Department of Computer Science & Engineering



# 计算机系统结构实验指导书-LAB1

#### **1.** OVERVIEW

#### 1.1 实验名称

FPGA 基础实验: LED Flow Water Light

- 1.2 实验目的
- 1. 掌握 Xilinx 逻辑设计工具 Vivado 的基本操作
- 2. 掌握使用 Verilog HDL 进行简单的逻辑设计
- 3. 掌握功能仿真
- 4. 使用 1/0 Planing 添加管脚约束
- 5. 生成 Bitstream 文件
- 6. 上板验证
- 1.3 实验预计时间 120分钟
- 1.4 实验报告与验收办法
  - 1) 实验报告和工程文件在第十一周星期二晚上 23 点前提交

2) 无法验收,但报告中需有仿真结果的截图和实验过程中有收获的心得或 反思文字

PS: 1) 云主机只有 C 盘可用, 可事先在 C 盘建立一个工作目录, 比如: C:\Arch1abs 文件 夹, 用于存放实验工程文件和预先给定的 IP 核等数据

2)本实验1和实验2皆是按部就班的验证实验,只要认真细致按照指导步骤都能得出实验结果(用于观察分析的仿真波形图),目的是通过简单的基础实验熟悉实验开发环境、用硬件描述语言来设计基本逻辑、通过仿真检验电路设计是否预期,掌握硬件开发的基本实验流程

# 2. EXPERIMENTAL STEPS

# 2.1 新建工程

1. 启动桌面 Vivado 2018.3 开发工具或点击左下角窗口选择 Xilinx Design Tools-> Vivado 2018.3,如图所示



图 2-1 运行 Vivado

2. 点击 Create Project

🔥 Vivado 2018.3	
Eile Flow Iools Window Help Q- Quick Access	
HLx Editions	
Quick Start Create Project > Open Project > Open Example Project >	
Tasks Manage IP > Open Hardware Manager > Xillinx Tcl Store >	
Learning Center Documentation and Tutorials > Quick Take Videos > Release Notes Guide >	

图 2-2 新建工程

3. 弹出 New Project 向导,由此建立一个新工程,点击 Next

🝌 New Project		×
HLX Editions	Create a New Vivado Project This wizard will guide you through the creation of a new project. To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.	
<b>£</b> XILINX.		
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel	

图 2-3 新建工程向导

4. 输入工程名称 lab01,建议工程存放位置 E:/Archlabs,确认勾选 Create project subdirectory 后,点击 Next

PS:工程名称和存储路径中不能出现中文和空格,建议工程名称以字母、数字、下划 线来组成

🝌 New Project				×
Project Name Enter a name for your project and specify a directory where the p	roject data files will be sto	ored.		4
Project name: ab01 Project location E:/archlabs				8
Project will be created at: E:/archlabs/lab01				
?	< <u>B</u> ack	<u>N</u> ext ≻	Einish	Cancel

图 2-4 工程名称和路径

5. 选择 **RTL Project** 工程类型, 勾选 **Do not specify sources at this time** 在创建 工程时不决定 sources 文件, 点击 **Next** 

A New Project	×
Project Type Specify the type of project to create.	4
RTL Project You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.	
20 not specify sources at this time	
<ul> <li>Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.</li> </ul>	
Do not specify sources at this time	
<ul> <li>I/O Planning Project</li> <li>Do not specify design sources. You will be able to view part/package resources.</li> </ul>	
<ul> <li>Imported Project</li> <li>Create a Vivado project from a Synplify, XST or ISE Project File.</li> </ul>	
Example Project Create a new Vivado project from a predefined template.	
	Cancel

图 2-5 RTL 工程

6. 选择 SWORD4.0 的 FPGA 参数: Family 选 Kintex-7, Package 选 ffg676, Speed grade 选-2; 接着具体型号中选 xc7k325tffg676-2, 点击 Next

Parts   Reset All	Boards Filters								
Category:	All		~	Package: ffg	676	∼ Temper	ature: All Ren	maining	~
<u>S</u> earch:	Q-		~	*					
Part		I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	G
xc7k160	ttfg676-2	676	400	203800	202800	325	0	600 840	8
xc7k410	tffg676-2	676	400	254200	508400	795	0	1540	8

图 2-6 FPGA 型号参数

7. 弹出新工程信息综述,点击 Finish 结束工程创建

🍌 New Project		×
VIVADO Hur Editor	New Project Summary         A new RTL project named Tab01* will be created.         The default part and product family for the new project.         Default Part ac/R1280th/6762         Product Kintes-7         Parkage: tbp76         Speed Grade: -2	
E XILINX.	To create the project, click Finish	
	图 2-7 新建工程完毕	

#### 2.2 Vivado 整体界面

如图 2-8 是 Vivado2018.3 的整体界面,大致分为四个区:

- 左侧区 Flow Navigator 包含整个开发流程,像 Project Settings、Run Simulation、Run Synthesized 和 Generate Bitstream 等;
- 2. 中间区 通常显示当前工程包含的文件树结构,提供工程文件的管理;
- 3. 右侧区 会显示工程信息、打开的编辑文件等;
- 4. 下部区 显示各种信息状态。

🍌 lab01 - [D:/archlabs/lab01/lab01.xp	or] - Vivado 2018.3											-	a ×
Eile Edit Flow Icols Rep	ogrts Window Layout View Help Q.Q	uick Access											Ready
■ + + E B X	▶, # \$ ∑ % % %											🗮 Default Lay	out 🗸
Flow Navigato = 0 ?	PROJECT MANAGER - lab01												? >
· PROJE T MANAGER	Sources	? _ 0 6 X	Project Summar	1									? 🗆 🗆 X
Sattings	9 2 0 + 0 00	0	Overview   Da	shboard									
Add Sources	Design Sources						-						^
Language Templates	> Constraints		Settings Ed										
P IP Catalog	Simulation Sources		Project name:	lab	01								
1	🖾 sim_1	1	Project location	D.A	archlabs/lab01								
* IP INTEGRATOR	> Guinty Sources		Product family.	Kin	tex-7								t
Create Block Design			Proje z part	xc7	k325ttfg676-2								
Open Block Design			Farget language	me: No	ilon								
Generate Block Design			Simulator lang	lage: Mix	ed								N
													1
SIMULATION			Synthesis							Implementation			
Run Simulation			Chatur	Noteta	ted					Otahue-	Not staded		
Y RTI ANALYSIS			Messages:	No erro	rs or warnings					Messages:	No errors or warnings		
Coren Flahorated Design			Part	xc7k325	5ttfg676-2					Part	xc7k325ttfg676-2		
	Hierarchy Libraries Compile Order		Strategy.	Vivado	Synthesis Default					Strategy:	Vivado Implementation Defaults	/	
Y SYNTHESIS	Properties		Report Strateg	Vivado	Synthesis Default	Reports				Report Strategy:	Vivado Implementation Default Repo	orts	
Run Synthesis	ropites	7 = 0 0 ×		-						Incremental implementation:	None		
> Open Synthesized Design		<b>o</b>	DDC Mediations		-					Timina	-		
			DRC violations			-	_			Timing			
MPLEMENTATION	Select an object to see properti	85		F	Run Implementati	on to see DR	C results				Run Implementation to see timin	ig results	
Run Implementation													~
> Open Implemented Design	Tcl Console Messages Log Reports	Design Runs											? _ 🗆 🖾
	Q + + + + + +	%										and the second	
<ul> <li>PROGRAM AND DEBUG</li> </ul>	Name Constraints Status	WNS TNS WHS	THS TPWS	Total Power	Failed Routes	LUT FF	BRAMs	URAM DSP	Start Elap	sed Run Strategy		Report Strategy	
Generate Bitstream	synth_1 constrs_1 Not started									Vivado Synthesis Defau	ilts (Vivado Synthesis 2018)	Vivado Synthesis Default Reports	s (Vivado Synure
> Open Hardware Manager	> impl_1 constrs_1 Not started									Vivado Implementation	Defaults (Vivado Implementation 2018)	Vivado Implementation Default R	eports (Vivado
	1												,
	In New York												<b>`</b>

图 2-8 是 Vivado2018.3 的整体界面

#### PS: 上图部分功能区域会随当前进行的操作而显示不同的内容

# 2.3 添加文件

 如下图,点击左侧区 Flow Navigator 下的 Project Manager->Add Sources 或中间区 Sources 的 "+"号,打开 Add Sources 对话框





 选择第二项 Add or Create Design Sources,用来添加或新建 Verilog HDL 源文件,点击 Next

🝌 Add Sources		×
	Add Sources This guides you through the process of adding and creating sources for your project Add or greate constraints Add or create design sources Add or create gimulation sources	
▲ XILINX₀		
<b>?</b>	< <u>Back</u>	Cancel

3. 若已有代码模块文件或 IP 核文件,可选 Add Files 以添加所需文件。本 实验这处是要新建代码文件,故选择 Create File 项:

🝌 Add Sources	×
Add or Create Design Sources Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.	4
$ +_{\lambda}  =  \pm  +$	
Use Add Files, Add Directories or Create File buttons below	
Add Files Add Directories Create File	_
Copy sources into project	
Add sources from subdirectories	
	ncel

4. 弹框 Create Source File 中 文件名输入 flowing\_light, 点击 OK

$ +_{z}  =  + +$	▲ Create Source File	
	Create a new source file and add it to your project.	
	Eile type:	
	File name:       flowing_light       Image: Control of the second	
	Cancel	

#### 5. 点击 Finish

Add Sources							
Add or Creat Specify HDL, nei lisk and add it t	<b>e Design S</b> tlist, Block De o your project.	sign, and IP files,	or directories co	ontaining those file typ	es to add to your proj	ect. Create a new sou	rce file on 🛛
+ -	+   +						
	Index	Name	Library	Location			
•	1	flowing_light.v	xil_defaultlib	<local project="" to=""></local>			
			<u>A</u> dd Files	A <u>d</u> d Directories	<u>C</u> reate File	]	
Scan and	add RTL <u>i</u> nclu	ide files into proje	ct			-	
Copy <u>s</u> our	ces into proje	ct					
Add so <u>u</u> rc	es from subdi	irectories					
?				< [	ack Next	> <u>F</u> inish	Cancel

6. 在弹出的 Define Module 中的 I/O Port Definition,输入设计模块所需的端口,并设置端口方向,若端口为总线型,勾选 Bus 选项,并由 MSB 和 LSB 确定总线宽度;完成后点击 OK

🍌 D	🍌 Define Module										
Def For N P	Define a module and specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.										
Module Definition											
	Module name: flowing_light										
VO Port Definitions + - ↑ ↓											
	Port N	Direction		М							
	clock	input 🗸		0	0		2				
	reset	input 🗸		0	0						
	led	output 🗸		7	0						
?	)	R				ОК	Cancel				

7. 新建的 flowing\_light 文件显示在 Sources 中的 Design Sources 下;



8. 双击 flowing\_ligh,添加如下设计代码(模块中文件默有的信息和代码不要删除),并点击保存按钮

```
2
         reg [23 : 0] cnt_reg;
 3
         reg [7 : 0] light reg;
 4
         always @ (posedge clock)
 5
             begin
 6
                  if (reset)
 7
                      cnt reg <= 0;
 8
                   else
                      cnt_reg <= cnt_reg + 1;</pre>
 9
10
             end
11
         always @ (posedge clock )
12
                begin
13
                     if (reset)
14
                           light reg <= 8'h01;
15
                     else if (cnt reg == 24'hffffff)
16
                          begin
                             if (light_reg == 8'h80)
17
                                  light_reg <= 8'h01;</pre>
18
19
                             else
20
                                  light_reg <= light_reg << l;</pre>
21
                          end
22
                  end
23
         assign led = light reg;
```

# 2.4 功能仿真

1. 创建激励测试文件。在中间区 Source 中右击选择 Add Source 或于左侧区 PROJEU MAHAGER 下选择 Add Source



2. 在 Add Source 中选择第三项 Add or Create Simulation Source, 点击 Next

À Add Sources		×
HLx Editions	Add Sources This guides you through the process of adding and creating sources for your project Add or greate constraints Add or create design sources Add or create <u>simulation sources</u>	
<b>E</b> XILINX.		
?	< <u>Back</u> <u>Next</u> > Einish Cancel	]

3. 选择 Create File 创建一个仿真激励文件

À Add Sources	×
Add or Create Simulation Sources Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.	4
Specify simulation set: 📮 sim_1 🗸	
$ +_{j}  =  \pm  \pm$	
Use Add Files, Add Directories or Create File buttons below	
Add Files Add Directories Create File	
Scan and add RTL include files into project	
Copy sources into project	
Add sources from subdirectories	
✓ Include all design sources for simulation	
? ≤Back Next> Einish Ca	ncel

4. 输入激励文件名称可以是 flowing\_light\_tb, 点击 OK

🝌 Create Source File 🛛 🗙							
Create a new source file and add it to your project.							
<u>F</u> ile type:	<ul> <li>Verilog</li> </ul>						
F <u>i</u> le name:	flowing_light_tb	3					
Fil <u>e</u> location:	■ <local project="" to=""> 文件名</local>						
?	OK Cancel						

5. 完成之后点击 Finish, 创建激励文件是不需要对外端口, 再点击 OK

🍌 D	efine Module									×
Define a module and specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.								4		
Mo	dule Definition	I								
	Module name	: flowing_li	ght_tb							$\otimes$
	I/O Port Defini	tions								
	+  -	<b>†</b>   <b>†</b>								
	Port Name	Direction	Bus	MSB	LSB					
	<	input 🗸		。 不	。 设置	端口				>
?	)							ОК		Cancel

6. 在弹出的对话框中点击 Yes

🍌 Define Module					
?	The module definition has not been changed. Are you sure you want to use these values?				
	Yes <u>N</u> o				

7. 在 Source 区 Simulation Sources 下,打开仿真测试文件 flowing\_light\_tb,在 其中对要进行仿真的模块作实例化并编写激励代码(并点击保存),如下 图所示

```
1
2
          clock;
     req
3
     reg reset ;
4
     wire [7:0] led;
5
6
     flowing_light u0(
7
          .clock(clock),
8
           .reset(reset),
9
          .led(led));
10
11
     parameter PERIOD= 10;
12
13
     always #(PERIOD*2) clock = !clock;
14
15
     initial begin
16
        clock = 1'b0;
17
        reset = 1'b0;
18
        #(PERIOD*2) reset = 1'bl;
19
        #(PERIOD*4) reset = 1'b0;
20
21
        //#580; reset = 1'bl;
22
     end
```

8. 在左侧 Flow Navigator 中点击 Simulation 下的 Run Simulation 选项,并选择 Run Behavioral Simulation。以下 Figure1 和 Figure2 为基于以上模块代码和激励测试文件运行仿真后所得到的波形:



在时钟上升沿时且 reset 置1时,图2中的 led[0]=1 表示第一个灯亮 (若 led 等于 0 表示该灯没亮)、另7 盏灯不亮 图1或图2是此次实验的仿真样例,也是实验的检查点,可截自图保存。



观察仿真波形时常用到下图几个按钮以便有适当的视野:



#### 观察仿真波形时还需掌握:

- a) 通过键盘 Ctrl+"-"和 Ctrl+"+"可以对波形图进行缩放;
- b) 对长信号而言,有时十六进制看起来更顺。选择该信号并点击右键,通
- 过 Radix 菜单可以改变信号不同进制的显示方式;

c) 下图 Scopes 窗口里可选中需要查看信号的模块,选中感兴趣的信号,

点击右键并选 Add To Wave Window, 可把该信号增加到仿真波形图中:



d) 增加信号后需要使用 Run 菜单下的 Restart, 重新开始或用工具条按 钮实现;

e) 使用 Run for…仿真指定时间长的波形;

f)可通过选择工具栏中的工具条按钮来进行波形的仿真时间控制。如下图 工具条,分别是复位波形(即清空现有波形)、运行仿真、运行特定时长 的仿真、仿真时长设置、仿真时长单位、单步运行、暂停、重新仿真:

<u>/</u> iew	<u>R</u> u	n	<u>H</u> elp	Q- QU	ick Acce	ss				4	<b>重新</b> 估百
W.	×	H	•	▶.m	10	ns	~	≛	Ш	C	里利门具

9. 在 8. 的仿真波形下添加了 cnt\_reg 信号的波形如下:

Q. 💾 🤁 🛛	0, 23 +	H H H H H H	a  ]∉			\$
					003	.000 ns
Name	Value	10 ns   100 ns	200 ns  3	00 ns  400 ns	500 ns	600 ns
🕌 clock	1					
🕌 reset	0					
> 💀 led[7:0]	0000001	ххххххх		000000	01	
> v cnt_reg[23:0]	000004	XXXXXX	000000	000001	000002 000003	000004
> 💐 light_reg[7:0]	00000001	хххххххх		000000	01	
> DELY[31:0]	20			20		
点击">"]	则展开led[	7:0]变量的8个分量信号	led[0], led[1	<b>]</b>		

观察本次仿真波形,可知道:

a) 当 reset 信号为1 时, 计数器 cnt\_reg 被初始化为零, 输出信号 led[0] 被始化为 00000001;

b) 当 reset 信号为 0 时,计数器在每个时钟信号上升沿时加1计数,直至 加到 24 位值全为1 时,输出信号左移1位(即 led[1]为 00000010); c) 本仿真运行周期不够,计数器并没加到 24 位全是1 而波形显示早已结 束。我们可以通过改变计数器的位数或者改变计数器计数值等参数,以便 较快速达到左移条件。如果愿意,你能给出如下两盏 led 灯亮时的仿真截图 吗:

√ 😽 led[7:0]	00000010	00000001	00000010	'X
[7]	0			
[6]	0			
[5]	0			
]₫ [4]	0		表示led[1]=1。	灯亮
14 [3]	0	表示led[0]=1, 灯亮		
14 [2]	0	1 1		
18 [1]	1			
] <b>å</b> [0]	0			

## 2.5 工程实现

由于实验板板载了 200MHz 时钟振荡器,属高频时钟,做下载验证时则需用 到差分时钟以更好适应工程上的需要。原 flowing\_light 代码模块需做时 钟方面的修改。

1. flowing\_light.v 的修改如下图橙色方框中:

1	
2	module flowing light(
3	input clock p,
4	input clock n,
5	input reset,
6	output [7:0] led
7	);
8	reg [23:0] cnt reg;
9	reg [7:0] light reg:
10	
11	IBUFGDS IBUFGDS inst (
12	.O(CLK i).
13	.I(clock p).
14	.IB(clock n)
15	
16	
17	always @ (posedge CLK i)
18	begin
19	[if (!reset)] //板子上的按钮实际上按下去是低电平,故要取反
20	cnt reg <= 0;
21	else
22	cnt reg <= cnt reg + 1;
23	end
24	always @ (posedge CLK i )
25	begin
26	if (!reset)
27	light reg <= 8'h01;
28	else if (cnt reg == 24'hffffff)
29	begin
30	if (light_reg == 8'h80)
31	<pre>light_reg &lt;= 8'h01;</pre>
32	else
33	<pre>light reg &lt;= light reg &lt;&lt; l;</pre>
34	end
35	end
36	assign led = light_reg;
37	-
38	endmodule

2. 添加管脚约束文件

有下面两种方法可添加约束文件:

- 一是利用 Vivado的IO planning功能;
- 二是直接新建类型为xdc的约束文件,手动输入约束命令。

1) 利用IO planning: 点击 Flow Navigator 中 Synthesis 中的 Run Synthesis, 先对工程进行综合。综合完成之后, 弹出以下对话框, 选 择Open Synthesized Design, 点击OK

Synthesis Completed						
Synthesis successfully completed.						
O Run Implementation						
Open Synthesized Design						
◯ <u>V</u> iew Reports						
Don't show this dialog again						
OK Cancel	]					

#### 点击OK后,应看到如下界面。若无则在下图示位置选择方框的 IO planning:



在下方区的选项卡中选中 I/O ports,并在对应的信号后,输入对应的FPGA 管脚标号(或将信号拖拽到右上方 Package 图中对应的管脚上),并指定I/O Std。 具体的 FPGA 约束管脚和IO电平标准可参考对应板卡的硬件手册。

Eile Edit Flow Icols Report	s Window Lageut Yew Help Q-Quick Access		Synthesis Complete
🕒 🖬 🐟 🖉 🗄 X	× × × Σ Φ Ε Φ Η Η Φ		III IO Planning 🗸 🗸
Flow Navigation 4 3 0 ? -	SYNTHE SIZED DE SIGN * - xc7k3258fg676-2 (active)		? )
Y PROJECT MANAGER	Sources Notist Device Constraints	rkana u Danira u Boainn Inhi u	a. 19.77
Settings		craye X beauty X manufature X	708
Add Sources	Q ± 0 = 0	- + u u z z o	0
Language Templates	<ul> <li>Internal VREF</li> </ul>	1 2 3 4 5 8 7 8 9 10 11 12 13 14 15 1	17 18 19 20 21 22 23 24 25 26
O IP Catalon	= 0.6V		
T II Calary	= 0.5/5v		
Y IP INTEGRATOR	= 0.9V		
Create Block Design	∽ □ NONE (8)		
Open Block Design	NO Bank 12		
Generate Block Design	I/O Bank 13		
	NO Bank 14		
✓ SIMULATION	Drop I/O banks on voltages or the "NONE" folder to setlunset Internal		
Run Simulation	VREF.		
	farmer and a second second second		
Y RTL ANALYSIS	TO PORT Properties × CLOCK Regions ? _ [] []		
> Open Elaborated Design	€ lad[7]		
	Name: led(7)		
✓ SYNTHESIS	Direction: OUT		
Run Synthesis	Parkana nin W23 CE Errad		
<ul> <li>Open Synthesized Design</li> </ul>	Possege price and a second sec		
Constraints Wizard	Site type: IO_LBP_11_12		
Edit Timing Constraints	General Properties Configure Power		
🗃 Set Up Debug			
C Report Timing Summary	Tci Consolo Messages Log Reports Design Runs Packa	ns IO Ports ×	? _ 0 0
Report Clock Networks	Q 풒 ● 📢 + ⅓		0
Report Clock Interaction	Name Direction Neg Diff Pair Package P	Fixed Bank I/O Std 👩 Vcco Vref Drive Strength Slew Type	Pull Type Off-Chip Termination IN_TERM
	All ports (12)	× ×	<u> </u>
<ul> <li>Report wethodology</li> </ul>	Jed (8) OUT	✓ 12 LVCMOS33* 3300 12 ✓ SLOW	V NONE V FP_VTT_50 V
Report DRC	@ 14d(7) OUT W23	3365 12 VOMOS33* * 3365 12 VSLOW	V NORE V PP_VTT_50 V
Report Noise	4 lades out 925		
Report Utilization	a ledial OUT AA23	v V 12 LVCMO833* * 3300 12 v SLOW	V NONE V FP VTT 50 V
Seport Power	@ led[3] OUT Y23	✓ ♥ 12 LVCMOS33 <sup>4</sup> → 3,300 12 ✓ SLOW	V NONE V FP VTT 50 V

本次实验用到的对应管脚及10电平标准如下:

1							
2	led[7]	对应的管脚是	W23	:	I/0	Std为	LVCMOS33
3	led[6]	对应的管脚是	AB26	:	I/0	Std为	LVCMOS33
4	led[5]	对应的管脚是	Y25	:	I/0	Std为	LVCMOS33
5	led[4]	对应的管脚是	AA23	:	I/0	Std为	LVCMOS33
6	led[3]	对应的管脚是	¥23	:	I/0	Std为	LVCMOS33
7	led[2]	对应的管脚是	¥22	:	I/0	Std为	LVCMOS33
8	led[1]	对应的管脚是	AE21	:	I/0	Std为	LVCMOS33
9	led[0]	对应的管脚是	AF24	:	I/0	Std为	LVCMOS33
10							
11	clock_p	对应的管脚是	AC18	:	I/0	Std为	LVDS
12	reset	对应的管脚是	W13	:	I/0	Std为	LVCMOS18

# 全部管脚锁定及I/O Std指定后,点击左上方工具栏中的保存按钮,弹出对话框提示新建约束文件,可输入lab01\_xdc,并点击OK

TCI Console Messa	ages Log	Reports	esign Runs	Раск	age Pins	101	orts ×										
Q ₹ ≑ €	H H																
Name	Direction	Neg Diff Pair	Package Pin		Fixed	Bank	I/O Std		Vcco	Vref	Drive Strength	Slew Type		Pull Type		Off-Chip Termination	IN_
🕢 led[7]	OUT		W23	~		12	LVCMOS33*	Ŧ	3.300		12 🗸	SLOW	~	NONE	~	FP_VTT_50	~
- Ied[6]	OUT		AB26	$\sim$		12	LVCMOS33*	~	3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	~
🕢 led[5]	OUT		Y25	~		12	LVCMOS33*	~	3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	~
- Ied[4]	OUT		AA23	~		12	LVCM0533*	•	3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	~
- Ied[3]	OUT		Y23	~		12	LVCMOS33*		3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	×
Ied[2]	OUT		Y22	~		12	LVCMOS33*	*	3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	×
🕢 led[1]	OUT		AE21	~		12	LVCMOS33*		3.300		12 🗸 🗸	SLOW	$\sim$	NONE	~	FP_VTT_50	¥
- Ied[0]	OUT		AF24	~		12	LVCMOS33*	•	3.300		12 🗸 🗸	SLOW	$\mathbf{v}$	NONE	~	FP_VTT_50	~
🗸 🕞 Scalar ports (	(3)																
clock_p	IN	clock_n	AC18	~		32	LVDS*	•						NONE	~	NONE	~
reset	IN		W13	$\sim$	$\checkmark$	32	default (LVC		1.800					NONE	$\mathbf{v}$	NONE	~

🝌 Save Constraints	×
Select a target file to write new unsaved constraints to. Choosing an existing file will update that file with the new constraints.	4
<u>C</u> reate a new file	
Eile type: 🚺 XDC 🗸 🗸	
File name: lab01_xdd	
Fil <u>e</u> location: 🕞 <local <sub="" f="" to="">文件名 🗸</local>	
Select an existing file	
<select a="" file="" target=""> V</select>	
	Cancol
OK C	Jancer

2)像创建模块代码文件一样新建约束文件。打开Add Sources对话框,如图选择第一项:

À Add Sources		$\times$
E XILINX.	Add Sources This guides you through the process of adding and creating sources for your project Add or greate constraints Add or create design sources Add or create gimulation sources	
?	< Back Next > Einish Cance	

Add Sources Add or Create C Specify or create cor	onstraints Istraint files for physical and timing constraint to add	d to your project.	Create Cons	trainte File	×
Specify constrain	t set 📄 constrs_1 (active) 🗸 🗸		Create a new c project	onstraints file and add it	to your
			<u>F</u> ile type:	T XDC	~
lab01_xdc.xdc	D:\archlabs\lab001\lab001.srcs\constrs_1\new		F <u>i</u> le name:	lab01_xdc	$\otimes$
			Fil <u>e</u> location:	😜 <local project="" to=""></local>	~
			?	ОК	Cancel
	<u>A</u> dd Files	<u>C</u> reate File			
Co <u>p</u> y constrair	nts files into project				
(?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel

在Sources区双击打开新建好的空的约束文件,并按照规则输入符合相应的 FPGA管脚约束信息与电平标准的约束语句:

<pre>set_property</pre>	PACKAGE_PIN W23 [get_ports {led[7]}]
<pre>set_property</pre>	PACKAGE_PIN AB26 [get_ports {led[6]}]
<pre>set_property</pre>	PACKAGE_PIN Y25 [get_ports {led[5]}]
<pre>set_property</pre>	PACKAGE_PIN AA23 [get_ports {led[4]}]
<pre>set_property</pre>	PACKAGE_PIN Y23 [get_ports {led[3]}]
<pre>set_property</pre>	PACKAGE_PIN Y22 [get_ports {led[2]}]
<pre>set_property</pre>	PACKAGE_PIN AE21 [get_ports {led[1]}]
<pre>set_property</pre>	PACKAGE_PIN AF24 [get_ports {led[0]}]
<pre>set_property</pre>	PACKAGE_PIN AC18 [get_ports clock_p]
<pre>set_property</pre>	PACKAGE_PIN W13 [get_ports reset]
<pre>set_property</pre>	<pre>IOSTANDARD LVCMOS33 [get_ports {led[7]}]</pre>
<pre>set_property</pre>	IOSTANDARD LVCMOS33 [get_ports {led[6]}]
<pre>set_property</pre>	<pre>IOSTANDARD LVCMOS33 [get_ports {led[5]}]</pre>
<pre>set_property</pre>	IOSTANDARD LVCMOS33 [get_ports {led[4]}]
<pre>set_property</pre>	IOSTANDARD LVCMOS33 [get_ports {led[3]}]
<pre>set_property</pre>	<pre>IOSTANDARD LVCMOS33 [get_ports {led[2]}]</pre>
<pre>set_property</pre>	IOSTANDARD LVCMOS33 [get_ports {led[1]}]
<pre>set_property</pre>	<pre>IOSTANDARD LVCMOS33 [get_ports {led[0]}]</pre>
<pre>set_property</pre>	IOSTANDARD LVDS [get_ports clock_p]
<pre>set_property</pre>	IOSTANDARD LVDS [get_ports clock_n]
set property	IOSTANDARD LVCMOS18 [get ports reset]

# 2.6下载验证(暂不做)

 在 Flow Navigator 中点击 Program and Debug 下的 Generate Bitstream 选项,系统会自动完成综合、实现、生成 FPGA 配置文件(bit文 件)。如出现以下框,点Yes





🝌 Launch Runs	$\times$						
Launch the selected synthesis or implementation runs.	~						
Launch directory: Solution Launch Directory>							
Options							
● Launch runs on local host: Number of jobs: 4 ~							
◯ <u>G</u> enerate scripts only							
Don't show this dialog again							
OK Cance	I						

Bitstream生成后,可点击 Open Implemented Design 来查看实现的结果,也可点击Cancel退出,也可选第三项直接去"烧写"(即生成电路的配置文件)

Bitstream Generation Completed X
Bitstream Generation successfully completed. Next
Open Implemented Design
○ <u>V</u> iew Reports
Open <u>H</u> ardware Manager
O Generate Memory Configuration File
Don't show this dialog again
OK Cancel

接下来点击Yes 或No



2. 按下图一连接 SWORD 实验板的 12V 电源、通过 JTAG 下载器使得 实验板连上计算机,然后开启电源开关,并对要通过防火墙的相关通信 服务允许访问。



图一(电源接口 JTAG 插口 电源开关)



图二(led 灯 拨码开关 rest 按钮)

🔗 Windows 安全中	心警报		$\times$
🔶 Windo	ws Defende	r 防火墙已经阻止此应用的部分功能	
Windows Defende	r 防火墙已阻止所	有公用网络和专用网络上的 hw_server 的某些功能。	
	名称(N):	hw_server	
	发布者(P):	未知	
	路径(H):	C:\xilinx_vivado\vivado\2018.3\bin\unwrapped\win64.o \hw_server.exe	
允许 hw_server 在读	这些网络上通信:		
☑ 专用网络,例 100 ☑ 公用网络,例 (U)	如家庭或工作网络 如机场和咖啡店F	各(R) 中的网络(不推荐,由于公用网络通常安全性很小或者根本不安全)	
允许应用通过防火墙	有何风险?		
		2 ◆ 允许访问(A) 取消	

3. 点击 Flow Navigator 中 Open Hardware Manager 一项,进入硬件编程 管理界面



4. 连接成功后,在目标芯片(xc7k325t\_0(1))上右键选择 Program Device、 或者点击下图方框的 Program Device

HARDWARE MANAGER - localhost/xilinx_tcf/l	Digilent/2017063	00081
1 There are no debug cores. Program dev	ice Refresh dev	ice
Hardware	2 - 0 C ×	flowing_light.v × lab01_xdc.xdc ×
Q   素   ♦   ∅   ▶   ≫   ■	•	D:/archlabs/lab001/lab001.srcs/sources_1/imports/ba
Name	Status	Q   ₩   ♠   ★   ★   ■   ■   ★   //
V 📱 localhost (1)	Connected	
✓ ■ ✓ xilinx_tcf/Digilent/2017063000	Open	
<ul> <li>xc7k325t_0 (1)</li> <li>XADC (System Monito</li> </ul>	Hardware Devic	e Properties Ctrl+E
	Program Device	·····
	Verify Device	-
C	Refresh Device	
<	Add Configurati	on Memory Device utput [15:0]
Hardware Device Properties	Boot from Confi	guration Memory Device

5. 在弹出的对话框中 系统会自动加载本工程生成的比特流文件,点击Program 对FPGA芯片进行编程

🍌 Program Device		×
Select a bitstream prog select a debug probes programming file.	gramming file and download it to your hardware device. You can optionally file that corresponds to the debug cores contained in the bitstream	4
Bitstre <u>a</u> m file:	D:/archlabs/lab001/lab001.runs/impl_1/flowing_light.bit	•
Debu <u>a</u> probes file:		•
✓ Enable end of st	tartup check	
(?	<u>P</u> rogram Cancel	

6. 观察实验板上那块子板 led 的实验结果,看 8 位流水灯的显示是否预期。

#### 2.7 主板16位LED流水灯的实现(暂不选做)

SWORD 平台提供了 4 种 GPIO 接口: 4X4 按键矩阵、16 位滑动开关、16 位 LED、8 位 7 段数码管。其中为了节省 I/O,仅 16 位的滑动开关采用了和 FPGA 直 连的方式, 16 位 LED 和 8 位7段数码管采用了SN74LV164移位寄存器进行串行转并 行的处理。

在原工程中的flowing\_light.v和lab01\_xdc.xdc文件加入串行转并行机制 (parallel2serial软核)并进行相应添加代码行就可实现。

```
1. 模块文件修改如下:
module flowing_light(
    input clock_p,
    input clock_n,
    input reset,
    output [7:0] led,
    output LED_CLK, //为实现主板16位led流水而新增
    output LED_DO, //新增
    output LED_CLR //新增
    );
reg [23:0] cnt_reg;
reg [7:0] light_reg;
reg [15:0] light_reg2;//新增
IBUFGDS IBUFGDS_inst (
        .0(CLK_i),
```

```
. I (clock_p),
       .IB(clock n)
    );
always @ (posedge CLK_i)
    begin
          if (!reset) //板上复位按钮按下是底电平
                 cnt_reg <= 0;
          else
          cnt_reg <= cnt_reg + 1;</pre>
    end
 always @ (posedge CLK i )
    begin
          if (!reset)
          begin//新增
                light reg \langle = 8' h 01;
                light reg2<=16'b01;//新增
          end //新增
          else if (cnt reg == 24'hffffff)
               begin
                  light_reg2<={light_reg2[14:0], light_reg2[15]};//新增
                  if (light_reg == 8'h80)
                      light reg \langle = 8' h 01;
               else
                      light_reg <= light_reg << 1;</pre>
               end
    end
assign led = light_reg;
  //以下所有语句为实现主板16位1ed流水而新增
  wire led clr;
  assign
          LED CLR=~led clr;
  reg [23:0] clkcnt;
  always@(posedge CLK i)
       clkcnt<=clkcnt+l;
  parallel2serial #(
       .P CLK FREQ(200),
       .S_CLK FREQ(20),
       .DATA BITS(16),
       .CODE ENDIAN(1))
     P2S LED (
       .clk(CLK i),
       .rst(~reset),
       .data(light_reg2),
       .start((clkcnt==24'b0)?1'b1:1'b0),
       .busy(),
       .finish(),
       .s clk(LED CLK),
       .s clr(led clr),
       .s dat(LED DO));
```

Endmodule

 flowing\_light.v 调用了 parallel2serial.v 这个 IP 内核,需添加该模块 (我们将之已封装成网表文件, parallel2serial.edif 和仅含端口的 parallel2serial.v 这两个文件对应原来的parallel2serial 的 IP 内核文件,如下 图圈 3 处)到工程中

Flow Navigator	≭ ≑ ? _	PROJECT MANAGER - lab001				
✓ PROJECT MANAGER		Sources		?_0Ľ×	:	Project Summary
Settings		Q ₹ ≑ + 2	• 0	0	1	Overview   Dashboard
Add Sources		Design Courses (1)				
Language Templa	À Add Sources					
👎 IP Catalog	Add or Creat	e Design Sources		À Add Source F	iles	
✓ IP INTEGRATOR	Specify HDL, ne disk and add it t	tlist, Block Design, and IP files, or o your project.	directories contain	Look <u>i</u> n: 📄 eo	dif_la	ab01
Create Block Des				Ŋ parallel2seri	al.ed	lif
Open Block Desig	$ +_{2}  =  $	*   <b>+</b>		parallel2seri	al.v	3
Generate Block D						
Run Simulation		Use /	Add Files, Add Dire			
RILANALISIS						
<ul> <li>Open Elaborated</li> </ul>						
Y SYNTHESIS		Ac	ld Files A			
Run Synthesis	Scan and	add RTL include files into project	2			
	Copy sour	rces into project	-			

#### 两个文件添加后,点击Finish

A	Add Sources				>	<
A SI di	<b>dd or Creat</b> becify HDL, ne sk and add it tr	<b>e Design S</b> tlist, Block De o your project.	iources sign, and IP files, or di	rectories contai	ining those file types to add to your project. Create a new source file on 🛛 💫	
	+  =	<b>1</b>				
		Index	Name	Library	Location	
	•	1	parallel2serial.v	xil_defaultlib	D:/archlabs/edif_lab01	
	Ń	2	parallel2serial.edif	N/A	D:/archlabs/edif_lab01	
	Scan and Copy Sour Add Source	add RTL inclu ces into proje	Add Ide files into project ct irectories	Files	A <u>d</u> d Directories	
(	?)				< Back Next > Finish Cancel	

3. 管脚约束文件添加如下
#161eds 以下为161ed显示流水新增
set\_property PACKAGE\_PIN N26 [get\_ports LED\_CLK]
set\_property PACKAGE\_PIN N24 [get\_ports LED\_CLR]
set\_property IOSTANDARD LVCMOS33 [get\_ports LED\_CLR]
set\_property IOSTANDARD LVCMOS33 [get\_ports LED\_CLR]
set\_property IOSTANDARD LVCMOS33 [get\_ports LED\_CLR]

4. 同样: 生成bit文件、下载验证, 观察 16 个 1ed 灯亮灭变化的运行规律是否 符合预期