Zynq 7000裸机的lwip 样例程序echo server 实验



本文是在Alinx黑金Zynq 7000开发平台配套教程的学习过程中产生的。硬件平台是黑金的AX 7010/ax7020,核心板AC7010/AC7020,其他类似 平台应该也可以。

预备知识是你先学会了helloWorld实验。如果你还没有学会helloworld实验,那你看看我的另一篇文章helloworld 实验,或者你学习学习板提供的helloworld实验,那应该更接近你的实际硬件。

工程文件下载处: http://pan.baidu.com/s/1pKEqsS7

如果想学习怎么pc端编程,可以看我的另一个文章: MFC 下的网络编程 socket编程

1: 硬件设计

Vivado工程创建:与helloworld 工程的创建一样,只是工程名为lwip.

Zynq 的配置: 与helloworld 工程的配置基本一样,唯一不同的是需要选择Ethernet。

下面的配置是针对黑金的学习板AX7010/AX7020或者核心板AC7010/AC7020,如果你的硬件网络口与此不一样,那你就要查看并做相应改变。

串口的配置是一样的:

Page Ilevigator Peripheral I/O Pins Summary Report SP-FL Configuration 9 11 32 33 34 35 36 37 38 39 40 11 42 43 44 45 46 47 48 49 50 11 52 53 EMO NO Peripheral I/O Pins 9 11 32 33 34 35 36 37 38 39 40 11 42 43 44 45 46 47 48 49 50 11 52 53 EMO EMO N2C Configuration 9 Ethernet 1 Image: Configuration Im	IV Documentation 🌇 Prese	ets 🦲 IP Location 🧤 I	npor	rt XP	S Se	tting	s																					
Zyng Block Design Search: Q- Bord Configuration 9 31 32 33 34 35 36 37 36 39 40 41 42 43 44 45 46 47 46 49 50 51 52 55 EMO Configuration B Ethernet 0 B Ethernet 1 Enet1 I I I I I I I I I I I I I I I I I I I	Page Navigator 🛛 «	Peripheral I/O Pins																							S	unmaz	ry Repor	<u>rt</u>
PS-PL Configuration Peripheral T/O Pins WID Configuration Clock Configuration DSC Configuration DSS D DSS D DSS D DSD Configuration DSS D DSS D DSD SDO DSD SPH DSD SPH DARTO DARTO </td <td>Zynq Block Design</td> <td>🗲 Search: 🔍</td> <td></td>	Zynq Block Design	🗲 Search: 🔍																										
Peripheral I/O Pins Peripherals 0 31 32 33 34 35 36 37 38 39 40 41 42 44 45 46 47 45 45 50 51 52 53 EMO Peripherals Periphera	PS-PL Configuration	<mark>♀</mark>																										
MID Configuration E Hthernet 0 Enet1 Enet1 EMD Clock Configuration WSB 0 USB 1 USB 0 USB 1 EMD SMC Tissing Calculation WSB 0 SD0 SD0 SD0 SD0 SD0 SD0 SD0 SD0 SD0 EMD Enterrupts SD0 SD0 SD1 SD1 SD1 EMD EMD Interrupts SD1 SD1 SD1 SD1 EMD EMD Image: SPI 0 mos SPI 0 SPI 0 SPI 0 EMD EMD Image: SPI 1 ART0 UART0 UART0 UART0 UART0 EMD Image: SPI 1 ART0 UART1 UART1 UART1 UART1 UART1 UART1 UART1 EMD Image: SPI 1 ART0 UART1 UART1 UART1 UART1 UART1 UART1 EMD Image: SPI 1 ART0 UART1 UART1 UART1 UART1 UART1 UART1 EMD Image: SPI 1 ART0 UART1 UART1 UART1 UART1<	Peripheral I/O Pins	Peripherals	0	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	EMIO	
Clock Configuration Ethernet 1 Enet1 USB 0 USD 0 U	MIO Configuration	🚺 🕂 🔲 Ethernet O	ŀ																								EMIO	Î
DDR Configuration DDR Configuration SMC lising Calculation Interrupts	Clock Configuration	🗄 📃 Ethernet 1				En	et1																				EMIO	
DDR Configuration SMC Lining Calculation Interrupts Interupts	CIOCK CONTIGM ACTON		1			US	6B0																					
SMC Timing Calculation Image: SD 0 SD 0 SD 0 SD 0 SD 0 SD 0 EMI0 Interrupts SD 0 SD 0 SD 0 SD 0 SD 0 SD 0 EMI0 Image: SD 0 EMI0 Image: SD 0 SD 0 SD 0 SD 0 SD 0 SD 0 EMI0 Image: SPI 0 mos SPI0 Mos SPI0 SPI1 EMI0 Image: SPI 1 SPI1 SPI1 SPI0 SPI1 EMI0 Image: SPI 1 JART0 UART0 UART0 UART0 UART0 UART1 UART1 UART1 UART1 UART1 EMI0 Image: WART 1 UART1 UART1 UART1 UART1 UART1 UART1 UART1 EMI0 Image: SIZE 0 IZC0 IZC0 IZC0 IZC0 IZC0 IZC0 IZC1 IZC1 IZC1 IZC1 EMI0 Image: SIZE 0 IZC0 IZC0 IZC1 IZC1 IZC1 IZC1 IZC1 IZC1 EMI0	DDR Configuration		Т															US	SB1									
Interrupts Image: SD 0	SMC Timing Calculation	056 1		300		-								_	q	00	_	_							-		EMIO	
B SD 1 SD 1 SD 1 EMIO B SFI 0 mos SFI 0 mos SFI 0 EMIO B SFI 1 SFI 1 SFI 0 UART0 UART0 UART0 UART0 UART0 UART0 UART0 EMIO CART 0 JART0 UART1 EMIO VART 1 UART1 UART1 UART1 UART1 UART1 UART1 UART1 UART1 EMIO ICC0	Interrupts		ŀ										_												-		Enno	
B SPI 0 mos SPI0 mos SPI1 SPI0 mos SPI1		🗄 📃 SD 1	:	h	ttr	:/	10.00	U.S.	SE	01		107	lec	'n	eer	g0					s	D1					EMIO	Ε
Image: SPI 1 Image: SPI 1 <td< td=""><td></td><td>⊕. [] SPI 0</td><td></td><td></td><td>1</td><td>mos</td><td></td><td></td><td></td><td></td><td></td><td></td><td>S</td><td>P10</td><td></td><td>-</td><td></td><td>mos</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>EMIO</td><td></td></td<>		⊕. [] SPI 0			1	mos							S	P10		-		mos									EMIO	
Image: Start of the start		🕀 🔲 SPI 1	Ŀ					SPI1												SPI1							EMIO	
UARTI EMIO Modem Sign Modem Sign I2C0 EMIO ICC0 I2C1 I2C1 I2C1 I2C1 I2C1 I2C1 I2C1 I2C1 EMIO ICANO CANO CANO CANO CANO CANO CANO CANO CANO EMIO ICANO CANO CANO CANO CANO CANO CANO CANO CANO EMIO		⊕ 🛄 VARI 0	JA	ARTO			UAF	RT0			UA	RT0			UA	RTO			UA	RTO			UA	RT0			EMIO	
Image: Modem Sign Im		- VARI 1			UA	RT1			UAF	RT1			UA	RT1			UA	RT1			UA	RT1			UA	RT1	EMIO	
Image: International conditional condite conditional conditional conditional conditional conditio		Modem Sig	" "																								EMIO	
Image: International control of the second contro			В	2C0			120	C0			12	C0			12	C0			12	CO			12	C0			EMIO	
CANO CANO CANO CANO CANO CANO CANO CANO		± 🔲 I2C 1			12	2C1			120	D1			12	C1			12	C1			12	C1			12	C1	EMIO	
<pre></pre>		E CAN O	с	AN0			CA	NO			C/	ANO			C/	N0			C/	N0			CA	AN0			EMIO	Ŧ
0K Cancel			1	(_										•	
																								0K	_		Cancel	-

选择Ethernet 0和MDIO,Ethernet0癿管脚为MIO16~MIO27, MDIO癿管脚为MIO52,MIO53。 Peripheral I/O Pins

+	Search: Q-																					
Q 							Ba	nk 1	LV	/CMOS	3.3	V	•									
	Peripherals		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Û	🕂 🔲 Quad SPI Flas	L																				
	. SRAM/NOR Flas						SRAM/NOR Flash, :										lash, addr[0-24]					
	🛨 🔲 NAND Flash			ht	tp	://	b1	og.	СS	dn.	ne	t/1	eo	n_2	en	g0						
	🗐 🗹 🗹 Ethernet O											En	et0									
	📝 MDIO																					
	🖽 🔲 Ethernet 1																					



选择Ethernet 0的接口信号管脚的速度都为"fast",电压为LVCMOS1.8V。参见黑金学习板教程1的P287

Page Navigator 🛛 «	MIO Configuration					Summary	Report
Zynq Block Design	🗲 Bank O I/O Voltage LVCMOS 3.3V	▼ Ba	nk 1 I/O Volta	age LVCMOS 1.8V 💌			
PS-PL Configuration	Search: Q.						
Peripheral I/O Pins	😝 Peripheral	10	Signal	IO Type	Speed	Pullup	Dire
MIO Configuration	📑 🕂 Memory Interfaces						
	🔓 🗐 I/O Peripherals						
Clock Configuration	ENET O	MIO 16 27 🛛 💌					
DDR Configuration	HIO MDIO	MIO 52 53 💌			\frown		
	··· Enet O	MI0 52	mdc	LVCMOS 1.8V -	fast 🗟	enabled 🔻	out
SMC Timing Calculation	Enet O	MI0 53	mdio	LVCMOS 1.8V	fast 🔻	enabled 🔻	inou =
Interrupts	···· Enet O	MIO 16	tx_clk	LVCMOS 1.8V	fast 🔻	enabled 🔻	out
	··· Enet O	MIO 17	txd[0]	LVCMOS 1.8V	fast 🔻	enabled 🔻	out
	- Enet 0 NUUP:	MI0 ¹⁸ og. csan. net	txd[1]	IVCMOS 1.8V -	fast 🔻	enabled 🔻	out
	··· Enet O	MIO 19	txd[2]	LVCMOS 1.8V -	fast 🔻	enabled 🔻	out
	··· Enet O	MI0 20	txd[3]	LVCMOS 1.8V -	fast 🔻	enabled 🔻	out
	··· Enet O	MIO 21	tx_ctl	LVCMOS 1.8V -	fast 🔻	enabled 🔻	out
	··· Enet O	MI0 22	rx_clk	LVCMOS 1.8V -	fast 🔻	enabled 🔻	in
	··· Enet O	MI0 23	rxd[0]	LVCMOS 1.8V -	fast 🔻	enabled 🔻	in
	··· Enet O	MI0 24	rxd[1]	LVCMOS 1.8V	fast 🔻	enabled 🔻	in
	··· Enet O	MIO 25	rxd[2]	LVCMOS 1.8V	fast 🔻	enabled 🔻	in
	··· Enet O	MIO 26	rxd[3]	LVCMOS 1.8V	fast 🔻	enabled 🔻	in
	···· Enet O	MIO 27	rx_ctl	LVCMOS 1.8V -	fast y	enabled 💌	in
					\bigtriangledown		
							Ψ •

这就是所有的硬件设计要点,与helloworld一样,选择菜单File->Export->Export Hardware,你就完成硬件导入SDK。

我第一次操作的时候,没有完全照helloworld 实验做,结果中间报错了,注意教程1的P92写了:

因为本实验中没有用到PL部分外设,所以就没有PL的外设需要挂到AXI的总线上,也就不需要AXI GP0的Master总线了,这里我们把 FCLK_RESET0_N和M AXI GP0 interface的选项去掉。

 软件设计 我开始做实验时,也许测试方法不对,结果老不通,黑金说: 黑金修改后的lwip库,做lwip实验时可以替换掉,解决无法识别连接速度的问题。 怎么替换呢?

就是用他提供的lwip141_v1_3\src

C:\Xilinx\SDK\2015.4\data\embeddedsw\ThirdParty\sw_services\lwip141_v1_3\src 如果你没有这个,也可以用我提供的项目里的lwip\lwip.sdk\lwip_test_bsp\ps7_cortexa9_0\libsrc\lwip141_v1_3\src

做了这个替换后,Launch SDK,就进入软件设计界面。 和helloworld 实验一样,File ->New -> Application Project

这里我取的工程名是lwip_test

重要的是下面的Next, 在Next界面里,选择的不是Hello World, 而是 IMP Echo Server

SDK	New Project										
Т((Templates Create one of the available templates to generate a fully-functioning application project.										
A	wailable Templates:										
	Dhrystone Empty Application Hello World <u>WIP Echo Server</u> Memory Tests OpenAMP echo-test OpenAMP matrix multiplication Demo OpenAMP RPC Demo Peripheral Tests RSA Authentication App Zynq DRAM tests Zynq FSBL http://blog.cs	The IwIP Echo Server application provides a simple demonstration of how to use the light-weight IP stack (IwIP). This application sets up the board to use IP address 192.168.1.10, with MAC address 00:0a:35:00:01:02. The server listens for input at port 7 and simply echoes back whatever data is sent to that port.									
	Reack New	xt > Finish Cancel									

像hello World 实验一样,你不用做任何修改,软件设计就完成了。 Build Project 然后 Debug As->Launch on hardware 或者 Run As-> 就进入了测试阶段

3:测试

其实硬件设计,软件设计都是很简单,但我在测试上花了好多天时间。因为测试不出结果,所以自然会认为软硬件设计哪里有问题。不断地去 修改,查找软硬件的问题。

和hello world 实验一样,程序运行就会在终端上显示很多信息:

G Serial-COM3 - SecureCRT	23
File Edit View Options Transfer Script Tools Help	
41 XI L7 41 XI == (= Q F2 F3 A 12 X 1 ? 2	
Serial-COM3	×
lwIP TCP echo server TCP packets sent to port 6001 will be echoed back WARNING: Not a Marvell or TI Ethernet PHY. Please verify the initialization se ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 3:/100 og. csdn. net/leon_zeng0 DHCP Timeout Configuring default IP of 192.168.1.10 Board IP: 192.168.1.10 Netmask : 255.255.0 Gateway : 192.168.1.1 TCP echo server started @ port 7	2qu E
Ready Serial: COM3 17, 1 24 Rows, 80 Cols VT100 NU	JM a

我的程序第一次运行就是这个结果,也许第一次的时候,软硬件设计就是对的。但是本实验的核心不是显示这些简单信息,而是要echo,要看到回声。

很自然,我开始就用黑金提供的网络助手做连接测试,可总是连接不上。这个不成功的问题太多,网线,网卡设置,软硬件,测试软件。当然 网卡需要设置ip4 为静态网址,我用的是直连线,我当时认为比较简单,不需要hub。

我估计用hub ,而不是直连线可能容易成功一点。程序里如果能够自动获取,那么zynq 7000会优先选用自动获取的ip地址,当然会显示处理。 这样你的本机就容易和他在一个网络里。

设置好后,我用ping 测试是成功的,但就是不能连接上。

我不再相信网络助手,在网上找了tcp socket 通讯的例子做测试,有个例子只是server,要求我用telnet 连接做测试。 终于我用黑金提供的终端软件用telnet 可以连接上zynq 7000了。其连接测试是这样的: Quick Connect

Protocol: Hostname:	Telnet
Port	7 Firewall None -
http	://blog.csdn.net/leon_zeng0
🔲 Show quick	connect on star 🔽 Save session
_	Connect Cancel

连接上之后,我就看到回声了,压抑我十多天的痛苦终于过去了。

Table 192.168.1.10 - SecureCR	r	
File Edit View Options	Transfer Script Tools Help	
🖏 🖏 🗔 🖏 🖄 🖻 I	ª Q G 등 ⊕ £ XX ↑ १ 2	
192.168.1.10		×
wert wert www.liwensoft.com zeng liwen zeng liwen 曾立文 曾立文 1234567 1234567 黑金 www.heijin.org 黑金 www.heijin.org	http://blog.csdn.net/leon_zeng0	*
		E
<u> </u>		-
Ready	Telnet 13, 1 24 Rows, 80 Cols VT100	NUM

```
4: 简单的代码解释
```

```
这个解说大部分来自xilinx xapp1026.pdf
这里主要有2个程序文件main.c echo.c
```

```
main.c 的主循环程序是:
```

```
/* receive and process packets */
while (1) {
    if (TcpFastTmrFlag) {
        tcp_fasttmr();
        TcpFastTmrFlag = 0;
    }
    if (TcpSlowTmrFlag) {
        tcp_slowtmr();
        TcpSlowTmrFlag = 0;
    }
    xemacif_input(echo_netif);
    transfer_data();
}
```

TcpFastTmrFlag 和 TcpSlwTmrFlag 是TCP TX中断所必需的,每隔250ms 和 500ms 定时器中断里设定的。 应用循环的功能是不断接受包(xemacif_input) 然后传送给lwip.

```
在进入这个循环前,echo Server 设置必要的回调函数。
这些设置就在echo.c 里的start_application函数里。
recv_callback是数据接收中断程序。
accept_callback是连接建立接收中断程序。
start_application 解释如下:
/* create new TCP PCB structure 建立新的TCP 的PCB结构*/
pcb = tcp_new();
if (!pcb) {
    xil_printf("Error creating PCB. Out of Memory\n\r");
    return -1;
    }
    /* bind to specified @port 绑定在端口port */
    err = tcp_bind(pcb, IP_ADDR_ANY, port);
```

```
if (err != ERR_OK) {
```

```
xil_printf("Unable to bind to port %d: err = %d\n\r", port, err);
  return -2;
}
/* we do not need any arguments to callback functions */
tcp_arg(pcb, NULL);
```

```
/* listen for connections侦探连接*/
pcb = tcp_listen(pcb);
if (!pcb) {
  xil_printf("Out of memory while tcp_listen\n\r");
  return -3;
}
```

/* specify callback to use for incoming connections 设定输入连接中断函数 */ tcp_accept(pcb, accept_callback);

而接收数据的中断函数是在accept_callback里设定的。

/* set the receive callback for this connection */

tcp_recv(newpcb, recv_callback);

5: 对软件做点修改

是不是应该对软件做点什么修改,什么也不做,有点不好意思。

到底zynq7000收到数据了吗?我有种怀疑态度,所以我想让监控的终端上显示我输入的字符串,而不仅仅是echo.

```
看看,我做到了,你也试一下吧。
```

Serial-COM3 - SecureCRT	
File Edit View Options Transfer Script Tools Help	
x3 X3 C7 X3 X8 🖻 🛍 🔍 😼 👺 🎒 🗳 X 🕴 💡 📰	a 192.168.1.10 (4) - SecureCRT
Serial-COM3	File Edit View Options Transfer Script To
lwIP TCP echo server TCP packets sent to port 6001 will be echoed back WARNING: Not a Marvell or TI Ethernet PHY. Please verify th ence Start PHY autonegotiation Waiting for PHY to complete autonegotiation. autonegotiation complete link speed for phy address 3: 100 DHCP Timeout Configuring default IP of 192.168.1.10 Board IP: 192.168.1.10 Netmask : 255.255.255.0 Gateway : 192.168.1.1 TCP echo server started @ port 7 123456 zengliwen 曾立文	Solution Constraints of the second state of the second s

6: 后续

echo server 只是lwip应用的一个开始,看看xapp1026, lwip实验还有 Web Server **TFTP Server** TCP RX 速度测试 TCP TX 速度测试 当然我们最大的目的是实际应用