

Vivado设计套件是 Xilinx 公司于 2012 年发布的集成设计环境,是一个基于 AMBA AXI4 互联规范、IP-XACT IP 封装元数据、工具命令语言(TCL)、Synopsys 系统约束 (SDC)符合业界标准的开放式环境,能够支持多达1亿个等效 ASIC 门的设计。

基于 Vivado 的 FPGA 设计开发流程如图 3.1 所示,主要包括以下步骤。



图 3.1 Vivado 设计的流程

(1) 创建工程。

(2) 编辑源设计文件,包括 HDL 文本、IP 核、模块文件、网表输入等方式。

(3) 行为仿真(Behavioral Simulation), 在别的软件中也被称为功能仿真、前仿真, 即 不包含延时信息的仿真; Vivado 自带仿真器, 也可以选择采用第三方仿真工具 ModelSim 等工具进行仿真。

(4) 添加引脚约束。

① 综合(Synthesis): 根据设定的编译策略,对工程进行综合,生成网表文件。

② 引脚约束:通过 I/O Planing 或者直接编辑.xdc 文件添加引脚约束信息。

③ 实现(Implimentation):指针对某一具体的目标器件经布局布线(Place & Route),或者称为适配(Fitting),产生延时信息文件、报告文件(.rpt),以供时序分析、时序仿真使用。

(5) 生成 Bitstream 文件,产生. bit 和. bin 等编程文件。

(6) 将生成的 Bitstream 文件下载至 FPGA 芯片。

设计步骤的次序并非一成不变,可根据个人习惯及实际情况进行调整和修改;同时,在 设计过程中,如果出现错误(Error),需改正错误或调整电路后重复相应的步骤;如果出现 严重警告信息(Critical Warning),也需引起注意,要不断调整和优化,直至达成设计 目标。

3.1 Vivado 流水灯设计

本节以使用 Verilog 语言设计流水灯为例,介绍在 Vivado 环境下运行 Verilog 程序的流程,包括源程序的编写、编译、仿真及下载。本例基于 Vivado 18.2版本,其他不同版本的 Vivado 使用方法与此类似。

3.1.1 流水灯设计输入

1. 创建新工程

首先建立一个工作目录,本例的工作目录为 D:/exam。

(1) 双击启动 Vivado 2018.2,出现如图 3.2 所示的 Vivado 启动界面,单击 Quick Start 栏中的 Create Project(或者选择菜单 File→New Project...命令),启动工程向导,创 建一个新工程。





(2) 在工程向导(如图 3.3 所示)界面中单击 Next 按钮。

(3) 在图 3.4 所示的界面中命名工程名和存储路径,此处项目命名为 led,其存放位 置为 D:/exam,勾选 Create project subdirectory 复选框,可为此工程在指定路径下建立 独立的文件夹,最终整个项目存在 D:/exam/led 文件夹中。设置完成后,单击 Next 按钮。

注意: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 画线来组成。

(4) 选择项目类型(如图 3.5 所示)界面,选择 RTL Project 类型,单击 Next 按钮。

注意:如果在图 3.5 中勾选 Do not specify sources at this time 复选框,则跳过后面的(5)和(6),表示当前工程无须添加源文件和约束文件。

New Project	×
	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.
€ XILINX.	
?	< <u>B</u> ack <u>Einish</u> Cancel



New Project				×
Project Name Enter a name for yo	our project and specify a directory	where the project data files will be store	ed.	A
<u>P</u> roject name:	led			0
Project location:	D:/exam			⊗
Create project will be cr	ct subdirectory eated at: D:/exam/led			
 ? 		< Back	lext > Einish	Cancel

图 3.4 工程名称、路径设定界面

(5) 在图 3.6 所示的 Add Sources 界面中添加源文件并选择设计语言,其中 Target language 和 Simulator language 均选择 Verilog,单击 Next 按钮。

(6) 不添加约束文件,所以在 Add Constraints 界面直接单击 Next 按钮。

(7) 在图 3.7 所示的器件选择界面中,根据使用的 FPGA 开发板,选择相应的 FPGA 目标器件。本例中,以 Xilinx EGO1 为目标板,故 FPGA 选择 xc7a35tcsg324-1,即 Family 选择 Artix-7,封装形式(Package)为 csg324,单击 Next 按钮。

第13章 Vivado使用指南

roje	ct Type	
Specify	y the type of project to create.	1
۲	<u>R</u> TL 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.	
	Do not specify sources at this time	
0	Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.	
	Do not specify sources at this time	
0	J/O Planning Project	
	Do not specify design sources. You will be able to view parupackage resources.	
0	Imported Project Create a Vivado project from a Synplify, XST or ISE Project File.	
0	Example Project Create a new Vivado project from a predefined template.	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel
	图 3.5 选择工程类型	
Ne	图 3.5 选择工程类型 w Project	
Add Specify ile on	图 3.5 选择工程类型 w Project Sources y HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source disk and add it to your project. You can also add and create sources later.	ce
Add Specify lile on	图 3.5 选择工程类型 w Project Sources y HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source disk and add it to your project. You can also add and create sources later.	ce
Add S Bpecify ille on	图 3.5 选择工程类型 w Project Sources y HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source disk and add it to your project. You can also add and create sources later. 	ce
Ner Add S Specify ile on	图 3.5 选择工程类型 w Project Sources y HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source disk and add it to your project. You can also add and create sources later. 」 = * * * Use Add Files, Add Directories or Create File buttons below <u>Add Files</u> <u>Add Directories</u> <u>Create File</u> Scan and add RTL include files into project	ce
Net	图 3.5 选择工程类型 w Project Sources y HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source disk and add it to your project. You can also add and create sources later. 」 =	ce

图 3.6 添加源文件并选择设计语言

< Back

<u>N</u>ext >

<u>F</u>inish

Cancel

?

oose a de	rt fault Xilinx p	art or board for you	r project. This can	be changed	later					à
Parts	Boards									
Reset All	Filters									
Category:	All		~	Package:	csg	324	✓ Temperat	ure: All Rema	aini	~
Family:	Artix-7		~	Speed:	-1	,	~			
Search:	Q-		~							
Part		I/O Pin Count	Available IOBs	LUT Eleme	nts	FlipFlops	Block RAMs	Ultra RAMs	DSPs	
	csg324-1	324	210	10400		20800	25	0	45	2
xc7a15t		324	210	20800		41600	50	0	90	
xc7a15t	csg324-1	Table Office A					75	0	400	
xc7a15t xc7a35t xc7a50t	csg324-1 csg324-1	324	210	32600		65200	15	U	120	\sim

图 3.7 器件选择界面

(8) 最终出现图 3.8 所示的界面,对工程信息进行汇总,确认相关信息正确与否,包括工程类别、源文件、所用的 FPGA 器件等。如果没有问题,则单击 Finish 按钮完成工程的创建;如果有问题,则返回前面界面进行修改。

new Project	×
	New Project Summary
HLx Editions	A new RTL project named 'led' will be created.
	No source files or directories will be added. Use Add Sources to add them later.
	No constraints files will be added. Use Add Sources to add them later.
	The default part and product family for the new project. Default Part xc7a35tcsg324-1 Product Artix-7 Family. Artix-7 Package: csg324 Speed Grade: -1
€ XILINX.	To create the project, click Finish
	Rook Note Cities Concel
•	

图 3.8 工程信息汇总

第3章 Vivado使用指南

2. 输入源设计文件

(1) 如图 3.9 所示,选择 Flow Navigator 下 PROJECT MANAGER 中的 Add Sources 选项,打开设计文件导入窗口。

F led - [D:/exam/led/led.xpr] - \	/wado 2018.2	- 🗆 ×
Elle Edit Flow Tools Rep	grts Window Layout View Help Q- Oulce Access	Ready
■, + + = = × 1	▶ # O Σ ± Ø x	📰 Default Layout 🗸 🗸
Flow Navigator = 0 ?	PROJECT MANAGER - Ied	?
Y PROJECT MANAGER	Sources 2 D IS X Project Summary	2 D K X
Settings		
Add Sources	Design Sources	
Language Templates	> Constraints Project name: led	
₽ IP Catalog	V Can Simulation Sources Project location. D./exam/led	
	sim_1 Product family. After7	
V IP INTEGRATOR	Hierarchy Libraries Compile Order Project part: sc7a35lcsg3	24-1
Create Block Design	Presenting of the second secon	
Open Block Design	Simulator language: Verilog	
Generate Block Design		
✓ SIMULATION	Synthesis	Implementation
Run Simulation	Select an object to see properties Status Not started	Status:
	(c	
Y RTL ANALYSIS	Tel Consola Massanes Lon Danorts Design Pans	2 5 7
> Open Elaborated Design		7 - 0 6
✓ SYNTHESIS	Name Constraints Status WMS TNS WHS THS TPMS Total Power Failed	Contes LUT FE BRAMS LIRAM DSP
Run Synthesis	✓ ▷ synth_1 constrs_1 Not started	
> Open Synthesized Design	▷ impl_1 constrs_1 Not started	
✓ IMPLEMENTATION		
Run Implementation		
> Open Implemented Design		

图 3.9 工程管理窗口

(2) 在 Add Sources 界面(如图 3.10 所示)中选中 Add or create design sources 单选 按钮,表示添加或创建 Verilog(或 VHDL)源文件,单击 Next 按钮。

Add Sources		×
VIVADO HLx Editions	Add Sources This guides you through the process of adding and creating sources for your project Add or greate constraints <u>A</u> dd or create design sources Add or create gimulation sources	
€ XILINX.		
•	<back cancel<="" einish="" td=""><td></td></back>	

图 3.10 添加或创建源文件

(3) 在图 3.11 中单击 Create File 按钮,在弹出的 Create Source File 对话框中输入 File name 为 flow_led,单击 OK 按钮。

and add it to your project	Philes, or directories	containing those file types	to add to your proj	ect. Create a new source	file on 🛛 🗖
			🍐 Create So	urce File	×
+, - * *			Create a new s project.	ource file and add it to yo	ur 🍌
			Eile type:	Verilog	~
			File name:	flow_led	\otimes
	Use Add Files, Ad	dd Directories or Create Fil	File location:	Local to Project>	~
				ОК	Cancel
	Add Files	Add Directories	<u>C</u> reate File]	
Scan and add RTL include files int	o project				
Copy sources into project					

图 3.11 创建源文件

注意: 文件名中不可出现中文和空格; 如果有现成的. v 或. VHD 文件, 可单击 Add Files 或者 Add Directories 按钮进行添加。

(4) 单击图 3.12 中的 Finish 按钮,完成源文件的创建。

Add Source	es				×
Add or Creat Specify HDL, ne disk and add it t	t e Design tlist, Block D o your projec	Sources lesign, and IP fi ct.	les, or directorie	es containing those file types to add to your project. Create a new source file on	A
+, -	+ 1 +				
	Index	Name	Library	Location	
	1	flow_led.v	xil_defaultlib	<local project="" to=""></local>	
		Г	Add Files	Add Directories Create File	
Scan and	add RTL (no	ude files into p	roject		
Copy sour	rces into pro	ject			
Add source	es from sub	directories			
~					
(?)				<back next=""> Einish C</back>	ancel

图 3.12 完成源文件创建

(5) 在弹出的 Define Module 对话框中填写模块名称,此处模块命名为 flow_led,如图 3.13 所示。还可以在 I/O Port Definitions 栏中填写模块的端口并设置端口方向,如果端口为总线型,勾选 Bus 选项,并通过 MSB 和 LSB 确定总线宽度。完成后单击 OK 按钮。

第13章 Vivado使用指南

			9 S. 199			
Define a modu	le and spec	city I/O Po	rts to add to you	ur source file.		
For each port s	pecified:	II he issee	and unless its F	has a share in sha	a also al	- A.
NSB and LS	B values wi	will not b	o written	sus column is che	ecked.	
1 0110 1111 01	annenannea	mininorb	o minicon.			
Module Definit	tion					
	1000 a					
Module na	me: flow_	led				8
Module na	me: flow_	led				8
Module na	me: flow_	led				8
Module na	me: flow_	led				8
Module na	finitions	led ₽				8
Module na	finitions	_ied				8
Module na	finitions	.ied ↓				8
Module na	finitions	.ied ↓ □ 0				

图 3.13 Define Module 对话框

(6)当前 Vivado 界面如图 3.14 所示,在中间的 Sources 窗格的 Design Sources 中出 现新建的设计文件 flow_led.v,双击打开该文件,利用 Vivado 自带的文本编辑器(Text Editor)输入设计代码。本例 LED 流水灯的代码如例 3.1 所示。

File Edit Flow Tools Rr	eports Window Layout View Help Q- Quick Access			Ready
B & A B B X	λ Ε Ο Σ Κ Ø Ø		Default Lavout	
Flow Navigator = = ? .	PROJECT MANAGER - led			?
Y PROJECT MANAGER				
Settings	Q ± + 0 0 0 D/examinative scattering v intervention judy			705 ×
Add Sources Language Templates	∨ ⊡ Design Sources (1) Q. ∐ ↑ ↑ X. □ ■ X. / // ■ 0.1 Torus Left Their Left (0) Image: Sources (1) Image: Sources (1)	0		•
IP Catalog				~
 IP INTEGRATOR Create Block Design Open Block Design Generate Block Design 	> □ sim_1(t) 9 // Project Bas: 10 // Farger Derices: 11 11 // Toul Terrices: 12 12 // Decription: 13 14 // Dependencies: 14			
 SIMULATION Run Simulation 	Source File Properties ? □ □ □ × 16 // Xerisian 17 // Nerisian 0.01 - File Greated ● fow_ledv ◆ ◆ ◆ ◆ 13 // Additional Consents: 19 ///			
 RTL ANALYSIS Open Elaborated Design 	Enabled Location: D/exam/tedied.srcs/sources_tinew D/ex			
 SYNTHESIS Run Synthesis Open Synthesized Design 	Type: Verlog LUrary el.(defautilit) General Properties			, ,
	Tel Consolo Messanos Los Renorts Design Runs		?	_ 0 0
IMPLEMENTATION Run Implementation				
IMPLEMENTATION Run Implementation Open Implemented Design	Q X Image: Status V V Image: Status V Name Constraints Status WNS TNS WHS THS TDB V > psrth_1 constraint Not stated	IT FF BI	RAMS URAM	DSP
V IMPLEMENTATION Run Implementation Copen Implemented Design V PROGRAM AND DEBUG	Q X ♦ Id >> + % Name Constraints Status WNS THS WHS THS THS Total Power Failed Routes LU* >> >> status WNS THS WHS THS THS Total Power Failed Routes LU*	IT FF BI	RAMS URAM	DSP

图 3.14 Verilog 代码编辑窗口

【例 3.1】 8 位流水灯源代码。

module flow_led(clk,clr,led);
input clk,clr;

```
output reg [7:0] led;
  reg [28:0] counter;
always @(posedge clk)
begin
if(!clr) begin counter <= 0;led <= 8'h01; end</pre>
   else
   if( counter < 5000000)
                                        //2Hz
      counter <= counter + 1;</pre>
   else
      begin
           counter < = 0;</pre>
           led < = {led[6:0],led[7]};</pre>
       end
  end
endmodule
```

3.1.2 行为仿真

至此,已完成源文件输入,此时可对源文件进行行为(功能)仿真,以测试其功能。

(1) 创建激励测试文件,在 Sources 中右击选择 Add Sources,在出现的 Add Sources 界面中(见图 3.10)选择第三项 Add or create simulation sources 单选按钮,单击 Next 按钮。

(2) 在如图 3.15 所示的界面中单击 Create File 按钮,创建一个仿真激励文件,在弹出的 Create Source File 对话框中输入激励文件名称为 tb_led,文件类型为 Verilog,单击 OK 按钮,确认添加完成后单击 Finish 按钮。

Specify simulation set	Ei iii sim_1	e Add Files, Ad	id Directories or Create	Create a new project. Eile type: File name: Filg location:	• Verilog • Verilog b_led • <local project-<="" th="" to=""><th>ur A</th></local>	ur A
		Add Files	Add Directories	<u>C</u> reate File		
Scan and add RTL	include files into proje	Add Files	Add Directories	<u>C</u> reate File		
Scan and add RTL	include files into proje project	<u>A</u> dd Files	Add Directories	<u>C</u> reate File		

图 3.15 创建仿真激励文件

第

高章 Vivado使用指

(3) 在如图 3.16 所示的仿真模块定义界面中填写仿真模块的名字为 tb_led,因为是 激励文件不需要对外端口,所以 I/O Port Definitions 部分无须填写,单击 OK 按钮。

For each port s MSB and LSE	le and spe specified: B values wi	cify I/O Ports to	add to your source file. Iess its Bus column is checked.	À
Forts with the	ans names	will not be write	UT I.	
Module Definit	tion			
Module na	me: the le	d		0
Module na	ime: tb_le	d		۵
Module na	me: tb_le	d		0
Module na	finitions	d #		0
Module na NO Port De + — Port N	finitions	d + 		8

图 3.16 仿真模块定义界面

(4) Vivado 界面如图 3.17 所示,在 Sources 窗格的 Simulation Sources 中出现新建的仿真文件 tb_led.v,双击打开该文件,利用 Vivado 的文本编辑器输入激励代码。本例 LED 流水灯的 Test Bench 激励代码如例 3.2 所示。

PROJECT MANAGER - led			?
Sources	? _ 🗆 🖾 ×	Project Summary × tb_led.v × flow_led.v ×	? 🗆 1
Q ¥ ≑ + ∅	• · •	D:/exam/led/led.srcs/sim_1/new/tb_led.v	3
Design Sources (1) Ilow led (flow let)	1.v)	Q 🖬 ♠ ≫ X 🖬 🗈 X // 88 0	
Constraints		6 // Create Date: 2019/10/24 13:08:18 7 // Design Nume:	,
 Consts_1 Simulation Sources (2) sim_1 (2) b_led (tb_led) 	v) _	8 : // Module Hame: tb_led 9 : // Project Hame: 10 : // Target Devices: 11 : // fool Versions: 12 : // Description:	
Hierarchy Libraries Co	ompile Order	13 // Dependencies: 15 // Revision:	
Source File Properties	? _ □ Ľ × ◆ ⇒ ✿	16 : // Revision 0.01 - File Created 17 : // Additional Comments: 10 ://	
Z Enabled	î		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Location: D:/exam/le	d/led.srcs/sim_1/new	21 🗇 module tb_led(22	
Type: Verilog		23 :): 24 🖨 endmodule	
General Properties		95.1	

图 3.17 Vivado 工程管理界面

【例 3.2】 LED 流水灯的 Test Bench 激励代码。

`timescale 1ns/1ns
<pre>module tb_led();</pre>
parameter DELY = 20;

```
req clk;
reg clr;
wire [7:0] led;
flow led i1(
    .clk (clk),
    .clr (clr),
    .led (led));
initial begin
clk = 1'b0; clr = 1'b0;
#(DELY * 2) clr = 1'b1;
end
always
begin
\# (DELY/2) clk = \sim clk;
end
endmodule
```

(5) 在 Flow Navigator 中选择 SIMULATION 下的 Run Simulation 选项,并选择 Run Behavioral Simulation,启动仿真界面,如图 3.18 所示。



图 3.18 仿真界面

端口信号自动出现在波形图中,此外,可通过左侧 Scope 一栏中的目录结构定位到 想要查看的 Module 内部寄存器,在 Objects 对应的信号名称上右击选择 Add To Wave Window,将信号加入波形图中查看。

(6)可通过仿真工具条来对仿真进行设置和操作。仿真工具条如图 3.19 所示,包括 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、暂停等操作。本例中仿真时长设置为 500ms。 第3章

EDA技术与Verilog HDL

A led	- [D;	/exam/l	ed/led.	xpr]	- Viva	do 2	018.2												1			×
Eile	Edit	Flow	Tools	R	ep <u>o</u> rts	W	indow	La	yout	⊻iew	R	un	Help	Q,-	Quick A	ccess		w	rite_bi	tstream	n Comple	ete 🗸
۵,	+	-	10	×	▶,	48	٥	Σ	z	Ø	×	H	•	₽ m	500	ms	~	≛	11	С		X

图 3.19 仿真工具条

(7)最终得到的仿真波形如图 3.20 所示,检查此波形是否与预想的功能一致,以验 证源设计文件的正确性。

▶ Mn 50	00 ms	▼ 王	C					📰 Default Layo	out 🗸
ation - Functional -	sim_1 - tb_le	d							?
flow_led.v ×	flow_led.xde	c x tb_led.v	× Untitled 1	×					? 🗆 🖸
Q. 📕 Q.	Q. 33	•F [4)-F	1≞ ≞r + Г	Fe +F I-	4				٥
Name	Value	0 ms	500 ns	1,000 ms	1,500 ms	2,000 ns	2, 500 ms	3,000 ps	3, 500 ms
14 clk	0								
14 cir	1								
> 164 [4:0]	01	01	02	04	08	10	20	40	80
> M# DELY[31:0	0000000a				0000000				
									~
	$\langle $	<							>

图 3.20 行为仿真波形图

3.1.3 综合与引脚的约束

1. 综合编译

(1)如图 3.21 所示,在 Flow Navigator 中选择 SYNTHESIS 下的 Run Synthesis 选项,对当前工程进行综合,弹出 Launch Runs 对话框,在 Options 中选中 Launch runs on local host 单选按钮,在 Number of jobs 下拉列表中选择最大值,以缩短编译时间,此处选择 8。

(2)编译成功后,在 Flow Navigator 中选择 Synthesis 下的 Schematic 选项,可以查看综合后的电路图。本例的综合后的电路如图 3.22 所示。

2. 添加引脚约束文件

有两种方法可以添加引脚约束,第一种是利用 Vivado 中的 I/O Planning 功能(需先 对工程进行综合,在综合后选择打开 Open Synthesis Design,然后在右下方的选项卡中 切换到 I/O Ports 栏,在对应的信号后输入对应的 FPGA 引脚号);第二种是直接新建 XDC 约束文件。本例采用方法二。XDC(Xilinx Design Constraints)是 Vivado 采用的约 束文件格式,它是在业界广泛采用的 SDC(Synopsys Design Constraints)格式的基础上, 加入 Xilinx 自身的一些物理约束来实现的。

Dj ← → E III X →	μ 🗘 Σ	I CONTRACT ON A CONTRACT OF							
low Navigator 🗧 ⊕ ? _	-	1. I X					🗮 Default I	_ayout	~
PINTEGRATOR	PROJECT MANAG	SER - led						1	? >
Create Block Design	Sources	? _ 🗆 🖾 X	Project Summa	iry x	low_led.	v ×		? 🗆	Ľ
Open Block Design	Q ₹ \$	+	D:/exam/led/led	l.srcs/sour	ces_1/ne	ew/flow_l	ed.v		×
Generate Block Design	∼ 🚍 Design S	ources (1)	Q 🗐 +	-	XE	i in	× // I		¢
_	●∴ flov	/_led (flow_led.v)	22 - module f	low_led(cl	k, clr, le	ed):			^
SIMULATION	📥 Launch Runs		×	lk, clr:					
Run Simulation				eg [7:0	led;				
	Launch the selected	synthesis or implementation runs.	p.						
RTL ANALYSIS			A 15	osedge o	1k)				
> Open Elaborated Design	Launch directory:	Si <default directory="" launch=""></default>	~	begin o	ounter ((= 0:1ed	<= 8' h01; end		L
SYNTHESIS	Options			unter <	50000000) //	2Hz		۰.
Run Synthesis	G Laurahau	a se lasal hast . Number at takes 0		unter <*	counter	r + 1:			~
P Run Synnesis	Launch run	s on local host. Number of jobs. 8	*	-				>	
 Open Synthesized Design 	O Generate s	cripts only						2	15
									-
	Don't show this	s dialog again			-	-			
 Run Implementation 		2470b		WHS	THS	TPWS	Total Power	Failed Route	IS o
> Open Implemented Design		ок	Cancel						_

图 3.21 Synthesis 综合编译



图 3.22 综合后的电路图

第3章 Vivado使用指

(1) 在 Flow Navigator 中选择 PROJECT MANAGER 下的 Add Sources 选项(或右击约束子目录下的文件夹,选择 Add Sourses...),打开如图 3.23 所示的 Add Sources 界面,选中第一项 Add or create constraints 单选按钮,单击 Next 按钮。

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 simulation sources
€ XILINX.	
?	< <u>B</u> ack Einish Cancel

图 3.23 创建约束文件

(2) 在图 3.24 所示的界面中,单击 Create File,在弹出的 Create Constraints File 对 话框中输入 XDC 文件名,本例填写 flow_led,单击 OK 按钮,再单击 Finish 按钮。

Add or Create Constraints	A Create Constraints File ×
Specify or create constraint files for physical and timing constraint to add	Create a new constraints file and add it to your project
Specify constraint set: a constrs_1 (active)	Eile type: 🗈 XDC 🗸
+, - + +	File name: flow_led 💿
	File location: 😜 <local project="" to=""> 🗸</local>
Use Add Files or Cre	Cancel
Add Files	Create File
Copy constraints files into project	
	Sack Next> Finish Cano

图 3.24 输入约束文件名

(3) 如图 3.25 所示,在 Sources 窗格中双击 flow_led.xdc 文件名,打开该文件,填写 引脚约束文件的内容。本例的引脚约束文件内容如例 3.3 所示。

注意:具体的 FPGA 约束引脚号和 I/O 电平标准,应参考目标板卡的用户手册或原理图。



图 3.25 编辑引脚约束文件

【例 3.3】 LED 流水灯的. XDC 引脚约束文件。

3.1.4 生成比特流文件并下载

(1) 如图 3.26 所示,在 Flow Navigator 中选择 PROGRAM AND DEBUG 下的 Generate Bitstream 选项,工程会自动完成综合、实现比特流文件的生成过程。完成后, 选中 Open Hardware Manager 单选按钮,进入硬件编程管理界面。

(2) 进入如图 3.27 所示的 HARDWARE MANAGER 对话框,将目标板通过 USB 连接至计算机,打开电源开关,单击图 3.27 中的 Open target,选择 Auto Connect 选项, 使软件连接到目标板。

(3)软件和目标板连接成功后,软件界面如图 3.28 所示。

在目标芯片上右击,选择 Program device,在弹出的 Program Device 对话框中, Bitstream file 一栏已经自动加载本工程生成的比特流文件 flow_led. bit,单击 Program 按钮对 FPGA 芯片进行编程下载。 第3章

Vivado使

用

指

Flow Navigator 🛛 🚊 🗘 _	SYNTHE SIZED DE SIGN - xc7a35tcsg324-1 (active)	? :
 SYNTHESIS Run Synthesis Open Synthesized Design Constraints Wizard Edit Timing Constraints Set Up Debug Report Timing Summary Report Clock Interaction Report Clock Interaction Report Methodology Report DRC Report Noise Report Ollization Report Power Schematic IMPLEMENTATION Run Implementation Open Implemented Design PROGRAM AND DEBUG Generate Bitstream Open Hardware Manager 	Source N × ? _ □ ™ Project Summary × Device × flow_led.v × Schematic × X X X Image: Schematic x Itstream Generation Completed X Itstream Generation successfully completed. X Next Image: Schematic x Open Hardware Manager Image: Schematic x Image: Schematic x X View Reports X Image: Schematic x X <th>?</th>	?
HARDWARE MANAGER - unconnected ● No hardware target is open Hardware Q × + + + + + + + + + + + + + + + + + +	图 3.26 生成比特流文件	? D C

图 3.27 连接到目标板

(4)下载完成后,在目标板上观察实际运行效果。

3.1.5 将配置数据烧写至 Flash 中

如果将程序烧写到 Flash(ROM)中,则程序会固化到板卡中,可脱机独立运行且掉 电不丢失。

(1) 生成烧录至 Flash 中的. bin 文件,选择菜单 Tools 中的 Settings 命令,在弹出的 Settings 对话框(如图 3.29 所示)中选择 Bitstream,在右面勾选-bin_file,单击 OK 按钮。

ion nangatos _ = = r	HARDWARE MANAGER - localhost/xilinx_td	f/01inx/1234-tulA	?	? >
O Report Timing Summary	There are no debug cores. Program de	wice Refresh devic	e	
Report Clock Networks	Hardware	? _ 🗆 🖒 X	flow_led.v ? 🗆 🖾	×
Report Clock Interaction	Q, ¥ ≑ ∅ ⊳ ≫ ■	•	D:/exam/led/led.srcs/sources_1/new/flow_led.v	×
C Report Methodology	Name	Status		39
Report DRC	V 📓 localhost (1)	Connected		~
Report Utilization	√ m ø xilinx_tcf/Xilinx/1234-tulA(1)	Open	22 🗇 module flow_led(clk, clr, led):	
E Paport Power	v @ xc7a35t_0 (1)	Programmed	23 input elk, elr:	
Cepoirrower	亚 XADC (System Monitor)		24 output reg [/:0] led; 25 reg [28:0] counter:	
A Schematic	A Program Device		×	
 IMPLEMENTATION Run Implementation Open Implemented Design 	Select a bitstream programming file and do select a debug probes file that correspond: programming file.	ownload it to your ha s to the debug cores	rdware device. You can optionally contained in the bitstream 21/2	1
				~
PROGRAM AND DEBUG	Bitstream file: D/exam/led/led.rur	ns/impl_1/flow_led.t	a ©	•
PROGRAM AND DEBUG	Bitstream file: D:/exam/led/led.rur Debug probes file:	ns/impl_1/flow_led.t	xi (2)	~ [5
PROGRAM AND DEBUG III Generate Bitstream Open Hardware Manager	Bitstream file: D/exam/led/led/ru Debug probes file:	ns/impl_1/flow_led.t	at S - 2	~
PROGRAM AND DEBUG Comparison	Bitstregm file: D/exam/led/led.ru Debug probes file: Debug brobes file: Enable end of startup check ?	ns/impl_1/flow_led.t	erogram Cancel	
PROGRAM AND DEBUG Concerned Bitstream Open Hardware Manager Open Target Program Device Add Configuration Memory D	Bitstregm file: D/exam/ledited.ru Debug probes file: Enable end of startup check Type a Tcl comand here	ns/impl_1/flow_led.t	erogram Cancel	

图 3.28 芯片编程下载

۹	Bitstream	
Project Settings General	Specify various settings related to writing bitstream	<i>f</i> ~
Elaboration	Write Bitstream (write_bitstream)	once you open an implemented design.
Synthesis	tcl.pre	
Ritetream	tcl.post	
> IP	-raw_bitfile	
,	-mask_file	
Tool Settings	-no_binary_bitfile	
Project	-bin_file	2
IP Defaults	-readback_file	
Display	-logic_location_file	
WehTalk	-verbose	
Help	More Options	
Y Text Editor 3rd Party Simulators Colors Selection Rules	-bin_file Write a binary bit file without header (.bin).	

图 3.29 勾选_bin_file

(2) 在 Flow Navigator 中选择 PROGRAM AND DEBUG 下的 Generate Bitstream 选项(见图 3.26), 启动编译并自动生成. bit 文件和用于固化的. bin 文件。

(3) 将目标板连接至计算机,打开电源,进入 HARDWARE MANAGER 对话框,如 图 3.30 所示,选中芯片 xc7a35t,右击选择 Add Configuration Memory Device。

第3章 Vivado使用指南



图 3.30 添加 Flash 芯片

(4) 在 Add Configuration Memory Device 对话框(如图 3.31 所示)的搜索框中输入 n25q64,选择 n25q64-3.3v(根据所用的目标板,选择相应的 Flash 芯片型号),单击 OK 按钮。

Add conligt	uration Memor	ry Device					
Choose a co Device: (a) xc7a3:	nfiguration memo	ory part. This can be ch	anged later.				
Manufacturer	All			Type	All		~
Density (Mb)	All		1	Width	All		~
			<u>R</u> eset Al	I Filters			
ect Configuration	Memory Part						
ect Configuration	Memory Part	C) (2 matches	5)			
ect Configuration Search: Q- n2 Name	Memory Part	Part	(2 matches Manufact	s) Alias	Family	Туре	Density
Search: Q- n2 Name n25q64-1.8v	• Memory Part 25q64 -spi-x1_x2_x4	Part n25q64-1.8v	(2 matches Manufact Micron	s) Alias	Family n25q	Type spi	Density 64
ect Configuration Search: Q- n2 Name P n25q64-1.8v n25q64-3.3v	• Memory Part 25q64 •spi-x1_x2_x4 •spi-x1_x2_x4	Part n25q64-1.8v n25q64-3.3v	(2 matches Manufact Micron Micron	s) Alias	Family n25q n25q	Type spi spi	Density 64 64

图 3.31 选择 Flash 芯片型号

(5) 在 HARDWARE MANAGER 对话框中(如图 3.32 所示),选中 Flash 芯片 n25q64-3.3v,右击选择 Program Configuration Memory Device,弹出如图 3.33 所示的对 话框,确认配置文件为 flow_led. bin,单击 OK 按钮,完成对 Flash 芯片的编程。

Hardware			2	
0 ₹ ≜	al⊳l≫l∎i		6	
Name	•• • •• ••	Status	-	
localhost (1)		Conne	ected ^	
✓ ■Ø xilinx_tcf	/Xilinx/1234-tulA (1)	Open		
✓	it_0 (2)	Progra	ammed	
亚 XAD	C (System Monitor)			
9/ n25	q64-3.3v-spi-x1_x2_x4	-	Configuration Memory Device Properties	Ctrl+E
Configuration Mem	nory Device Properties		Remove Configuration Memory Device	Delete
n25q64-3.3v-spi	-x1_x2_x4		Program Configuration Memory Device	
			Readback Configuration Memory Device	
Name:	n25q64-3.3v-spi-x1_x2_	Export to Spreadsheet		
Memory Part	n25q64-3.3v-spi-x1_x	c2_X4		
Memory type: spi Memory density: 64				
		ni 1/fio	w led bin 🕲 📖	

图 3.32 选中 Flash 芯片

elect a configuratio	on file and set pr	ogramming options.	J
Memory Device:	@ n25q64-3.3	v-spi-x1_x2_x4	
Configuration file:	D:/exam/led/le	d.runs/impl_1/flow_led.b	in 💿 -
PRM file:			-
State of non-confi	g mem I/O pins:	Pull-none 🗸	
Program Operation	ons		
Address Rang	ge: Config	uration File Only	~
✓ Erase			
✓ Erase Blank Che	ck		
✓ Erase Blank Che ✓ Program	ick		
 ✓ Erase ☐ Blank Che ✓ Program ✓ Verify 	rck		
Erase Blank Che Program Verify Verify Che	cksum		
Erase Blank Che Program Verify Verify Che SVF Options	cksum		
Erase Blank Che Program Yerify Verify Che SVF Options Create SV	cksum F Only (no progr	am operations)	
Erase Blank Che Program Verify Verify Verify Verify SVF Options Create SVI SVF File:	cksum F Only (no progr	am operations)	

图 3.33 对 Flash 芯片编程

第3章 Vivado使用指南

1

(6) 编程完成后,将开发板断电再重新上电,开发板会从 Flash 中启动,观察开发板的实际运行效果。

3.2 IP 核的创建和封装

基于 IP 核的设计对提高设计的复用具有优越性。Vivado 本身自带了丰富的 IP 核, 还允许设计者自己定义和封装 IP 核。本节以设计和封装功能类似 74LS161 和 74LS00 的 IP 核为例,介绍基于 Vivado 的 IP 核封装流程。

1. 创建工程

启动 Vivado 2018.2,单击 Quick Start 栏中的 Create Project,启动工程向导,创建一 个新工程,将其命名为 ip_161,存于 D:/exam/ip_161 文件夹中,如图 3.34 和图 3.35 所 示。工程创建的过程可参考 3.1 节,此处不再赘述。

Project Name		,
inter a name for yo	ur project and specify a directory where the project data files will be :	stored.
Project name:	ip_161	0
Project location:	D:/exam	
🖌 Create proje	tsubdirectory	
Project will be cr	ated at: D:/exam/ip_161	
\sim		

图 3.34 工程名称、路径设定

A New Project	×
	New Project Summary
	 A new RTL project named 'ip_161' will be created. The default part and product family for the new project. Default Part xc?a35tcsg324-1 Product: Artix-7 Family: Artix-7 Package: csg324 Speed Grade: -1
€ XILINX.	To create the project, click Finish
(?)	< <u>B</u> ack Next > Einish Cancel

图 3.35 工程信息汇总

2. 输入源设计文件

在 Flow Navigator 中选择 PROJECT MANAGER 下的 Add Sources 选项,在弹出的界面中选中 Add or Create Design Sources 单选按钮(参见图 3.10),创建一个名为 ls161.v 的源文件,其代码如例 3.4 所示,输入源文件后的 Vivado 界面如图 3.36 所示。

Flow Navigator \pm \oplus ? .	PROJECT MANAGER - ip_161	? >
Y PROJECT MANAGER	Sources 2 D FLX	Project Summary × Is161.v* × 2 □ 13
Settings Add Sources	Q ★ ♦ + 2 0 ¢	D:/exam/ip_161/ip_161.srcs/sources_1/new/is161.v × Q
P IP Catalog	is161 (is161.v) Constraints · Simulation Sources (1)	22 ⇔ module 1s161 23
 IP INTEGRATOR Create Block Design Onen Block Design 	> □ sim_1 (1) ✓ Hierarchy Libraries Compile Order	25 input wire CET, CEP, LD, 26 input wire DD, D1, D2, D3, 27 output wire CD, 28 output wire Q0, Q1, Q2, Q3);
Generate Block Design	Source File Properties ? □ □ □ × ● Is161.v ← → ◆	29 30 reg [3:0] Q = 0: 31 ⊖ always@posedge CLX or negedge CLR) 32 → if((CLR)
 SIMULATION Run Simulation 	Z Enabled Location: D:/exam/ip_161/ip_161.srcs/sour	33 ; Q <= 4'h0; 34 ⊖ else if("LD) 35 ; Q <= {D3, D2, D1, D0};
 RTL ANALYSIS Open Elaborated Design 	General Properties	36 (c) else if(CET # CEF) 37 { 0 <= 0 + 1'b1; <

图 3.36 输入源设计文件

【例 3.4】 ls161 核的源代码。

```
module ls161
\ddagger (parameter DELAY = 3)(
         input wire CLK, CLR,
         input wire CET, CEP, LD,
         input wire D0, D1, D2, D3,
         output wire CO,
         output wire Q0,Q1,Q2,Q3);
reg [3:0] Q = 0;
always@(posedge CLK or negedge CLR)
   if(!CLR)
      Q <= 4'h0;
   else if(\simLD)
      Q <= \{D3, D2, D1, D0\};
   else if(CET & CEP)
      Q <= Q + 1'b1;
   else Q <= Q;</pre>
assign # DELAY Q0 = Q[0];
assign # DELAY Q1 = Q[1];
assign # DELAY Q2 = Q[2];
```

第3章 Vivado使用指

EDA技术与Verilog HDL

```
assign # DELAY Q3 = Q[3];
assign CO = ((Q == 4'b1111)&&(CET == 1'b1))? 1 : 0;
```

endmodule

在 Flow Navigator 中选择 SYNTHESIS 下的 Run Synthesis 选项,对当前工程进行 综合。综合完成后在弹出的 Synthesis Completed 对话框中单击 Cancel 按钮,表示不再 继续进行后续操作。

3. 创建 IP 核

(1) 在 Flow Navigator 中选择 PROJECT MANAGER 下的 Settings 选项,弹出 Settings 对话框,如图 3.37 所示,在左侧选中 IP 下面的 Packager,在右侧的 Packager 标 签页中定制 IP 核的库名和目录。

Q-	IP > Packager			X	
Project Settings	Specify settings	related to IP Packager.		-	
General					
Simulation	Default Values	Default Values			
Elaboration	The followi	ng values will be automat	ically applied after finishing		
Synthesis	the IP Pack	ager Wizard.	ioniy appres and michning		
Implementation	Vendor	viliny com	0		
Bitstream	Vendor.	Annacom	6		
∨ IP	Library:	UIP	0		
Repository	Category:	74IP	0		
Packager	IP location:	_/ip_repo	0		
Tool Settings					
Project	Automatic Beh	avior			
IP Defaults	After Pack	aina			
Source File					
Display	✓ Cĩe	ate archive of IP			
WebTalk	🖌 Add	IP to the IP Catalog of the	current project		
Help	Clos	e IP Packager window			
> Text Editor		in a denager window			
3rd Party Simulators		ide Source project archive	e		
> Colors	Edit ID in ID	Dackagor			
Selection Rules	cut in in in	Fackager			
	✓ Dele	ete project after packaging	2		

图 3.37 定制 IP 核属性

在 Library(库名)处填写 UIP, Category 处填写 74IP, 勾选 After Packaging 下的 Create archive of IP、Add IP to the IP Catalog of the current project 复选框,其他按默认设置。 设置完成后单击 Apply 按钮,再单击 OK 按钮。

(2) 在 Vivado 主界面中,选择菜单 Tools 中的 Create and Package New IP 命令,如 图 3.38 所示,启动创建和封装新 IP 的过程。此过程的启动界面如图 3.39 所示。

ools	Rep <u>o</u> rts	Window	Layout	View	Help
~	Create and	Pac <u>k</u> age Ne	w IP		
	Create Inter Enable Part Run Tcl Scri Property Ed Associate E Generate M Compile Sir	face Definition ial Reconfigu ipt itor itE Files emory Configu mulation Libg	on uration guration Fil garies	e	Ctrl+J
	Xilinx <u>T</u> cl Sto C <u>u</u> stom Co	ore mmands			,
Q	Language <u>1</u>	emplates			
۰	Settings				

图 3.38 创建和封装新的 IP

Create and Package	e New IP
	Create and Package New IP This wizard can be used to accomplish following tasks:
	Package a new IP for the Vivado IP Catalog This wizard will guide you through the process of creating a new Vivado IP using source files and information from your current project, block design or specified directory.
	Create a new AXI4 Peripheral This wizard will guide you through the process of creating a new AXI4 peripheral which includes HDL, driver, software test application, IP Integrator VIP simulation and debug demonstration design.
€ XILINX.	Click Next to continue
?	< <u>B</u> ack <u>N</u> ext> Einish Cancel

图 3.39 创建和封装新 IP 的启动界面

(3) 单击 Next 按钮,弹出如图 3.40 所示的封装选项界面,选中 Packaging Options 下的 Package your current project 单选按钮,表示将当前的工程封装为 IP 核,单击 Next 按钮。

reate Pe	ripheral, Package IP or Package a Block Design	
lease sele	ct one of the following tasks.	pi pi
Packag	jing Options	
۲	Package your current project Use the project as the source for creating a new IP Definition.	
	Package a block design from the current project Choose a block design as the source for creating a new IP Definition.	
0	Package a specified directory Choose a directory as the source for creating a new IP Definition.	
~	- Dack North	Cancel

图 3.40 封装选项界面

第3章 Vivado使用指南

(4) 如图 3.41 所示,此界面中的 IP location 指示 IP 核的路径,以便设计者到此路径 下将 IP 核导入别的工程中,也可通过单击右侧带省略号的按钮来给 IP 核指定新的位置, 单击 Next 按钮。

A Create and	Package New IP	×
Package You	r Current Project	
Select the directo	ny where the IP Definition will be created and the associated options for packaging the current project.	₽°-
IP location:	d/exam/ip_161/ip_161.srcs/sources_1/new	©
Packaging	IP in the project	
) Inclu	ide xci files	
🔘 Inclu	ide IP generated files	
🔿 Indu	ide IP generated files	
(?)	< Back Next> Finish	Cancel
0		

图 3.41 IP 核的路径

(5) 单击 Finish 按钮,完成 IP 核的创建,如图 3.42 所示。

New IP ×
New IP Creation
The following pieces of information will be gathered:
 Identification information based on top module name
 Family compatibility based on part in the project
 File(s) from Synthesis and Simulation file sets
 Ports from the file containing the top module
 Parameters from the file containing the top module
 Bus Interfaces based on port names
 Address Spaces and Memory Maps based on inferred bus interfaces
< <u>B</u> ack <u>N</u> ext > <u>Einish</u> Cancel

图 3.42 IP 核创建完成

4. 封装 IP 核

(1) 完成 IP 核的创建后,在 Vivado 主界面中,选择 Sources 窗格下的 Hierarchy 标签页,此时在 Design Sources 下方出现一个名为 IP-XACT 的图标,其下有一个 component. xml 的文件,其中保存了封装 IP 核的信息,如图 3.43 所示。

(2) 在 Vivado 主界面右侧窗格中的 Package IP 标签页下,单击 Identification 可查 看并修改 IP 核的相关信息,如图 3.44 所示。



图 3.43 IP 核封装信息文件

Proje	ct Summary X Is161.v X	Package IP - Is161 ×		
Pac	kaging Steps	Identification		
1	Identification	Vendor:	xilinx.com	٢
1	Compatibility	Library:	UIP	٢
~	File Groups	Name:	ls161	\otimes
~	Customization Parameters	Version:	1.0	\odot
0	Ports and Interfaces	Display name:	ls161_v1_0	\otimes
	Addressing and Memory	Description: Vendor display name:	ls161_v1_0	0
1	Customization GUI	Company url:		
	Review and Package	Root directory: Xml file name:	d:/exam/ip_161/ip_161.srcs/sources_1/new d:/exam/ip_161/ip_161.srcs/sources_1/new/componer	itxml
		Categories		_
		+ - *	ŧ	_
-	>	74IP		

图 3.44 封装 IP 核的 Identification 界面

(3) Compatibility 界面显示 IP 核支持的 FPGA 系列,可以继续添加 IP 核支持的 FPGA 器件,单击右侧的加号,选择第一项 Add Family Explicitly...,如图 3.45 所示。

(4) 在弹出的 Add Family 对话框中可添加除已支持的 artix7(Artix-7)外的其他器 件系列,如图 3.46 所示,勾选完毕单击 OK 按钮。

(5) 单击 Customization GUI 界面,在右侧可以预览 IP 核的信号接口,同时可以在 Component Name 文本框中修改 IP 核的名称,如图 3.47 所示。

第3章 Vivado使用指

EDA技术与Verilog HDL

roje	ect Summary × Is161.v)	< Package IP - Is1	51 ×				
Pac	kaging Steps	Compatibility					
1	Identification	Family Sime	ulator				
1	Compatibility	$ +_{2} = + $	+				
×	Company	Family	Life C	ycle			
1	File Groups	virtex7	Produ	ction			
	52.	qvirtex7	Produ	+	Add		Add Family Explicitly
~	Customization Parameters	kintex7	Produ	-	Remove		Add Family using Regular Expression.
0	Ports and Interfaces	kintex71	Produ	1	Move Up	Г	
	Addressing and Memory	qkintex7	Produ	4 #	Move Down		
	Addressing and memory	@ qkintex7l	Produ	ction	1		
1	Customization GUI	artix7	Produ	ction			
Review and Package @ artix71 Pr	Produ	ction					
		aartix7	Produ	ction			
		@ qartix7	Produ	ction			
		zynq	Produ	ction			
		azyną	Produ	ction			
		azynq	Produ	ction			
-		@ coodoo7	Dradu	ation			

图 3.45 IP 核的 Compatibility 界面

Choose a set of families or parts and specify a ife-cycle to apply to them	4
Families	
¥ \$	
✓ ■ All Families and Parts	^
> 🗹 aartix7 (Artix-7)	- 11
> artix7 (Artix-7)	- 11
> 🖌 artix7l (Artix-7)	- 11
> 🖌 aspartan7 (Spartan-7)	- 81
> 🖌 azynq (Zynq-7000)	- 11
> 🗹 kintex7 (Kintex-7)	- 11
> 🗸 kintex7l (Kintex-7)	- 11
> kintexu (Kintex UltraScale)	- 11
V Vinterrinkus (Vinter IltraCasta :)	
	ancel

图 3.46 添加 IP 核支持的器件系列

(6) 单击 Review and Package 界面,可查看 IP 核的最终信息,其中,Root directory 表示 IP 核的存储目录,信息确认无误后单击下方的 Package IP 按钮,完成 ls161 核的封装,如图 3.48 所示。

(7)回到 Vivado 主界面,选择 PROJECT MANAGER 中的 IP Catalog 选项,出现 IP Catalog 窗格,在其中的 User Repository 下可找到刚创建的 ls 161_v1_0,说明该 IP 核

Packaging Steps	Customization GUI		
✓ Identification	Layout	Preview	
Compatibility	Q ¥ ≑ +, *	Show disabled ports	Component Name Is161_0
✓ File Groups	Window Component Name		Delay 3
 Customization Parameters 	∽ 🖻 Page 0		
Ports and Interfaces	Delay	- CLK	
24. H. 2764	Hidden Parameters	- CLR	
Addressing and Memory		- CET CO-	
Customization GUI		CEP Q0	
Deview and Dackage			
Review and Package		_ D0 Q2 _	
		- D1 Q3 -	
		- D2	
		D 3	



roject Summary × Is161.	V × Package IP - Is161 ×
Packaging Steps	Review and Package
✓ Identification	1 warning 2 info messages
✓ Compatibility	Summary
✓ File Groups	Display name: Is161_v1_0
 Customization Parameter 	S Root directory: d/exam/ip_161/ip_161.srcs/sources_1/new
Ports and Interfaces	After Packaging
Addressing and Memory	Create archive of IP - d:/exam/ip_161/ip_161.srcs/sources_1/new/xiilinx.com_UIP_Is161_1.0.zip
 Customization GUI 	edit
Review and Package	IP will be made available in the catalog using the repository - d/exam/ip_161/ip_161.srcs/sources_1/new Edit packaging settings
	Package IP

图 3.48 Review and Package 界面

已创建和封装成功,可以调用了,如图 3.49 所示。

5. 创建和封装另一 IP 核 741s00

采用与上面 74ls161 核相同的步骤,创建和封装功能类似 74ls00(2 输入与非门)的 IP 核,以供调用。ls00 核的源代码如例 3.5 所示。

第3章 Vivado使用指南

IP Catalog					? 🗆 🖒 X
Cores Interfaces					
₹ ≑ ≇ •€, ⊁ ∂	0 0 Q-				0
Name	^1	AXI4	Status	License	VLNV
✓	161/ip_161.srcs/source:	s_1/new)			
✓					
<pre># ls161_v1_0</pre>			Production	Included	xilinx.com:UIP:Is161:1.0
🗸 🖆 Vivado Repository					
> 🚍 Alliance Partners					
> 🚍 Audio Connectivity & Proce	ssing				
< ==					
Details					
Name: Is161 v1 0					
Version: 1.0 (Rev. 2)					
Description: Is161_v1_0					

图 3.49 查看 IP 核

【例 3.5】 ls00 核(2 输入与非门)的源代码。

```
module ls00
#(parameter DELAY = 3)(
    input a, b,
    output y
    );
nand #DELAY(y,a,b);
```

```
endmodule
```

ls00 核创建后,对其进行封装,其中 Identification 界面信息如图 3.50 所示。

Pac	kaging Steps	Identification		
1	Identification	Vendor:	xilinx.com	0
~	Compatibility	Library:	UIP	0
1	File Groups	Name:	Is00	0
1	Customization Parameters	Version:	1.0	0
,	Ports and Interfaces	Display name:	ls00_v1_0	0
	Addressing and Memory	Description: Vendor display name:	ls00_v1_0	0
1	Customization GUI	Company url:		
	Review and Package	Root directory: Xml file name:	d/exam/ip_is00/ip_is00.srcs/sources_1/new d/exam/ip_is00/ip_is00_srcs/sources_1/new/c/	omponent xm

图 3.50 ls00 核的 Identification 界面

章 Vivado使用指南

第 3

单击 Review and Package 界面,可查看 ls00 核的最终信息,信息确认无误后单击下 方的 Package IP 按钮,完成 ls00 核的封装,如图 3.51 所示。

Packaging Steps	Review and Package
✓ Identification	Summary
 Compatibility 	Display name: Is00_v1_0
✓ File Groups	Description: Is00_v1_0 Root directory: d:/exam/ip_Is00/ip_Is00.srcs/sources_1/new
 Customization Parameters 	After Packaging
 Ports and Interfaces 	Create archive of IP - d/exam/ip_Is00/ip_Is00.srcs/sources_1/new/xilinx.com_UIP_Is00_1.0.
Addressing and Memory	edit
Customization GUI	IP will be made available in the catalog using the repository - d:/exam/ip_ls00/ip_ls00.srcs/sources_1/new
Review and Package	Edit packaging settings

图 3.51 ls00 核的 Review and Package 界面

3.3 基于 IP 集成的计数器设计

本节利用 3.2 节创建和封装的 ls161 和 ls00 两个 IP 核,采用原理图设计的方式实现 一个模 9 计数器,以说明基于 IP 集成的 Vivado 设计的流程。

1. 创建工程

启动 Vivado 2018.2,单击 Quick Start 栏中的 Create Project,启动工程向导,创建一个新工程,将其命名为 count_bd,存于 D:/exam/count_bd 文件夹中。此过程不再详述。

2. 添加 IP 核

(1) 将 3.2 节中生成的 IP 封装目录中的压缩包"xilinx.com_UIP_ls161_1.0.zip"和 "xilinx.com_UIP_ls00_1.0.zip"复制到当前工程目录中,并解压到新建的 UIP 目录下, 解压后的文件目录如图 3.52 所示。

(2) 在 Flow Navigator 中选择 PROJECT MANAGER 下的 Settings 选项,在弹出的 Settings 对话框的左侧选中 IP,单击 Repository,出现 Repository 标签页,单击+,进入当前工程目录,选中 UIP 文件夹(其中放置 ls161、ls00 两个 IP 核封装文件),单击 Select 按钮,在弹出的窗口中单击 OK 按钮,上述过程如图 3.53 所示。

(3) 如图 3.54 所示, D:/exam/count_bd/UIP 文件夹已出现在 IP Repositories 下, 单击 Apply 按钮, 再单击 OK 按钮。





Q-	IP > Repository	
Project Settings	Add directories to the list of repositories. You may then add additional IP to a selected	-
General	repository. If an IP is disabled then a tool-tip will alert you to the reason.	
Simulation	IP Repositories	
Elaboration	IP Repositories	
Synthesis	Recent D:/exam/ip_161/ip_1 > 1 🏠 🖵 🕹 🎝 🖵 🗋 🗙	6
Implementation		
Implementation	Directory: D:/exam/count_bd/UIP	
Bitstream		_
∨ IP	✓ ■ exam	
Repository		_
Packager		
	> T xilinx com LIP Is161 10	
Tool Settings	> count bd cache	
Project	> count bd.hw	
IP Defaults	> count bd.ip_user_files	
Source File	> Count_bd.sim	
Display		_
WebTalk	Select Ca	ncel
Liele		

图 3.53 将 IP 核封装文件加入 IP 库

(4) 在 PROJECT MANAGER 下选择 IP Catalog 选项,在右侧的 IP Catalog 标签页 中,展开 User Repository,可以看到用户自定义的 IP 核 ls161_v1_0 和 ls00_v1_0 已经出 现在 IP 库中,可以调用了,如图 3.55 所示。

3. 基于 IP 集成的原理图设计

EDA技术与Verilog HDL

(1) 进入 Vivado 主界面,在左侧的 Flow Navigator 中选择 IP INTEGRATOR 下的 Create Block Design 选项,在弹出的 Create Block Design 对话框的 Design name 栏中输 入设计名 count_bd,表示新建一个名为 count_bd 的原理图文件,如图 3.56 所示。

(2) 单击 OK 按钮,进入 BLOCK DESIGN 设计界面。在原理图中添加 IP 核,可采 用如下方式:

Q		IP > Repository
Project Settings General	î	Add directories to the list of repositories. You may then add additional IP to a selected repository. If an IP is disabled then a tool-tip will alert you to the reason.
Simulation	- 11	ID Descriteries
Synthesis	- 11	
Implementation		+ - +
Bitetroom		D :/exam/count_bd/UIP (Project)
✓ IP		
Repository		
Packager		
		Refresh All

图 3.54 指定 IP 库

Flow Navigator 🗧 🕸 ?	PROJECT MANAGER - count_bd		? :
Y PROJECT MANAGER	Sources 2 D D X	Project Summary × IP Catalog ×	200
Settings	Q <u>¥</u> ≑ + <u>∂</u> ● 0 "	Cores Interfaces	
Add Sources Language Templates	Design Sources	± ≑ ≇ •5 ≁ ⊘ ⊕ 0 Q.	•
ft in outside		Name	^1 AX14
Y IP Catalog	V Simulation Sources		
	⊜ sim_1		
IP INTEGRATOR		# 1s00 v1 0	
Create Block Design	Hierarchy Libraries Compid > =	R Intel ut 0	
Open Block Design			
	IP Properties ? _	Vivado Repository	
Generate Block Design	# ls161_v1_0 ← → 🌣	Alliance Partners	>
SIMULATION	Version: 10 (Rev 2)	Details	
Run Simulation	Description: Is161_v1_0	Name: Is161_v1_0	
	Status: Production	Version: 1.0 (Rev. 2)	
RTL ANALYSIS	times builded	Description: Is161 v1 0	
> Onen Elaborated Design			~

图 3.55 将 ls161 和 ls00 核添加到 IP 库

① 单击原理图中间区域的+按钮。

② 在 Diagram 图形界面上侧工具栏中单击+按钮。

③ 在原理图空白区域右击,在弹出的快捷菜单中选择 Add IP 命令。

在弹出窗口的 Search 搜索栏中输入 ls,在列表中选择 ls161_v1_0,如图 3.57 所示。

(3) 双击 ls161_v1_0,或者按 Enter 键,将其添加到原理图中。采用同样的方法将 IP 核 ls00_v1_0 也调入原理图中,左击选中 ls00_v1_0 模块,右击,选择菜单 Orientation 中的 Rotate Clockwise 命令,连续执行两次,使其旋转 180°,并将其移动到原理图上合适的 位置,如图 3.58 所示。

(4) 连线:将鼠标指针移至 ls161 模块的 Q0 接口处,待其变成铅笔形状后,按下鼠标左键并拖曳到 ls00 模块的 a 接口处,释放鼠标左键后可看到两个接口信号已被连接起

第3章 Vivado使用指



图 3.56 新建原理图并输入文件名



图 3.57 在原理图中添加 IP 核

来。采用同样的方式进行其他连线。

(5) 创建端口。创建端口有以下两种方式。

① 在原理图空白处右击,在弹出的快捷菜单中选择 Create Port...命令,在弹出的 Create Port 对话框中设置端口的名称、方向和类型。图 3.59 所示是创建了一个名为 PT 的输入端口;图 3.60 所示是创建了一个名为 Q0 的输出端口。



图 3.58 添加 IP 核并合理布局

ale port and conne	a it to selected plifs a		
Port name:	PT		Q
Direction:	Input 🗸		
Туре:	Other ~		
Create vector:	from	31 🌲 to	0 🤹
Frequency (MHz):			
Interrupt type:	Level	O Edge	
Sensitivity:	(e) Active High	C Active Low	
Connect to mat	tching selected ports		

图 3.59 创建一个名为 PT 的输入端口

② 单击选中模块的某一引脚,右击,在弹出的快捷菜单中选择 Make External 命令,可自动创建与引脚同名、同方向的端口。

第3章 Vivado使用指

eate port and conne	ct it to selected pins a	nd ports
Port name:	20	Ø
Direction:	Output 🗸	
Тұре:	Other 🗸	
Create vector:	from	31 🗘 to 0 🌻
Frequency (MHz):		
Interrupt type:	(i) Level	O Edge
Sensitivity:	Active High	O Active Low
Connect to mal	Iching selected ports	

图 3.60 创建一个名为 Q0 的输出端口

(6) 连线完成后的原理图如图 3.61 所示,单击原理图工具栏中的 Regenerate Layout,自动对模块和连线进行优化布局。执行 Regenerate Layout 后的原理图如图 3.62 所示。在完成后将对原理图存盘。



图 3.61 完成后的原理图

(7)完成原理图后,生成顶层文件。

① 在 BLOCK DESIGN 对话框的 Hierarchy 标签页中,在 Design Sources 下的



图 3.62 执行 Regenerate Layout 后的原理图

count_bd. bd 图标上右击,在弹出的快捷菜单中选择 Generate Output Products 命令,如图 3.63 所示。

BLOCK DESIGN - count_bd *		?
Sourc × Design Signals ? _ □ Q ≍ ♦ + ? □	⊡ Diagram Q Q X S2 O Q X ⊕ + □	?□¤× ⊮ ⊠ ⊁ °
✓ ☐ Design Sources (1)		
	Source Node Properties Ctrl+E Open File Alt+O Create HDL Wrapper	
> Simulation Sources (1)	View Instantiation Template	D Q0
· · · · · · · · · · · · · · · · · · ·	Generate Output Products LD Q1	
Hierarchy IP Sources Libraries	Reset Output Products D1 Q3	Q2
Source File Properties ? _ [Replace File D2 Copy File Into Project	
🚓 count_bd.bd 🗧 🗧	Copy All Files Into Project Alt+I Is161_v1_0	
☑ Enabled	Remove File from Project Delete Is00_v1_0	
Location: D:/exam/count_bd/	Enable File Alt+뚜号 b	
Type: Block Designs -	Disable File Alt+報号 y a	
General Properties C	Hierarchy Update Refresh Hierarchy	

图 3.63 选择 Generate Output Products 命令

第3章 Vivado使用指南

② 弹出 Generate Output Products 对话框,如图 3.64 所示。Synthesis Options 选项组中有如下选项。

ne foll	owing	output p	products	will be	genera	ated.	
Previe	w						
Q,	¥	\$					
×	E cou	unt_bd.l	bd (000	per IP)		
) s	ynthesi	s				
	🗊 Ir	npleme	entation				
	🗊 S	imulatio	on				
Synthe	esis O	ptions					
0	Globa	al					
۲	Out of	f contex	t per IP				
0	Out of	f contex	t per Blo	ck Des	ian		
Run S	ettings	5					
N	umber	of jobs	: 8		~		

图 3.64 Generate Output Products 对话框

Global: 表示全局综合,选择此选项,则 IP 核生成的文件将会和其他文件一起进行 综合,即每一次设计文件被修改后,IP 核文件都会跟着一起综合一遍。

Out of context per IP:即 OOC 选项,此选项是 Vivado 的默认选项,选择此选项, Vivado 将会把生成的 IP 核当作一个单独的模块来综合,生成.dcp(design checkpoint)文件;在运行实现(Implementation)时,Vivado 会将 OOC 模块的综合网表插入到顶层网 表中,从而完成设计。此选项还会生成一个以 stub 为扩展名的存根文件,类似黑盒子,即 文件中只有输入输出端口。

Out of context per Block Design(此处略)。

本例选择 Out for context per IP 单选按钮,然后单击 Generate 按钮,如图 3.64 所示,完成后单击 OK 按钮。

③ 输出文件生成后,再次在 BLOCK DESIGN 对话框中的 count_bd. bd 图标上右击,在弹出的快捷菜单中选择 Create HDL Wrapper 命令,如图 3.65 所示。

④ 在弹出的 Create HDL Wrapper 对话框中选中 Let Vivado manage wrapper and auto-update 单选按钮,单击 OK 按钮,如图 3.66 所示。

至此,已完成原理图设计。从图 3.67 可看到原理图源文件层次结构图,在 Design Sources 的 count_bd. bd 图标之上已生成 count_bd_wrapper.v 顶层文件。



图 3.65 选择 Create HDL Wrapper 命令

ou can either add or copy the l	HDL wrapper file to the proje	ect. Use copy	option if
ou would like to modily this life	•		
Options			
O Copy generated wr	apper to allow user edits		
• Let Vivado manage	wrapper and auto-update		
_			
?		ок	Cancel

图 3.66 Create HDL Wrapper 对话框

4. 添加引脚约束文件

添加引脚约束有两种方法:第一种是利用 Vivado 中的 I/O Planning 功能;第二种 是直接新建 XDC 约束文件。3.2节中采用了方法二,本例采用方法一来完成此任务。

(1) 在 Vivado 主界面中,在 Flow Navigator 中选择 SYNTHESIS 下的 Run Synthesis 选项,单击 OK 按钮,完成后在弹出的 Synthesis Completed 对话框中选中 Open Synthesized Design 单选按钮,并单击 OK 按钮,如图 3.68 所示。 第 3

|章 Vivado使用指







图 3.68 选中 Open Synthesized Design 单选按钮

(2) 如图 3.69 所示,选择菜单 Window 中的 I/O Ports 命令,使 I/O Ports 标签页出 现在主窗口下方。

(3) 在 I/O Ports 标签页中对输入输出端口添加引脚约束,首先在 Package Pin 栏中



图 3.69 使能 I/O Ports 标签页

输入各端口对应的 FPGA 芯片的引脚号(对应关系可查看目标板说明文档或原理图),本例的 Q0~Q3 锁至 EGO1 开发板的 4 个 LED 灯,CLK 锁至按键 S1,PT 锁至拨码开关 SW0; 然后在 I/O Std 栏中通过下拉菜单选择 LVCMOS33,将所有信号的电平标准设置 为 3.3V,如图 3.70 所示。

Q	H I								4
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std		Vcco	Vref
~ 🗇 CLK.CLK_4055	IN			1	14	LVCMOS33*	•	3.300	
✓ Scalar ports (1)									
CLK	IN		R17 •	/	14	LVCMOS33*	•	3.300	
✓ Scalar ports (5)									
PT	IN		R1 .	/	34	LVCMOS33*	•	3.300	
@ Q0	OUT		К3 ч	/	34	LVCMOS33		3.300	
-@ Q1	OUT		M1 .	/	34	LVCMOS18	•	3.300	
-@ Q2	OUT		L1 ,		34	LVCMOS25	•	3.300	
/F 02	OUT		K6 .	17	34	LVCMOS33		3 300	

图 3.70 对输入输出端口添加引脚约束

(4) 引脚约束完成后单击保存按钮,如图 3.71 所示,可看到 Sources 标签页中已出现引脚约束文件 count_bd_wrapper.xdc,双击该文件,其内容如图中所示。

第

高章 Vivado使用指



图 3.71 引脚约束文件

5. 生成比特流文件并下载

(1) 在 Vivado 主界面的 Flow Navigator 中选项 PROGRAM AND DEBUG 下选择 Generate Bitstream 选项,此时会弹出 No Implementation Results Available 对话框,提 示工程还没有经过 Run Implementation 等过程,如图 3.72 所示,单击 Yes 按钮,再单击 OK 按钮,软件会自动执行 Run Implementation 并生成比特流文件。



图 3.72 生成比特流文件

(2) 生成比特流文件后,选择 Open Hardware Manager 并单击 OK 按钮,用 Micro USB 线连接计算机与板卡,并打开电源开关。在 HARDWARE MANAGER 界面单击 Open Target,选择 Auto Connect,连接成功后,在目标芯片上右击,选择 Program device, 在弹出的 Program Device 对话框中单击 Program 按钮,对 FPGA 芯片进行编程,上述过 程如图 3.73 所示。

	arcount_banpi] v	17400 2010.2			- u ^
Eile Edit Flow Tools Repo	orts <u>W</u> indow Lay	out ⊻iew <u>H</u> elp	Q- Quick Access	s wri	ite_bitstream Complete
	Σ Φ 👫	1% Ø ¥	Dashboard 👻	=	Default Layout 🗸 🗸
Flow Navigator 🗧 🗧 🤉 -	HARDWARE MANA	GER - localhost/xilinx	tcf/Xilinx/1234-tulA		?
Y RTL ANALYSIS	There are no de	ebug cores. Program	device Refresh devic	e	
> Open Elaborated Design	Hardware		? _ 🗆 🗆 X	count_bd_wrapper.xdc	? 🗆 🖒 X
SYNTHESIS	Q ₹ ♦		•	D:/exam/count_bd/count_bd.srcs/cons	trs_1/new/count_bd_v ×
Run Synthesis	A Program Device	e		×	5 × // 88 *
	select a debug probe	s file that corresponds	to the debug cores of	antsinad in the hitstream	et_ports PT]
IMPLEMENTATION Run Implementation Open Implemented Design	programming file. Bitstre <u>a</u> m file:	D:/exam/count_bd/	count_bd.runs/impl_1	(count_bd_wrapper.bit)	et_perts Q0] et_perts Q2]
V IMPLEMENTATION Run Implementation Open Implemented Design PROGRAM AND DEBUG	programming file. Bitstream file: Debug probes file	D:/exam/count_bd/	count_bd.runs/impl_1	/count_bd_wrapper.bit	et_ports 00] et_ports 02] ? _ 0 0

图 3.73 对 FPGA 芯片进行编程

(3)下载完成后,观察开发板实际运行效果。

3.4 Vivado 的综合策略与优化设置

1. 设计的调试和可视化

在 Vivado 设计流程中,一般可查看三个阶段的网络表(如图 3.74 所示),实现设计的可视化:

- 详细设计(Elaborated Design)。
- 综合后的设计(Synthesized Design)。
- 实现后的设计(Implemented Design)。

所谓的网络表(Netlist),是对设计的一种描述,用部件、端口和连线等来表达设计。

(1) 选择 Open Elaborated Design(打开详细设计),该网表是在综合之前表述设计, 一般由复用器、加法器、比较器、寄存器组等较大的部件来表述设计,也称为 RTL 级 设计。 第 3

|章 Vivado使用指



图 3.74 查看不同设计阶段的网络表

(2)选择 Open Synthesized Design(打开综合设计),该网表是在综合之后表达设计, 由 LUT、缓冲器、触发器、进位链等基础元件(BEL)构成的网络表来实现设计。

(3) 选择 Open Implemented Design(打开实现设计),该网表是在布局布线之后具体 实现设计的网表。

2. Language Template(语言模板)

Vivado 提供多种语言模板代码供用户参考,选择菜单 Tools 中的 Language Templates 命令,便会出现 Language Templates 对话框,如图 3.75 所示。从图中可以看出,Vivado 提供 Verilog、VHDL、SystemVerilog 语言模板,还提供 XDC 约束文件模板和 Debug 调试模板,选择 Verilog 模板,可看到包括可综合模板、激励代码模板和 IP 集成器 模板等。

模板代码实现的设计不仅规范,而且更优化,可有效节省 FPGA 资源。图 3.76 所示 是一个固定深度的移位寄存器(SRL)LUT 的可综合模板的参考代码。

建议在设计时尽可能参考语言模板。

3. 综合设置选项

在 Vivado 主界面的 Flow Navigator 中,选择 PROJECT MANAGER 下的 Settings 选项,在弹出的 Settings 对话框中选中 Synthesis 标签页,如图 3.77 所示。以下介绍 Synthesis 标签页中的各项设置。

(1) Constraints 栏:选择用于综合的约束集。约束集是一组 XDC 约束文件,默认选择 active 约束集。约束集包括时序约束和物理约束。

• 时序约束(Timing Constraints):定义设计的时序需求。如果没有时序约束, Vivado 会根据布线长度和布局拥挤度优化设计。





Image: Second analyzing of Winplace Q X ♦ ▲ Image: Write CE and Sync Active High Reset Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image: Second Sync Active Low Reset Image: Second Sync Active Low Reset V Image	<pre>Preview 1 1 2 3 4 vire/reg (data_in>, (data_out>: 5 reg [clock_cycles=1:0] (shift_reg> = {clock_cycles{1'b0}}: 6 7 always @(posedge (clock)) 8 if ((clock_enable)) 9 (ahift_reg> (= {(shift_reg>[clock_cycles=2:0], (data_) 10 11 assign (data_out> = (shift_reg>[clock_cycles=1]: 13 </pre>
Dynamic Depth SRL /w register on output (b Static Shift SRL (bus) Static Shift SRL (single-bit) State-Machines	

图 3.76 固定深度的 SRL 可综合模板代码

第3章 Vivado使用指

Q-	Synthesis			h
Project Settings General Simulation	Specify various set	ttings ass	sociated to Synthesis	<u> </u>
Elaboration	Default constr	aint set:	Constrs_1 (active)	~
Implementation	Report Options			
> IP	Strategy: A	Vivado S	Synthesis Default Reports (Vivado Synthesis 2018)	~
Tool Settings	Options			124
Project	Strategy:	h Vivad	o Synthesis Defaults (Vivado Synthesis 2018)	∼ ⊔ Î
IP Defaults Source File	Description:	User Viva	r Defined Strategies do Strategies	
Display	✓ Synth Desig	1	b Vivado Synthesis Defaults	
WebTalk	tcl.pre		Flow_AreaOptimized_high	
Help	tcl.post		Flow AreaMultThresholdDSP	
> Text Editor	-flatten_hie		Flow_AlternateRoutability	~
3rd Party Simulators	oh haten-		Flow_PerfOptimized_high	~ ~
> Colors Selection Rules	Select an option	1	has Flow_PerrThresholdCarry has Flow_RuntimeOptimized has been supported to the support of the	

图 3.77 Synthesis 设置页面

• 物理约束(Physical Constraints):包括引脚约束,物理单元(如块 RAM、查找表、 触发器)布局的位置等。

(2) Options 栏:用于选择综合运行时使用的策略(Strategy)。Vivado 提供了几种预定义的策略,也可以自定义策略,这需要对策略中每个选项单独设置,这些选项如图 3.78 所示。以下给出各选项的含义。

-flatten_hierarchy: 定义综合工具如何控制层次结构,选择将所有层次融为一体进行综合,还是独立综合再连接到一起。具体选项如下。

• none: 不展开层次结构。

• full: 全部展开层次结构。

• rebuilt:让综合工具展开层次结构,综合之后再重建层次结构。

-gated_clock_conversion:门控时钟转换使能。设计中应避免使用门控时钟,时钟信 号应尽可能由混合模式时钟管理器(Mixed-Mode Clock Manager,MMCM)或者锁相环 (Phase-Locked Loop,PLL)产生。

-fanout_limit: 信号最大驱动负载数量,如果超出了该数值,会复制一个相同的信号 来驱动超出的负载。

-directive:设置 Vivado 综合的优化策略,包括以下选项。

-flatten_hierarchy	rebuilt	
-gated_clock_conversion	off	
-bufg	12	
-fanout_limit	10,000	
-directive	Default	
-retiming		
-fsm_extraction	auto	
-keep_equivalent_registers		
-resource_sharing	auto	
-control_set_opt_threshold	auto	
-no_lc		
-no_srlextract		
-shreg_min_size	3	
-max_bram	-1	
-max_uram	-1	
-max_dsp	-1	
-max_bram_cascade_height	-1	
-max_uram_cascade_height	-1	
-cascade_dsp	auto	
-assert		

图 3.78 综合策略中的 Options 设置选项

- AreaOptimized_high: 面积最省选项。
- AreaOptimized_medium: 面积优化中等,在面积和速度间平衡处理。
- AreaMultThresholdDSP:更多地使用 DSP 模块资源实现设计。
- AlternateRoutability:提高布线能力,减少 MUXF 和 CARRY 的使用。
- PerfOptimized_high:性能最优,即速度最快,但耗用的资源会高一些。
- RuntimeOptimized:综合运行时间最短,会忽略一些优化以减少综合运行时间。

-retiming: 启用该功能,可通过在逻辑门和 LUT 之间的移动寄存器,降低最大路径时延,以提高电路时序性能。

-fsm_extraction:设定状态机的编码方式,默认值为 auto。

- one-hot: 一位热码编码方式。
- gray:格雷编码。
- johnson: 约翰逊编码。
- sequential: 顺序编码。
- auto:此时 Vivado 会自动推断最佳的编码方式。

注意: -fsm_extraction 设定的编码方式优先级高于 HDL 代码中自定义的编码方式。 -keep_equivalent_registers: 保留或合并等效寄存器。勾选该选项,等效寄存器 保留。

-resource_sharing:资源共享。

第

高章 Vivado使用指

-no_lc: 勾选该选项,表示不允许 LUT 整合。当两个或多个逻辑函数的输入变量总 数不超过 6 时,这些函数均可放置在一个 LUT 中实现,称为 LUT 整合。LUT 整合可以 降低 LUT 的资源消耗,但也可能导致布线拥塞。因此,Xilinx 建议,当整合的 LUT 超过 LUT 总量的 15%时,应考虑勾选-no_lc,关掉 LUT 整合。

-no_srlextract: 勾选该选项时,SRL 会用 FPGA 内的触发器实现,而不用 LUT 资源 去实现(Xilinx 的 LUT 可用于实现 SRL)。

-shreg_min_size: 当 SRL 的深度小于或等于-shreg_min_size 时,其实现方式采用触发器级联的方式;当其深度大于-shreg_min_size 时,实现方式则为"触发器+LUT+触发器"的形式。上面的-no_srlextract 选项,如果勾选,则是阻止将移位寄存器用 LUT 实现,其优先级高于-shreg_min_size。

-max_bram: 块 RAM(BRAM)的最大使用数量。默认值为一1,表示允许使用 FPGA 中所有的块 RAM。

-max_uram: 设置 UltraRAM 最大使用数量(对于 UltraScale 架构 FPGA 而言)。 默认值为一1,表示允许使用所有的 UltraRAM。

-max_dsp:设置 DSP 模块的最大使用数量。默认值为一1,表示允许使用该 FPGA 中所有的 DSP 模块。

-max_bram_cascade_height:设置可以将 BRAM 级联在一起的最大数量。

-max_uram_cascade_height:设置可以将 UltraRAM 级联在一起的最大数量。

-assert:将 VHDL 中的 assert 状态纳入评估。

(3) 自定义综合策略:除了 Vivado 提供的配置好的综合策略,还可以自定义综合策略。在 Settings 对话框中设置好各选项后,单击 Options 栏右侧的存盘按钮,弹出 Save Strategy As 窗口,在其中填写名称和描述,即可保存为用户自定义的综合策略,单击 Apply 按钮,综合策略列表中就会出现自定义的策略,可在后面的综合中使用该策略。

在 Settings 对话框的 Tool Settings 中,选择 Strategies 下的 Run Strategies 选项,在 右侧的页面中也可以设置综合策略,单击+按钮可新建策略,如图 3.79 所示。如果想在 已有策略的基础上修改,可选中一个策略,单击上方的 Copy Strategy 按钮,User Defined Strategies 中会出现该策略的备份以供修改(Vivado 提供的策略是不能修改的)。

4. 控制文件编译次序

Vivado 可以自动识别和设置顶层模块,同时自动更新编译次序。

如图 3.80 所示,在 Vivado 主界面的 Sources 窗格中右击 Design Sources,在弹出的 Hierarchy Update 级联菜单中选择 Automatic Update and Compile Order 选项,则设定 当源文件发生改动时,Vivado 会自动管理层次结构和编译次序,Hierarchy 标签页中会自 动调整各模块的层级,Compile Order 标签页中将显示编译顺序。

如果选择 Automatic Update 中的 Manual Compile Order 选项,则表示 Vivado 自动 调整各模块层级,但允许人工设定编译顺序,在 Compile Order 标签页中拖曳文件所处位 置即可完成设定。

	Strategies >	Run Strategies		h			
General	^ Choose a st	rategy based on various options.		₽ -5			
Simulation			•••••				
Elaboration	Flow: Viva	do Synthesis 2018	✓ Name: Vivado Sy	Name: Vivado Synthesis Defaults			
Synthesis			Description: Vivado Su				
Implementation	÷ 👳	+ - 6	Description. Vivado Sy				
Bitstream	User Def	fined Strategies	Options	Options			
P IP	Vivado S	trategies	-flatten_hierarchy	rebuilt			
ool Settings		vado Synthesis Defaults	-gated_clock_co	off			
Project	2011	low_AreaOptimized_nign	-bufg	12			
IP Defaults	200 FT	ow_AreaOptimized_medium	-fanout_limit	10,000			
Source File	2011	ow_AlemateReutability	-directive	Default			
Display	2911	ow_mematerconability	-retiming	1			
WebTalk	A New Strates	gy ×	-fsm_extraction	auto			
Help	Create any shall		-keep_equivalent				
Text Editor	Create new strate	egy.	-resource_sharing	auto			
3rd Party Simulators			-control_set_opt	auto			
Colors	Name:	Strategy 1	-no_lc				
Selection Rules			-no_srlextract				
Shortcuts	Type	Synthesize V		~			
 Strategies 	Tool version:	Vivado Synthesis 2018 🗸	Select an option above to	see a description of it			
Run Strategies	Description						
Report Strategies	Ecosiption.						

图 3.79 自定义综合策略



图 3.80 设置自动更新编译顺序

第3章 Vivado使用指

习题 3

3.1 用 Verilog 语言编写一个功能类似 74ls90 的程序,并用 Vivado 软件进行综合和仿真。

3.2 在 3.1 题的基础上,将 74ls90 的 Verilog 程序封装成一个 IP 核,并采用 IP 核 集成的方式设计一个模 12 计数器,进行引脚锁定和下载。

3.3 在 3.2 题的基础上,调用两个 74ls90 的 IP 核,采用 IP 核集成的方式设计一个 模 60 计数器,个位和十位均采用 8421BCD 码的编码方式,进行引脚锁定和下载。

3.4 基于 Vivado 软件,设计功能类似 74163 的 IP 核,并采用 IP 核集成的方式设计 一个模 24 计数器,进行引脚约束和下载。

3.5 用数字锁相环(PLL)实现分频,输入时钟频率为100MHz,用数字锁相环得到6MHz的时钟信号,用 Vivado 中自带的 IP 核(Clocking Wizard 核)实现该设计。