درس: آزمایشگاه معماری کامپیوتر

مریم محبتی

دانشکده کامپیوتر دانشگاه علم و صنعت

maryam_mohabbati@comp.iust.ac.ir

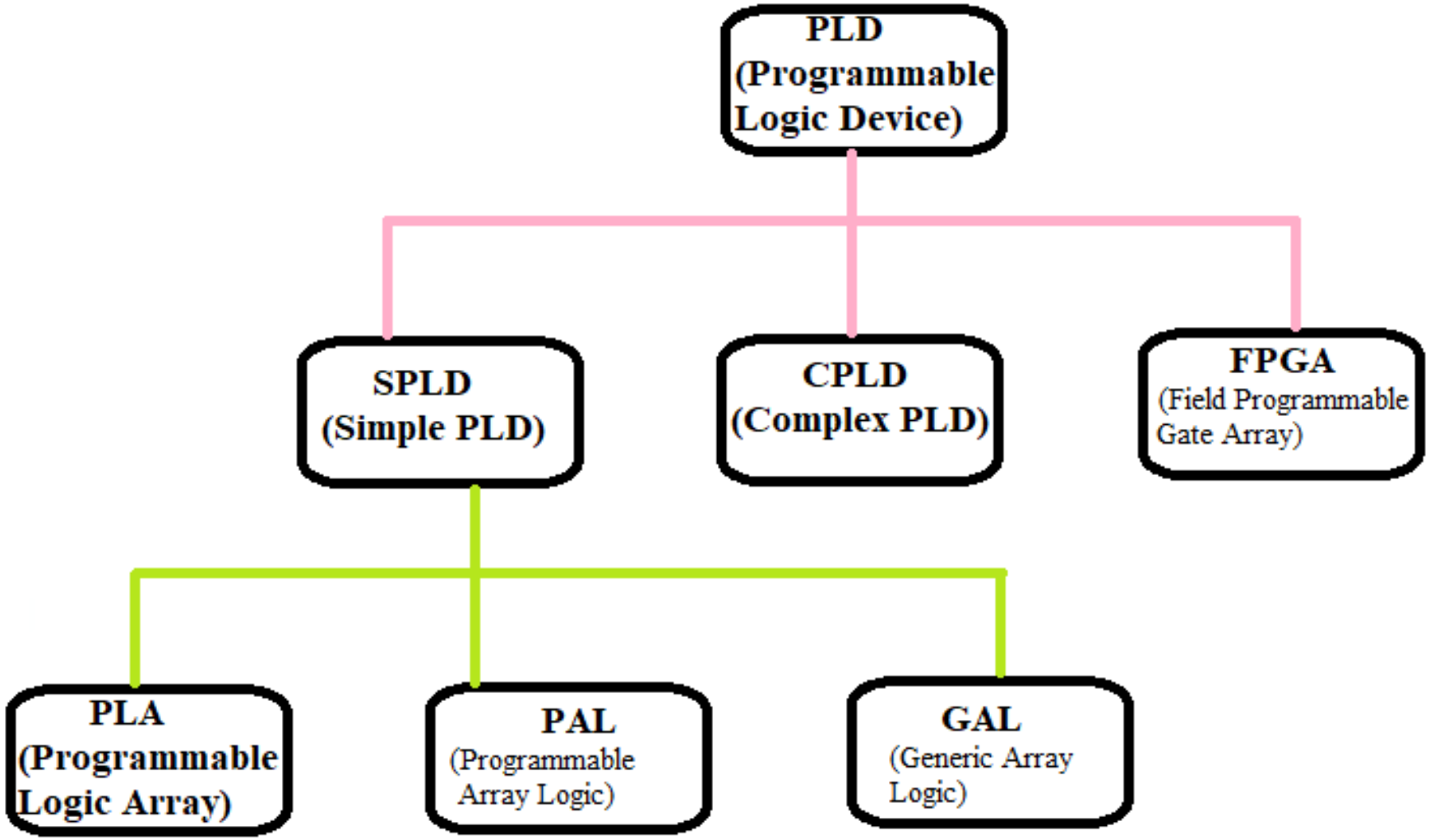
نیمسال دوم ۱۴۰۰

- گروه بندی
- یک جلسه غیبت مجاز است
- نرم افزار ISE 14.7 (تراشه fpga)
- زبان VHDL
- ۱۳ آزمایش(۱۳)
  - توصیف و هدف از انجام آزمایش
  - کد vhdl قابل اجرا
  - بررسی درستی عملکرد مدار
  - نتایج کامپایل و شبیه سازی
- پروژه نهایی (۳)
- امتحان(۴)
- حضور فعال در کلاس(۲)

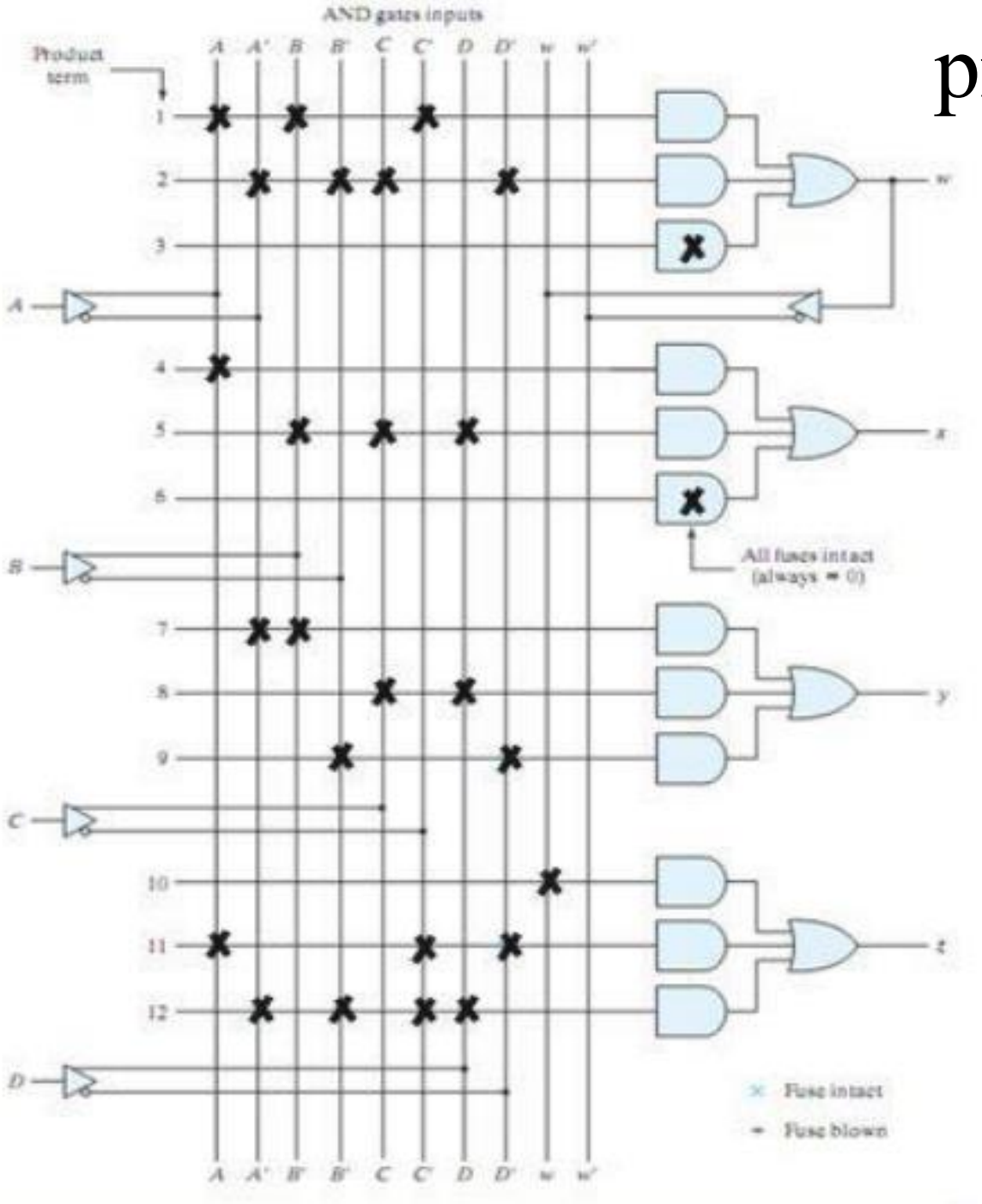
مزایای مدارهای دارای قابلیت برنامه ریزی

- امکان تغییر مدار به صورت دلخواه و به روزرسانی
- عدم نیاز به تهیه قطعات و ICهای تجاری
- فضای فیزیکی بسیار کوچکتر
- سرعت و عملکرد بهتر
- تلفات توان کم
- از نظر ساخت و تولید مقرون به صرفه تر
- دارای قابلیت اطمینان بالاتر به دلیل وجود ICها و مدار کمتر
- امکان شبیه سازی توسط نرم افزار قبل از پیاده سازی

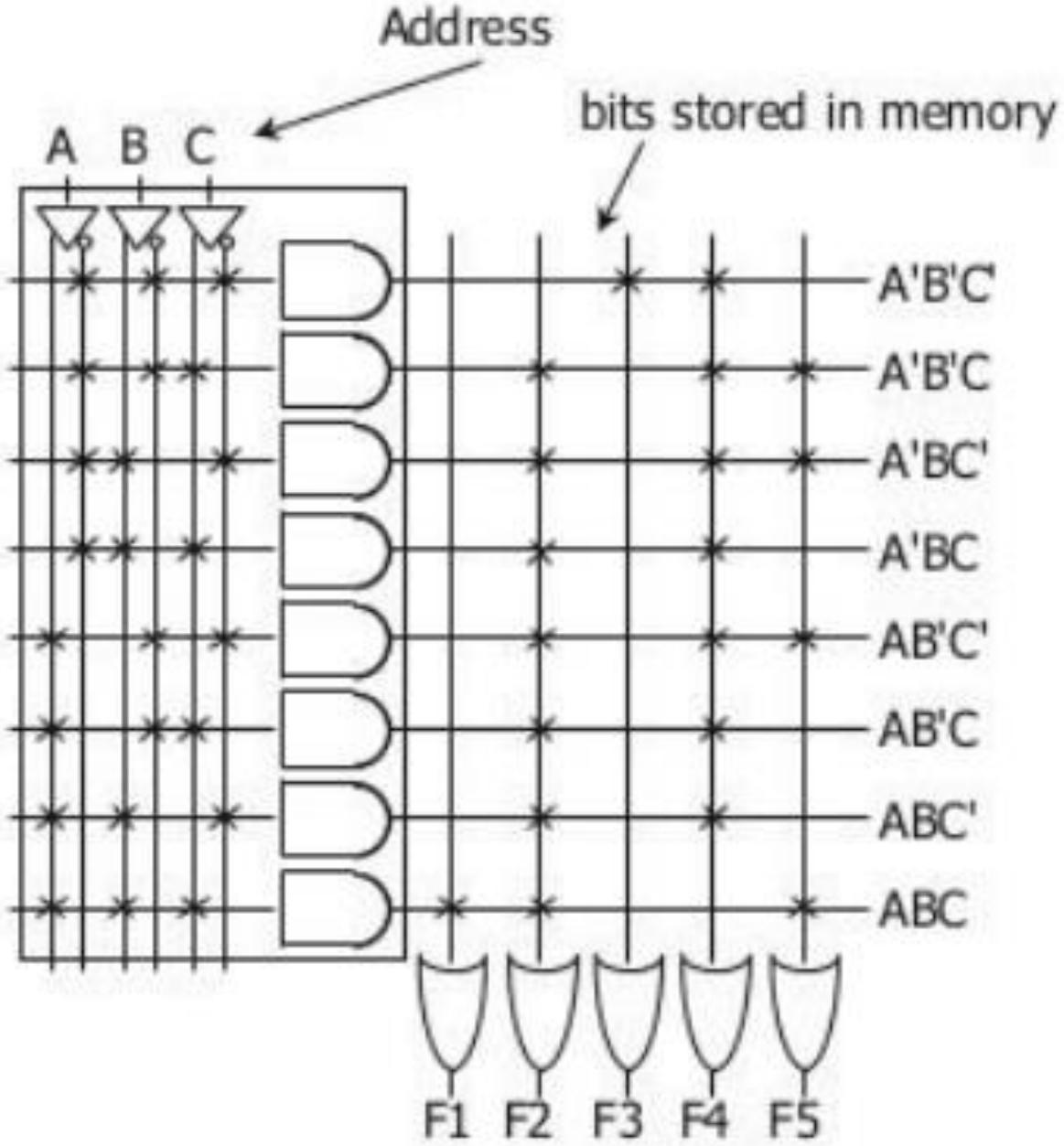
مروری بر مدار های قابل برنامه ریزی



programmable array logic



programmable logic array



مروری بر مدار های قابل برنامه ریزی(ادامه)

PLD (Programmable Logic Device) مدارهای مجتمع دیجیتال هستند که دارای قابلیت برنامه ریزی مجدد می باشند.

SPLDها ساده ترین، کوچکترین و ارزان ترین قطعات در بین PLDها هستند که شامل آرایه ای از گیت ها AND،OR و NOT و همچنین اتصالات یا interconnection و بلوکهای IO Buffer می باشند. برخی از آنها شامل ۴ تا ۲۲ عدد Macrocellهستند. این بلوک ها حاوی گیت های مزبور و همچنین Flip-Flop می باشند.

SPLDها در کل دارای کمتر از حدودا ۶۰۰ گیت می باشند.

Functionality اکثر SPLDها با استفاده از Fuseها یا حافظه هایnon-Volatile مانند EPROM EEPROM, FLASH و غیره تعیین و برنامه ریزی می شود.

مروری بر مدار های قابل برنامه ریزی(ادامه)

- CPLDها دارای بخش های متعددی مانند آرایه های تماما برنامه پذیر یا Programmable Interconnected Array) PIA)، گیت های ANDو OR، بلوکهای Macrocell، بلوکهای کنترل کننده IOو حافظه می باشند.
- مهم ترین بلوک CPLDها ماکروسل است که امکان ایجاد انواع مدارهای ترکیبی و ترتیبی و فیدبکهای بین آنها را از طریق تغییر اتصالات میانی به وجود می آورد.
- برنامه ریزی بر اساس EEPROM
- کاربرد در سیستم ها با تعداد FF کم و گیت های زیاد

مروری بر مدار های قابل برنامه ریزی(ادامه)

FPGA (Field Programmable Gate Array):

با افزایش میزان پیچیدگی سیستم های دیجیتال مجبور به تغییر معماری CPLDمی شویم؟

بدین ترتیب به معماری FPGAمی رسیم.

FPGAقطعه ای چند منظوره است که دارای قابلیت برنامه ریزی چند سطحی توسط کاربر می باشد. دارای Logic Cellهای بسیار و همچنین ماتریسی از اتصالات و سیم ها است که توسط سوییچها قابل برنامه ریزی هستند. به علاوه شامل بسیاری از بلوک های دیگر از جمله , Clk Mngm، ضرب کننده، DSP block،ALU، حافظه و ..

مروری بر مدار های قابل برنامه ریزی(ادامه)

انواع مختلفی از معماری و ساختارها وجود دارند که توسط شرکتهای مختلف در ساخت FPGAکاربرد دارند.

مرسوم ترین نوع پیاده سازی بلوکهای منطقی (LUT (Look Up Table وMux-Based می باشد.

همچنین مرسوم ترین تکنولوژی Programmingیا اتصالات درونی SRAMوEEPROM/Flash

است.

Hardware Description Language (HDL)

زبان توصیف سخت افزار یا زبان سخت افزاری ابزاری است برای توصیف عملکرد، ساختار و رفتار مدارهای الکترونیکی بخصوص مدارهای الکترونیکی دیجیتال

دارای قابلیت تجزیه و تحلیل مدار به صورت اتوماتیک، شبیه سازی و همچنین عیب یابی مدارهای الکترونیک

خروجی این زبان پس از Synthesis یا سنتز مدار یک Netlist است که شامل اطلاعات فیزیکی مدار طراحی شده است و در ادامه می توان آن را در تراشه مورد نظر مانند FPGA پروگرام نمود.

دارای Structure ، Statement Expression ،Syntax مانند دیگر زبان های برنامه نویسی نرم افزاری مانند C است؛ به علاوه موضوع زمان یا Timing که مستقیما در HDL مورد استفاده قرار می گیرد.

دو زبان توصیف سخت افزار اصلی : VHDLو Verilog

زبان های دیگری مانند SystemCنیز وجود دارند که در واقع مجموعه ای از Classها و Macroها در زبان C/C++ است که امکان شبیه سازی event-drivenرا ایجاد می کند. کاربرد آن شبیه سازی و مدلسازی است.

هدف از ایجاد زبانهای توصیف سخت افزار: ایجاد قابلیت برنامه پذیری و تغییر و پیکربندی مدارهای الکترونیک (دیجیتال)، عدم درگیر شدن در تکنولوژی پیاده سازی برای طراحی موردنظر و آسان تر شدن توصیف سطح بالای انواع مدارهای الکترونیک (دیجیتال)

Hardware Description Language (HDL)(cont.)

اهداف طراحی از جمله Latency (delay)، Functionality Test Coverage ، Throughput Power و Area Consumptionرا می توان با استفاده از HDLدر حین طراحی بررسی نمود.

در واقع امکان ارائه مفاهیم زبانی سخت افزار توسط زبان های برنامه نویسی سنتی مانند C ++وجود دارد البته لازم است Class Libraryهای خاصی به آن اضافه شود.

به طور کلی به دلیل عدم وجود زمان در زبان های نرم افزاری، نمی توان عملکرد آنها را مشابه HDL دانست.

Hardware Description Language (HDL)(cont.)

HDLs for Digital Circuit Design

VHDL, Verilog, SystemC, ABEL, AHDL, Bluespec, BSV, Chisel, MyHDL, HHDL, Confluence, CoWareC, CUPL, Handel-C, HJJ, JHDL, LavaHDL, Lola HDL, Mentor Graphics, PALASM, SystemVerilog

HDLs for Analog Circuit Design

AHDL, SpectreHDL, Verilog-AMS, VHDL-AMS, HDL-A

PHDL

HDL for Printed Circuit Design

آشنایی با شرکت های سازنده FPGAو نرم افزارهای طراحی، شبیه سازی و پیاده سازی

.

تا سال ۲۰۱۶، اولین و قدیمی ترین شرکت های سازنده FPGAیعنی Xilinxو (Altera (intel پیشتاز در بازار بوده اند.

شرکت Xilinxحدود ۵۰ درصد و (Altera (Intelحدود ۴۰ درصد از کل بازار دنیا را در دست داشتند.

در صنعت و دانشگاههای ایران اکثرا از FPGAهای ساخت همین دو شرکت استفاده میشود.

این دو شرکت دارای نرم افزارهایی قدرتمند به منظور طراحی، آنالیز، شبیه سازی و سنتز (کامپایل) هستند که به ترتیب Xilinx ISE / Vivado و Altera Quartusنام دارند و دارای نسخه های رایگان در محیطهای Windowsو Linux نیز هستند.

VHDL:

V = Very high-speed integrated circuit

HDL = Hardware Description Language

VHDLزبانی استاندارد به منظور توصیف، مدلسازی، سنتز و شبیه سازی مدارها و سیستم های دیجیتال است.

ویژگی ها و ابزارهای VHDL

ابزارهای موجود در VHDL

- ویژگی های کلی خواسته شده در VHSICمانند دسته بندی، توصیف سطح بالا، شبیه سازی، مدلسازی، سنتز و ارزیابی طراحی مورد نظر
- امکان طراحی همروند (Concurrentو ترتیبی (Sequential / Procedural
- طراحی سلسله مراتبی؛ به معنی طراحی چند سطحی. طراحی شامل دو بخش interfaceو description، subprograms، types و Primitivesو طراحی های مختلف با امکان
- پشتیبانی از Libraryو Package، شامل دسترسی کاربران
-طراحی Generic؛ یعنی بتوان پارامترها را از خارج طراحی تغییر داد و لازم نباشد برای هر تغییر پارامتری، کل طراحی مجدد انجام شود

ابزار های موجود در VHDL

- تعریف Typeعلاوه بر typeهای پیش تعریف HDLها مانند Bitو Boolean
- تعریف مجدد اپراتورها و توابع پیش تعریف مانند جمع، ضرب و غیره برای typeهای جدیدی مانند Floating-Pointو integer
- تعریف Subprogramمانند Functionو Procedure
- کنترل زمانی یا Timingدر تمامی سطوح طراحی؛ برای مثال تاخیر، مقداردهی در زمان خاص و ایجاد Clockیا PWM
- طراحی Structuralبه معنی طراحی سخت افزارهای کوچکتر و استفاده از آنها در طراحی های بزرگتر. امکان ترکیب و تکرار طراحی آنها

شروع کار با نرم افزار Xilinx ISE

با عنوان ISE Design Suite 14.7 یا ISE Project Navigator

اپراتورها

Boolean Operators	NOT	AND	OR	NAND	NOR	XOR	XNOR
Comparison Operators	=	/=	<	<=	>	>=	
Arithmetic Operators	+	-	*	/	MOD	REM	ABS

از Package زیر استفاده می‌شود:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL; -- or SIGNED
```

اپراتورها(ادامه)

اپراتورهای بولی یا منطقی بین دو یا چند عملگر واقع می شوند. تعداد بیت های عملگر ها می بایست برابر باشد، (بجز NOT)؛ در غیر این صورت نتیجه ممکن است اشتباه باشد. خروجی نیز با همان تعداد بیت است.

خروجی اپراتورهای مقایسه از نوع بولی true یا falseاست.

تعداد بیت عملگرها و نتیجه مهم است. مثلا در عملیات جمع اگر احتمال وجود بیت Carryوجود داشته باشد، می بایست لحاظ شود. یا مثلا اینکه خروجی ضرب دو عدد nو mبیتی دارای n +m بیت است باید لحاظ شود.

اپراتورهای ریاضیاتی در واقع برای typeهای integerو realتعریف شده اند، اما با اضافه کردن پکیج می توان برای typeهای دیگر نیز بعضا استفاده نمود. لازم است که هر دو عملگر از یک typeباشند

اپراتورها(ادامه)

در هنگام استفاده از اپراتورهای ضرب، تقسیم، توان و غیره لازم است هزینه های سخت افزاری و زمانی و دیگر مسائل را در نظر داشته باشیم (بهتر است خودمان طراحی کنیم).

اپراتورهای توان، تقسیم بر غیر از عدد ۲، mod و rem معمولا قابل سنتز نیستند مگر اینکه عملگرها عدد ثابت باشند.

در عملیات ریاضی، می توان علامت دار (signed) بدون علامت (unsigned) بودن عملگرها را لحاظ نمود.

اپراتورها(ادامه)

Shift Operator	توضیح
SLL	Shift Left Logical
SLA	Shift Left Arithmetic
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
ROL	Rotate Left Logical
ROR	Rotate Right Logical

این اپراتور ها برای typeهای bit_vector و تعدادی برای std_logic_vector تعریف شده اند.
بهتر است از Package زیر استفاده شود: (علاوه بر قبلی ها)

```
use IEEE.NUMERIC_STD.ALL;
```

Type های پر کاربرد

Bit 0 & 1 Logic	Bit_Vector Array of Bit
Std_Logic 0, 1, Z, X, U, ... Logic	Std_logic_Vector Array of Std_logic
Integer اعداد صحیح	Real اعداد اعشاری
Character کاراکترهای ASCii	String Array of Char
Time زمان فیزیکی	Boolean True / False

Logic یا منطق‌های استاندارد در VHDL

0 LOW	1 HIGH	Z HIGH IMPEDANCE	X UNKNOWN	- DON’T CARE	H WEAK HIGH	L WEAK LOW	U UNINITIALIZED	W WEAK UNKNOWN

توصیف یک سیستم با VHDL : موجودیت سخت افزاری

```
Entity mobile is
...
End entity mobile;
```

هر سیستم شامل دو جزء اصلی: ورودی/خروجی و بدنه سیستم

- مثل: RAM، CPU، مودم، ماوس، کامپیوتر، موبایل و مانند آن.

توصیف یک سیستم با VHDL : پورت‌های ورودی و خروجی

```
Entity and_gate is
   Port(
       A, B: in bit;
       C   : out bit
   );
End entity and_gate;
```

توصیف یک سیستم با VHDL : بدنه سیستم

```
Entity and_gate is
   Port(
       A, B: in bit;
       C   : out bit
   );
End entity and_gate;


Architecture behavioral of and_gate is
Begin
   C <= A and B;
End behavioral;
```

نام ماجولی که آن را توصیف می‌کنیم. این ماجول یک گیت AND است.	And_gate
دو ورودی گیت AND	A, B
خروجی گیت AND	C
سطحی که این کد را در آن توصیف می‌کنیم.	behavioral
یک نوع (type) است. مثل int در زبان C. سیگنال‌های از این نوع مقادیر '0' یا '1' می‌گیرند.	bit
عملگر مقداردهی است.	<=

معرفی کتابخانه IEEE

توصیف یک سیستم با VHDL : بدنه سیستم

```
library IEEE;
use IEEE.std_logic_1164.all;

Entity and_gate is
   Port(
       A, B: in std_logic;
       C   : out std_logic
   );
End entity and_gate;

Architecture behavioral of and_gate is
Begin
   C <= A and B;
End behavioral;
```

استفاده از std_logic_vector

```
Port(
    A: in  std_logic_vector(9 downto 0);
    C: out std_logic_vector (9 doento 0)
);

Signal b: std_logic_vector(9 downto 0);
```

مقداردهی به std_logic_vector

```
B(9 downto 1) <= B(8 downto 0); -- shift
B(0) <= '1';
B(2 downto 0) <= "110";
B <= A;
B(9 downto 8) <= A(2 downto 1);
B(4) <= B(5);
B <= (others => '0');
B(5 downto 0) <= A(9 downto 7) & "011";
```