RISC-V 处理器阿里平头哥玄铁 E902 与 wujian100 的 FPGA 实现

目录

RISC-V 处理器阿里平头哥玄铁 E902 与 wujian100 的 FPGA 实现1
实验准备2
实验一: Wujian100的仿真 4
实验目标4
实验步骤
实验二: Wujian100的 SoC 实现10
实验目标10
实验步骤10
实验三: I/O 口与 SD 卡读写实验
实验目标
实验步骤
实验四: RT-thread Nano OS 移植
实验目标
实验步骤
实验五:在总线上挂载 IP
实验目标
实验步骤
一、建立 Block Design
二、挂 IP 到 wujian100 44
三、实现与验证
附: xdc 文件修改说明58

实验准备

- 1. 开发板
- 2. 参考资料和源代码下载

邸老师主页: <u>http://www.dizhixiong.cn/class5/</u>

课程slides

1-SoC概述

2-玄铁E902处理器与无剑100 SoC体系架构.pdf 3-玄铁E902处理器异常与中断.pdf 4-无剑100 SoC软硬件开发通用工具介绍.pdf 5-基于NexysVideo板卡的FPGA实现和I/O LAB实验.pdf 6-RT-Thread Nano移植.pdf 7-案例:无剑100SoC与softmax硬件加速IP的集成.pdf

课程源代码

基于NexysVideo板卡的FPGA实现
 基于NexysVideo板卡的I/O LAB实验
 RT-Thread Nano移植
 无剑100SoC与softmax硬件加速IP的集成

课程参考资料

1.玄铁E902_R2S2用户手册
 2.RISC-V 指令集手册
 3.RISC-V Core-Local Interrupt Controller
 4.RISC-V platform-level interrupt controller
 5.NexysVideo板卡资料
 6.RT-Thread Nano内核原理与应用
 7.跟我一起写makefile

3. 工具: Vivado, CDK(百度查找下载方法)

Vivado: 可以下载最新版,也可以下载 2018 版,占内存小;

CDK:

平头哥开源工程

平头哥新开源的 OpenXuantie 系列 RISC-V 处理器,包括玄铁 E902、E906、C906、C910 等4 款量产处理 器,以及基于 OpenXuantie 的多操作系统(AliOS、FreeRTOS、RT-Thread、Linux、Android 等)的全栈软 件及工具。 1.Wujian100 SoC 2.玄铁E902内核 3.玄铁E906内核 4.玄铁C906内核 5.玄铁C910内核 **Ar**头哥工具链和CDK集成开发软件 **Nexys-n4-ddr (100T) 移植案例** Verimake成功将玄铁E902与无剑100 SoC移植到Digilent Nexys100T开发板,可给同学们学习RISC-V、 FPGA提供更丰富的学习资源。访问可戳此链接

CPU	~				
SoC	^	资源名称	更新时间	资源大小	操作
无剑100		CSkyDebugServer-V5.8.6	2019-11-20 20:16:01	12.39MB	下载
工具	~	RISC-V+Toolchain-V1.2.2	2019-11-20 20:34:55	454.61MB	下载
开发板		cdk-windows-V1.18.10-20191018	2019-11-20 20:26:43	517.73MB	下载
05	~				

4. 下载器:

亘自

- FPGA USB Cable (连接 Vivado 和板子,将 wujian100 下载到板子上)
- Clink (调试器,用来连接 CDK 和开发板,下载 LED、SD 卡等测试程 序到开发板)
- USB转串口(用来读串口数据)
- 5. 邸老师主页的源代码相关文件有:

kysvideo_wujian100-main > Nexysvideo	_wujian100-main > v
^ 名称 ^	修改日期
IO LAB nexysvideo	2022/6/1 19:06
rtthread_nexysvideo	2022/6/1 19:06
📕 softmax_nexysvideo	2022/6/1 19:07
📙 wujian100_nexysvideo	2022/6/1 19:07
C readme.md	2021/12/4 15:45
vujian100_nexysvideo	2022/6/1 19:07 2021/12/4 15:45

实验一: Wujian100的仿真

实验目标

Wujian100_open 是阿里平头哥在 Github 上开源了 RISC-V 项目。本实验主要完成在 Linux 环境下实现 Wujian100 的仿真。

实验步骤

1.下载官方代码

无剑的仿真需要 Linux 环境,这里使用的是 MobaXterm 的 SSH 终端连接 Ubuntu 的工作站。

eda@abc-MW51-HP0-00:~\$ cd Desktop eda@abc-MW51-HP0-00:~/Desktop\$

建一个文件夹,作该为项目的文件夹: mkdir wujian

切换工作目录到该文件夹下: cd wujian

将 Wujian100 的代码从 github 上下载到该文件夹下: git clone http://github.com/T-

head-Semi/wujian100_open.git



2.下载工具链:

在平头哥的官网上下载工具链。版本可能不同,下载最新的就好。

地址: https://occ.t-head.cn/community/download?id=3913221581316624384

		and the second se			
(m. 198		the second s	And an Aller of	and the second second	u li ni
CPU	~				
SoC	~	资源名称	更新时间	资源大小	操作
工具	^	ReleaseNote.pdf	2021-05-13 11:14:02	95.53KB	下载
开发-软件开发指南		riscv64-elf-i386-20210512.tar.gz	2021-05-13 11:15:09	142.89MB	下载
工具链-800系列	~	riscv64-elf-mingw-20210512.tar.gz	2021-05-13 11:15:22	123.59MB	下载
工具链-900系列	^	riscv64-elf-x86_64-20210512.tar.gz	2021-05-13 11:15:27	139.88MB	下载
V2.4.0		riscv64-linux-i386-20210512.tar.gz	2021-05-13 11:15:42	392.11MB	下载
V2.2.6		riscv64-linux-mingw-20210512.tar.gz	2021-05-13 11:03:05	368.47MB	下载
V2.2.5		riscv64-linux-x86_64-20210512.tar.gz	2021-05-13 11:04:23	388.59MB	下载

创建工具链文件夹: mkdir riscv_toolchain

切换工作目录到该文件夹下: cd riscv_toolchain

eda@abc-MW51-HP0-00:~/Desktop/wujian\$ mkdir riscv_toolchain eda@abc-MW51-HP0-00:~/Desktop/wujian\$ cd riscv_toolchain eda@abc-MW51-HP0-00:~/Desktop/wujian/riscv toolchain\$

将下载好的工具链压缩包放到该文件夹下,这里直接拖拽复制,将文件由 Windows 本机上传到了该 Ubuntu 系统的工作站上。也可用 cp 复制命令复制到 文件夹下。



解压文件: tar -zxvf riscv64-elf-x86_64-20210512.tar.gz

riscv64-elf-x86_64-20210512.tar.gz eda@abc-MW51-HP0-00:~/Desktop/wujian/riscv_toolchain\$ tar -zxvf riscv64-elf-x86_64-20210512.tar.gz 等待解压完成。

3.修改脚本:

由于原本的脚本是 setup.csh 文件,在 bash 环境下有一些不兼容的地方。这里重新建立一个 setup.sh 脚本,如下图所示,放入 wujian100_open/tools 目录下(原 setup.csh 目录)



Cd 到 wujian100_open/tools 目录下:

eda@abc-MW51-HP0-00:~/Desktop/wujian/riscv_toolchain\$ cd ../wujian100_open/tools eda@abc-MW51-HP0-00:~/Desktop/wujian/wujian100_open/tools\$

这里也是将 Windows 本机的 setup.sh 文件上传到工作站上。



输入: source setup.sh

eda@abc-MW51-HP0-00:~/Desktop/wujian/wujian100_open/tools\$ ls
run_case setup.csh setup.sh Srec2vmem Srec2vmem.py
eda@abc-MW51-HP0-00:~/Desktop/wujian/wujian100_open/tools\$ source setup.sh

4.运行仿真

进入到 wujian100_open/workdir 目录下运行仿真: cd ../workdir

开始运行仿真: ../tools/run_case -sim_tool iverilog ../case/timer/timer_test.c



仿真成功。

5.仿真波形的查看

在运行完仿真后,在 wujian100_open/workdir 目录下会生成 test.vcd,如下图所示。VCD 文件包含了信号的变化信息,就相当于记录了整个仿真的信息,我们可以用这个文件来再现仿真,也就能够显示波形。

	-				
eda@abc-MW5	1-HP0-00:~/De	sktop/wujian/wujia	n100_open/workdir	\$ ls	
config.h	getc.c	_lltostr.o	putchar.o	test.pat	v_printf.c
crt0.o	getchar.c	_ltostr.c	putc.o	test.vcd	vprintf.c
crt0.s	getchar.o	_ltostr.o	puts.c	test.vvp	v_printf.o
datatype.h	getc.o	Makefile	puts.o	timer_test.c	vprintf.o
dtostr.c	isinf.c	minilibc_stdio.h	rtl_gpr.log	timer_test.elf	vsnprintf.c
dtostr.o	isinf.o	printf.c	<pre>run_case.report</pre>	timer_test.hex	vsnprintf.o
fprintf.c	isnan.c	printf.h	snprintf.c	timer_test.o	vsprintf.c
fprintf.o	isnan.o	printf.o	snprintf.o	timer_test.obj	vsprintf.o
fputc.c	linker.lcf	putc.c	sprintf.c	vfprintf.c	vtimer.h
fputc.o	_lltostr.c	putchar.c	sprintf.o	vfprintf.o	
eda@abc-MW5	1-HP0-00:~/De	sktop/wujian/wujia	n100_open/workdir	\$	

因为 VCD 是 Verilog HDL 语言标准的一部分,因此所有的 verilog 的仿真器都能够查看该文件。

这里介绍用 Modelsim 软件查看波形。将生成的 test.vcd 文件传输到 Windows 主

К	0 10 0	⑪ ↓ 排序 - 三 查看 -			Session Servers Tools Games Ses
	« Project > wujian100_sim	~ C ,0 t	E wujian100_sim 申	搜索	Quick connect
:)	名称 ^ ① test.vcd	修改日期 2022/7/14 21:16	美型 VCD 文件	大小 14,775 KB	 Imme immediate in the second se

机上,存放于 E:/Project/wujian100_sim 文件夹下。

打开 Modelsim 软件。

点击 File->Change Directory,将目录更改至 test.vcd 的目录下;或者直接在在

Modelsim 中的控制台输入: cd E:/Project/wujian100_sim 直接定为到该文件夹下。



因为 Modelsim 只支持.wlf 波形文件,所以需要做格式转换。

在 Modelsim 中的控制台输入: vcd2wlf test.vcd test.wlf



M Open File Layout NoDesign ColumnLay × $\leftarrow \rightarrow$ \checkmark \uparrow \uparrow \uparrow Project \rightarrow wujian100_sim ~ C / 在 wujian100_sim 中搜索 = • 💷 😗 组织 ▼ 新建文件夹 大小 名称 修改日期 美型 > 🔀 图片 test.wlf 2022/7/14 22:18 WLF 文件 6,552 > 📔 文档 > 🛓 下载 > 👩 音乐

在 Modelsim 中打开 test.wlf, 点击 File->Open->test.wlf

> 🔚 桌面 > 🏪 Windows (C:) 📗

> 新加卷 (D:)	
> 新加卷 (E:)	
文件名(N):	✓ Log Files (*.wlf) ✓
	HDL Files (*.v,*.vl,*.vhd,*.vhdl,*.vho,*.hdl,*.vo,*.vp,*.sv,*.svh,*.svp,*.psl,*.vt,*.vht,*.vqm)
	C/C++ Files (*.c,*.h,*.cpp,*.hpp,*.cxx,*.hxx,*.cc,*.c++,*.cp,*.i,*.ii)
W averation for the second sec	Log Files (".wit)
axi_cdma_v4_1_25 Library E:/program/net/vivado/dds_ex/dds_e	Project Files (*.mpf)
axi_chip2chip_v5_0Library E:/program/net/vivado/dds_ex/dds_e	GZ Files (*.gz)
axi_dock_converterLibrary E:/program/net/vivado/dds_ex/dds_e	Macro Files (*.do,*.tcl)
axi_crossbar_v2_1 Library E:/program/net/vivado/dds_ex/dds_e	Verilog Files (*.v,*.vl,*.vo,*.vp,*.vt,*.vqm)
axi_data_fifo_v2_1Library E:/program/net/vivado/dds_ex/dds_e	SystemVerilog Files (*.sv,*.svh,*.svp)

在 object 标签中选取需要观察的信号添加到波形窗即可。



实验二: Wujian100 的 SoC 实现

实验目标

主要完成 wujian100 SoC 在 FPGA 上的部署。学会使用 Xilinx vivado 工具。

实验步骤

1. 把 wujian100 用 Vivado 打开

1) 创建工程:

Cree Op Op	uick Start ante Project > 1. en Froject >	
🔥 N	ew Project	×
Pro	ject Name	10
Ente	r a name for your project and specify a directory where the project data files will be stored. 2.project的名字和位置	A
В	roject name: project_wujinan	8
P	roject [ocation] E:/software/vivado/pro	⊗
	Create project subdirectory	
Proje Specify	net Type y the type of project to create. 3.选RTL type 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.	A
	□ Do not specify sources at this time	
	Project is an extensible Vitis platform	
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	I/O Planning Project Do not specify design sources. You will be able to view part/package 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.	

next 后添加 verilog 源码:

	À Add	Source Directories						×	
- QUICK ACCOSS	Recent:	E:/software/vivado/p	v on	1 🗠 🖣	ė ±	λ. 🖵	o x		-
💫 New Project	Directory	/SOC设计/课设/Nexy	svideo_wuji	an100-main	/Nexysvi	deo_wuj	ian100-n	nain/v	×
Add Sources Specify HDL, netlist, Block Design, and IP files, or dir also add and create sources later.			 Nexys 	video_wujia xysvideo_v I IO_LAB_ rtthread_ softmax_ wujian100 v 3 wujia v 1 th v 3 wujia	n100-ma vujian100 nexysvid nexysvid nexysvid)_nexysv n_src nead_file soc_file	in I-main eo eo ideo]	î	add it to your project. You can
		> 』商: > ■ 团: > ■ 小	> I openet > I vivado 英马作 支部材料 程序培训	> 3 902-main -boards-ma	ster	olect	Can	cel	
Scan and add RTL include files into project Copy gources into project		Add Files	gd Directoria	25	<u>C</u> reate F	ile			
Add sources from subdirectories Target language: Verlog Simulator	language:	Mixed ~		添加	JUł	lea	ad	利	l soc 文件
•						< <u>B</u> a	ck	N	ext > Enish Cancel

	INJEX	Name	Cibrary	HOL SOURCE FOR		LOCATION					
•	1	head_file	xil_defaultlib	Synthesis & Simulation	-	D:/1723307039/	leRecv/2022	/SOC设计/	课设/Nexysvide	o_wujian100-r	nain/Nexysvi
	2	soc_file	xil_defaultib	Synthesis & Simulation	*	D:/1723307039/F	leRecv/2022_	/SOC设计/	课设/Nexysvide	o_wujian100-r	nain/Nexysvi
	2			Add Files	Ac	d Directories	<u>C</u> reate File		_		
Scan (3.and add RTI	. jnclude files ir	to project	Add Files	A	d Directories	<u>C</u> reate File		_		
Scan a	3. and add RTI	. jnclude files ir project	ito project	Add Files	Ac	d Directories	Create File		_		
Scan a Copy s	and add RTI	. include files ir project subdirectories	to project	Add Files	Ac	d Directories	<u>C</u> reate File		_		
Scan a Copy s Add so get lar	and add RTI sources into ources from nguage: Ve	. include files in project subdirectories enlog V	to project	Add Files	Ac	id Directories	<u>Q</u> reate File		_		

点击 next 后添加 xdc 文件

= + +	🔥 Add Constraint Files	×
	Look jn: 🕞 xdc	✓ t ◊ ₽ ± ∧ ₽ × 0 ⅢΞ
	NexysVideo.xdc	Recent Directories
		□ D./1723307039/FileRecv/2022上/SOC设计/课设/ ~
		File Preview
	2.选定.XdC又们	www.This file is a general . Mdc for the Nexrs V \bigcirc
		,
	File game Nexysvideo xdc	
	Add Files	Create File
logy constraints area into proje	a.	< Back Next > Einish Ca
Constraints (ontional)		
Constraints (optional)	physical and timing constraints.	
Constraints (optional)	physical and timing constraints.	

1) 选择开发板

emaining emaining emaining	all Re	nperature: tic power:	Temp	~	56	Package: ftg2			Filters	Reset All
emaining •	: All Re	mperature: tic power:	Tempe	~	56	Package: ftg2				
emaining	: All Re	tic power:				· ·	~		All	Category:
			Static	~		Speed: -1	~		Artix-7	Family:
Transceivers GT	Ps Gb	Is DSP:	Ultra RAMs	Block RAMs	FlipFlops	LUT Elements	Available IOBs	I/O Pin Count	u-	Part
								~	Q-	Search:
0	0	45	0	25	20800	10400	170	256	ftg256-1	xc7a15tf
0	0	90	0	50	41600	20800	170	256	ftg256-1	xc7a35tf
0	0	120	0	75	65200	32600	170	256	ftg256-1	xc7a50tf
0	0	180	0	105	94400	47200	170	256	ftg256-1	xc7a75tf
0	0	240	0	135	126800	63400	170	256)tftg256-1	xc7a100
	0 0 0 0	45 90 120 180 240	0 0 0 0	25 50 75 105 135	20800 41600 65200 94400 126800	10400 20800 32600 47200 63400	170 170 170 170 170	256 256 256 256 256	ftg256-1 ftg256-1 ftg256-1 ftg256-1 ltfg256-1	xc7a15tf xc7a35tf xc7a50tf xc7a75tf xc7a100

文件添加进来后,VIVADO 会自动识别、编译、分析,VIVADO 分析文件中的错误,用红色波浪线标识(错误原因是没有识别出头文件,将这四个文件类型改为头文件类型即可)



3) 调用时钟 IP



ocking Wizard (6.0)			1
ocumentation I IP Location C Switch	3. 27 输入时轴频率		
and a second			
Symbol Resource			
Show disabled ports	Board Clocking Options Output Clocks Port Renaming MMCM Setti	ngs Summary	
	I MINON O PEL		
	Clocking Features Jitter Optimization		
	C Francisco Darbitación C Malarica Denue		
	Frequency Synthesis Minimize Power		
+ s_ast_lds	Phase Alignment Spread Spectrum Minimize Out	put Jitter	
+ CLK_N2_D	Dynamic Reconfig Dynamic Phase Shift Maximize Inp	t Jitter filtering	
+ CLKFB_N_D CLKFB_OUT_D +	Safe Clock Startup		
 s_axi_areanth ck_stap(3.0) 			
- reset interrupt -	Dynamic Reconfig Interface		
nut_clk clk_out1	Options Phase Duty Cycle Config Write DRF	registers	
- user_cikt	(AXI4Lite () DRP		
- user_ck2 - user_ck3			
- clk_in1	Input Clock Information		
	Input Clock Port Name support requency(innz)	Jitter Options Input Jitte	r Source
	Primary clk_in1 50.000 ③ 10.000 - 800.000	UI • 0.010	Single
	Secondary clk_in2 100.000 30 000 - 00 000	0.010	Single

The phase is	s calculated relative to	the active input cl	ock.			Durte Quelle (%)
Output Clo	ock Port Name	Output Freq (M	IHz)	Phase (degrees) Requested	Actual	Requested
clk_out1	1 clk_out1 🔇	20.000	200000	0.000 🛞	0.000	50.000
clk_out2	2 clk_out2	100.000	N/A	0.000	N/A	50.000
clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000
clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000
clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000
clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000
alk out7	clk_out7	100.000	N/A	0.000	N/A	50.000

2. 根据 perfv 开发板手册添加管脚约束

详细见文件"xdc 的更改".

3. 综合工程,综合无误后进行 bit 流下载

<u>N</u> ame:	synth_1							🝌 Launch Runs	×
Part:	@xc7a50ticsg324-1	1L (active)						Launch the selected synthesis or implementation runs.	
Description:	Vivado Synthesis D)efaults		\otimes					
Status:	Not started							Launch directory: Contract Contract Launch Directory >	
General P	roperties Option	ns Log I	Reports	Mess	sages			Options	ırd
Tcl Console	Messages Log	g Reports	Desi	gn Run	s ×			Launch runs on local host: Number of the second	
Q ¥1	点击《 ▶	» +	%					Generate scripts only	
Name	Constraints S	Status	WNS	TNS	WHS	THS	WBSS	(?) ОК Сапсеі	1
✓ ▷ synth_1	constrs_1	Not started							
⊳ impl_	1 constrs_1	Not started							_

可能出现 error-某某 module 未找到, 注释掉这个 module 就行

- [Synth 8-2654] second declaration of PIN_EHS ignored [wujian100_open_fpga_top.v:159]
- [Synth 8-439] module 'ddr3_mig_wrapper' not found [ahb_matrix_top.v:1026]

综合结果如下:

9 32 Implemented DRC	warnings Report		Total Negative Slack (TNS): Number of Failing Endpoints: Total Number of Endpoints: Implemented Timing Report	0 ns 0 39072
Utilization	Post-Synthesis	Post-Implementation	Power	Summary On-Chip
LUT - LUTRAM - 1 FF - BRAM - IO - BUFG - MMCM -	% 22% 9% 20% 25 50	Graph Table	Total On-Chip Power: Junction Temperature: Thermal Margin: Effective &JA: Power supplied to off-chip devices Confidence level: Implemented Power Report	0.2 W 26.0 °C 59.0 °C (12.1 W) 4.9 °C/W © 0 W Low
Orts Design Runs Df Design Timing Surget Setup Setup Worst Negative Sla Total Negative Sla Number of Failing Total Number of Failing Total Number of E Timing constraints at	AC Methodology ECCEST ack (WNS : 0.377 ns ck (TNS): 0.000 ns Endpoints: 0 ndpoints: 37529 re not me	Power Timing × 满足就行 Hold Worst Hold Slack (WHS): Total Hold Slack (THS): Number of Failing Endpoint Total Number of Endpoints	Pulse Width -0.500 ns Worst Pulse Width Slack (WP -0.500 ns Total Pulse Width Negative Sla 1 Number of Failing Endpoints: 37529 Total Number of Endpoints:	WS): 7.000 ns ick (TPWS): 0.000 ns 0 13609

如果生成 bit 流时出现"没有约束的逻辑端口的错误": 可以添加 tcl 文件消除 错误

100 V2 V3			
📕 .Xil	2022/7/14 0:52	文件夹	
project_wujian100.cache	2022/7/15 17:50	文件夹	
📒 project_wujian100.gen	2022/7/15 17:50	文件夹	
project_wujian100.hw	2022/7/15 17:50	文件夹	
project_wujian100.ip_user_files	2022/7/15 17:50	文件夹	
project_wujian100.runs	2022/7/15 17:50	文件夹	
📕 project_wujian100.sim	2022/7/15 17:50	文件夹	
project_wujian100.srcs	2022/7/15 17:50	文件夹	
nroiect wuijan100.xpr	2022/7/15 17:52	Vivado Project F	***
Diprottcl.tcl 1.建立文	2022/7/15 18:10	Altium Script Do)
U vivado.jou	2022/7/13 19:20	JOU 文件	
📄 vivado.log	2022/7/14 0:52	文本文档	
vivado_6232.backup.jou	2022/6/1 20:46	JOU 文件	
vivado_6232.backup.log	2022/6/1 20:46	文本文档	
		v -	u x
d-i2事本 2.添加语	a	-	
	 project_wujian100.cache project_wujian100.gen project_wujian100.ip_user_files project_wujian100.runs project_wujian100.srg project_wujian100.srg protiect_wujian100.spr proticl.tcl vivado.log vivado.log vivado.6232.backup.jou vivado.6232.backup.log 	ora 2022/7/15 17:50 project_wujian100.cache 2022/7/15 17:50 project_wujian100.ip 2022/7/15 17:50 project_wujian100.ip user_files 2022/7/15 17:50 project_wujian100.ip user_files 2022/7/15 17:50 project_wujian100.ip 2022/7/15 17:50 project_wujian100.ip 2022/7/15 17:50 project_wujian100.ip 2022/7/15 17:50 project_wujian100.sm 2022/7/15 17:50 project_wujian100.xpr 2022/7/15 17:52 prottcl.tcl 1.建立文文件 2022/7/15 18:10 vivado.jou 2022/7/14 0:52 vivado.log 2022/7/14 0:52 vivado_6232.backup.jou 2022/6/1 20:46	project_wujian100.cache 2022/7/15 17:50 文件夹 project_wujian100.gen 2022/7/15 17:50 文件夹 project_wujian100.hw 2022/7/15 17:50 文件夹 project_wujian100.ip_user_files 2022/7/15 17:50 文件夹 project_wujian100.runs 2022/7/15 17:50 文件夹 project_wujian100.runs 2022/7/15 17:50 文件夹 project_wujian100.runs 2022/7/15 17:50 文件夹 project_wujian100.sim 2022/7/15 17:50 文件夹 project_wujian100.spr 2022/7/15 17:50 文件夹 project_wujian100.xpr 2022/7/15 17:52 Vivado Project F protect.tcl 1.建立文文件 2022/7/15 18:10 Altium Script Do vivado.log 2022/7/14 0:52 文本文指 vivado.log 2022/6/1 20:46 JOU 文件 vivado_6232.backup.jou 2022/6/1 20:46 JOU 文件 vivado_6232.backup.log 2022/6/1 20:46 文本文指



λ.	Bitstream					
Project Settings	Specify various settings related to writin	g Bitstream 🗾				
General						
Simulation	(i) Note: Additional bitstream settings will	 Note: Additional bitstream settings will be available once you open an implemented design. 				
Elaboration	VWrite Bitstream (write bitstream)					
Synthesis	tcl.pre*	re\vivado\pro\project_wujian100\prottcl.tcl				
Implementation	tcl.post					
Bitstream	-raw bitfile	Π				
> IP	-mask_file					
Tool Settings	-no_binary_bitfile					
Project	-bin_file*	V				
IP Defaults	-readback_file					
> Vivado Store	-logic_location_file					
Source File	-verbose					
Display	More Options					
Help						
> Text Editor						
3rd Party Simulators	tcl.pre*					
> Colors	pre-step tcl hook					
Selection Rules						

- 4. 下载到开发板上
 - 1) 接口连接电脑和开发板



2) 固化 FLASH:

	1								
		D DEPUC	Q .						
	Constate Pi	DEBUG		Hardware			? _ 🗆 🗆	i × wujia	an100_o
	Generate bi	Stream		Q 🗄 🛊	e ø 1	> > III		3307	7039/File
2	V Open Hard	ware Manager	e e	Name		Statu	s	Q	i 🖬 i
2		Auto Connect	li l	 Iocalhost 	4	Conn	ected	211	ino
	Progr	Recent Targets		- vilio	v_tcf/Digilant	2102033 Oper	•	212	ino
	Add C	Open New Targ	et	∨ @ xc	7a50t 0 (1)	Progr uware Device Pr	operties	213 Ci	trl+E 10
					Pro	gram Device			10
				Hardware Dev	ic Veri	fy Device			10
ld Configuration	n Memory Device (ompleted	×	@ xc7a50t_0	C Ref	resh Device -			10
					She	5			10
O you way	ant to program the co	Infigution memory d	evice now?	Name:	She	W Bus Flot			10
Don't show t	his dialog again			Part:	Add	Configuration M	emory Device	2	10
		OK	Cancel	ID code:	Boo	t from Configura	tion Memory	Device	_
	L	UK	Cancer	General P	rc Pro	gram BBR Key	10 A		
4				Tel Come i	Cle	ar BBR Key			
				Tcl Console	Pro	gram eFUSE Re	gisters		
				Q X 4	Exp	ort to Spreadshe	eet		
🝌 Add Configura	ation Memory De	vice							×
Device: @ xc7a5	0t_0								
Manufacturer	All		~		Туре	All			~
Density (<u>M</u> b)	All		~		Width	All			~
			Baaa	t All Filtere					
			Rese	t All Filters					
Select Configuratio	n Memory Part								
Select Comgutatio	in Memory Part								
<u>S</u> earch: Q- q64		8	(2 matches)						
Name		Part	Manufac	Alias		Family	Туре	Density	Widt
জ n25c <mark>64</mark> -1.8v-	-spi-x1_x2_x4	n25q64-1.8v	Micron			n25q	spi	64	x1_x
▶ n25q64-3.3v-	-spi-x1_x2_x4	n25q64-3.3v	Micron			n25q	spi	64	x1_x
<							<u> </u>		>
2							OK	Can	cel
\bigcirc							ÖN		
Add Configuratio	n Memory Device	Completed		×					
🕜 Do you w	ant to program the	configuration memo	ry device now?						
Don't show	this dialog again	3.							
		ОК	Cancel						
-									

🔔 Program Configuration Memory Device	×
Select a configuration file and set programming options. 添加.bin文件	-
Memory Device: @ n25q64-3.3v-spi-x1_x2_x4	
Configuration file E:/software/vivado/pro/project_wujian100/project_wujian100.runs/impl_1/wujian100_open_top.bin	
PRM file:	
State of non-config mem I/O pins: Pull-none	
Program Operations	
Address Range: Configuration File Only	
✓ Erase	
Blank Check	
✓ Program	
✓ Verify	
Verify Checksum	
SVF Options	
Create SVF Only (no program operations)	
SVF File:	
OK Cancel Apply	
▶ Program Flash ×	
5.	
Flash programming completed successfully.	
ОК	

显示 5 表明 wujian100 已经下载到了开发板里,并且上电复位就可以运行。

实验三: I/O 口与 SD 卡读写实验

实验目标

(1) 在 wujian100 SoC 在 FPGA 上完成部署之后,通过设置该 SoC 的 I/O 引脚 为高电平或者低电平,观察 LED 的亮灭,学会使用 E902 的 I/O。

(2) 学会使用 E902 处理器读写 SD 卡, 需要 SD 卡和 SD 卡的拓展板。

注意: SD 卡相关功能需要 Pref 的 SD 卡拓展板,借插口如下。SD 卡的程序和 LED 的程序是一起的,所以一起就可以测试。(不要热插拔)





实验步骤

1. 连接电脑和 wujian100

Clink与 PMOD 连接:





如下对应关系:

CKLink 接口								
JTAG 引 脚	信号名称	对应引脚	JTAG 引 脚	信号名称	对应引脚			
1	Verf	12(-)	2	NC	6(-)			
3	GND	11(-)	4	NC	5(-)			
5	TMS	10(F13)	6	TDO	4(D11)			
7	TCK	9(F12)	8	TRST	3(E11)			
9	TDI	8(D15)	10	NRST	2(E13)			

JP1 (PMOD) 接口								
JP1 引脚	信号名称	FPGA 引脚	JP1 引脚	信号名称	FPGA 引脚			
1	B15_L13P	E12	2	B15_L13N	E13			
3	B15_L14P	E11	4	B15_L14N	D11			
5	GND	-	6	3V3	-			
7	B15_L15P	D14	8	B15_L15N	D15			
9	B15_L16P	F12	10	B15_L16N	F13			
11	GND	-	12	3V3	-			

2. 连接 USB 转串口下载器和开发





3. 编译调试



Project Settings Build Type: BuildSet			2	> Include Files > Global Functio
Target Output User Compiler Asse Connector Configurations	mbler Linker Debug Fl	 Use: ICE 		Settings
Init File: \$(ProjectPath)////utilities/ Load Configurations I Load Application to Target	ICE Configuration			Connected Debug Target
After Load: ☑ Auto Run ☑ Stop at: main	ICE Clock: NReset Delay:	12000	KHz x10us	WORD[1]: 0x1000000 WORD[1]: 0x24204038 MISA : 0x40001014 Tabat Chin Info:
Misc Configurations Reset CPU Type: Hard Reset V	Reset Wait: CPU Number:	50 0	ms •	CPU Type is E902M, Endian=Little, ISA Patch: 0x0, Revision: 0x0, MGU zone num is 256. MGU zone size is 1288.
After Reset:	Use DDC	Enable TRST		HWBKPT number is 5, HWWP number is 2. MISA: (RV32MCE, Imp M-mode)
Click Help button to confirm the sequen				Update
	Debug			

显示出来了 E902 才可以进行下一步,没有显示出来可以去查一下 PMOD 和 CLINK 的连接是不是错的,或者没连上。

4. CDK 的串口窗口

🚥 [wujian´	100_op	en-hell	lo_world	E:\soft	ware\SO(C\E902\pro\N	exysvide	o_wujian10
<u>F</u> ile <u>E</u> dit	<u>V</u> iew	<u>S</u> DK	<u>P</u> roject	F <u>l</u> ash	<u>D</u> ebug	Pe <u>r</u> ipherals	<u>T</u> ools	<u>W</u> indows
🖹 📔 🗖	× .	Show St	tatus Bar					
Project Vie	~	Show To	oolBar					
√ □ .		Toggle	Current F	old			Alt-RIG	нт
wujian100_o		Toggle	All Folds			Ctrl	-Alt-RIG	нт
uiian1		Toggle	All topmo	ost Fold	s in Selec	tion		201
v 📄 wuji		Toggle	Every Fol	d in Sel	ection			
> 🖬 b		Display	EOL					
> 📒 c		Show W	/hitespace	е				
> 🔤 o								. low
> 📒 li		full Scre	een				Alt-	M p
F	: :	Show W	/elcome F	age				Jan kokoko
~	\checkmark	Load W	elcome P	age at	Startup			
`	~	Output	Pane				Ctr	1-` ad, h'
	~	Project	Pane				Ctrl-Alt-	w threa
Project	~	, Navigat	ion Bar				Ctrl-Alt	-N
Output Vie		Debuad	er Pane				Ctrl-Alt	-D
		France	勾洗	. 8	比者c	rtl+3	Ctr	-1
niscy64 au			e	-			Ctr	-2 1.0
riscv64-u	~	Serial P	ane	- ±	TŦŦŚ	日本	Ctr	-3 _gcc
/mair.c		Cit Dam						
32		OSTrac	e Pane				Ctr	-4
niccy64-u		Analysis	Pane				Ctr	-5
riscv64-u		T = =						rrup
riscv64-u		loolbai	5					W1,-
riscv64-u size of t		Toggle	All Panes				Ctrl-	M elf
text	dat	a	hss	dec	hex fi	lename		

Serial Pane 1.位于右下角 @× rial P	。 → Settings… Port对应上文查到的COM
Close All	Port:
Select All	Baud Rate: 115200
Connect	Data Bits: 8比特率一般不变
Settings	Stop Bit: 1
Log Settings > terminal buffer >	Parity: None ~
	Flow Control: None ~
2.石键,打开setting	ОК
Send: ∨ \r\n ∨ rd: [~
· · · · · · · · · · · · · · · · · · ·	2 Hart #2 Dahua/arint)
× Serial Pane 5. 成功连接	
Connected.	

5. 下载程序

[wujian100_open-hello_world]E:\software\So File Edit View SDK Project Flash Debug	DC\E902\pro\Nexysvideo_wujian100-main\Ne g Pe <u>r</u> ipherals <u>T</u> ools <u>W</u> indows <u>H</u> elp	xysvideo_wujian100-ma	in\IO_LAB_nexysvideo\
New	> = = = = Q 🔂 🔍	🔏 🞯 🔛 🛗 🔘 🖬	🖬 @ @ 🕘 🎻
Open	> Open File	Ctrl-O	
Reload File	Ctrl-R Open Folder		
Load A Group of Tabs	Open Multi-Project Workspa	∞… 1.单击	
Save File	Ctrl-S LED_OFF(); ndelay(500);		
Save as Ctrl	-Shift-S		1.
Save All Files	return 0;		0
Templates Management	d sd test()		
$\leftarrow \rightarrow - \uparrow$ « examples »	hello_world > CDK >	ڻ ~	
组织▼ 新建文件夹	心活旦合莱玫汉		8==
□ 此电脑 ^ 名称	必须走主夹焰住	修改日期	
🧊 3D 对象 🛛 .cdk	E	2022/6/.	
📴 视频 🔤 Lst	2 萬士	2022/6/	
- 图片 Obj	2. 半击	2022/7/	
🖗 文档 🔤 Wuji	an100_open-hello_world.cdkws	2022/7/	
↓ 下载			洗择裏砌塔的マ
♪ 音乐			100+3411/3481/A
三 桌面			
🏪 Windows-SSD			
windows (D:)			
I Seagate Basic			
		>	
File name:		-	CDK Workspace fil
			Open

	5-0-	
Project View		oid)
√ ⋒ ∦ ¥	of all.c	main.c × ff.h led.h led.c lib.c
wujian100 ope ~ BuildSet	~ 16	#include "oled128 32.b"
uujian100_open-hello_world	17	#include "ff.h" /* Declarations of FatFs
v wujian100_open-hello_worl	d 18	FATFS FatFs; /* FatFs work area needed
> 🚞 board	20	FIL Fil; /* File object needed for
> csi_core	21 22	extern void mdelay(uint32_t ms);
> csi_driver	23	<pre>void sd_test();</pre>
	24	int main(void)
	26	<pre>{ rmintf("Halls Wenld()n"); </pre>
hello world	28	key_gpio_intr(PA8);
> configs	29	LED_InitO, while里添加printf函数
> 📩 key_gpio_intr	30	sd test(); 电口给屮合容包丢此
> 🧮 LED	32	while(1) 中口制山云谷勿有一三
> 📩 oled128_32	33	printf("Hello World!\n");
	35	
C main.c	30	ndelay(500); LED OFF():
	38	mdelay(500);
Project		
		HILL FAILS
		all.c
wujian loo_ope V Buildset		
wujian100_open-hello_work	Ontions for	wuijan100 open-belk
v wujian100_open-h	options for	wajian oo_open nene
> board	Packages Pa	th Setting
> csi core		
	Switch SDK	
> 🧮 csi_driver	Switch SDK Create a new	v SDK
> 📩 csi_driver > 📒 libs	Switch SDK Create a new Create a Cou	v SDK mmon Package
> <mark>= csi_driver</mark> > <mark>=</mark> libs > <mark>=</mark> projects	Switch SDK Create a new Create a Con	v SDK mmon Package r "wwiian100 open-be
> isi_driver > ibs > projects	Switch SDK Create a new Create a Con Packages fo	v SDK mmon Package r "wujian100_open-hel
> isi_driver > ibs > projects	Switch SDK Create a new Create a Con Packages fo	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
> isi_driver > ibs > projects	Switch SDK Create a new Create a Con Packages fo Packages fo Build Order.	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
> csi_driver > libs > projects	Switch SDK Create a new Create a Cou Packages fo Packages fo Build Order. Build	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
> csi_driver > libs > projects	Switch SDK Create a new Create a Con Packages fo Build Order. Build Bebuild	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
> csi_driver > libs > projects	Switch SDK Create a new Create a Con Packages fo Build Order. Build Rebuild Clean	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
> ibs > projects 1	Switch SDK Create a new Create a Cou Packages fo Wuj Build Order. Build Rebuild Clean	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wc
> isi_driver > ibs > projects 1	Switch SDK Create a new Create a Co Packages fo Build Order. Build Order. Build Clean Build "wujiar	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
<pre>> csi_driver > libs > projects 1 </pre>	Switch SDK Create a new Create a Co Packages fo Build Order. Build Order. Build Clean Build 'wujiar	v SDK mmon Package r "wujian100_open-hel ian100_open-hello_wo
<pre>> csi_driver > libs > projects 1</pre>	Switch SDK Create a new Create a Co Packages fo Build Order. Build Order. Build Clean Build 'wujian wideo_wujian10	v SDK mmon Package r *wujian100_open-hel ian100_open-hello_work 0-main\Nexysvideo_wujian
<pre>> csi_driver > libs > projects 1 </pre>	Switch SDK Create a new Create a Co Packages fo Build Order. Build Order. Build Clean Build "wujiar wideo_wujian100 Windows	v SDK mmon Package r *wujian100_open-hel ian100_open-hello_work D-main\Nexysvideo_wujian Help

或者用这两个:





6. 复位后就会正确现象是灯闪烁,以及

		🗙 Serial Pane	
۵i×	Serial Pane	Hello World!	
Nexysvi	sd card success Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World!	Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Hello World! Mello SPI INIT Sd card success Hello World! Hello World! Hello World!	1:

表明 SD 卡读取正确,可以在 Debug 调试界面查看寄存器中具体的值。

实验四: RT-thread Nano OS 移植

实验目标

亘自

RT-thread Nano 是一种实时的嵌入式操作系统。RTT 的代码也包括在从邸老师主 页下载的 Nexysvideo_wujian100-main 文件里,这个文件里有着 IO 实验和 RTT 移植要用的所有代码,只需要在 CDK 里打开工程就行了。

<pre></pre>	ujian100-main > 🛛 🗸 🗸
^ 名称	修改日期
IO LAB nexysvideo	2022/6/1 19:06
rtthread_nexysvideo	2022/6/1 19:06
softmax_nexysvideo	2022/6/1 19:07
📙 wujian100_nexysvideo	2022/6/1 19:07
C readme.md	2021/12/4 15:45

(邸老师主页: <u>http://www.dizhixiong.cn/class5/</u>)

课程源代码 1.基于NexysVideo板卡的FPGA实现 2.基于NexysVideo板卡的I/O LAB实验 3.RT-Thread Nano移植 4.无剑100SoC与softmax硬件加速IP的集成

实验步骤

1、在CDK中打开串口面板



2、设置串口

	~				L.0
Serial Pane	1.位于右下角	٦°×	rial Pa	Brt Settings	Port对应上文查到的COM
				Port:	COM12 ~
	Clear All Select All			Baud Rate:	115200 ~
	Copy Connect			Data Bits:	。比特率一般不变
	Disconnect Settings			Stop Bit:	1 ~
	Log Settings >			Parity:	None ~
				Flow Control:	None ~
2.3	右键,打开set	ting			ОК
Send:		✓ \r\n	id:		~
		¥		2 11-	rt #2 Debug(print)
×	Serial Pane 5. 局	切连接			
	Connected.				

3、添加针对 E902 处理器移植的 RTT nano 代码修改版



实验五: 在总线上挂载 IP

实验目标

Wujian100 中有留有许多 dummy 模块,可供用户自定义设计。本文档使用 AHB 总线上的 Dummy0 模块,通过写寄存器的方式控制 RGB LED 外设。



本次实验在 vivado 中建立自己的 IP, IP 为 AXI4 接口。基于三色 LED 灯的控制 需要 9 位,这里用 32 位寄存器 0 的低 9 位作为控制输出。挂载 IP 到 wujian100、 实现与验证功能。



实验步骤

一、 建立 Block Design

Wujian100 软核为 AHB 总线, VIVADO 支持 AXI 总线 IP, 需要用到 AHB-Lite to AXI Bridge 转接模块。

点击 Create Block Design,输入设计名,打开 Block Design 界面。

	,		> 🌒 x_aou_top : aou_to	op (aou_top.v) (3)		
~	IP INTEGRATOR		À Create Block Desi	ign	×	
	Create Block Design	Hierarc	Please specify name	of block design.	2	
	Open Block Design					
	Generate Block Design	Source I	<u>D</u> esign name:	ahb_axi	\otimes	
~	SIMULATION	ahb_n	D <u>i</u> rectory:	😜 <local proj<="" th="" to=""><th>ect> 🗸</th><th></th></local>	ect> 🗸	
	Run Simulation	🕑 Ena	Specify source set:	🗅 Design Sourc	es 🗸	
		Locatio	?	ОК	Cancel	10
~	RTL ANALYSIS	Type:				

点击加号添加 IP, 搜索 AHB-Lite to AXI Bridge, 选择双击添加。

Diagram × NexysVideo.xdc × wujian100)_open_fpga_top.v × ahb_matrix_top.v ×	205
$\textcircled{\begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c } \hline \begin{tabular}{c c c c c } \hline \begin{tabular}{c c c c c c } \hline \begin{tabular}{c c c c c c c } \hline \begin{tabular}{c c c c c c c c c c c c c c c c c c c $	+ ▷ 1 2 3 4 C 1 = Default View V	¢
L		
Search: Q- Abb 📀 (2	matches)	
AHB-Lite to AXI Bridge		
🌻 AXI AHBLite Bridge		
	gn is empty. Press the 🛨 button to add IP.	

点击模块,按图标或 Ctrl+T 引出端口。该桥接模块作为 AXI 主机,自定义 IP 为 AXI 从机。



后面需要对源文件中的端口名进行修改,这里将后缀去掉,待会编辑的时候方

便一点



创建自己的 AXI IP 核, 自定义 AXI IP。点击 Tools->Create and Package New IP。



点击 Next。

	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.
E XILINX.	Click Next to continue
•	<back cancel<="" einish="" th=""></back>

选择创建 AXI 外围的 IP, 点击 Next。

A Create and Package New IP	×
Create Peripheral, Package IP or Package a Block Design Please select one of the following tasks.	4
Packaging Options	
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. Select a block design: a block design: a block design: Package a specified directory Choose a directory as the source for creating a new IP Definition. Create AXIA Peripheral	
Create a new AXI4 peripheral Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.	
? Einish Car	ncel

可对 IP 自己命名,写描述。

Name:	myio	e	
Version: Display name: Description:	1.0	6	
	myio_v1.0	E	
	My new AXI IP		
IP location:	E:/Project/SOC/project_wujian100_50T//ip_repo		
Overwrite e	xisting		

桥是 master,我们就创建 slave 的模式。如下图,点击 Next。

Enable Interrupt Support	+ -		Name	S00_AXI	¢
	Interfaces		Interface Type	Lite	~
	- S00_A	XI	Interface Mode	Slave	~
			Data Width (Bits)	32	~
	<	<	Memory Size (Bytes)	64	~
:: - S00_AXI	>	>	Number of Registers	4	⊗ [451
mvio v1.0					
,					

点击左侧 IP Catalog, 打开 IP 管理器, 看到刚才添加的 myio→ 右击 myio_v1.0 → Edit in IP Packager 选项, 单击 OK, 此时系统会自动打开另一个 Vivado IDE 来对用户 IP 核进行编辑。

Eile Edit Flow Tools Reports	Window Layout View Help	Q- Quick Access		
	🗴 🗹 🕨 🗰 😓 🗶	10 🔀		
Flow Navigator	BLOCK DESIGN - ahb_axi *			
Y PROJECT MANAGER	Sources Design × Signals	2 0 6	Diagram × NexysVideo.xdc	× wujian100_open_fpga_top.v × ah
Settings				
Add Sources	Q ± 3	Q	Cores Interfaces	
Language Templates		^	Q ≚ ≑ ≇ щ ≯	P 0
P IP Catalog	> >> AHB_INTERFACE		Search: Q-	
	> Interface Connections		Name	^1 AXI4 S
> IP INTEGRATOR	√ □ Ports		V 🗅 User Repository (e:/Project/SC	DC/ip_repo/myio_1.0)
	s_ahb_hclk		AXI Peripheral	
✓ SIMULATION	▶ s_ahb_hresetn		2右键 👎 myio_v1.0	Properties Ottat
Run Simulation	> 🖻 Nets	*	Vivado Repository	Guite
			> 🗅 Alliance Partners	IP Settings
✓ RTL ANALYSIS	IP Properties	? _ 🗆 🖒 ×	> 🗅 Audio Connectivity & Pr	Add Repository
> Open Elaborated Design	≢ myio_v1.0	← → ∞	> 😑 Automotive & Industrial	Refresh All Repositories
	Version: 1.0 (Rev. 1)	^		Customize IP
✓ SYNTHESIS	Interfaceo: AVId		Details 3	Edit in IP Packager
Run Synthesis	Interfaces. AAI4		Name: myio_v1.0	Disable IP
P Hun oynarcaia	Description: My new AXI IP		Version: 1.0 (Rev. 1)	

在本模块中,设定有4个寄存器,每个寄存器位宽32位。这里通过往寄存器中 写数据,来达到控制输出端口电平,从而控制三色灯的输出。将输出端口连接 寄存器寄存器。

双击进入 myio_v1_0_S00_AXI,设计文件里面有用户代码区域。

根据需求增加端口声明和添加用户信号。这里增加一个 output [31:0]myio 用于 控制三色灯。

Sources ? _ □ Ľ × Q ≚ + P P o Project Summary × Package IP - myio × myio_v1_0_S00_AXI.v* × Project Summary × Package IP - myio × myio_v1_0_S00_AXI.v* × e:/Project/SOC/ip_repo/rmyio_1.0/hdl/myio_v1_0_S00_AXI.v V □ Design Sources (2) Midow v1_0_S00_AXI inst: myio_v1_0_S00_AXI (myio_v1_0_S00_AXI (myio
Q X + Design Sources (2) > myio_v1_0_(myio_v1_0,v) (1) + X <t< th=""></t<>
Q Image: Constraints Image: Constraints
Impio_v1_0_S00_AXI_inst: myio_v1_0_S00_AXI (myio_v1_) myio_v1_0_S00_AXI_inst: myio_v1_0_S00_AXI (myio_v1_) Impio_v1_0_S00_AXI_inst: myio_v1_0_S00_AXI_Inst: myio_v1_0_S00_AXI_Inst: myio_v1
Constraints Hierarchy Libraries Compile Order
Hierarchy Libraries Compile Order 11 Output [31:0] 12 Output [31:0] 13 Output [31:0] 14 Output [31:0] 15 Output [31:0] 16 Output [31:0] 17 Output [31:0] 18 Output [31:0] 19 Output [31:0] 19 Output [31:0] 10 Not modify the ports beyond this line
Interarchy Libranes Compile Order 19 // User ports ends 20 // Do not modify the ports beyond this line
Source File Properties ? _ D D X 21 22 0 // Global Clock Signal
● myjo_v1_0_S00_AXI.v ← → ✿ 23 input wire S_AXI_ACLK,
✓ Enabled ✓ Enabled 24 // Global Reset Signal. This Signal is Active LOW 25 input wire S_AXI_ARESEIN, 26 // Write address (issued by master, acceped by Slave)

根据需求,添加用户逻辑代码。这里增加 assign myio = slv_reg0;将输出连上寄存器 0。



双击击进入 IP 的顶层, 增加 output [31:0]myio

PROJECT MANAGER - myio_v1_0_project						
Sources ? _ D B ×	Project Summary × Package IP - myio × myio_v1_0_S00_AXI.v × myio_v1_0.v ×					
$\mathbf{Q} \mid \mathbf{X} \mid \mathbf{a} \mid \mathbf{+} \mid \mathbf{C} \mid 0 $	e:/Project/SOC/ip_repo/myio_1.0/hdl/myio_v1_0.v					
✓ □ Design Sources (2) ✓ ● ♣ myio_v1_0 (myio_v1_0.v) (1)						
<pre>myio_v1_0_S00_AXI_inst : myio_v1_0_S00_AXI (myio_v1_ > C IP-XACT (1)</pre>	<pre>// User parameters ends // Do not modify the parameters hereond this line</pre>					
> Constraints	10 11					
Kernel Karley Ka	12 // Parameters of Axi Slave Bus Interface S00_AXI parameter integer C_S00_AXI_DATA_WIDTH = 32, parameter integer C_S00_AXI_DATA_WIDTH = 4					
Source File Properties $? - \Box \ \square \times$	15) 16 (
• myio_v1_0.v	17 (c) // Users to add ports here 18 output [31:0] myio,					
✓ Enabled	19 // User ports ends 20 // Do not modify the ports beyond this line					
Location: e:/Project/SOC/ip_repo/myio_1.0/hdl	21 22 22 22 22 22 22 22 22 22 22 22 22 2					
General Properties	24 input wire s00_axi_aclk,					
	<					

例化中增加.myio(myio),



将更改刷新。切换到 Package IP-myio 窗口,单击如图所示链接,对刚才修改过的顶层文件进行更新。

Identification	Merge changes from File Gro	ups Wizard							
Compatibility			lerge changes from File Groups Wizard						
	≪ x ♥ [•] 4 T	C							
File Groups	Name	Library Name Typ	e Is Include	File Group Name	Model Name				
Customization Parameters	Standard								
、 、	Advanced								
Ports and Interfaces	> 🗁 Verilog Synthesis (2)				myio_v1_0				
Addressing and Memory	> 📄 Verilog Simulation (2)				myio_v1_0				
	> 🚍 Software Driver (6)								
Customization GUI	> 📄 UI Layout (1)								
Review and Package	> 📄 Block Diagram (1)								



重新打包一下 IP



现在可以使用这个 IP 了,将 IP 添加进 Block Design。

🍌 project_wujian100_50T - [E:/Project/S	OC/project_wujian100_50T/project_wujian100_50T.xpr]	Vivado 2021.2	- 0	×
Eile Edit Flow Tools Reports	Window Layout View Help Q- Quick Acces	s	Implementation Complete, Failed Tir	ming! 🗸
	∞ 🖻 🕨 👭 💠 Σ ± # # #		I Default Layout	~
Flow Navigator 🗧 🔍 🚬	BLOCK DESIGN - ahb_axi *			? ×
V PROJECT MANAGER	Sources Design × Signals	2 - 0 0	Diagram x NexysVideo.xdc x wujian100_open_fpga_top.v x ahb_matrix_top.v x IP Catalog x IP Catalog (2) x	200
Settings	요 준 네	0		0
Add Sources		~	2	
Language Templates	> > AHB INTERFACE			
P IP Catalog	> <= M_AX_0		Search: Q- my (1 match)	
	> 😑 Interface Connections		₱ myio_vt.0	
> IP INTEGRATOR			3	
	D s_ahb_hclk		abblite axi bridge 0	
 SIMULATION 	s_ahb_hresetn			
Run Simulation	> 🖾 Nets	~	+ AHB_INTERFACE	
✓ RTL ANALYSIS	External Port Properties	? _ 🗆 🖾 ×	s_ahb_tek M_AXI + M_AXI_0	
> Open Elaborated Design	D s_ahb_hresetn	• • •	AHB-Lite to AXI Bridge	

但它们的 AXI port 仍然有点差异,连接不上,需要加一个互联的 IP



添加 AXI Interconnect 模块

Diagram × Address Editor ×	Address Map ×
Q Q X ∑ ⊖ Q	꽃 ♠ + ▷, ୬ ☑ ★ C 안
Search: Q- axi inter	(3 matches)
AXI4-Stream Interconnect	to and believe O
AXI Interconnect	te_axi_bridge_0
🌻 AXI Interrupt Controller	TERFACE
	د M_AXI +
	setn

双击打开 AXI Interconnect 模块, Slave 和 Master 的接口都为 1。

🍌 Re-customize IP			×	
AXI Interconnect (2.1)			4	☑ ★ C 앱
Ocumentation P Location				
Component Name axi_interconnect_0				axi_interconnect_0
Top Level Settings Slave Interfaces				ARESETN S00_ACLK
Number of Slave Interfaces Number of Master Interfaces Interconnect Optimization Strategy	1 2. 1 2. 3	改为1		MOD_ARESETN MOD_ACLK MOD_ACLK MOD_ARESETN MOD_ARESETN MOD_ARESETN MOD_ARESETN AXI Interconnect
AXI Interconnect includes IP Integrator autom When the endpoint IPs attached to the in width, clock or protocol, a converter IP If a converter IP is inserted, IP integrato configures the converter to match the d To see which conversion IPs have been 'expand hierarchy' buttons to explore in	a uration. im fiffer IP e the intercomposition of trically de 7 sis 8 9 .	onnect.		1.XX±
NOTE:Addressing information for AXI Interco	on 10	editor.	~	
		ОК	Cancel	

可以鼠标拖拽,连接端口



直接点击自动连接



自动连接所以,点击确定。

💫 Run Connection Automation		×
Automatically make connections in your design by checking the boxe	is of the interfaces to connect. Select an interface on the left to display its configuration options on the right.	4
Q 王 章 全选 ∨ ✔ NI Automation (4 off 0f 4 selected) ∨ ♥ II axi_interconnect_0		
 ✓ ≫ ACLK ✓ ≫ M00_ACLK ✓ ≫ S00_ACLK ✓ ⇒ S00_ACLK ✓ ♥ myio_0 		
✓ ≈ s00_axi_ack		
	Select an interface pin on the left panel to view its options	
(?)	ОК Сан	cel

在自动布线后,出现了一个新的模块连接 rst 信号,这里把它删掉,自己手动连接。



将 reset 信号手动连好。







与之前一样修改名称,改为 myio



分配地址。去 Adress Editor 界面,找到 myio_0,右键点击 Assign。

Diagram x	Address Editor	× Ad	dress Map	×		
Q	♦ 1 1 6	🖌 Ass	signed (0)	Unassigned	(1)	Ex
Name				^	¹ Interface	:
	c O					
∨ 🗅 Exte	rnal Masters (1)					
∨ Ⅲ //	HB_INTERFACE (0 ad	dress	bits : 4G)			
~ =	Unassigned (1)	1				
2.右键	Ъ, /myio_0/S00_AXI				SUU VI	:
	г	-	Address Pa	th Properties	Ctrl+E	
	3	7	Assign			
			Exclude			
		ļļ	Assign All			
		*	Linaccian Al	I		

将地址修改为 dummy0 的首地址 0x4001_0000。

Diagram × Address Editor	× Address Map	×				
C ₹ \$ ↓ ↓	Assigned (1)	✓ Unassigned (0)	🖌 E	Excluded (0)	Hide All	
Name		^1 I	nterface	Slave Segment	Master Base Ad	dress F
✓ ► Network 0						
✓						
✓ ■ HB_INTERFACE (0 a	address bits : 4G)					
ॏऺॗ /myio_0/S00_AXI		S	00_AXI	S00_AXI_reg	0x4001_0000	e

回到 diagram 窗口,点击验证一下,没有错误,退出 block design。

3 ×
Ľ
¢

二、 挂 IP 到 wujian100

生成 IP 的 Wrapper。在 Source 里找到生成的 ahb_axi, 右键 Generate Output Products

PROJECT MANAGER - project_wujia	an1(00_50T		
Sources			? _	
Q ¥ ♦ + ? ●	0			۰
✓		Source Node Properties	Ctrl+E	^
> 📄 Verilog Header (4)	E	Open File	Alt+O	
> ● ≞ wujian100_open_tc		Create HDL Wrapper		1 1
> 👬 🔳 ahb_axi (ahb_axi.bd)				1.11
> 🗅 Constraints (1)		View Instantiation Template		
> 🚍 Simulation Sources (6)		Generate Output Products		~
Hierarchy IP Sources Lit		Reset Output Products		
		Replace File		
Source File Properties		Copy File Into Project		
≛ obh ovi hd		• • •		

再右键点击 Create HDL Wrapper。

Sources			? _	- 0 C ×	F
Q		D		٠	
✓		Source Node Properties	Ctrl+E	^	
> 🗅 Verilog Header (4)	-	Open File	Alt+O		
> 🔵 🚠 wujian100_open_		Create HDL Wrapper			
> 👬 🔳 ahb_axi (ahb_axi.ł		oreate fibe wrapper			
> 🗅 Constraints (1)		View Instantiation Template			
> 🚍 Simulation Sources (6)		Generate Output Products		~	
Hierarchy IP Sources		Reset Output Products			
Course File Descention		Replace File			
Source File Properties		Copy File Into Project			

点击 OK。

A Create HDL Wrapper	×
You can either add or copy the HDL wrapper file to the project. Use copy op you would like to modify this file.	otion if
Options	
O Copy generated wrapper to allow user edits	
Let Vivado manage wrapper and auto-update	
?	Cancel

可以看到生成了 wrapper,可以被调用,或者再加一层文件进行调用。



给自己的 design 加一层顶层文件。点击+号,新建设计源文件。

Sources1		? _ 🗆 🗠 🗙 Project Summary
Q ¥ ≑ + 2 ●	🝌 Add Sources	x
∨ □ Design Sources (6)		
> 🚍 Verilog Header (4)		Add Sources
> 🌒 🏥 wujian100_open_top (This guides you through the process of adding and creating sources for your project
> 🔵 ahb_axi_wrapper (ahb_axi	- CARLEN	
> 🖻 Constraints (1)		Add or <u>c</u> reate constraints
> Simulation Sources (6)		<u>A</u> dd or create design sources 2
Hierarchy IP Sources Librar		Add or create simulation sources
Source File Properties		
ahb_axi_wrapper.v		
Enabled		
General Properties		
Tcl Console Messages Log		
Q 素 ≑ I4 ≪ ►		
Name Con:		
✓ ✓ synth_1 (active) con:	🐔 XILIINÄ,	3
✓ impl_1 cons		
✓ □ Out-of-Context Module Runs	(?)	< Back Next > Finish Cancel
✓ clk_wiz_0_synth_1 clk_v		
> c abb avi	Submodulo Punc Co	ampleto

Create File。File name 为 myio_top。点击 Finish。

pecify HDL, netlist, Block Design, and dd it to your project.	IP files and destations and the set of the second destation respect. Create a new source file on disk and Create Source File X	
+, - + +	Create a new source file and add it to your project.	
	Eile type: Verilog V	
	File name: myio_top 2	
	File location: 😜 <local project="" to=""> 🗸 🗸 🗸</local>	
	OK Cancel	
	Add Files Add Directories Greate File 1	
Scan and add RTL include files	into project	
Copy sources into project		
Add sources from subdirectories	S	

点击 OK, 点击 Yes。

🍌 Define 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
Module Definition	
Module name: myio_top	8
I/O Port I A Define Module	
+ • Port Na • The module definition has not been changed. Are you sure you want to use these values? Yes	
?	OK Cancel

要将 main_dummy_top0 替换成 myio_top。在 myio_top 中例化 ahb_axi_wrappper, 并添加与原 dummy 模块文件中相同的端口。

Proje	ect Summary × myio_top.v	× dummy.v ×				
E:/Project/SOC/project_wujian100_50T/project_wujian100_50T.srcs/sources_1/new/myio_top.v						
Q	₩ ◆ > % ■	× // ■ ≂ ♀				
58	wire intr;					
59						
60						
61	wire [2:0] hburst;					
62						
63	wire hready_in;					
64	assign hready_in = hready;					
65						
66	wire [31:0] myio;					
67	- ll - ni mana a shh shi ma					
08 -	and_axi_wrapper u_and_axi_wra	(baddy)				
70	AHE INTERFACE houset	(haddr),				
70	AHB_INTERFACE_nourst	(hprot)				
72	AHB INTERFACE brdata	(hrdata)				
73	AHB INTERFACE bready in	(hready in).				
74	. AHB INTERFACE hready out	(hready).				
75	.AHB_INTERFACE_hresp	(hresp),				
76	.AHB_INTERFACE_hsize	(hsize),				
77	.AHB_INTERFACE_htrans	(htrans),				
78	.AHB_INTERFACE_hwdata	(hwdata),				
		74 · · · · ·				

相关 RTL 代码:

module myio_top(haddr, hclk, hprot, hrdata, hready, hresp, hrst_b, hsel, hsize, htrans, hwdata,

hwrite, hburst,//新增 intr, myio//新增); output [31:0] myio; input hburst; input [31:0] haddr; input [31:0] haddr; input [3:0] hprot; input [3:0] hprot; input hrst_b; input hsel; input [2:0] hsize; input [1:0] htrans;

input [31:0] hwdata;

input hwrite;

output [31:0] hrdata;

output hready;

output [1:0] hresp;

output intr;

wire [31:0] hrdata;

wire hready;

wire [1:0] hresp;

wire intr;

```
wire [2:0] hburst;
```

wire hready_in;

assign hready_in = hready;

wire [31:0] myio;

ahb_axi_wrapper u_ahb_axi_wrapper(//例化 ahb_axi_wrapper 模块

.AHB_INTERFACE_haddr (haddr),

.AHB_INTERFACE_hburst (hburst),

.AHB_INTERFACE_hprot (hprot),

.AHB_INTERFACE_hrdata (hrdata),

.AHB_INTERFACE_hready_in (hready_in),

.AHB_INTERFACE_hready_out (hready),

.AHB_INTERFACE_hresp (hresp),

.AHB_INTERFACE_hsize (hsize),

.AHB_INTERFACE_htrans (htrans),

.AHB_INTERFACE_hwdata (hwdata),

.AHB_INTERFACE_hwrite (hwrite),

.AHB_INTERFACE_sel (hsel),

```
.myio (myio),
.s_ahb_hclk (hclk),
.s_ahb_hresetn (hrst_b)
);
```

endmodule

要将 main_dummy_top0 替换成 myio_top。依次修改 ahb_matrix_top, pdu_top, wujian100_open_top 中的例化。

Sources ? _ □	Ľ X
	٥
Search: Q- main_dummy_top0 (2 matches)	
✓	^
✓ ● ♣ wujian100_open_top (vujian100_open_fpga_top.v) (66)	
x_pdu_top : pdu_top (pcu_top.v) (4)	
x_main_bus_top : ahb_matrix_top (ahb_matrix_top.v) (10)	
x_main_dummy_top0 : ahb_dummy_top (dummy.v)	
	~
Hierarchy IP Sources Libraries Compile Order	

双击打开修改 ahb_matrix_top 文件



找到 x_main_dummy_top0 的例化

Project	Summary × myio_top.v	× ahb_matrix_top.v ×
E:/Proje	ct/SOC/project_wujian100_50T/	project_wujian100_50T.srcs/sources_1/imports/wujian_src/so
Q	← ≁ X ■	Ů 🗙 🖊 🎟 ≂ ♀
851	.hwdata	(hmain0_dmemdummy0_s5_hwdata),
852	.hwrite	(hmain0_dmemdummy0_s5_hwrite),
853	.intr	(main_dmemdummy0_intr)
854 🔶);	
855 👳	ahb_dummy_top x_main_dumm	uy_top0 (
856	. haddr	(hmain0_dummy0_s7_haddr),
857	.hclk	(pmu_dummy0_hclk),
858	.hprot	(hmain0_dummy0_s7_hprot),
859	.hrdata	(dummy0_hmain0_s7_hrdata),
860	.hready	(dummy0_hmain0_s7_hready),
861	.hresp	(dummy0_hmain0_s7_hresp),
862	.hrst_b	(pmu_dummy0_hrst_b),
863	.hsel	(hmain0_dummy0_s7_hsel),
864	.hsize	(hmain0_dummy0_s7_hsize),
865	. htrans	(hmain0_dummy0_s7_htrans),
866	.hwdata	(hmain0_dummy0_s7_hwdata),
867	.hwrite	(hmain0_dummy0_s7_hwrite),
868	.intr	(main_dummy0_intr)
869 🖕);	
870 🖯	dmac_top x_dmac_top (
871	.ch0_etb_evtdone	(ch0_etb_evtdone),
1	<	/· · · · · · · · · · · · · · · · · · ·

将例化更改为 myio_top, 注意:

1.还要添加新增的 output 端口 myio, 信号需要引出到最顶层, myio
 2.新增了 hburst,之前的 dummy 空模块中未使用,这里需要加上。

Projec	t Summary	× myio_top.v × ahb_matrix_top.v ×
E:/Proj	ject/SOC/proj	ject_wujian100_50T/project_wujian100_50T.srcs/sources_1/import
Q,		→ X E K X // III F Q
866	// .hwda	ta (hmain0_dummy0_s7_hwdata),
867	// .hwri	te (hmain0_dummy0_s7_hwrite),
868	// .intr	(main_dummy0_intr)
869	1D;	更改
870 🤅	🤉 myio_top	x_main_dummy_top0 (//ahb_dummy_top
871	. haddr	(hmain0_dummy0_s7_haddr),
872	.hclk	(pmu_dummy0_hclk),
873	.hprot	(hmain0_dummy0_s7_hprot),
874	.hrdata	(dummy0_hmain0_s7_hrdata),
875	. hready	(dummy0_hmain0_s7_hready),
876	.hresp	(dummy0_hmain0_s7_hresp),
877	.hrst_b	(pmu_dummy0_hrst_b),
878	.hsel	(hmain0_dummy0_s7_hsel),
879	.hsize	(hmain0_dummy0_s7_hsize),
880	. htrans	(hmain0_dummy0_s7_htrans),
881	.hwdata	(hmain0_dummy0_s7_hwdata),
882	.hwrite	(hmain0_dummy0_s7_hwrite),
883	.hburst	(hmain0_dummy0_s7_hburst), 增加
884	.intr	(main_dummy0_intr),
885	.myio	(myio) 增加
886 E);	
	<	

在改文件开始处,增加输出引脚和定义:

Project	t Summary	/ × m	yio_top.v × ahb_matrix_top.v ×					
E:/Project/SOC/project_wujian100_50T/project_wujian100_50T.srcs/sources_1/ir								
Q,		*	% 🔲 🛍 🗙 🖊 🖩 🖛 ♀					
112	pmu_mdummy2_hrst_b,							
113	smc_h	main0_s2	2_hrdata,					
114	smc_h	main0_s2	2_hready,					
115	smc_h	main0_s2	2_hresp,					
116	smc_h	main0_s3	3_hrdata,					
117	smc_h	main0_s3	8_hready,					
118	smc_h	main0_s3	B_hresp,					
119	smc_h	main0_s4	L_hrdata,					
120	smc_h	main0_s4	L_hready,					
121	smc_h	main0_s4	L_hresp,					
122	myio							
123);							
124	output	[31:0]	myio;					
125	wire	[31:0]	myio;					
126	input	[31:0]	cpu_hmain0_m0_haddr;					
127	input	[2 :0]	cpu_hmain0_m0_hburst;					
128	input	[3 :0]	cpu_hmain0_m0_hprot;					
129	input	[2 :0]	cpu_hmain0_m0_hsize;					
130	input	[1 :0]	cpu_hmain0_m0_htrans;					
131	input	[31:0]	cpu_hmain0_m0_hwdata;					
132	132 input cpu_hmain0_m0_hwrite;							
1	<							

双击打开修改 pdu_top 文件



Project	Summary ×	myio_top.v	× ahb_matrix_top.v ×	pdu_top.v ×				
E:/Project/SOC/project_wujian100_50T/project_wujian100_50T.srcs/sources_1/imports/wujia								
Q,	-	λ 🖬 Ι	ŭ 🗙 🖊 🖩 🎟 🖛	Q				
946	wire	usi2_wic_	intr;					
947	wire	wdt_pmu_r	st_b;					
948	wire	wdt_wic_i	ntr;					
949 🖯	ahb_matrix_t	op x_main_bu	s_top (
950	.myio		(myio),				
951	.cpu_hmain	0_m0_haddr	(cpu_hmain0_m0_haddr),				
952	.cpu_hmain	0_m0_hburst	(cpu_hmain0_m0_hburst),				
953	.cpu_hmain	0_m0_hprot	(cpu_hmain0_m0_hprot),				
954	.cpu_hmain	0_m0_hsize	(cpu_hmain0_m0_hsize),				
955	.cpu_hmain	0_m0_htrans	(cpu_hmain0_m0_htrans),				
956	.cpu_hmain	0_m0_hwdata	(cpu_hmain0_m0_hwdata),				
957	.cpu_hmain	0_m0_hwrite	(cpu_hmain0_m0_hwrite),				
958	.cpu_hmain	0_m1_haddr	(cpu_hmain0_m1_haddr),				
959	.cpu_hmain	0_m1_hburst	(cpu_hmain0_m1_hburst),				
960	.cpu_hmain	0_m1_hprot	(cpu_hmain0_m1_hprot),				
	• • •		· · · · · · ·	×				

找到例化的 ahb_matrix_top,添加 myio 连接

同样, 增加输出引脚和定义

Project Summary × myio_top.v × ahb_matrix_top.v × pdu_top.v * ×
E:/Project/SOC/project_wujian100_50T/project_wujian100_50T.srcs/sources_1/imports/wujian
Q, 🔛 🛧 🕕 🐰 🗈 🗈 🗙 🖊 🖩 두亘 ♀
304 usi2_ioct1_sd0_ie_n,
305 usi2_ioctl_sd0_oe_n,
306 usi2_ioct1_sd0_out,
<pre>307 usi2_ioctl_sd1_ie_n,</pre>
308 usi2_ioctl_sd1_oe_n,
<pre>309 usi2_ioctl_sd1_out,</pre>
310 usi2_wic_intr,
311 wdt_pmu_rst_b,
312 wdt_wic_intr,
313 myio
314);
315 output [31:0] myio;
316 wire [31:0] myio;
317 input [31:0] cpu_hmain0_m0_haddr;
318 input [2:0] cpu_hmain0_m0_hburst;
319 input [3:0] cpu_hmain0_m0_hprot;
320 input [2:0] cpu_hmain0_m0_hsize;
321 input [1:0] cpu_hmain0_m0_htrans;

双击打开修改 wujian100_open_top 文件



找到例化的 pdu_top, 添加 myio 连接

Project	t Summary × myio_top.v	× ahb_matrix_top.v	/ × pdu_top.v	× wujian100_open_fpga_top.v ×
E:/Proj	ject/SOC/project_wujian100_5)T/project_wujian100_501	C.srcs/sources_1/imp	ports/wujian_src/soc_file/wujian100_open_fpga
Q,	₩ ★ ≯ & ■	■ × // ■	₽= 0	
768	.rtc_wic_intr	(rtc_wic_intr),	
769	.test_mode	(test_mode),	
770	.wdt_pmu_rst_b	(wdt_pmu_rst_b)	
771);			
772				
773				
774				
775 Ę	pdu_top x_pdu_top (
776	.myio	(myio),	
777	.apb0_dummy1_intr	(apb0_dummy1_intr),	
778	.apb0_dummy2_intr	(apb0_dummy2_intr),	
779	.apb0_dummy3_intr	(apb0_dummy3_intr),	
780	.apb0_dummy4_intr	(apb0_dummy4_intr),	
781	.apb0_dummy5_intr	(apb0_dummy5_intr),	
782	.apb0_dummy7_intr	(apb0_dummy7_intr),	

同样, 增加输出引脚和定义

Project	Summary × myio_top.v	× ahb_matrix_top.v	× pdu_top.v	× wujian100_open_fpga_to	p.v ×
E:/Proje	ect/SOC/project_wujian100_50	l/project_wujian100_50T	.srcs/sources_1/imp	orts/wujian_src/soc_file/wujian10	00_open_fpg
Q	- × × E		₽⊒		
97	PAD_USI2_SCLK,				
98	PAD_USI2_SD0,				
99	PAD_USI2_SD1,				
100	// PIN_EHS,				
101	clk,				
102	POUT_EHS,				
103	vadj_en,				
104	set_vadj,				
105	myio				
106);				
107	output [31:0] myio;				
108	wire [31:0] myio;				
109	input clk;	//100Mhz			
110 뒂	clk_wiz_0 u_clk_wiz_0				
111	(
112	// Clock out ports				
113	.clk_out1(PIN_EHS),				

现在就实现了IP的挂载。

在约束文件中添加对应功能的引脚,这里将 myio 的低 9 位连接到三色灯的 9 个 引脚上。



像之前的教程一样,综合,实现,再生成 BIT 文件,下载到开发板上。



三、 实现与验证

在 CDK 中编写简单的程序验证功能。



附: xdc 文件修改说明

说明:修改 xdc 文件(即该芯片的约束文件),要配置成所用板子一致的 xdc 需要的资料:开发板用户手册; wujian100 工程里的 xdc 文件

1. 开发板用户手册重点阅读内容:

1) 时钟:

2.4 有源晶振 开发板提供的 FPGA 系统时钟源为 50Mhz 有源晶振电路。晶振输出信号端 SYS_CLK 连接到 FPGA 的 BANK14 全局时钟管脚 N14 (IO_L12P_T1_MRCC_14), 这

2) IO 口电压:不同开发板用的电压不一样

余 4	XC7A35T 个 Bank I/	-1FTG256C 共 O 的连接情况	有 5 个 I/0 如下表。	<mark>Bank,</mark> 其中 U2E 是 FPGA 专用的酉	己置 B	ank,其
Ban	k	7		用途		
Ban	k14 (3.3V)		LED/KEY/P1/P2		
Ban	k15 (3.3V)		P1/P2/JP1		
Ban	k34 (3.3V)		USER_JTAG/USER_FLASH/P1		
Ban	k35 (1.5V)		DDR3		

设置电压的约束:

set_property CONFIG_VOLTAGE 3.3 [current_design]: 全部设定电压为 3.3

3) 引脚: 以灯和开关为例子:

	9个 LED	
LED	信号名称	FPGA 引脚
D0	LED0	M16
D1	LED1	N16
D2	LED2	P15
D3	LED3	P16
	D4B	M2
D4	D4G	L5
	D4R	P5
	D5B	N12
D5	D5G	Т9
	D5R	T10
	D6B	D10
D6	D6G	P6
	D6R	K12
D7	FPGA_DONE	H10
D8	电源指示灯	-

4) Flash

2.9 SPI FLASH

开发板上使用了两片 8MB(64Mbit)大小的 SPI FLASH 芯片,型号为 25Q064A,它

5) 接口:



al	<u>R60</u>	TCK1	12		
	1KΩ (1001) ±1%	TORT	13		-
	0.50	TDO1		4 123/2	1X0
3V3	R/9	TMS1	5	6 343	JTAG1_SRST
	10KΩ (1002) ±1%	JIAOI IKSI	9	10	KAU
3V3		TDI1	USER	JTAG	
	$10K\Omega(1002) \pm 19$	Yo	.002.1		
		USER J1	AG 接口		
USER			USER		
JTAG	信号名称	FPGA 引脚	JTAG	信号名称	FPGA 引脚
引脚		2 D	引脚		
erf-V	/ datasheet		_	Perf	Lab
erf-V	datasheet	N11	2	Perf/	Lab
erf-V	/ datasheet	N11 M1	2 4	Perf/	P9
1 3 5	TCK1 TDO1 TMS1	N11 M1 N3	2 4 6	TX0 TX0 ITAGL SPST	P9
1 3 5 7	TCK1 TDO1 TMS1 JTAG1_TRST	N11 M1 N3 L4	2 4 6 8	TX0 TX0 ITAGL SPST RX0	P9

两种接口区别:

UART: 是嵌入式开发所说的串口。通过它可以向设备烧写程序和调试程序。 JTAG: 用于设备调试,需要硬件支持。主要用于芯片内部测试,调试程序。可 以直接观察和修改寄存器和内存中的数据,方便找出程序中的问题。

- 2. xdc 约束更改
- 1) 时钟:

## Clock Signal	
<pre>set_property -dict { PACKAGE_PIN N14 IOSTANDA</pre>	<pre>RD_LVCMOS33 } [get_ports { clk }]; #IO_L13P_T2_MRCC_34 Sch=sysclk</pre>
create_clock -add -name sve_clk_pin -period 20.0	0 waveform {0 10} [get_ports clk]
2180	周期
2 Thek	19773

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets PAD_JTAG_TCLK] 这一句必须加上

2) LED:

注意这三列, 对照开发板资料给的引脚改

## LEDs						
<pre>set_property -dict { PACKAG</pre>	_PIN M16	IOSTANDAR	LVCMOS33	<pre>} [get_por</pre>	ts { POUT_EHS }];	<pre>#IO_L15P_T2_DQS_13 Sch=led[0]</pre>
<pre>set_property -dict { PACKAG</pre>	_PIN N16	IOSTANDAR) LVCMOS33	} [get_por	ts { PAD_GPIO_0 }]	#IO_L15N_T2_DQS_13 Sch=led[1]
<pre>set_property -dict { PACKAG</pre>	_PIN P15	IOSTANDAR	LVCMOS33	} [get_por	ts { PAD_GPIO_1 }]	#IO_L17P_T2_13 Sch=led[2]
<pre>set_property -dict { PACKAG</pre>	_PIN P16	IOSTANDAR	LVCMOS33	<pre>} [get_por</pre>	ts { PAD_GPIO_2 }]	#IO_L17N_T2_13 Sch=led[3]
<pre>set_property -dict { PACKAG</pre>	_PIN M2	IOSTANDARD	LVCMOS33)	[get_port	s { PAD_USI0_NSS }	; #IO_L14N_T2_SRCC_13 Sch=led[4]
<pre>set_property -dict { PACKAG</pre>	_PIN P5	IOSTANDARD	LVCMOS33 }	[get_port	s { PAD_USI0_SD1 }	; #IO_L16N_T2_13 Sch=led[5]
<pre>set_property -dict { PACKAG</pre>	_PIN K12	IOSTANDAR	LVCMOS33	} [get_por	ts { PAD_PWM_CH9 }	; #IO_L16P_T2_13 Sch=led[6]
<pre>set_property -dict { PACKAG</pre>	_PIN H10	IOSTANDAR	LVCMOS33	} [get_por	ts { PAD_USI2_NSS]; #IO_L5P_T0_13 Sch=led[7]
## Buttons						
<pre>set_property -dict { PACKAG</pre>	_PIN R15	OSTANDARD	.VCMOS33 }	[get_ports	{ PAD_GPIO_8 }];	‡IO_L20N_T3_16 Sch=btnc
<pre>set_property -dict { PACKAG</pre>	_PIN R16	OSTANDARD	.VCMOS33 }	[get_ports	{ PAD_GPI0_9 }];	#IO_L22N_T3_16 Sch=btnd
<pre>set_property -dict { PACKAG</pre>	_PIN T14	OSTANDARD	.VCMOS33 }	[get_ports	{ PAD_GPI0_10 }];	#IO_L20P_T3_16 Sch=btnl
<pre>set_property -dict { PACKAG</pre>	_PIN M15	OSTANDARD	.VCMOS33 }	[get_ports	<pre>{ PAD_GPI0_11 }];</pre>	#IO_L6P_T0_16 Sch=btnr
<pre>#set_property -dict { PACKAG</pre>	E_PIN F15	IOSTANDARD	LVCMOS33 }	[get_port	s { PAD_GPI0_12 }]	#IO_0_16 Sch=btnu
<pre>set_property -dict { PACKAG</pre>	_PIN L13	IOSTANDARD	LVCMOS33 }	[get_port	s { PAD_MCURST }];	#IO_L12N_T1_MRCC_35 Sch=cpu_resetr

3) 复位:

set_property PACKAGE_PIN_L13 [get_ports PAD_MCURST] set_property IOSTANDARD LVCMOS33 [get_ports PAD_MCURST]

复位引脚

4) UART JTAG:

set_property PACKAGE_PIN_T8 [get_ports PAD_US 0_SD0] set_property IOSTANDARD LVCMOS33 [get_ports PAD_USI0_SD0] set_property PACKAGE_PIN_T7 [get_ports PAD_USI0_SCLK] set_property IOSTANDARD LVCMOS33 [get_ports PAD_USI0_SCLK]

串口

5) JTAG:

set_property PACKAGE_PII F12 [get_ports PAD_JTAG_TCLK] set_property IOSTANDARD LVCMOS33 [get_ports PAD_JTAG_TCLK] set_property PACKAGE_PII F13 [get_ports PAD_JTAG_TMS] set_property IOSTANDARD LVCMOS33 [get_ports PAD_JTAG_TMS]

CLINK连接时的引脚设置

6) SD 卡:

注意这两列, 对照开发板资料给的引脚改

#SD card					
set_property PACKAGE	_PIN L4 [ge	_ports {	n=sd_cclk s	clk	PAD_GPIO_6
set_property PACKAGE	_PIN T2 [ge	_ports {	ch=sd_cd	ard det	ect
set_property PACKAGE	_PIN R1 [ge	:_ports {	n=sd_cmd	mosi	PAD_GPIO_5
set_property PACKAGE	_PIN H16 [g	et_ports {	ch=sd_d[0]	miso	PAD_GPIO_7
set_property PACKAGE	_PIN R2 [ge	:_ports {	ch=sd_d[3]	cs	PAD_GPIO_19
# set_property PACKA	E_PIN F12	get_ports {	#Sch=sd_reset	rst	PAD_GPIO_20

3.其他

Switches 20 30 31
and points () dist (DYKAGE PIN T15 IOSTANDARD LVCMOS33) [get norts (PAD PWM (H10)]) #IO L22P T3 16 Sch=sw[0]
set property -dict { PACKAGE PIN 114 TOSTANDARD LYCMOS33 } [gct ports { PAD PMM (H11)]; #TO 25 16 Sch sw[1]
set property -dict { PACKAGE PIN MAL TOSTANDARD LVCMOS33 } [get_ports { PAD PWM CHA }]; #TO 124P T3 16 Schesw[2]
set property -dict { PACKAGE PIN K13 TOSTANDARD LYCMOS33 } [get ports { PAD PUM (H1 }]; #TO 12/N T3 16 Sch=su[3]
set property dict / DACKAGE DIN H17 TOSTANDARD LYCHOS33 \ [ad norts / DAD WM (H) \]: #TO 16D H15 Sch-su[A]
#set property dict { DACKAGE DIN 116 TOSTANDARD EVENDS33 } [get norts { DAD GIT 31 }]; #IO 0 15 ch=su[4]
set property -dict { PACKAGE PTN K13 TOSTANDARD LVCMOS33 } [get norts { PAD RD 30 }]; #TO 110P T3 A22 TS Schesw[6]
set property dict / DACKAGE DIN M17 TOSTANDARD LUCHOS33 \ [ad norts / DAC DIO 20 \]; #10.25 15 Schew[7]
#3cc_broker(3, area (1, eccade_1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
OLED Display
PACKAGE PIN W22 _ TOSTANDARD LVCMOS33 } [pet ports { PAD GPT0 21 }]: #TO L7N T1 D10 14 Sch=oled dc
set property -dict { PACKAGE PIN U21 IOSTANDARD LVCMOS33 } [pet ports { PAD GPIO 22 }]; #IO L4N TO D05 14 Sch=oled res
set property -dict { PACKAGE PIN W21 IOSTANDARD LVCMOS33 } [pet ports { PAD GPIO 23 }]; #IO L7P T1 D09 14 Sch=oled sclk
set property -dict { PACKAGE PIN Y22 IOSTANDARD LVCMOS33 } [get ports { PAD GPIO 24 }]; #IO L9N T1 DOS D13 14 Sch=oled sdin
set property -dict { PACKAGE PIN P20 IOSTANDARD LVCMOS33 } [ret ports { PAD GPIO 25 }]; #IO 0 14 Sch=oled vbat
set property -dict { PACKAGE PIN V22 IOSTANDARD LVCMOS33 } [pet ports { PAD GPIO 26 }]; #IO L3N T0 DOS EMCLK 14 Sch=oled vdd
HDMI in
#set property -dict { PACKAGE PIN AA5 IOSTANDARD LVCMOS33 } [get ports { hdmi rx cec }]; #IO L10P T1 34 Sch=hdmi rx cec
<pre>#set property -dict { PACKAGE PIN W4 IOSTANDARD TMDS 33 } [get ports { hdmi rx clk n }]; #IO L12N T1 MRCC 34 Sch=hdmi rx clk n</pre>
<pre>#set_property -dict { PACKAGE_PIN V4 IOSTANDARD TMDS_33 } [get_ports { hdmi_rx_clk_p }]; #IO_L12P_T1_MRCC_34 Sch=hdmi_rx_clk_p</pre>
<pre>#set_property -dict { PACKAGE_PIN_AB12_IOSTANDARD_LVCMOS33 } [get_ports { hdmi_rx_hpa }]; #IO_L7N_T1_13 Sch=hdmi_rx_hpa</pre>
<pre>#set_property -dict { PACKAGE_PIN Y4 IOSTANDARD LVCMOS33 } [get_ports { hdmi_rx_scl }]; #IO_L11P_T1_SRCC_34 Sch=hdmi_rx_scl</pre>
<pre>#set_property -dict { PACKAGE_PIN AB5 IOSTANDARD LVCMOS33 } [get_ports { hdmi_rx_sda }]; #IO_L10N_T1_34 Sch=hdmi_rx_sda</pre>
<pre>#set_property -dict { PACKAGE_PIN R3 IOSTANDARD LVCMOS33 } [get_ports { hdmi_rx_txen }]; #IO_L3P_T0_DQS_34 Sch=hdmi_rx_txen</pre>
Hard and the follower official to the total following to the following to the total of

这些根据自己板子的资源多少和想实现的功能改,比如只想实现点灯,那么这些就可以全注释掉,但如果要实现 OLED,那先看自己板子有没有 OLED 的配置,然后根据开发板手册改 xdc 里相对应的引脚;

以下为移植到 A7 50T 的 xdc 代码:

set_property IOSTANDARD LVCMOS33 [get_ports]

#clock

set_property -dict {PACKAGE_PIN N14 IOSTANDARD LVCMOS33} [get_ports clk]
create_clock -period 20.000 -name sys_clk_pin -waveform {0.000 10.000} -add [get_ports clk]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets PAD_JTAG_TCLK]

#LED

set_property -dict {PACKAGE_PIN N16 IOSTANDARD LVCMOS33} [get_ports PAD_GPIO_0]

set_property -dict {PACKAGE_PIN P15 IOSTANDARD LVCMOS33} [get_ports PAD_GPIO_1] set_property -dict {PACKAGE_PIN P16 IOSTANDARD LVCMOS33} [get_ports PAD_GPIO_2]

#RESET button

set_property PACKAGE_PIN L13 [get_ports PAD_MCURST]

#USI0

set_property PACKAGE_PIN T7 [get_ports PAD_USI0_SCLK]
set_property PACKAGE_PIN T8 [get_ports PAD_USI0_SD0]
#JTAG
set_property PACKAGE_PIN F12 [get_ports PAD_JTAG_TCLK]
set_property PACKAGE_PIN F13 [get_ports PAD_JTAG_TMS]
#SD card
set_property PACKAGE_PIN L4 [get_ports { PAD_GPIO_6 }]; #Sch=sd_cclk sclk
PAD_GPIO_6
set_property PACKAGE_PIN T2 [get_ports { PAD_GPIO_18 }]; #Sch=sd_cd card detect
set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi
<pre>set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5</pre>
set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmdmosiPAD_GPIO_5set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0]miso
set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5 set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0] miso PAD_GPIO_7
set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmdmosiPAD_GPIO_5set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0]misoPAD_GPIO_7set_property PACKAGE_PIN R2 [get_ports { PAD_GPIO_19 }]; #Sch=sd_d[3]cs
set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5 set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0] miso PAD_GPIO_7 set_property PACKAGE_PIN R2 [get_ports { PAD_GPIO_19 }]; #Sch=sd_d[3] cs PAD_GPIO_19
<pre>set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5 set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0] miso PAD_GPIO_7 set_property PACKAGE_PIN R2 [get_ports { PAD_GPIO_19 }]; #Sch=sd_d[3] cs PAD_GPIO_19 # set_property PACKAGE_PIN F12 [get_ports { PAD_GPIO_20 }]; #Sch=sd_reset rst</pre>
<pre>set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5 set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0] miso PAD_GPIO_7 set_property PACKAGE_PIN R2 [get_ports { PAD_GPIO_19 }]; #Sch=sd_d[3] cs PAD_GPIO_19 # set_property PACKAGE_PIN F12 [get_ports { PAD_GPIO_20 }]; #Sch=sd_reset rst PAD_GPIO_20</pre>
<pre>set_property PACKAGE_PIN R1 [get_ports { PAD_GPIO_5 }]; #Sch=sd_cmd mosi PAD_GPIO_5 set_property PACKAGE_PIN H16 [get_ports { PAD_GPIO_7 }]; #Sch=sd_d[0] miso PAD_GPIO_7 set_property PACKAGE_PIN R2 [get_ports { PAD_GPIO_19 }]; #Sch=sd_d[3] cs PAD_GPIO_19 # set_property PACKAGE_PIN F12 [get_ports { PAD_GPIO_20 }]; #Sch=sd_reset rst PAD_GPIO_20</pre>