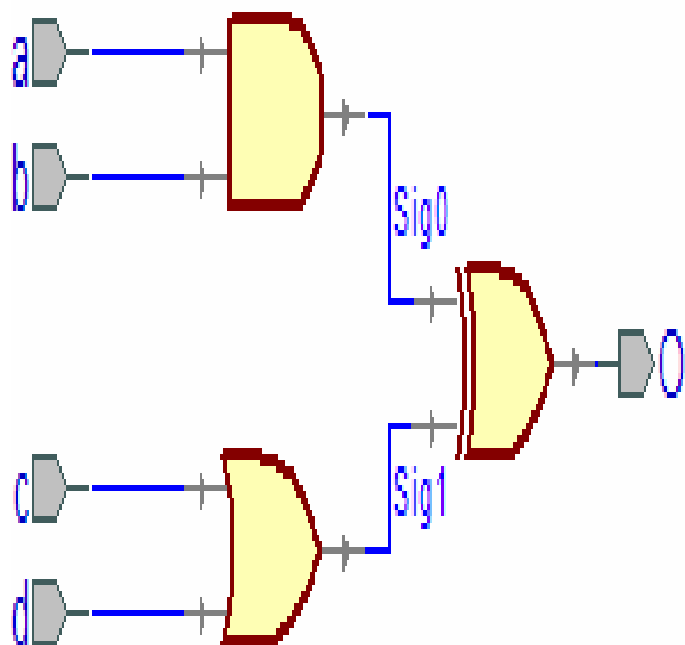
```
o <=(a and b) xor (c or d);
```


[

]

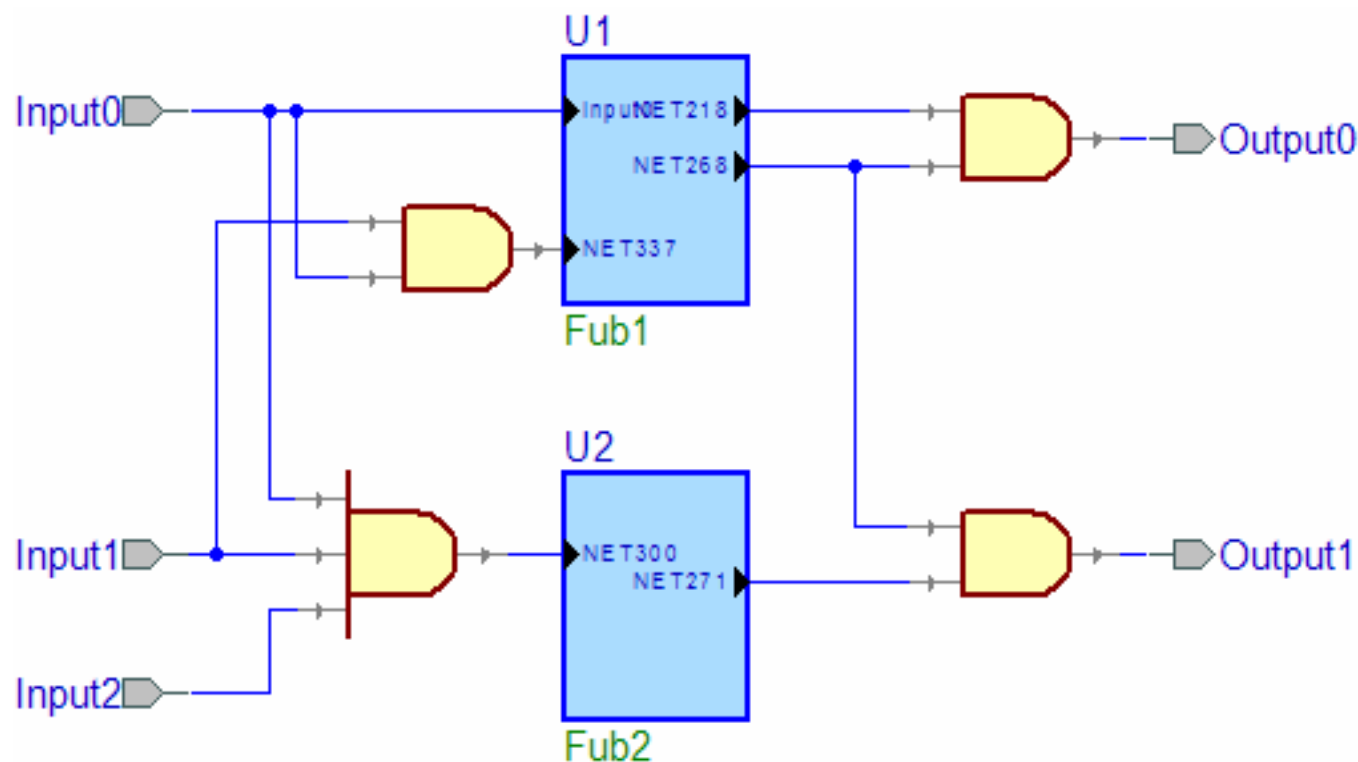
■ راه دوم:

استفاده از سیگنال به عنوان واسطه

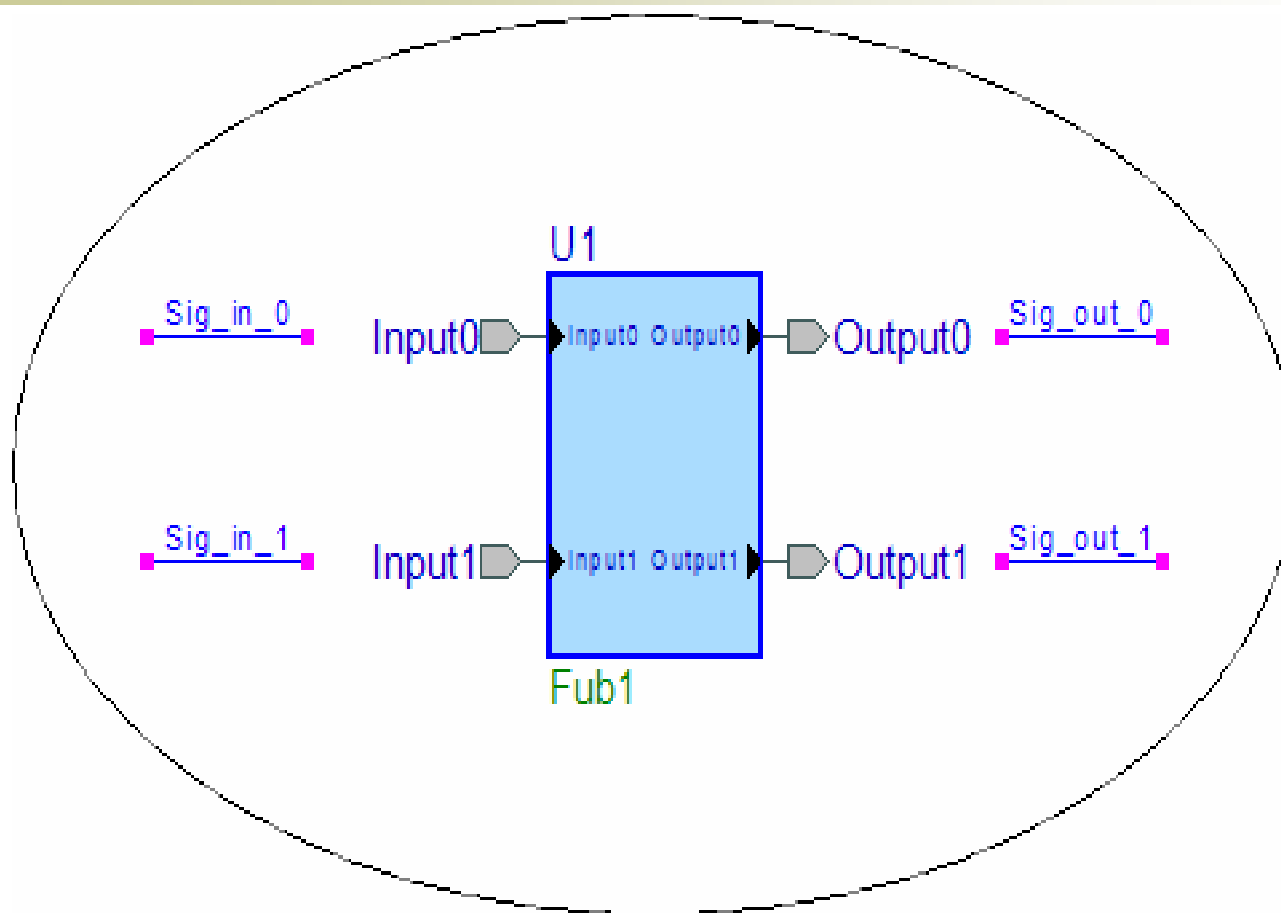


```
sig0<=a and b;  
sig1<=c or d;  
O<= sig0 xor sig1;
```

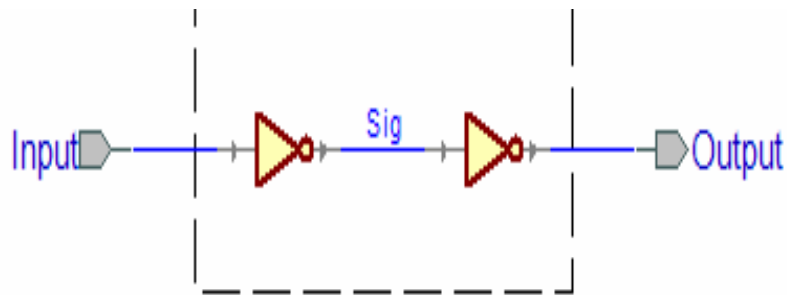
2. کاربرد سیگنال به عنوان اتصال دهنده قطعات



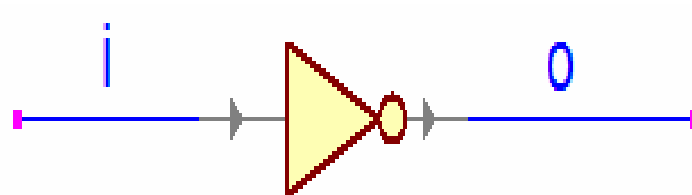
استفاده از یک بلوک در یک طراحی جدید
اتصال یک قطعه به دنیای خارج



مثال: ساخت بافر بوسیله گیت Not



■ مدار مورد نظر



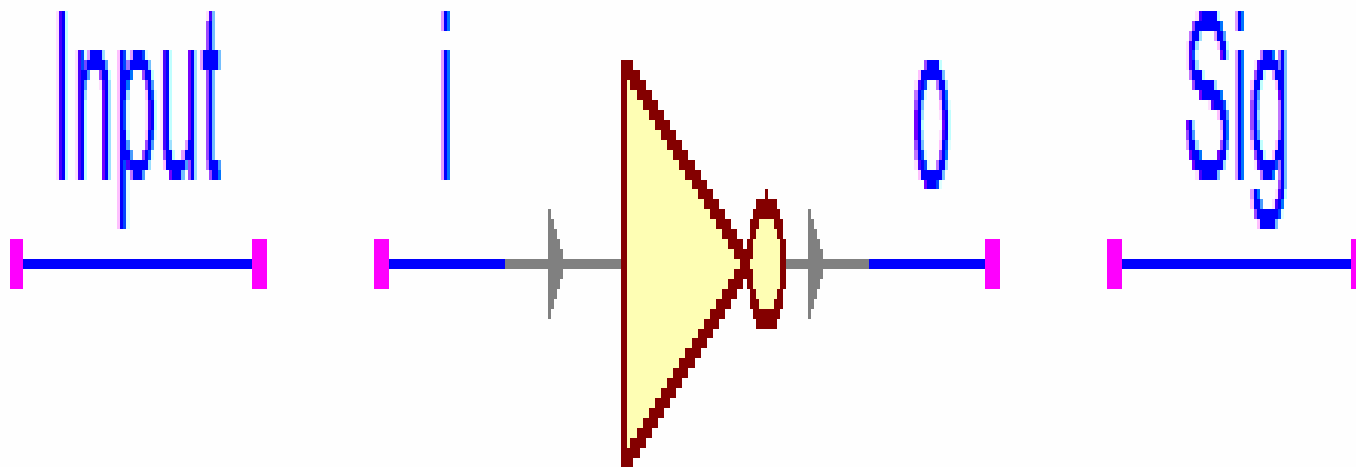
■ قطعه مورد استفاده:

گیت NOT

معرفی دستور Port Map

نام فایل: Buffer.vhd

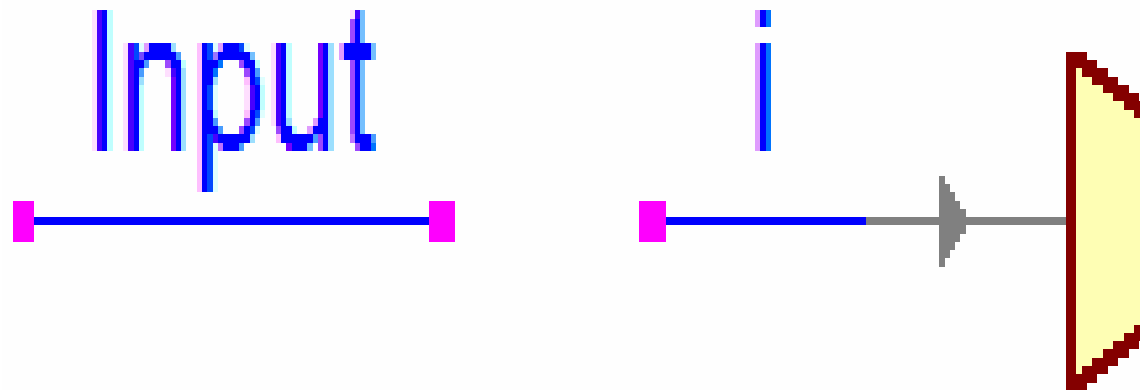
Lable1: entity inv port map (i=>Input, o=>sig);



[

$i \Rightarrow \text{Input}$

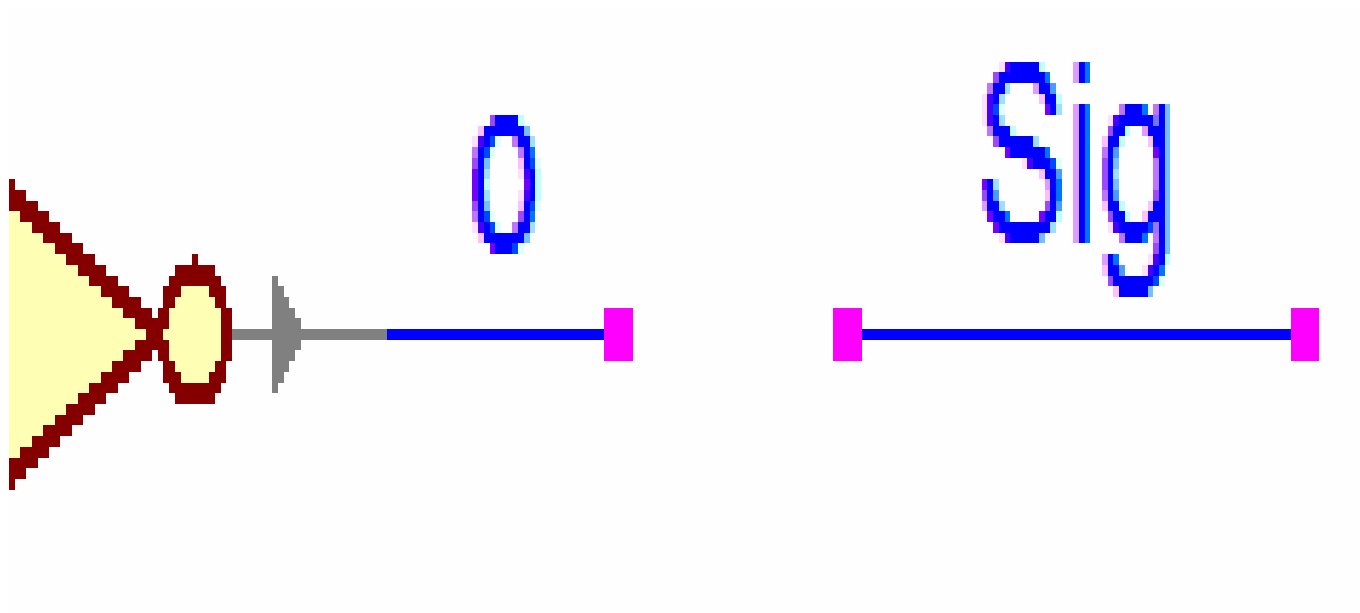
]



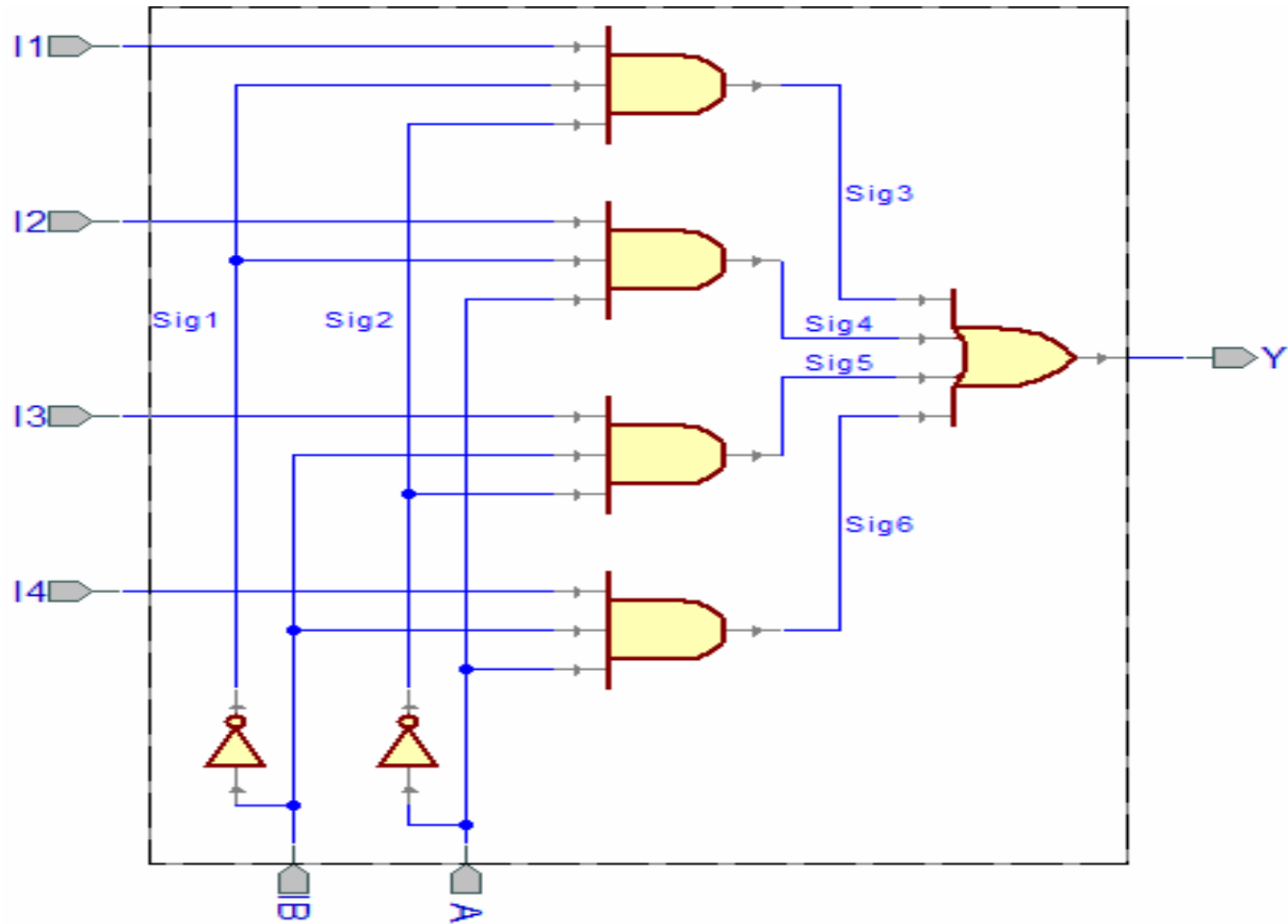
[

$o \Rightarrow \text{sig}$

]



مثال: 4 to 1 MUX



قطعات مورد نیاز

گیت NOT: 2 عدد

■ گیت AND با 3 ورودی: 4 عدد

■ گیت OR با 4 ورودی: 1 عدد

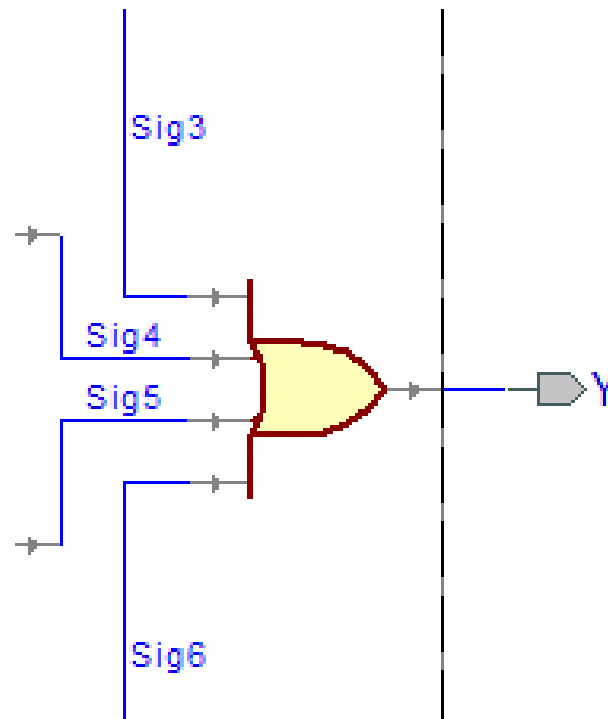
کد قطعات بالا باید از قبلا نوشته شده باشد.

■ فایل های مورد استفاده: Inv.vhd ، And3.vhd ، Or4.vhd

نمونه: Pot map گیت OR

نام فایل: Mux_4_to_1.vhd

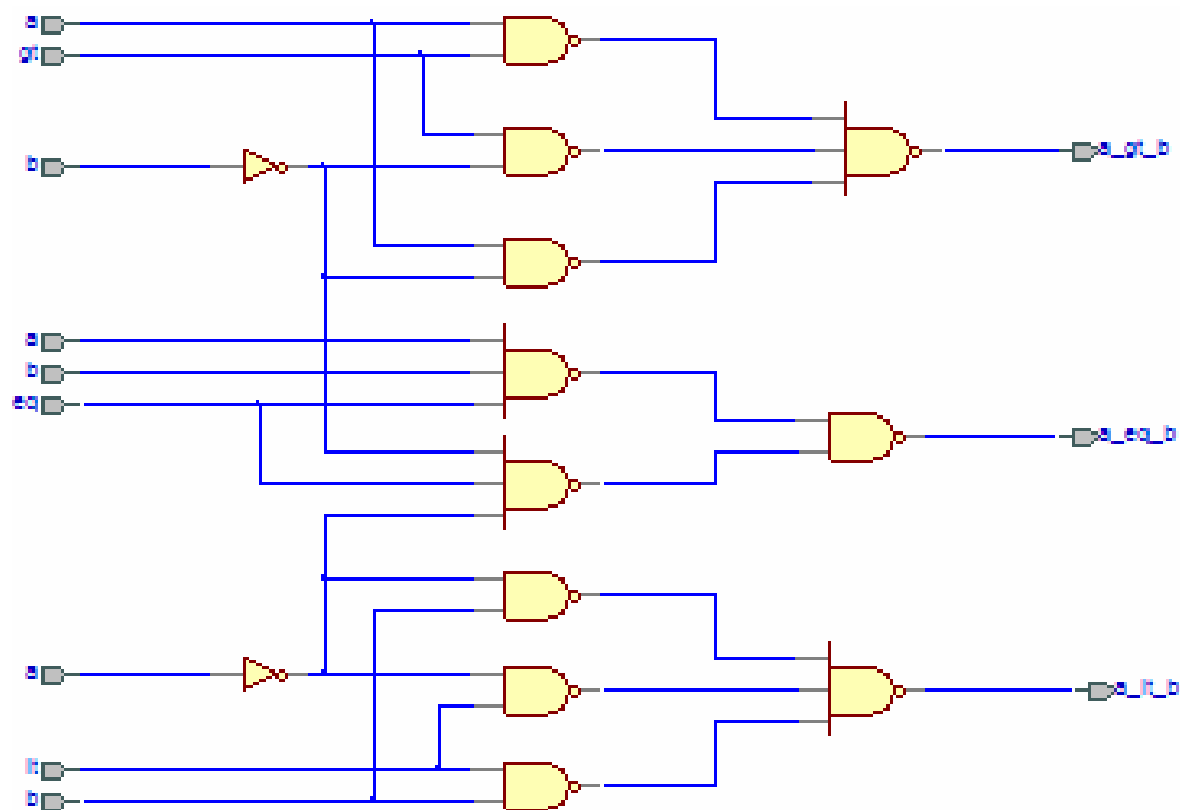
Lable7: entity or4 port map
(i1=>sig3, i2=>sig4,
i3=>sig5,i4=>sig6,
o=>y);



آشنایی با محیط ActiveHDL

- ایجاد پروژه جدید
- کد نویسی و Compile
- شبیه سازی و مشاهده نتایج

مثال: مقایسه گر تک بیتی



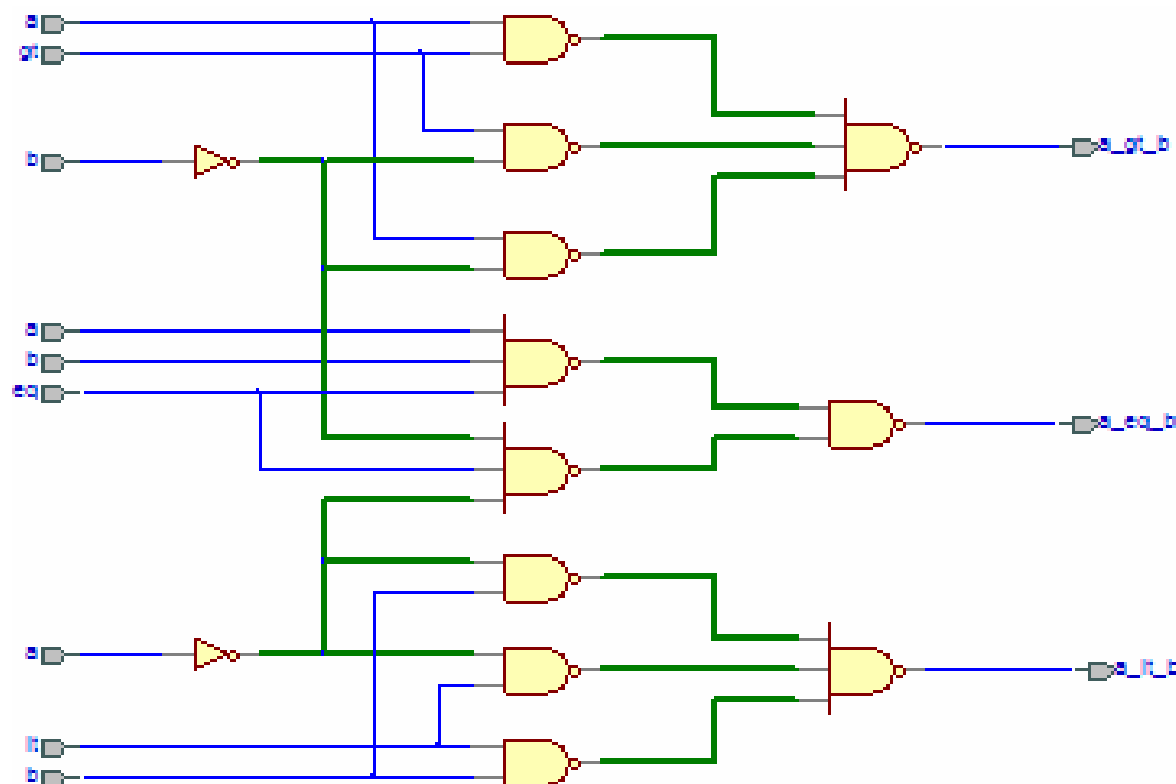
قطعات مورد نیاز

گیت NOT: 2 عدد

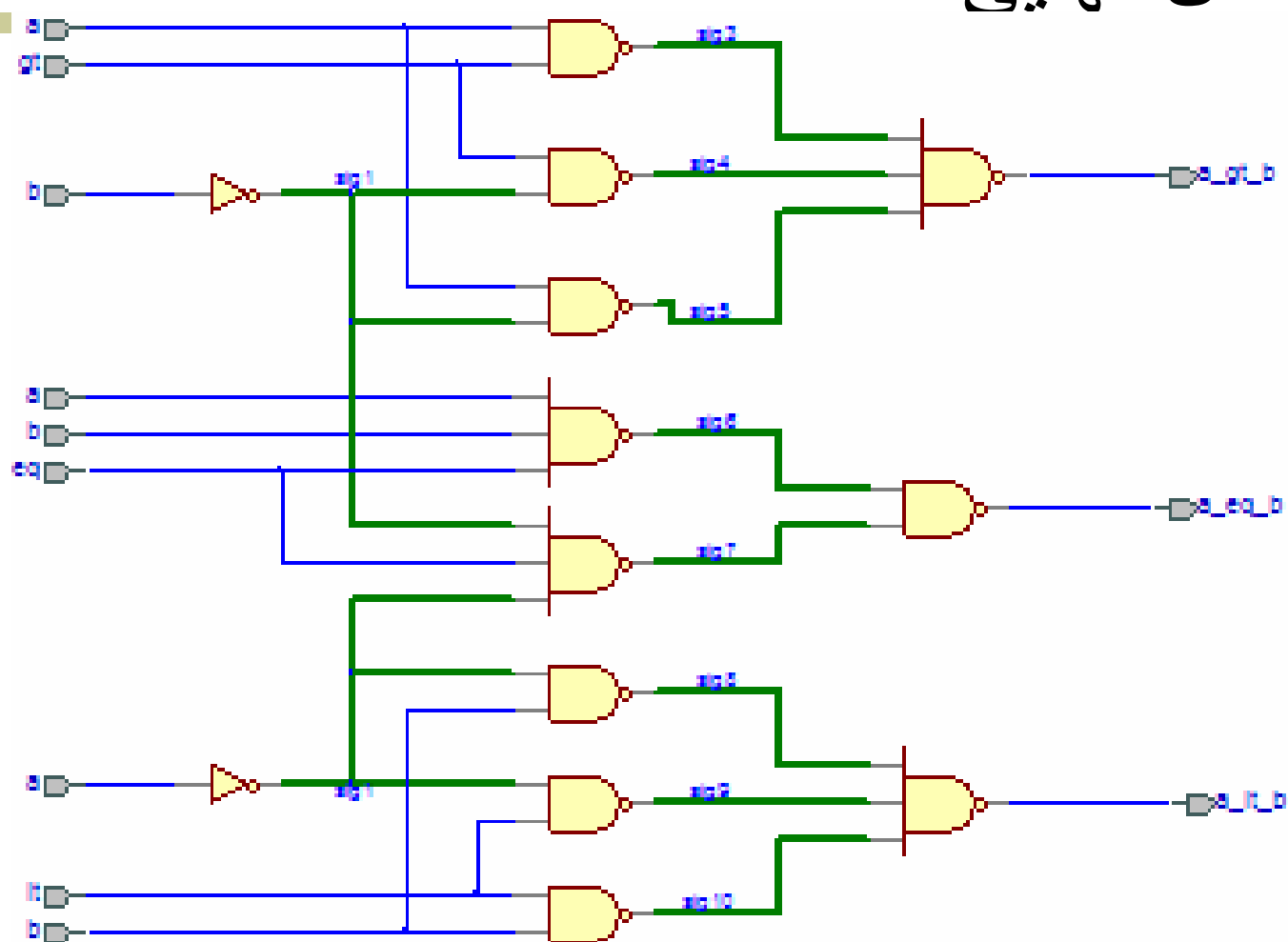
1. Nand دو ورودی: 7 عدد

2. Nand 3 ورودی: 4 عدد

خطوط سبز به هیچ پورتی متصل نیستند => سیگنال لازم دارند



شکل نهایی



یک نمونه از Port map مدار

نام فایل: 1bit_comp_sig.bde
single_bit_comp.vhd

```
L1:entity inv port map(i=>a ,O=>sig1)
```

[می توانیم Port map را ساده تر نیز انجام دهیم]

L1:entity inv port map (a , sig1)

اگر i و O را حذف کنیم نرم افزار، پورت ها را به ترتیب به مقادیر مورد نظر ما متصل می کند.

پس لازم نیست نام پورت های Entity مورد استفاده را بنویسیم.

ساخت Testbench رفتاری

شبیه سازی رفتاری در VHDL

چرا شبیه سازی:

کشف نقایص با کمترین هزینه و زمان

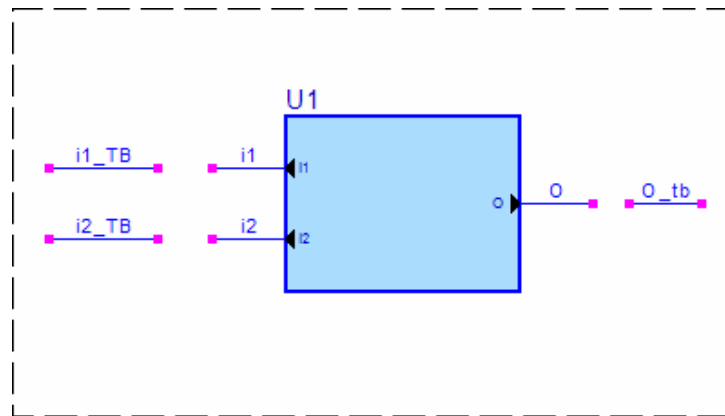
TestBench راه های ساخت

1. استفاده از امکانات نرم افزار

2. نوشتن کد برای TestBench

مزیت روش دوم: قابلیت استفاده در تمامی نرم افزار ها

نوشتن کد Testbench



1. طرح را در یک بلوک بزرگتر قرار می دهیم
2. به تمام پایه های آن سیگنال متصل می شود (Port map)
3. سیگنال ها را در زمان های دلخواه مقدار می دهیم

مثال: کد TestBench برای گیت Not

Entity ورودی و خروجی ندارد.

```
entity Inv_TB is  
end Inv_TB;
```

[architecture

architecture arch_inv_TB of Inv_TB is

signal i_tb,o_tb:std_logic; سیگنال برای مقداردهی

begin

L0:entity inv port map(i_tb,o_tb); اتصال سیگنال ها

i_tb <= '0', '1' after 100 ns, '0' after 300 ns; مقدار دهی سیگنال

end arch_inv_TB;

[منابع و مآخذ]

1. VHDL (ANALYSIS AND MODELING OF DIGITAL SYSTEMS)-----ZAINALABEDIN NAVABI