



درس: آزمایشگاه معماری کامپیوتر (جلسه دوم)

- مروری بر VHDL (ادامه)

- آزمایش اول

نیمسال دوم ۱۴۰۰

مفهوم هم روندی یا همزمانی یا Concurrency

طراحی سخت افزار یعنی طراحی به صورت همروند؛ در زبان های HDL ترتیب نوشتن کد عبارات و دستور های Concurrent مهم نیست. همگی همزمان با هم اجرا می شوند. اساس کار دستورات همروند، event است. اصطلاحاً event-driven. در شبیه ساز ها به صورت واقعی همروندی وجود ندارد، بلکه این تصور ما است. در عبارات Sequential دستورات پشت سر هم بررسی و اجرا می شوند در نتیجه ترتیب نوشتن کد ها مهم است. در VHDL بخش همروند Process نام دارد. اما در کل تمام Process ها نیز همروند هستند.

VHDL ۾ Object


<Class> object_name : <Type> := Initial Value ;

- Signal
- Variable
- Constant
- File
- Bit
- Bit_Vector
- STD_Logic
- STD_Logic_Vector
- String
- Integer
- Real

قواعد نام گذاری

- نام یا identifier هر object در VHDL از جمله Array.Port.Signal و ... می تواند شامل حروف انگلیسی، اعداد و

The following table shows the VHDL reserved words.

abs	configuration	impure	null	rem	type
access	constant	in	of	report	unaffected
after	disconnect	inertial	on	return	units
alias	downto	inout	open	rol	until
all	else	is	or	ror	use
and	elsif	label	others	select	variable
architecture	end	library	out	severity	wait
array	entity	linkage	package	signal	when
assert	exit	literal	port	shared	while
attribute	file	loop	postponed	sla	with
begin	for	map	procedure	slr	xnor
block	function	mod	process	sra	xor
body	generate	nand	pure	srl	
buffer	generic	new	range	subtype	
bus	group	next	record	then	
case	guarded	nor	register	to	
component	if	not	reject	transport	

علامت underline باشد.

- مانند سایر زبان ها لازم است به موارد زیر نیز دقت شود:
- ❖ نام ها همواره باید با حروف لاتین شروع شوند.
- ❖ از کلمات reserved یا از پیش تعریف شده نمی توان به تنهایی استفاده نمود.
- ❖ آخرین کاراکتر باید یک حرف یا عدد باشد.
- ❖ استفاده از دو یا بیشتر علامت underline پشت سر هم مجاز نیست.
- ❖ زبان VHDL حساس به بزرگ یا کوچک بودن حروف (Case Sensitive) نیست.

Signal Declaration

تعریف بدون مقدار دهی
(مقداردهی میتواند درون برنامه یا مقدار اولیه باشد)

```
--Declaration
signal one_bit : bit;

signal vector_of_bits_descending : bit_vector(7 downto 0);

--signal vector_of_bits_no_lenght : bit_vector;    --unconstrained is not valid

signal vector_of_bits_ascending : bit_vector(0 to 7);

signal one_stdlogic : std_logic;

signal vector_of_stdlogic_descending : std_logic_vector(3 downto 0);

signal vector_of_stdlogic_ascending : std_logic_vector(0 to 3);

signal vector_of_stdlogic_typical : std_logic_vector(2 to 7);

signal int_signal_no_lenght : integer;
```

نکات Signal Declaration

- محل تعریف Signal و Constant درون Architecture و قبل از begin است.
- Package مورد استفاده ما در اکثر برنامه‌ها std_logic است. از bit و bit_vector کمتر استفاده می‌کنیم.
- برای Comment کردن از دو خط تیره استفاده می‌کنیم.
- اندیس‌ها را معمولا به صورت نزولی می‌نویسند مثلا (0 downto 3) تا با ارزش مکانی بیت‌ها مشابه باشد.
- تعریف و کاربرد Constant فرقی با Signal ندارد، تنها تفاوت این است که مقدار Constant همانطور که از اسمش پیداست همواره ثابت است و درون برنامه قابل تغییر نیست.
- واحدهای زمانی در VHDL از فمتو ثانیه (fs) تا ساعت (hr) تعریف شده است.

مقداردهی اولیه Signal و Constant

```
--initialization
signal one_bit_init1 : bit := '1';

--signal one_bit_init2 : bit := 1;    --incorrect

signal vector_of_bits_desc_init : bit_vector(7 downto 0) := "11110000";

signal vector_of_bits_desc_init_hex : bit_vector(7 downto 0) := X"F0";

signal vector_of_bits_desc_init_octa : bit_vector(8 downto 0) := O"360"; --should be 3*3 bits because of OCTA

signal vector_of_bits_asc_init : bit_vector(0 to 7) := "01110100";

--signal one_stdlogic_init : std_logic := 1;    --incorrect

signal vector_of_stdlogic_desc_init : std_logic_vector(3 downto 0) := X"A"; --X'A' is incorrect
```

مقداردهی اولیه Constant و Signal

```
signal vector_of_stdlogic_asc_init : std_logic_vector(0 to 3) := "0011";
signal vector_of_stdlogic_tvp_init : std_logic_vector(2 to 7) := "111000";
signal int_signal_no_lenght_init : integer := -6;
--signal int_signal_pos_init : integer range 0 to 127 := -4;    --incorrect
signal int_signal_neg_init : integer range -16 to 15 := 10;
signal boolean_signal_init : boolean := true;
signal time_signal_init : time := 5 ps;    --10 ns    --20 fs;
signal string_signal_init : string(1 to 10) := "My Message";
signal char_signal_init : character := 'M';    -- "M" is wrong
```


نکات مقداردهی اولیه Signal و Constant

- در صورت یکسان بودن type چند Signal یا Constant یا ...، می‌توان آنها را یکجا تعریف نمود:
`signal a,b,c : std_logic := '1';`
داخل برنامه حتی اگر مقادیر یکسان باشند هم باید جدا جدا مقداردهی کرد و این مورد فقط در زمان تعریف است.
- جهت تعریف Constant کافی است در دو صفحه اخیر، به جای کلمه Signal، Constant جایگزین کنیم (مقدار اولیه حتما باید Assign شود).

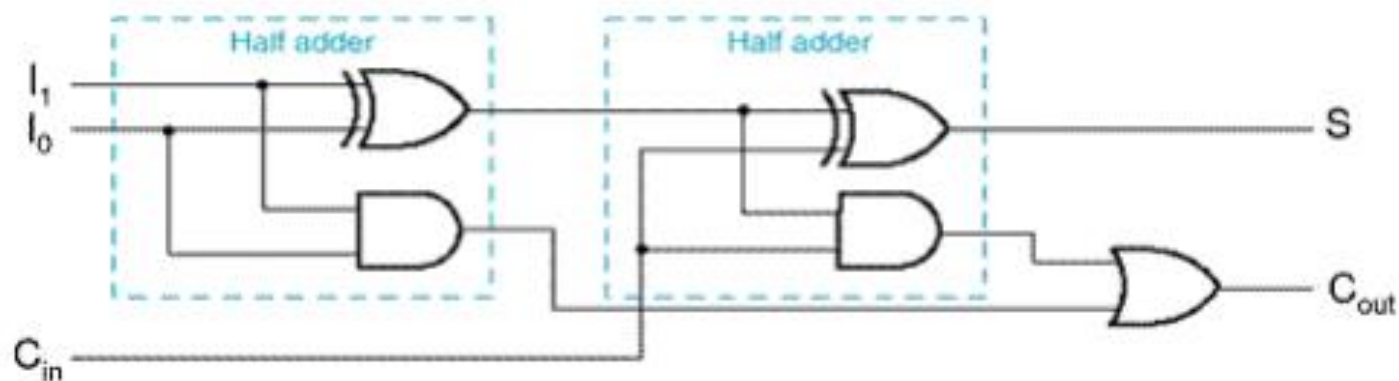
```
--CONSTANT
-- All the same
constant const_bit : std_logic := '1';
constant const_vector : std_logic_vector(3 downto 0) := "1011";
```

آزمایش اول:

- توصیف مدارهای پایه (or, and, xor)
- توصیف یک نیم جمع کننده (HA)
- توصیف یک تمام جمع کننده (FA)

طراحی ساختاری

در این روش طراحی، هر موجودیت از چند کامپوننت تشکیل شده است. مثلاً یک جمع کننده تک بیتی از دو نیم جمع کننده و یک گیت OR تشکیل شده است (مانند شکل ۱).



شکل ۱ جمع کننده تک بیتی

ISE Project Navigator (P.20131013) - D:\selfcheck\try2\try2.xise - [or_gate (RTL2)]

File Edit View Project Source Process Tools Window Layout Help

Design

View: ☒ Implementation ☐ Simulation

Hierarchy

- xc5vbx110t-3ff1136
 - FA - Gate_Level (FA.vhd)
 - a - Behavioral (a.vhd)
 - full_adder - Gate_Level (FA.vhd)
 - or_gate - Behavioral (or_gate.vhd)
 - FAXOR (FAXOR.sch)
 - TMR (TMR.sch)
 - XLXI_29 - TRV - RTL (tmrbuf.vhd)
 - lut (lut.sch)
 - testtmr (testtmr.sch)

No Processes Running

Processes: or_gate - Behavioral

- Design Summary/Reports
- Design Utilities
 - Create Schematic Symbol
 - View Command Line Log File
 - View HDL Instantiation Template
- User Constraints
- Synthesize - XST
 - View RTL Schematic
 - View Technology Schematic
 - Check Syntax

Start Design Files Libraries

View by Category

Design Objects of Top Level Block

Instances

- or_gate
- z_imp_z1

Pins

- or_gate

Signals

- or_gate

Properties of Instance: z_imp_z1

Name	Value
Verilog Model	OR2
VHDL Model	OR2
Type	or2

Console Errors Find in Files Results View by Category

[24,284]

New Source Wizard

Select Source Type

Select source type, file name and its location.

- BMM File
- ChipScope Definition and Connection File
- Implementation Constraints File
- IP (CORE Generator & Architecture Wizard)
- MEM File
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

File name:

Location:

D:\selfcheck\try2

☒ Add to project

More Info Next > Cancel

ISE Project Navigator (P.20131013) - D:\selfcheck\try2\try2.xise - [FA.vhd]

File Edit View Project Source Process Tools Window Layout Help

Design

View: ☒ Implementation ☐ Simulation

Hierarchy

- try2
 - xc5vbx110t-3ff1136
 - full_adder - Gate_Level (FA.vhd)
 - or_gate - Behavioral (or_gate.vhd)

No Processes Running

No single design module is selected.

Design Utilities

- Update All Schematic Files
- Compile HDL Simulation Libraries
- Regenerate All Cores
- Check All Core Versions

Start Design Files Libraries

Console

```
WARNING:ProjectMgmt - File D:/selfcheck/try2/FA.stx is missing.  
WARNING:ProjectMgmt - File D:/selfcheck/try2/TMR.stx is missing.  
WARNING:ProjectMgmt - File D:/selfcheck/try2/full_adder.spl is missing.
```

Ln 13 Col 1 VHDL

Type here to search

13

New Source Wizard

Define Module

Specify ports for module.

Entity name HA

Architecture name Behavioral

Port Name	Direction	Bus	MSB	LSB
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

More Info < Back Next > Cancel



```
1
2 ----- Library and Packages -----
3 library ieee;
4 use ieee.std_logic_1164.all;
```

New Source Wizard X

Project Navigator will create a new skeleton source with the following specifications.

Entity name: HA
Architecture name: Behavioral
Port Definitions:

Cancel

No single design module is selected.

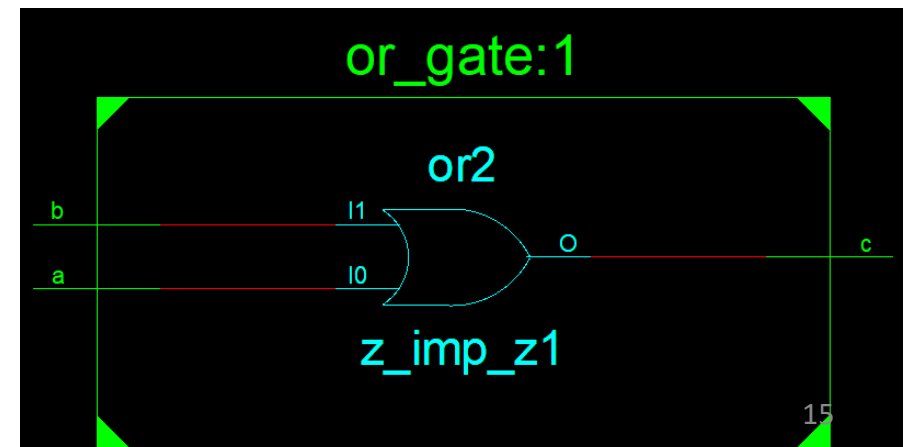
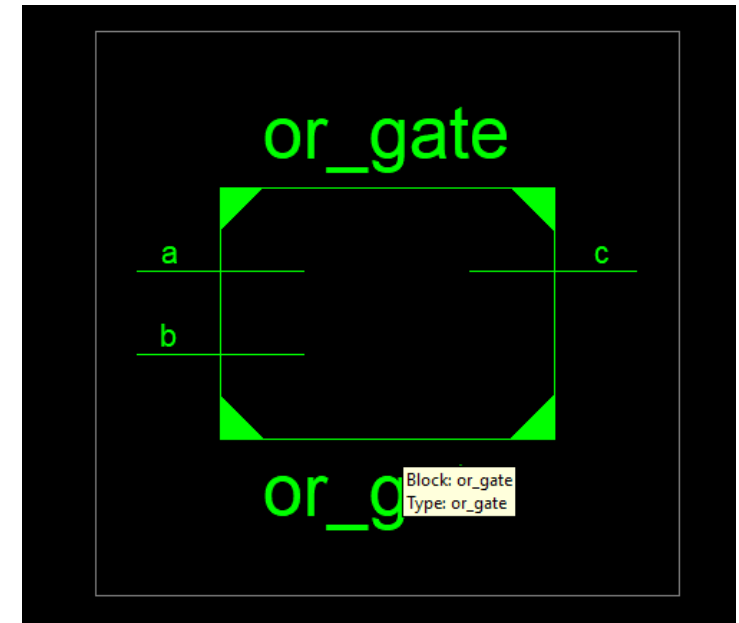
- Update All Schematic Files
- Compile HDL Simulation Libraries
- Regenerate All Cores
- Check All Core Versions

Console

12:17 AM
2/23/2022

Or: برای عدم استفاده مستقیم از گیت ها برای هر گیت مورد نیاز نیز یک توصیف رفتاری نوشتیم و از component مورد نظر در مواقع نیاز نمونه گیری کردیم.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity or_gate is
4   Port ( a : in STD_LOGIC;
5         b : in STD_LOGIC;
6         c : out STD_LOGIC);
7 end or_gate;
8 architecture Behavioral of or_gate is
9   signal z : STD_LOGIC;
10 begin
11   z <= a or b;
12   c <= z;
13 end Behavioral;
```





Design

View: ☒ Implementation ☐ Simulation

Hierarchy

try2

xc5v1x110t-3ff1136

HA - Behavioral (HA.vhd)

full_adder - Gate_Level (FA.vhd)

or_gate - Behavioral (or_gate.vhd)

New Source...

Add Source...

Add Copy of Source...

Open

Remove

Manual Compile Order

Set as Top Module

SmartGuide...

Implement Top Module

File/Path Display

Expand All

Collapse All

Find... Ctrl+F

Design Properties...

Source Properties...

No Processes Running

Processes: or_gate - Beh

Design Utility

Create So

View HDI

Check Syntax

Start Design

Console

WARNING:ProjectMgm

WARNING:ProjectMgmt - File D:/selfcheck/try2/TMR.stx is missing.

WARNING:ProjectMgmt - File D:/selfcheck/try2/full_adder.spl is missing.

Set the selected module as the top of the design

Ln 23 Col 10 VHDL

Type here to search

16

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity half_adder is
4   Port ( ah : in STD_LOGIC ;
5         bh : in STD_LOGIC ;
6         s : out STD_LOGIC;
7         cout : out STD_LOGIC );
8 end half_adder;
9 Carry
10 Sum A i
11 B i
12 architecture Behavioral of half_adder is
13   signal z : STD_LOGIC;
14   component xor_gate is
15   port (
16     a,b :in STD_LOGIC;
17     c : out STD_LOGIC);
18 end component xor_gate;
19 component and_gate is
20   port(
21     a,b :in STD_LOGIC;
22     c : out STD_LOGIC);
23 end component and_gate;
24 begin
25   xor_ins : xor_gate port map ( a => ah , b => bh ,c => s);
26   and_ins : and_gate port map ( a => ah , b => bh ,c => cout);
27 end Behavioral;

```

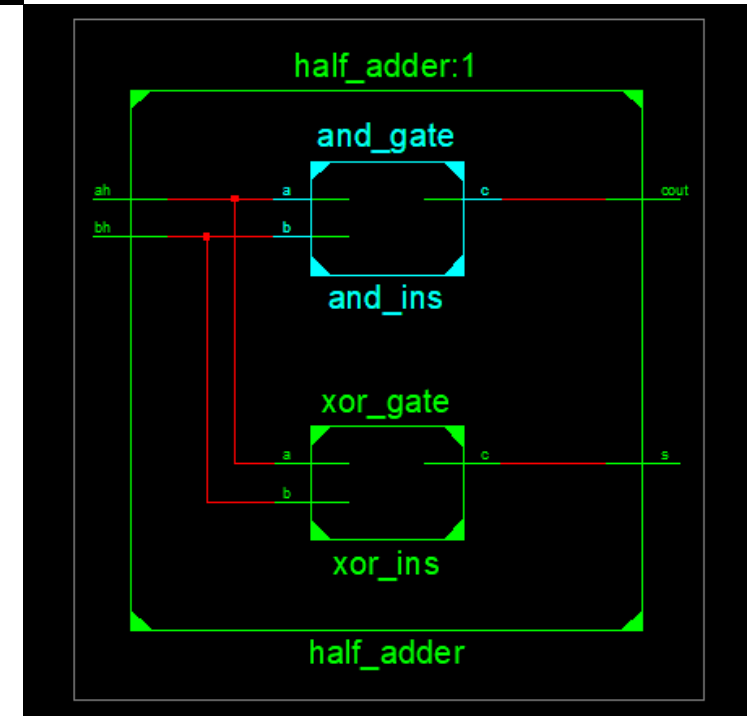
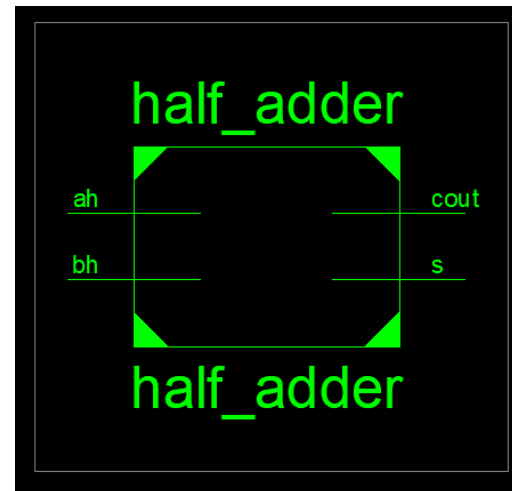
FA.vhd

HA.vhd*

AND , XOR =?

HA

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity half_adder is
4      Port ( ah : in STD_LOGIC ;
5            bh : in STD_LOGIC ;
6            s : out STD_LOGIC;
7            cout : out STD_LOGIC );
8  end half_adder;
9
10 architecture Behavioral of half_adder is
11     --signal z : STD_LOGIC;
12     component xor_gate is
13     port (
14         a,b :in STD_LOGIC;
15         c : out STD_LOGIC);
16     end component xor_gate;
17     component and_gate is
18     port(
19         a,b :in STD_LOGIC;
20         c : out STD_LOGIC);
21     end component and_gate;
22     begin
23     xor_ins : xor_gate port map ( a => ah , b => bh ,c => s);
24     and_ins : and_gate port map ( a => ah , b => bh ,c => cout);
25 end Behavioral;
```



FA=?

بررسی صحت عملکرد

