

از آنجا که هدف از ایجاد زبان VHDL توصیف کامل سیستمهای فیزیکی می باشد. باید از شیوه مناسبی برای تمایز بین حالات مختلفی که ممکن است یک سیگنال در مدار داشته باشد استفاده نمود. به منظور ایجاد یک مدل استاندارد انجمن IEEE یک سیستم ۹ مقداری را در بسته ای تحت عنوان IEEE std\_logic\_1164 ارائه کرده است.

به منظور درک بهتر از اجزای زبان ، توضیحات را با توصیف VHDL یک مدار تمام جمع کننده که در کد زیر آمده است شروع می کنیم. کلمات کلیدی زبان با حروف پررنگ نمایش داده شده است.

```
library ieee;  
use ieee.std_logic_1164.all;  
  
--definition of a full adder  
  
entity FULLADDER is  
    port(a,b,c:in std_logic;  
          sum,carry:out std_logic);  
  
end FULLADDER;  
  
architecture fulladder_behav of FULLADDER is  
  
    begin  
        sum<=(a xor b) xor c;  
        carry <=(a and b) or (c and (a xor b));  
  
    end fulladder_behav;
```

در ادامه به بررسی اجزای تشکیل دهنده و ساختارهای اصلی زبان می پردازیم

### **:Entity**

Entity یک طرح بیانگر نحوه ارتباط خارجی آن می باشد و عملکرد آن مشابه با بلوکهای مورد استفاده در طراحی شماتیک می باشد که پورتهای تعریف شده در آن معادل با پینهای در بلوکهای شماتیک می باشند.

قالب کلی Entity طرح مشابه با ساختار زیر می باشد.

```
entity NAME_OF_ENTITY is [generic (generic_declarations)];
```

```
    port (signal_names: mode type;
```

```
        signal_names: mode type;
```

```
        .  
        .
```

```
        signal_names: mode type);
```

```
end [NAME_OF_ENTITY];
```

عبارت **mode** بیانگر جهت پورت می باشد و به چهار شکل ممکن زیر بیان می شود :

**In**: نشان دهنده پورتهای ورودی می باشد.

**Out**: بیانگر پورتهای خروجی می باشد.

**Inout**: نشان دهنده پورتهای ورودی- خروجی با چند راه انداز (driver) فعال است.

**Buffer**: نشان دهنده پورتهای ورودی- خروجی با یک راه انداز می باشد.

نکته: باید توجه داشت که مقادیر پورتهای خروجی قابل خواندن نیست. برای حل این مشکل بایداز پورتهای دوطرفه یا بافر استفاده نمود. برای مثال در شکل زیر به منظور صحت توصیف مدار، باید پورت OUT1 از نوع بافر یا دوطرفه تعریف شود.

*entity example is*

```
port(in1,min2,in3 :in std_logic;
```

```
out1,out2 : out std_logic); -- out1: buffer std_logic or out1 : inout std_logic;
```

```
end example;
```

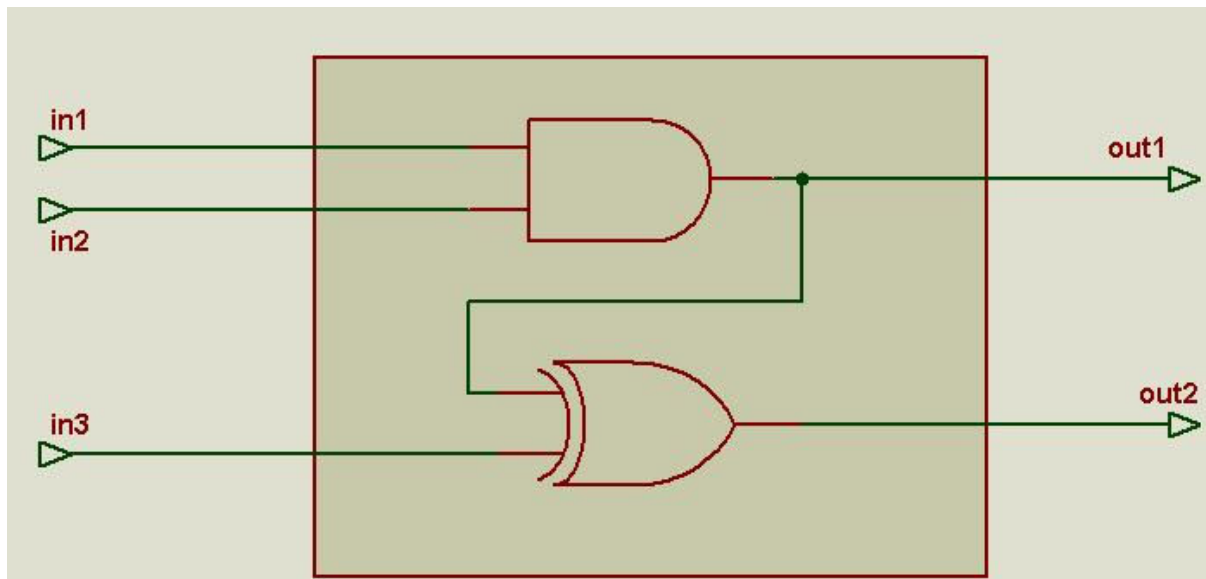
*architecture internal of example is*

```
    begin
```

```
out1 <= in1 and in2;
```

```
out2 <= out1 xor in3;
```

```
end internal;
```



شکل کاربرد پورت دوطرفه

Type بیانگر نوع پورت می باشد. همانگونه که در زبانهای برنامه نویسی کامپیوتری ، متغیرها می توانند انواع مختلفی از قبیل float,string,char, و ... اختیار کنند، پورتها و سیگنالها نیز در زبانهای توصیف سخت افزاری می توانند انواع مختلفی داشته باشند که از ممیان انواع رایج آن می توان به موارد زیر اشاره کرد :

**bit**: می تواند مقادیر 0 یا 1 را داشته باشد.

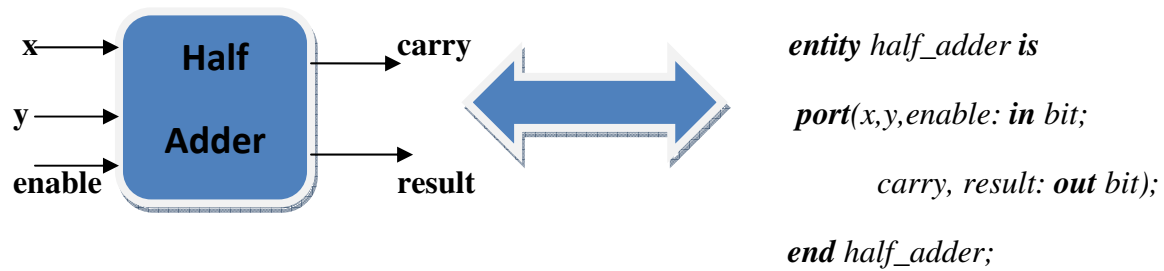
**bit\_vector**: آرایه ای از بیتها می باشد به عنوان مثال bit\_vector(0 to 7) بیانگر یک باس هشت بیتی است.

**std\_logic,std\_logic\_vectro** : توسط انجمن مهندسين برق و الكترونيك تعريف شده است و می تواند مقدار مختلف از جمله 0, 1 و امپدانس بالا اختیار کند.

**boolean**: مقادیر منطقی true,false را شامل می شود.

**integer**: به منظور تعریف مقادیر صحیح به کار می رود.

به عنوان مثال توصیف Entity یک مدار نیم جمع کننده به شکل زیر است.



از عبارت generic برای انتقال مقادیر استاتیک استفاده می شود و استفاده از آن اختیاری است، به عنوان نمونه برای تعیین خطای انتشار یا پهنای باس اطلاعات می توان از این عبارت استفاده نمود:

**entity AND2 is**

**generic** (DELAY : time:=5ns);

**port**(in1,in2 : **in** std\_logic; out1 : **out** std\_logic);

**end** AND2;

**entity counter is**

**generic** (BUS\_WIDTH: integer :=10);

**port** (clk,reset **in** :std\_logic;

count **out** : std\_logic\_vector(BUS\_WIDTH -1 **downto** 0);

**end** counter;