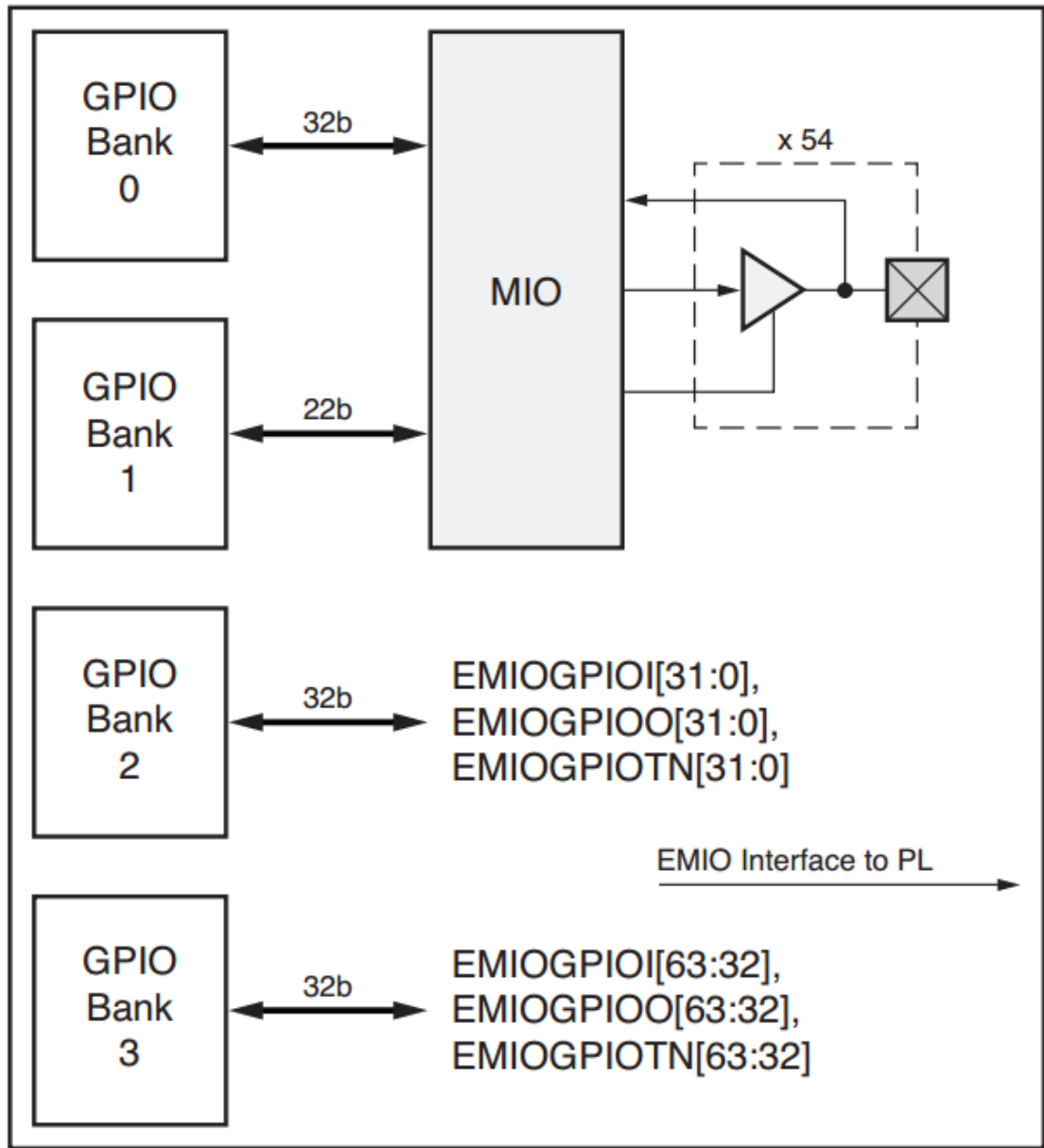


MYIR-ZYNQ7000系列-zturn教程(18)：基础教程gpio_mio做为输入口

原创 虚无缥缈vs威武 最后发布于2018-08-31 20:02:42 阅读数 773 ☆ 收藏

开发板环境：vivado 2017.4，开发板型号xc7z020clg400-1，这个工程主要介绍怎样将gpio_mo做为输入口使用

工程链接：<https://pan.baidu.com/s/1jewYR14IH0tsbvlMCIUswg> 密码：glpq



https://blog.csdn.net/UG585_c14_01_02221214520

从这个ug585手册可以看到gpio_mio主要分布在这个Bank0 和Bank1上。这里主要介绍gpio_mio暂时不介绍gpio_emio

这个Bank0主要是mio0~mio31

MIO_0
MIO_1
MIO_2
MIO_3
MIO_4
MIO_5
MIO_6
MIO_7
MIO_8
MIO_9
MIO_10
MIO_11
MIO_12
MIO_13
MIO_14
MIO_15
MIO_16
MIO_17
MIO_18
MIO_19
MIO_20
MIO_21
MIO_22
MIO_23
MIO_24
MIO_25
MIO_26
MIO_27
MIO_28
MIO_29
MIO_30
MIO_31

👍
2

🔗

💬

☆

📱

<

>

赏


展开


🔊


举报


Bank1主要是mio31~mio53


MIO_32
MIO_33
MIO_34
MIO_35
MIO_36
MIO_37
MIO_38
MIO_39
MIO_40
MIO_41
MIO_42
MIO_43
MIO_44
MIO_45
MIO_46
MIO_47
MIO_48
MIO_49
MIO_50
MIO_51
MIO_52
MIO_53


2

















这里继续介绍gpio_mio做为输入要涉及到的寄存器

Register Name	Address	Width	Type	Reset Value	Description
DIRM_0	0x00000204	32	rw	0x00000000	Direction mode (GPIO Bank0, MIO)

这个DIRAM_0主要设置gpio_mio的mio0~mio31的输入输出设置：设为0为输入、设置为1为输出

DIRM_1	0x00000244	22	rw	0x00000000	Direction mode (GPIO Bank1, MIO)
------------------------	------------	----	----	------------	----------------------------------

这个DIRAM_1主要设置gpio_mio的mio32~mio53的输入输出设置：设为0为输入、设置为1为输出

既然我们设置为输入，那么输入的数据必定有存储的寄存器，下面我来看下这些存储器

DATA_0_RO	0x00000060	32	ro	x	Input Data (GPIO Bank0, MIO)
---------------------------	------------	----	----	---	------------------------------

这个DATA_0_RO主要存储这个mio0~mio31的输入数据，这个是一个32位的寄存器，寄存器的偏移地址是0x00000060

DATA_1_RO	0x00000064	22	ro	x	Input Data (GPIO Bank1, MIO)
---------------------------	------------	----	----	---	------------------------------

这个DATA_1_RO主要存储这个mio32~mio53的输入数据，这个是一个22位的寄存器，寄存器的偏移地址是0x00000064

我们一般对一个寄存器进行写一般是：

Xil_Out32(基地址+偏移地址，数据)

比如对这个0x00000244进行写：Xil_Out32（0xE000A000 + 0x00000244,0x00）对这个0x00000244寄存器写入数据0x00,这里的基地址是0xE000A000

一般对一个寄存器进行读：

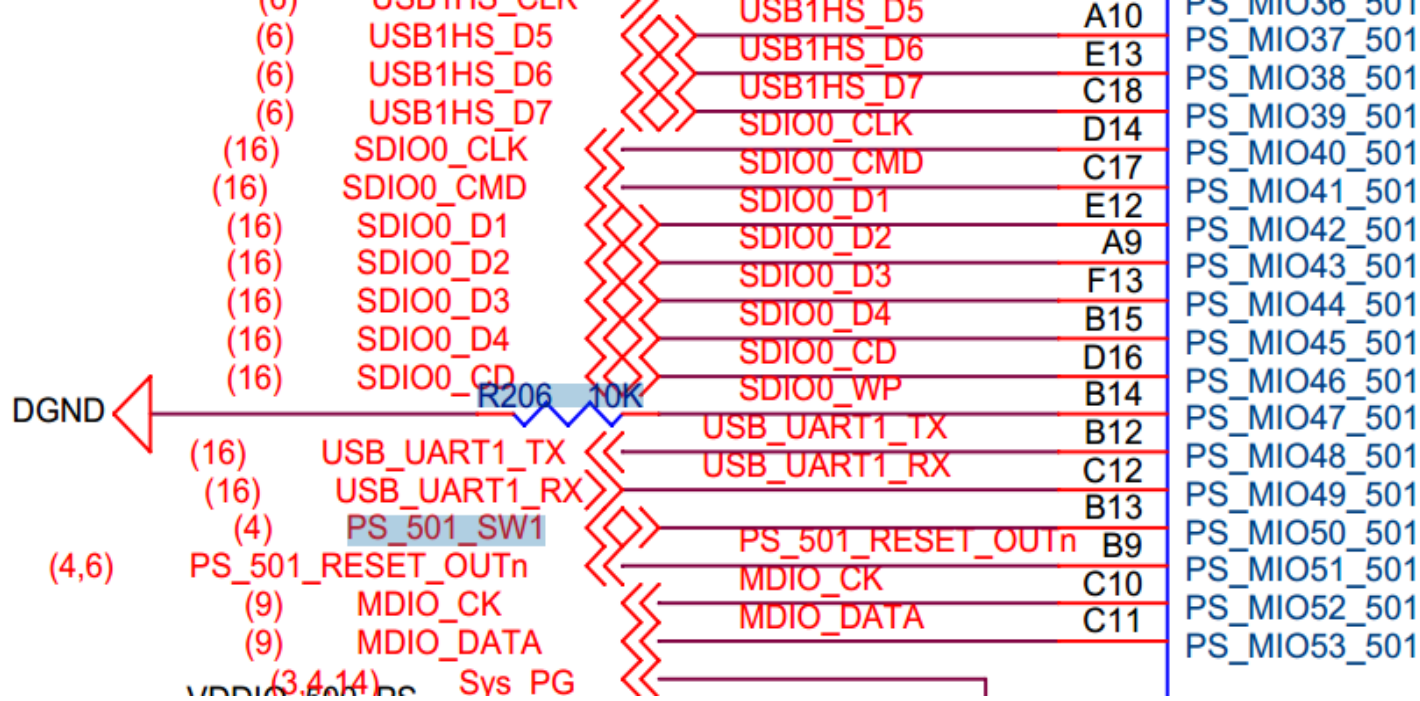
data = Xil_In32(基地址+偏移地址)

比如对这个0x00000064进行读：data = Xil_In32(0xE000A000 + 0x00000064)

硬件设置

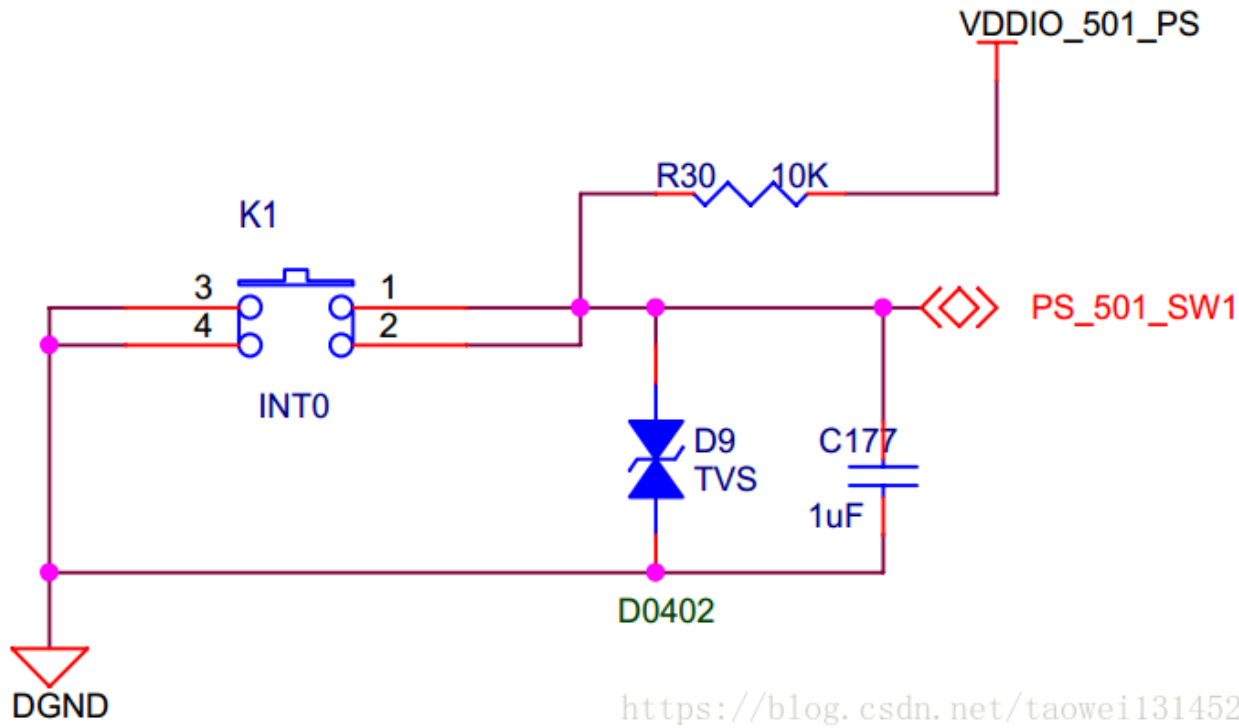


举报



<https://blog.csdn.net/taowei1314520>

因z-turn开发板的mio50上面接了一个开关，所以主要接收这个管脚的输入值



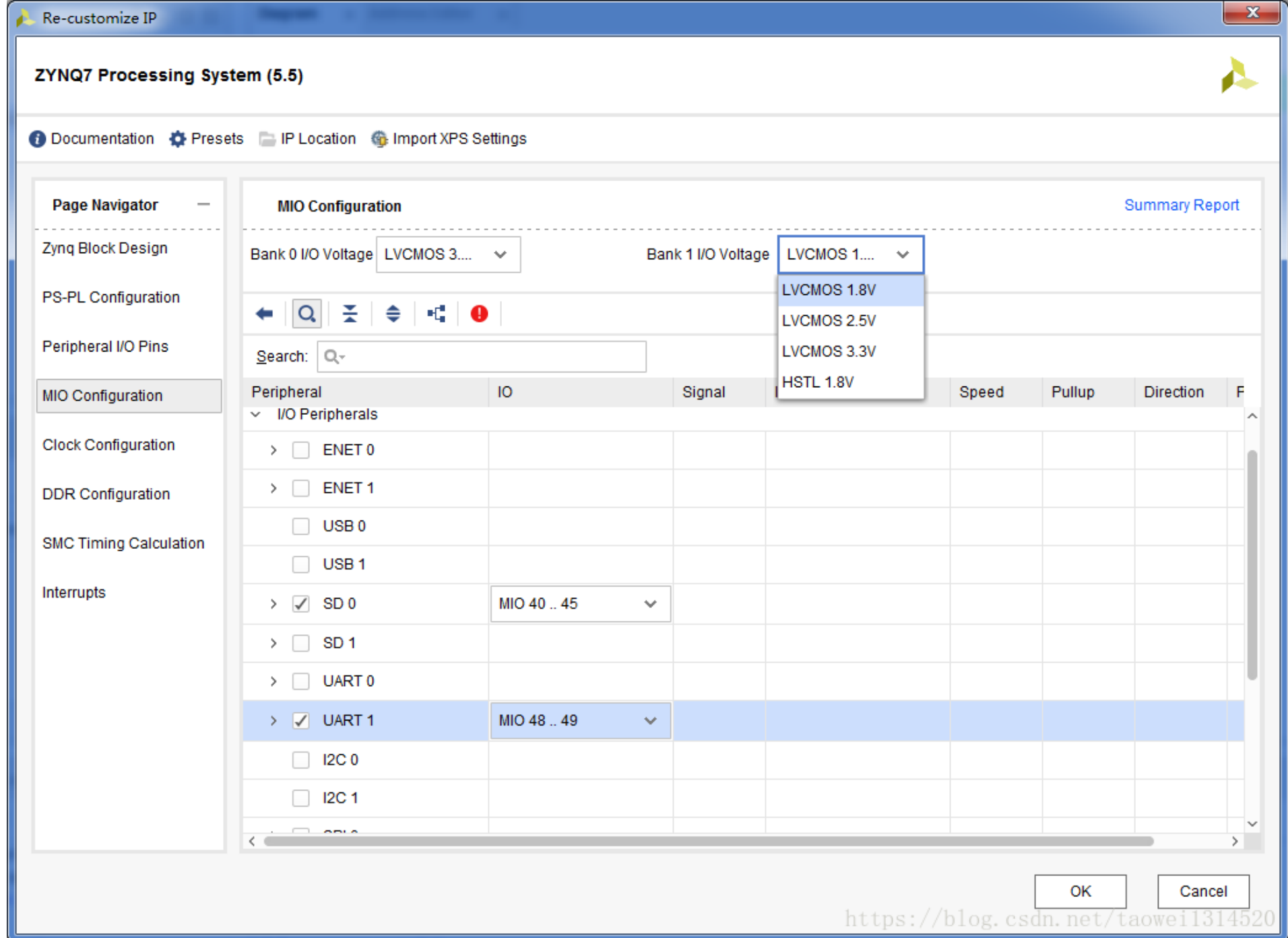
<https://blog.csdn.net/taowei1314520>

从这个图可以看出没有按开关输入为1，按下开关输入为0

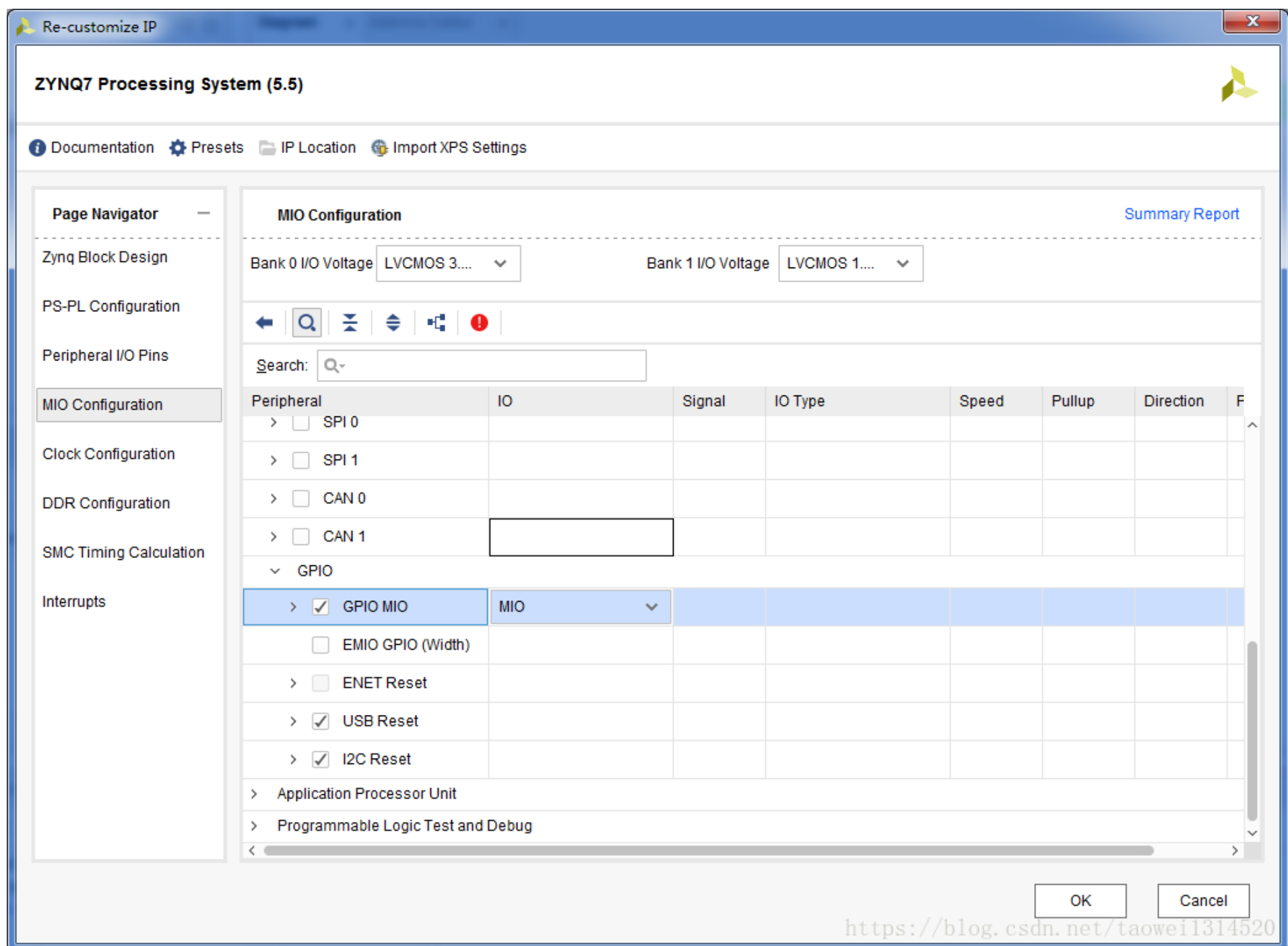
Step1 新建vivado 工程，我这里选择了SD卡和uart



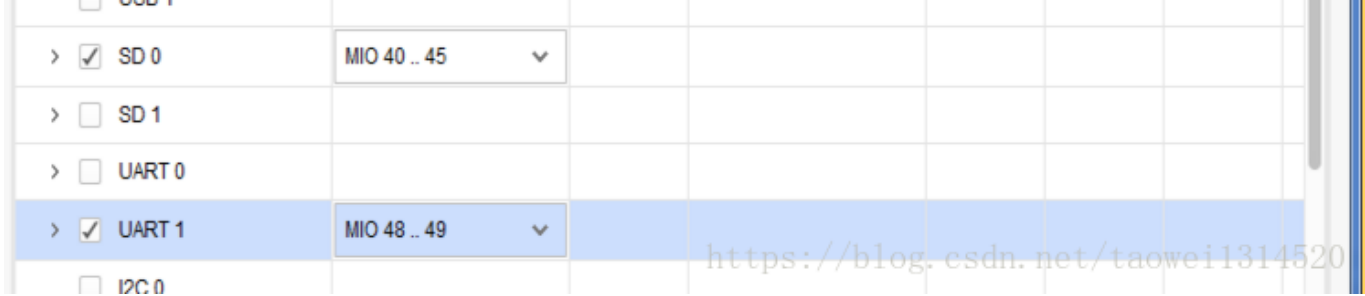
举报



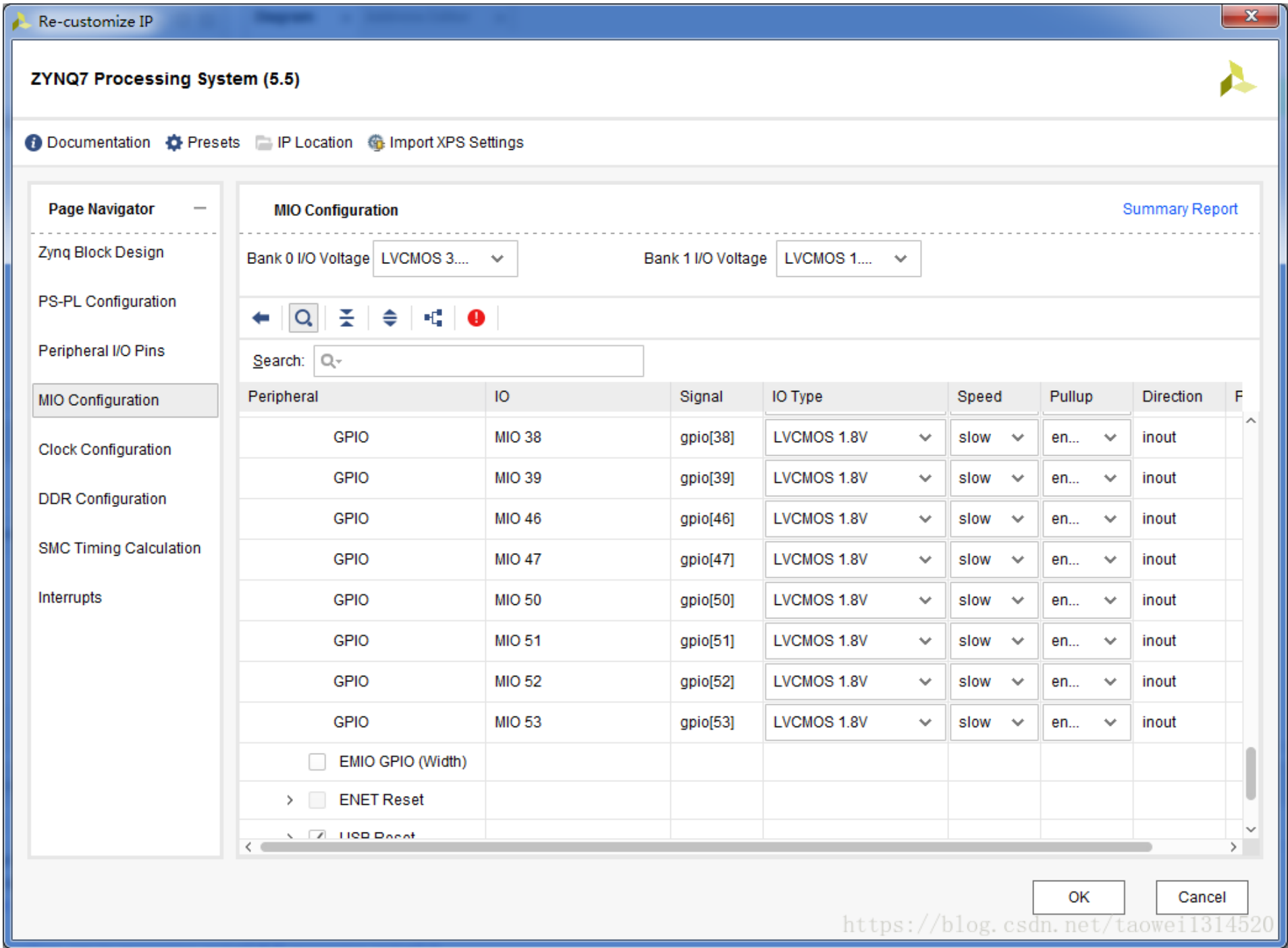
勾选gpio_mio




我们前面勾选了这个SD和uart使用了MIO40~MIO45、MIO48~MIO49





我们可以看到下面的gpio_mio中没有这个MIO40~MIO45和MIO48~MIO49，这个mio和gpio_mio是不能同时使用的，也就是说当你使用这个mio48那么你在gpio_mio中就无法再使用这个gpio_mio48这两个是互斥的，gpio_mio说得简单点就是将mio的管脚拿过来做为gpio_mio使用。





设置sd时钟为50M


2











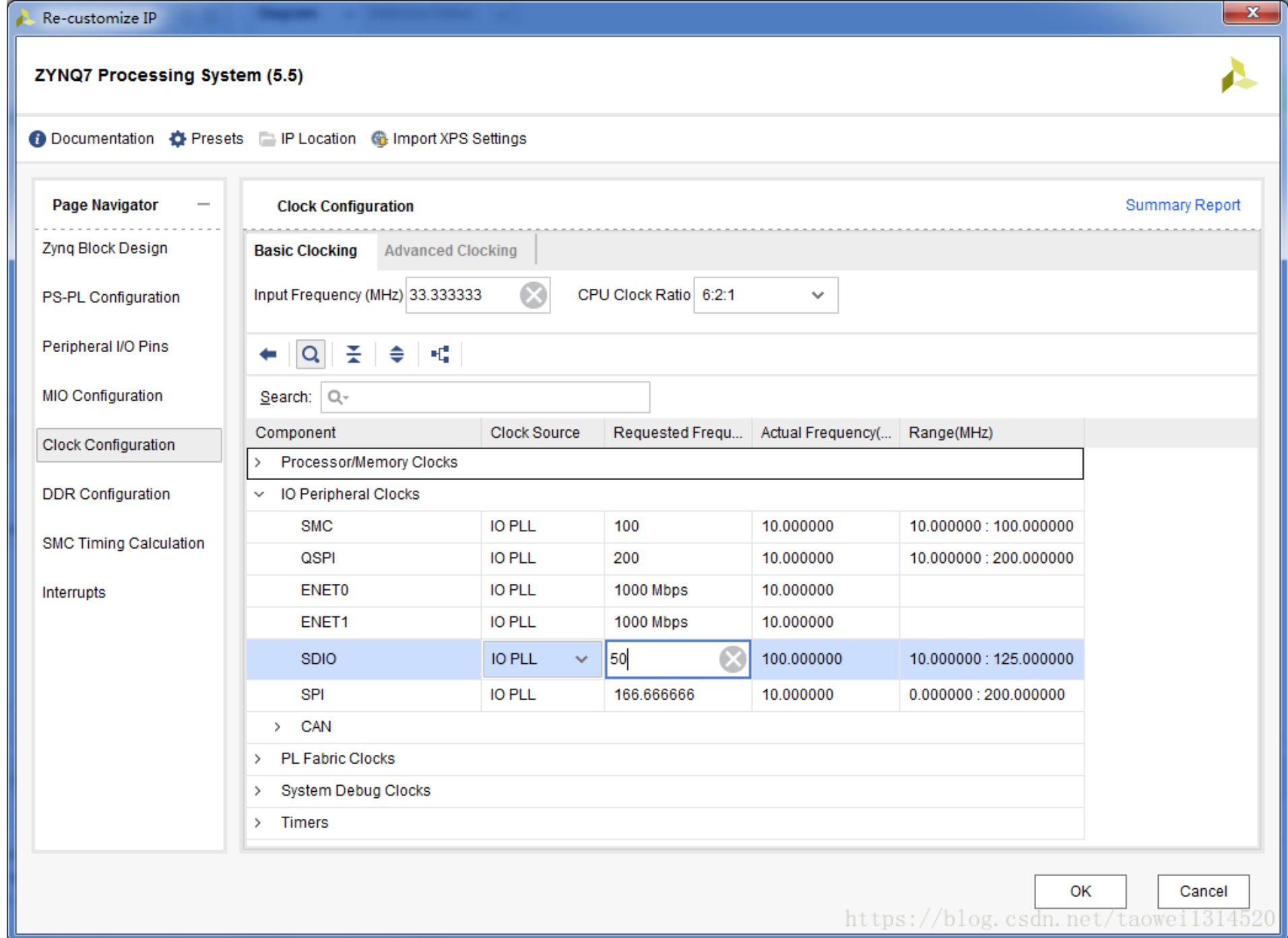




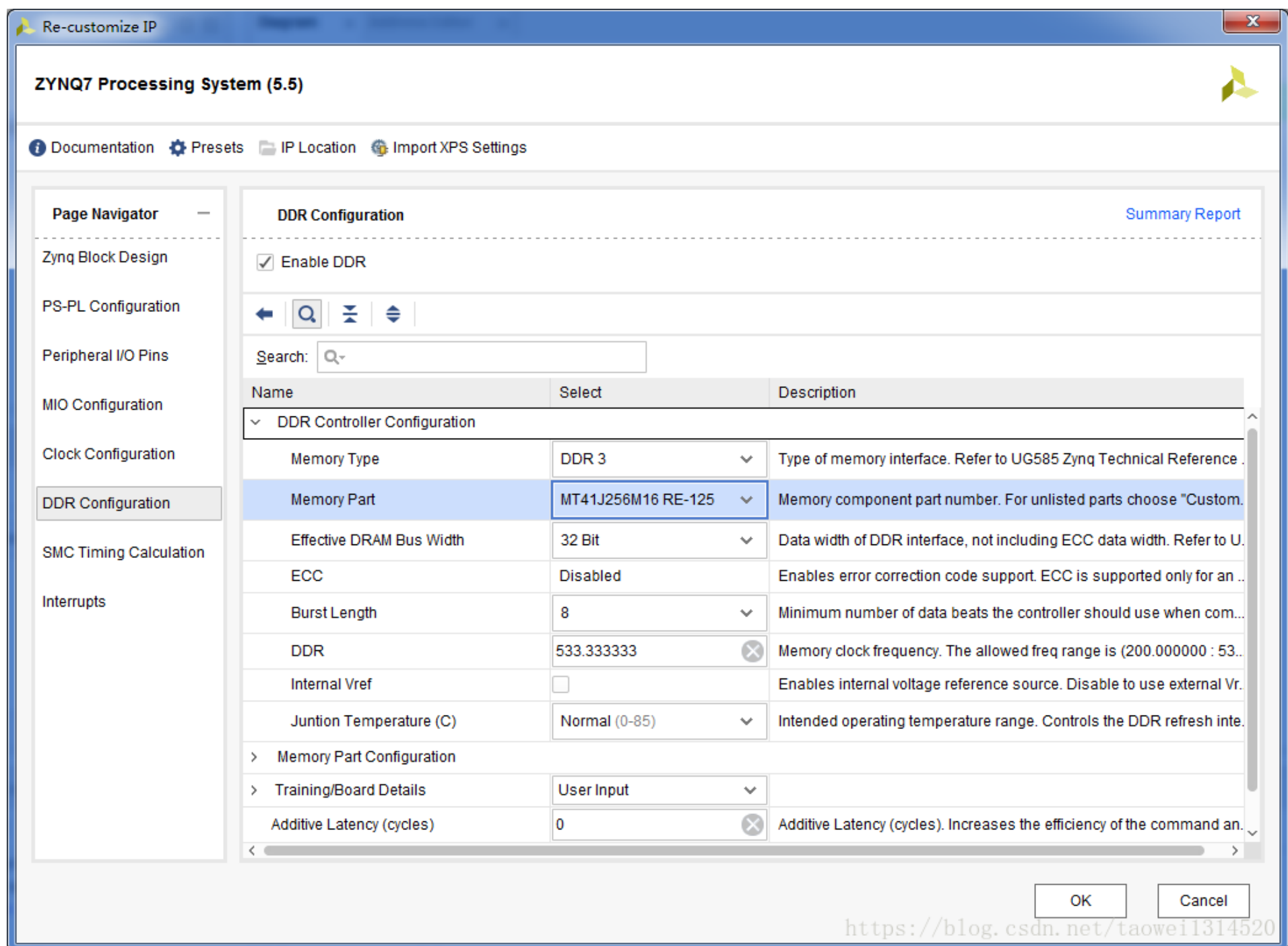




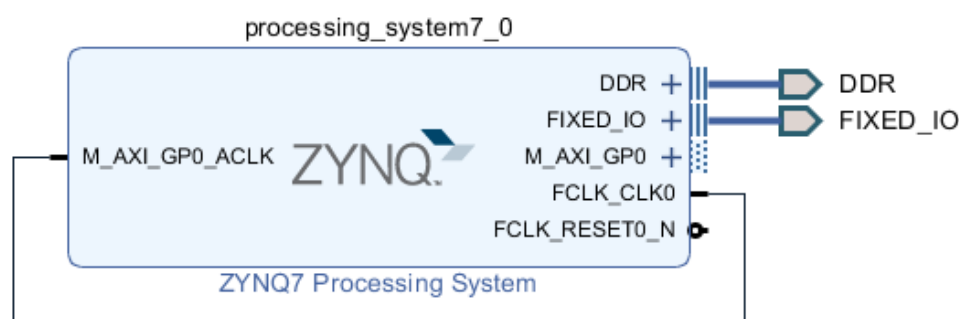
举报



设置DDR型号（不同的开发板有差异）

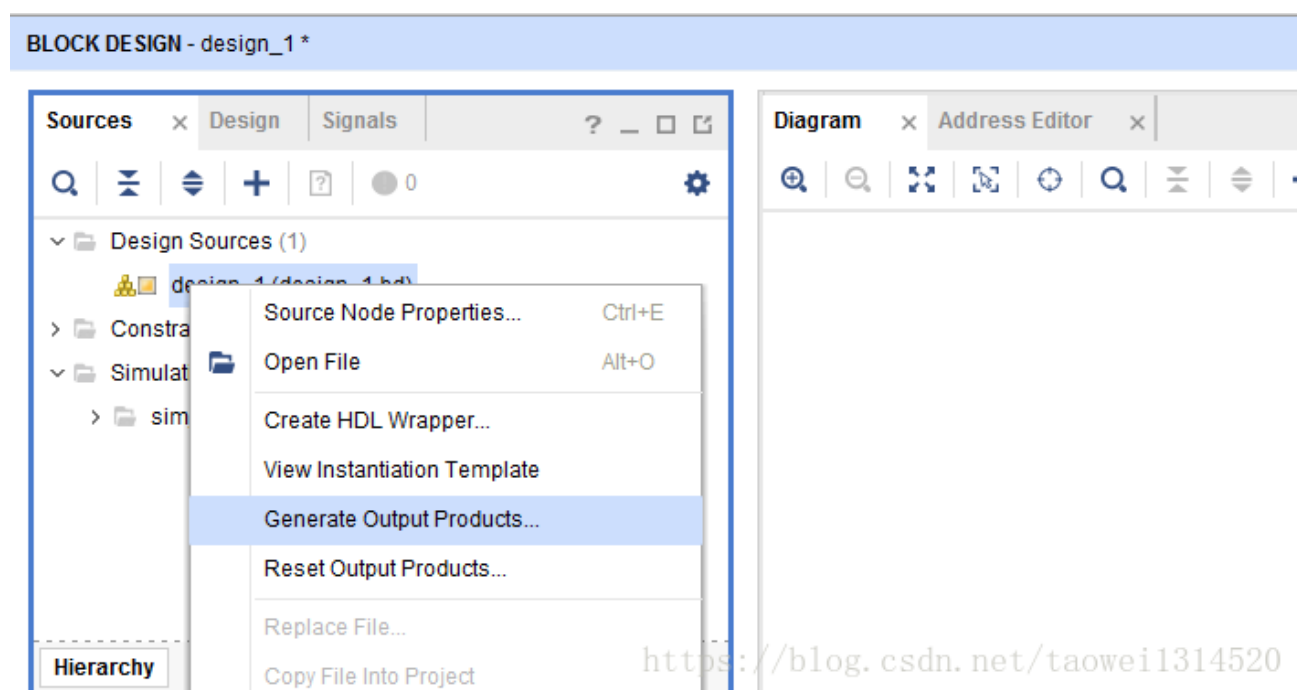


最后按照这个截图进行连接就可以了

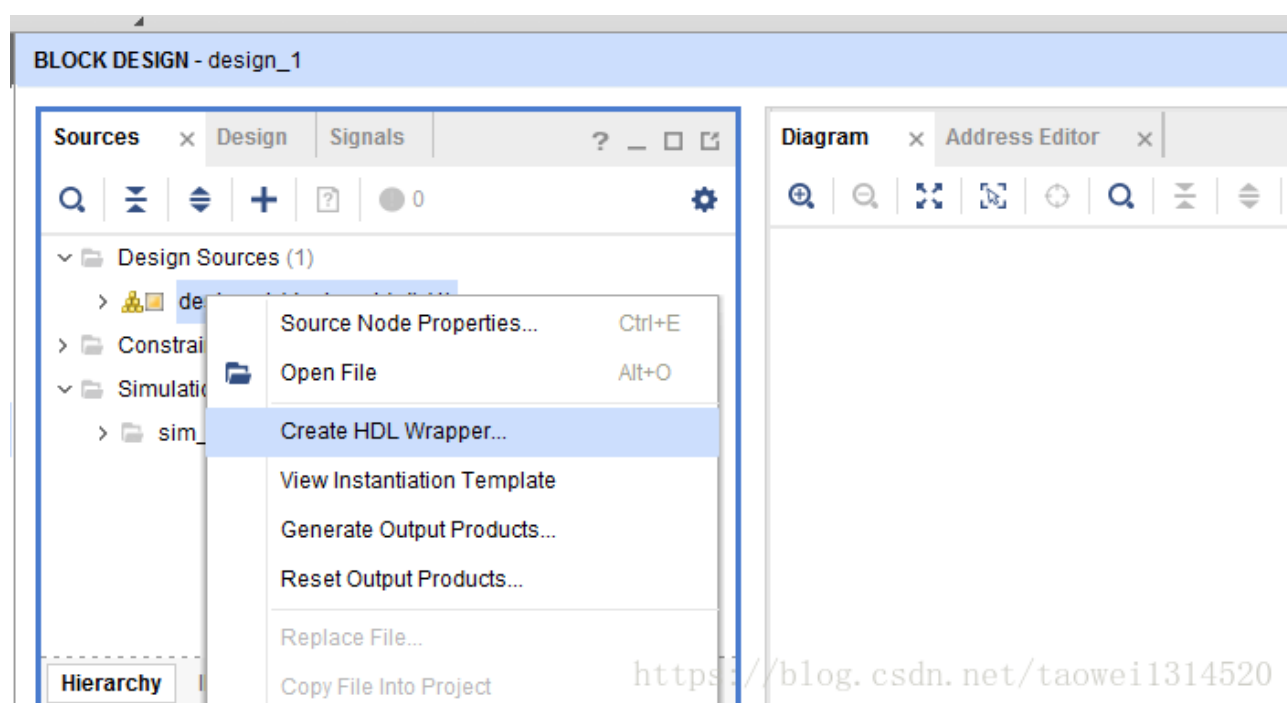


<https://blog.csdn.net/taowei1314520>

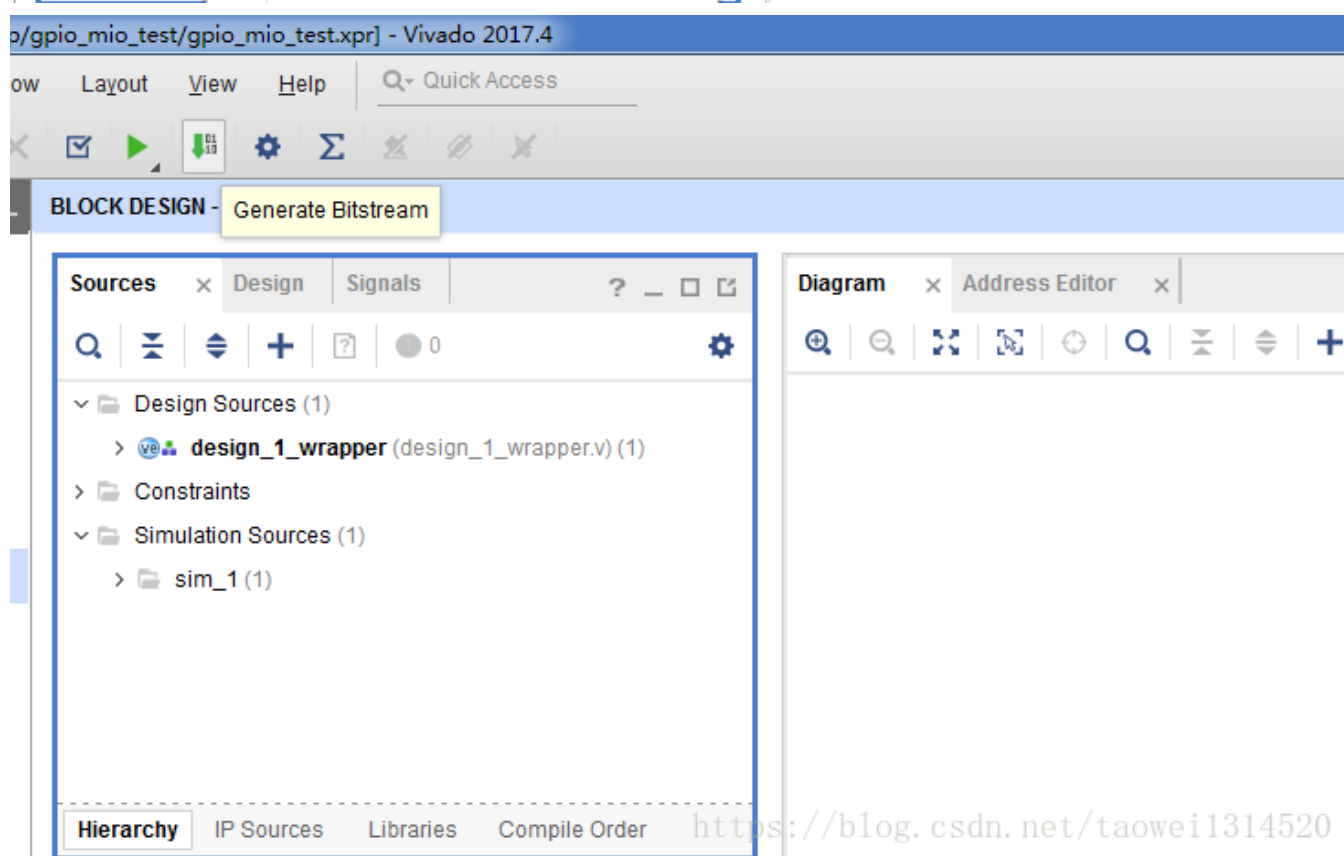
Step2 进行综合、生成顶层文件、生成bit文件



<https://blog.csdn.net/taowei1314520>



<https://blog.csdn.net/taowei1314520>



<https://blog.csdn.net/taowei1314520>

顶层文件如下图所示





2





举报


```
1 //Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.
2 //-----
3 //Tool Version: Vivado v.2017.4 (win64) Build 2086221 Fri Dec 15 20:55:39 MST 2017
4 //Date       : Fri Aug 31 19:22:18 2018
5 //Host       : taowei running 64-bit Service Pack 1 (build 7601)
6 //Command    : generate_target design_1_wrapper.bd
7 //Design     : design_1_wrapper
8 //Purpose    : IP block netlist
9 //-----
10 `timescale 1 ps / 1 ps
11
12 module design_1_wrapper
13     (DDR_addr,
14      DDR_ba,
15      DDR_cas_n,
16      DDR_ck_n,
17      DDR_ck_p,
18      DDR_cke,
19      DDR_cs_n,
20      DDR_dm,
21      DDR_dq,
22      DDR_dqs_n,
23      DDR_dqs_p,
24      DDR_odt,
25      DDR_ras_n,
26      DDR_reset_n,
27      DDR_we_n,
28      FIXED_IO_dds_vrn,
29      FIXED_IO_dds_vrp,
30      FIXED_IO_mio,
31      FIXED_IO_ps_clk,
32      FIXED_IO_ps_porb,
33      FIXED_IO_ps_srstb);
34   inout [14:0]DDR_addr;
35   inout [2:0]DDR_ba;
36   inout DDR_cas_n;
37   inout DDR_ck_n;
38   inout DDR_ck_p;
39   inout DDR_cke;
40   inout DDR_cs_n;
41   inout [3:0]DDR_dm;
42   inout [31:0]DDR_dq;
43   inout [3:0]DDR_dqs_n;
44   inout [3:0]DDR_dqs_p;
45   inout DDR_odt;
46   inout DDR_ras_n;
47   inout DDR_reset_n;
48   inout DDR_we_n;
49   inout FIXED_IO_dds_vrn;
50   inout FIXED_IO_dds_vrp;
51   inout [53:0]FIXED_IO_mio;
52   inout FIXED_IO_ps_clk;
53   inout FIXED_IO_ps_porb;
54   inout FIXED_IO_ps_srstb;
55
56   wire [14:0]DDR_addr;
57   wire [2:0]DDR_ba;
58   wire DDR_cas_n;
59   wire DDR_ck_n;
60   wire DDR_ck_p;
61   wire DDR_cke;
62   wire DDR_cs_n;
63   wire [3:0]DDR_dm;
64   wire [31:0]DDR_dq;
65   wire [3:0]DDR_dqs_n;
66   wire [3:0]DDR_dqs_p;
67   wire DDR_odt;
68   wire DDR_ras_n;
69   wire DDR_reset_n;
70   wire DDR_we_n;
71   wire FIXED_IO_dds_vrn;
72   wire FIXED_IO_dds_vrp;
73   wire [53:0]FIXED_IO_mio;
74   wire FIXED_IO_ps_clk;
75   wire FIXED_IO_ps_porb;
76   wire FIXED_IO_ps_srstb;
77
78   design_1 design_1_i
```


2



















举报

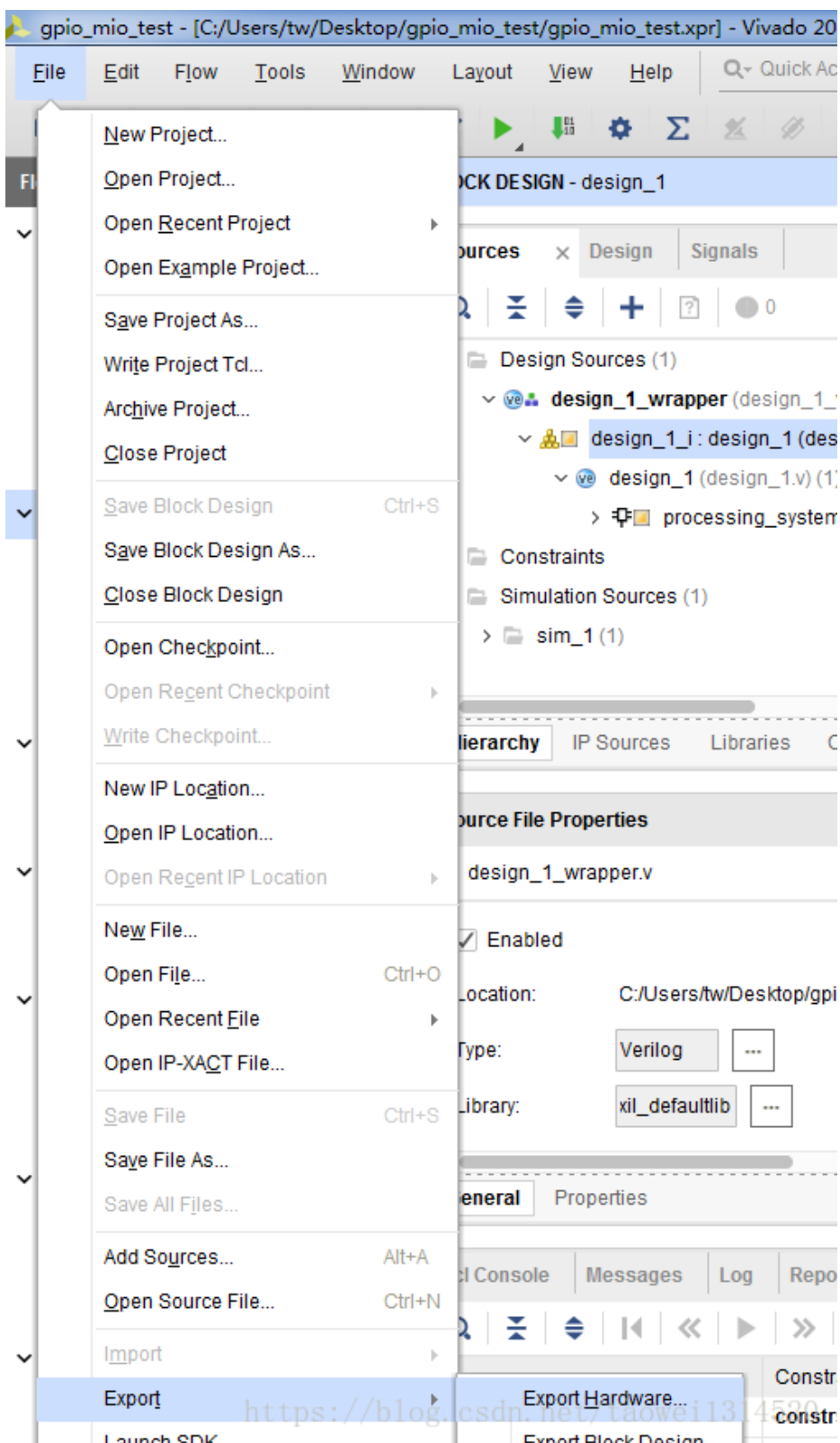

```
79 (.DDR_addr(DDR_addr),
80 .DDR_ba(DDR_ba),
81 .DDR_cas_n(DDR_cas_n),
82 .DDR_ck_n(DDR_ck_n),
83 .DDR_ck_p(DDR_ck_p),
84 .DDR_cke(DDR_cke),
85 .DDR_cs_n(DDR_cs_n),
86 .DDR_dm(DDR_dm),
87 .DDR_dq(DDR_dq),
88 .DDR_dqs_n(DDR_dqs_n),
89 .DDR_dqs_p(DDR_dqs_p),
90 .DDR_odt(DDR_odt),
91 .DDR_ras_n(DDR_ras_n),
92 .DDR_reset_n(DDR_reset_n),
93 .DDR_we_n(DDR_we_n),
94 .FIXED_IO_ddr_vrn(FIXED_IO_ddr_vrn),
95 .FIXED_IO_ddr_vrp(FIXED_IO_ddr_vrp),
96 .FIXED_IO_mio(FIXED_IO_mio),
97 .FIXED_IO_ps_clk(FIXED_IO_ps_clk),
98 .FIXED_IO_ps_porb(FIXED_IO_ps_porb),
99 .FIXED_IO_ps_srstb(FIXED_IO_ps_srstb));
100 endmodule
```



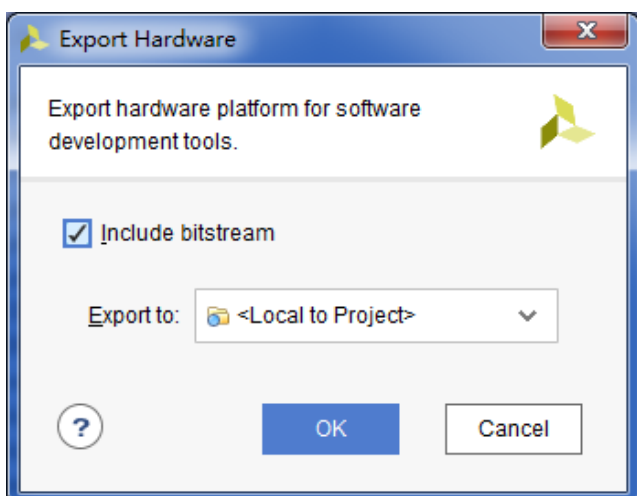
2



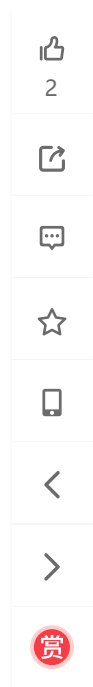
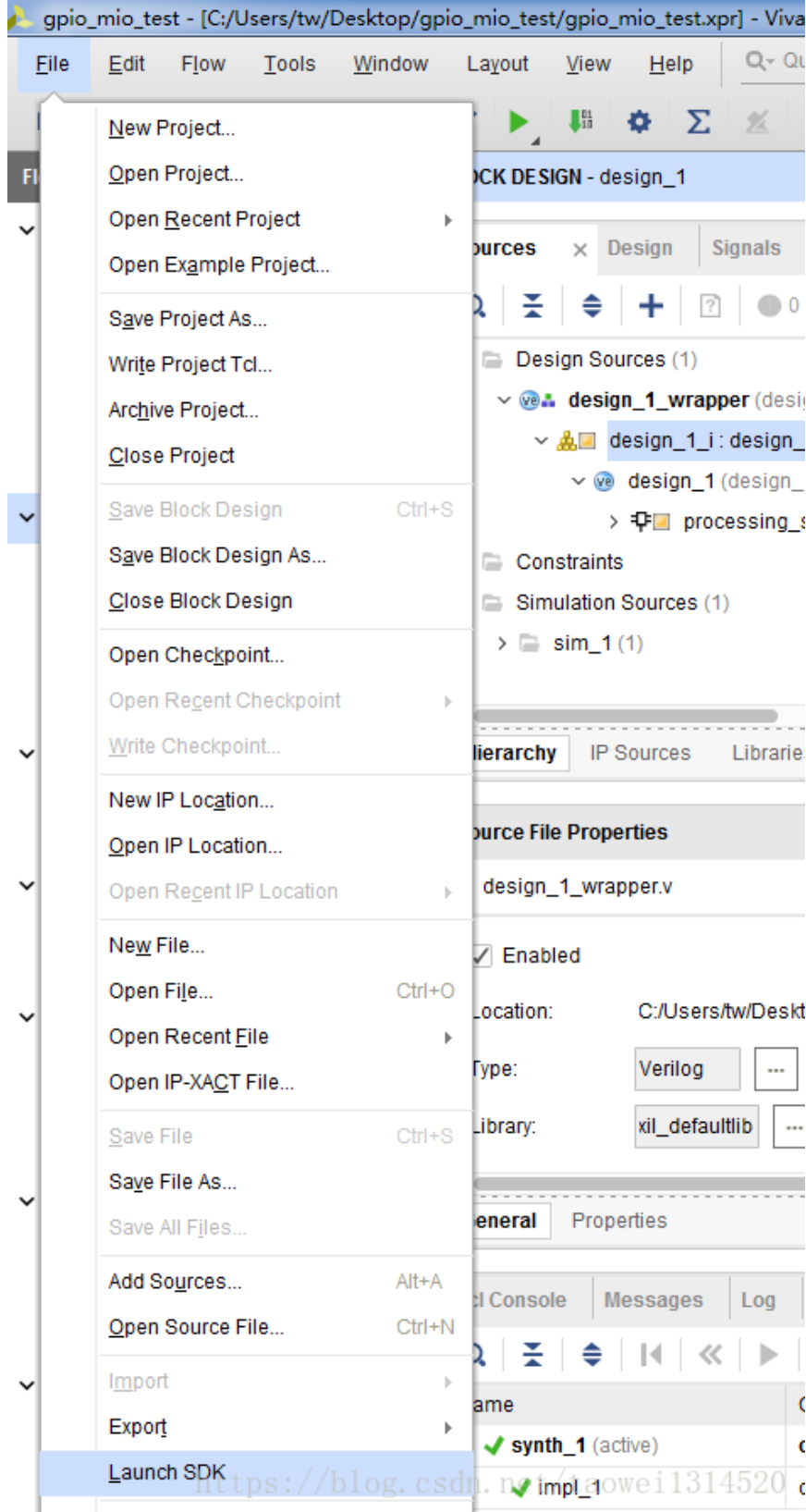
Step3 导出硬件配置、打开SDK



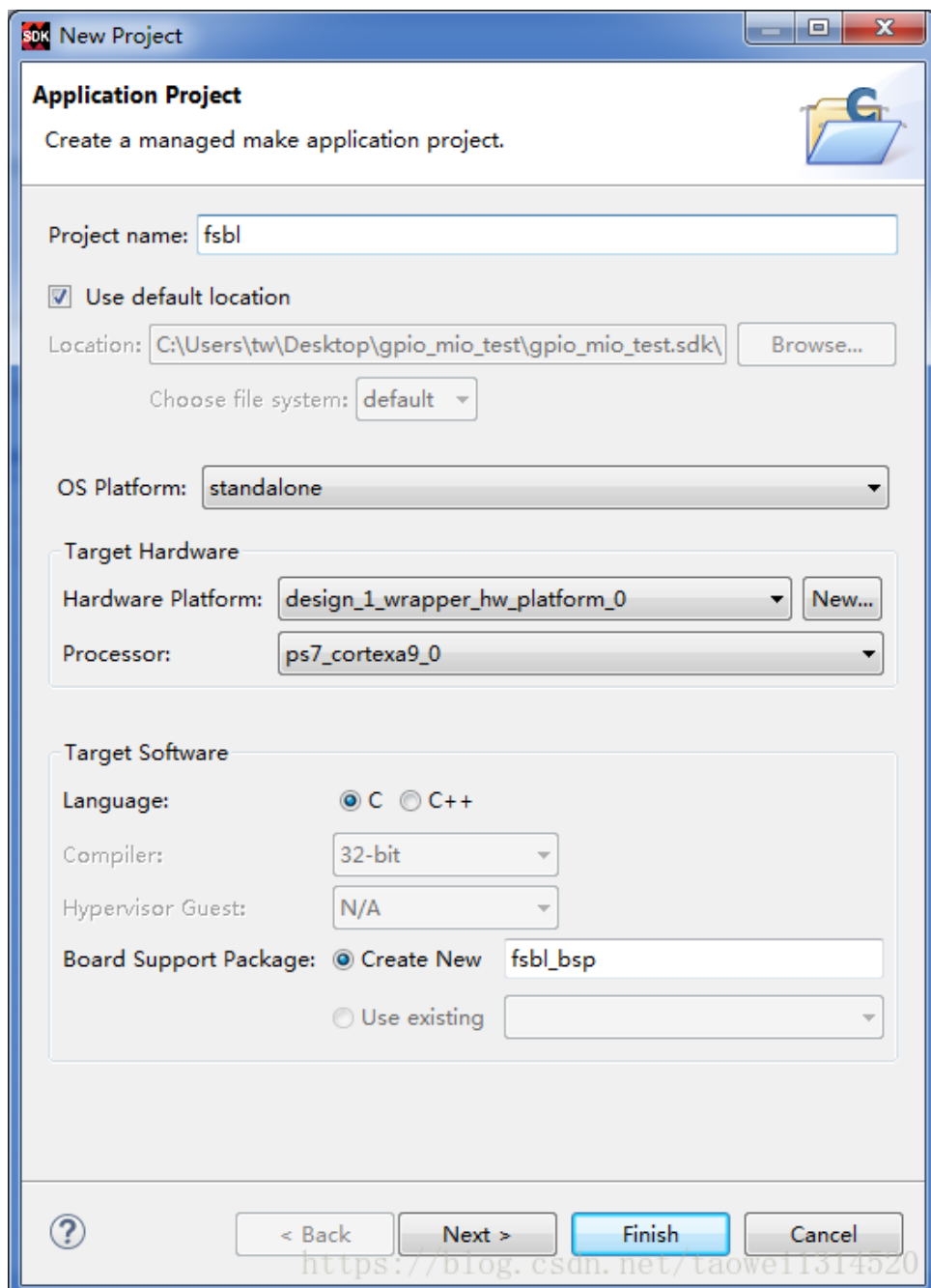
<https://blog.csdn.net/taowei1314520>

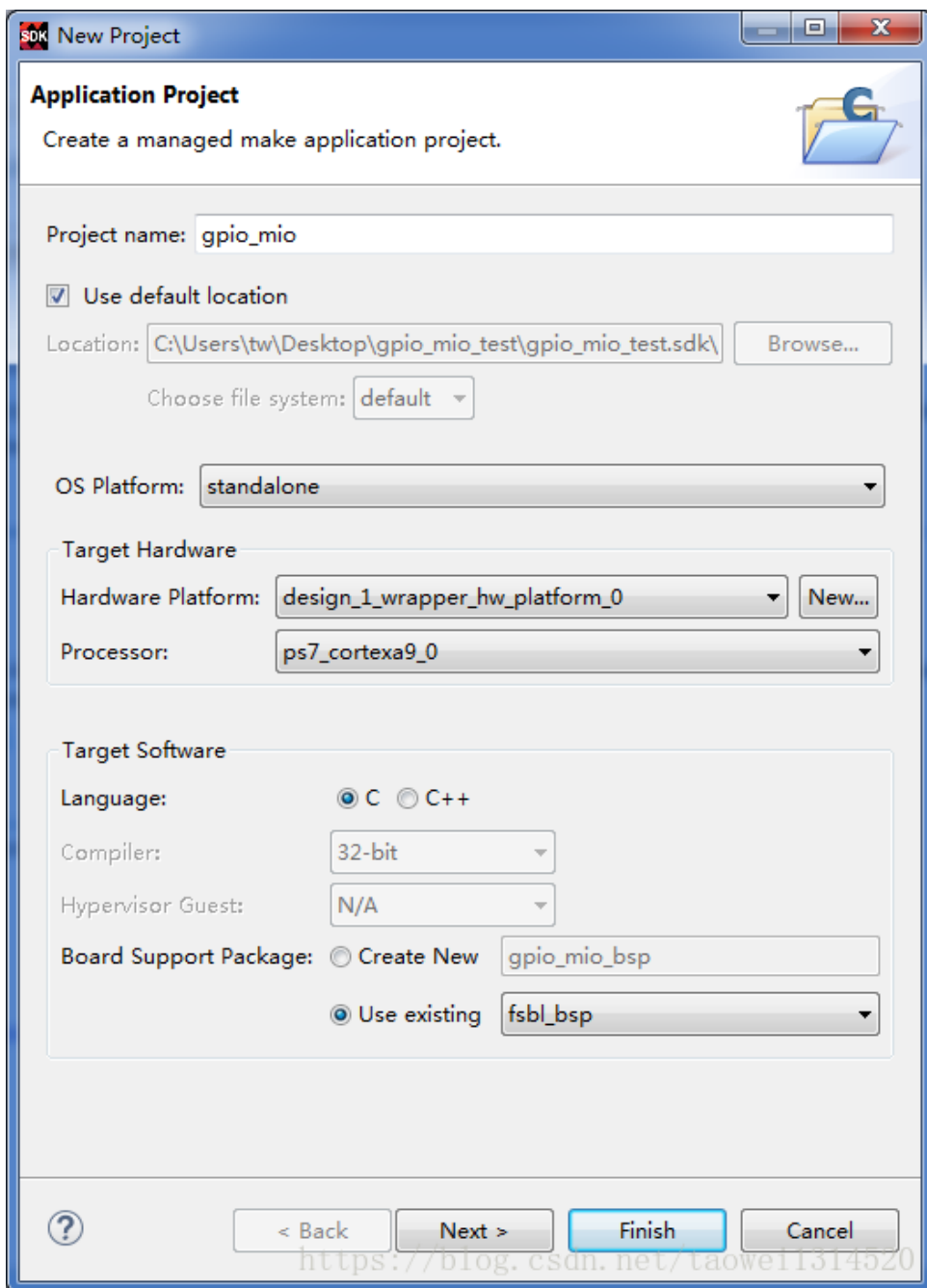
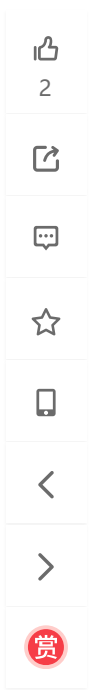
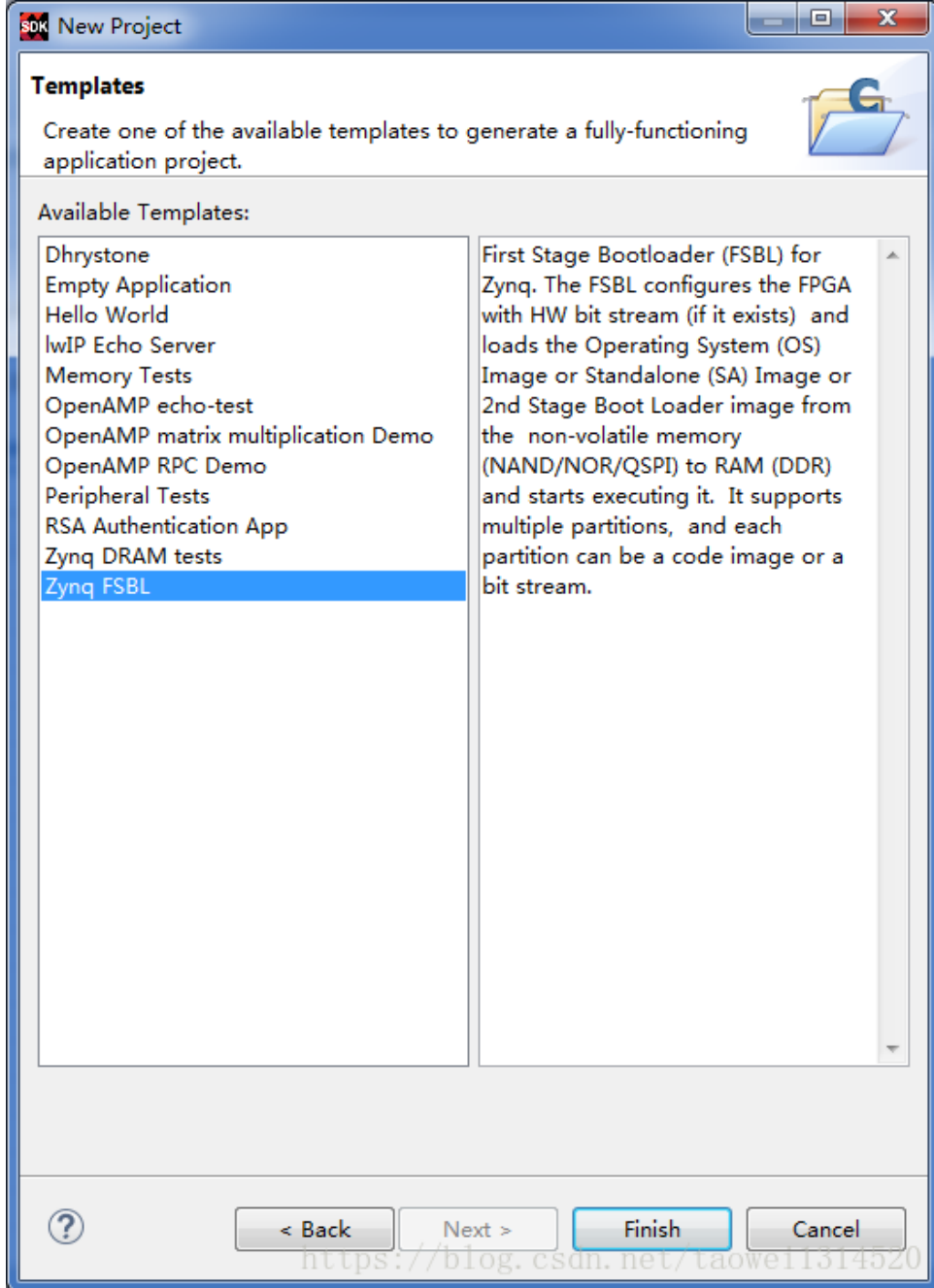


举报



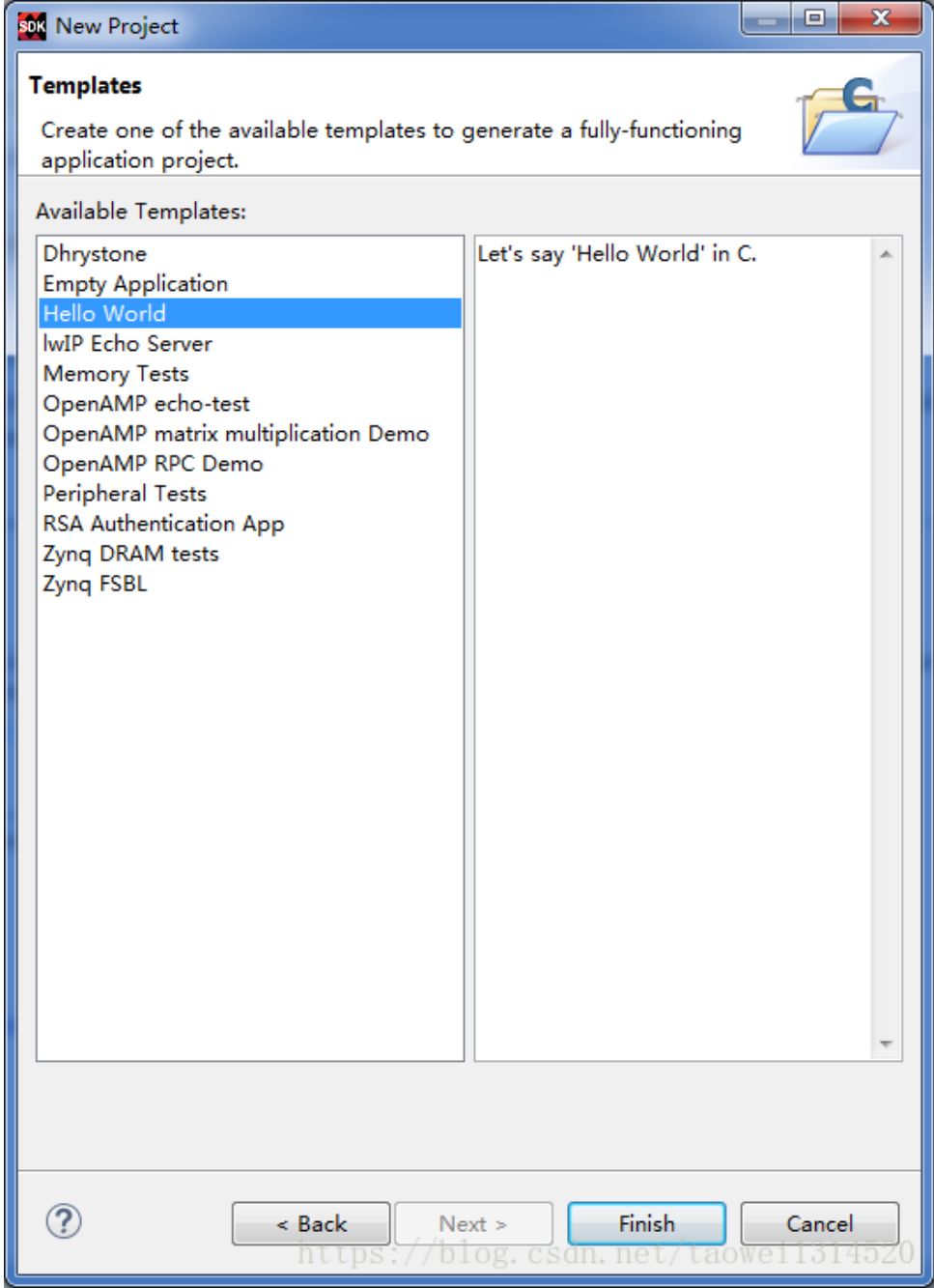
Step4 新建一个fsbl、新建一个gpio_mio工程








hello_world模板









2


















工程代码


```
1  /*****
2  *
3  * Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.
4  *
5  * Permission is hereby granted, free of charge, to any person obtaining a copy
6  * of this software and associated documentation files (the "Software"), to deal
7  * in the Software without restriction, including without limitation the rights
8  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9  * copies of the Software, and to permit persons to whom the Software is
10 * furnished to do so, subject to the following conditions:
11 *
12 * The above copyright notice and this permission notice shall be included in
13 * all copies or substantial portions of the Software.
14 *
15 * Use of the Software is limited solely to applications:
16 * (a) running on a Xilinx device, or
17 * (b) that interact with a Xilinx device through a bus or interconnect.
18 *
19 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 *****/
32
33 /*
34  * helloworld.c: simple test application
35  *
36  * This application configures UART 16550 to baud rate 9600.
37  * PS7 UART (Zynq) is not initialized by this application, since
38  * bootrom/bsp configures it to baud rate 115200
39  *
40  * -----
41  * | UART TYPE   BAUD RATE |
42  * -----
```





举报


```
43 | *      uartns550      9600
44 | *      uartlite      Configurable only in HW design
45 | *      ps7_uart      115200 (configured by bootrom/bsp)
46 | */
47 |
48 | #include <stdio.h>
49 | #include "platform.h"
50 | #include "xil_printf.h"
51 | #include "xil_io.h"
52 | #include "sleep.h"
53 |
54 |
55 | #define Base_Address      0xE000A000
56 |
57 | #define DIRM_0              0x00000204 //mio0~mio31
58 |
59 | #define DIRM_1              0x00000244 //mio32~mio53
60 |
61 | #define DATA_0_R0          0x00000060 //mio0~mio31  data
62 |
63 | #define DATA_1_R0          0x00000064 //mio32~mio53  data
64 |
65 |
66 |
67 |
68 | int main()
69 | {
70 |     u32 rxdb1;
71 |     u32 rxdb2;
72 |
73 |     Xil_Out32(Base_Address + DIRM_1,0x00);    //模式设置: 设置mio32~mio53为输入      (0为输入      1为输出)
74 |
75 |
76 | while(1)
77 | {
78 |     rxdb1 = Xil_In32(Base_Address + DATA_1_R0);    //接收mio32~mio53的输入数据(22位数据)
79 |
80 |     rxdb2 = (rxdb1 & 0x040000) ;                    //只取这个mio50的数据
81 |
82 |     xil_printf("rxdb2 = %08x\n\r",rxdb2);          //打印mio50的输出数据
83 |
84 |     sleep(1);
85 |
86 | }
87 |
88 | }
```


2











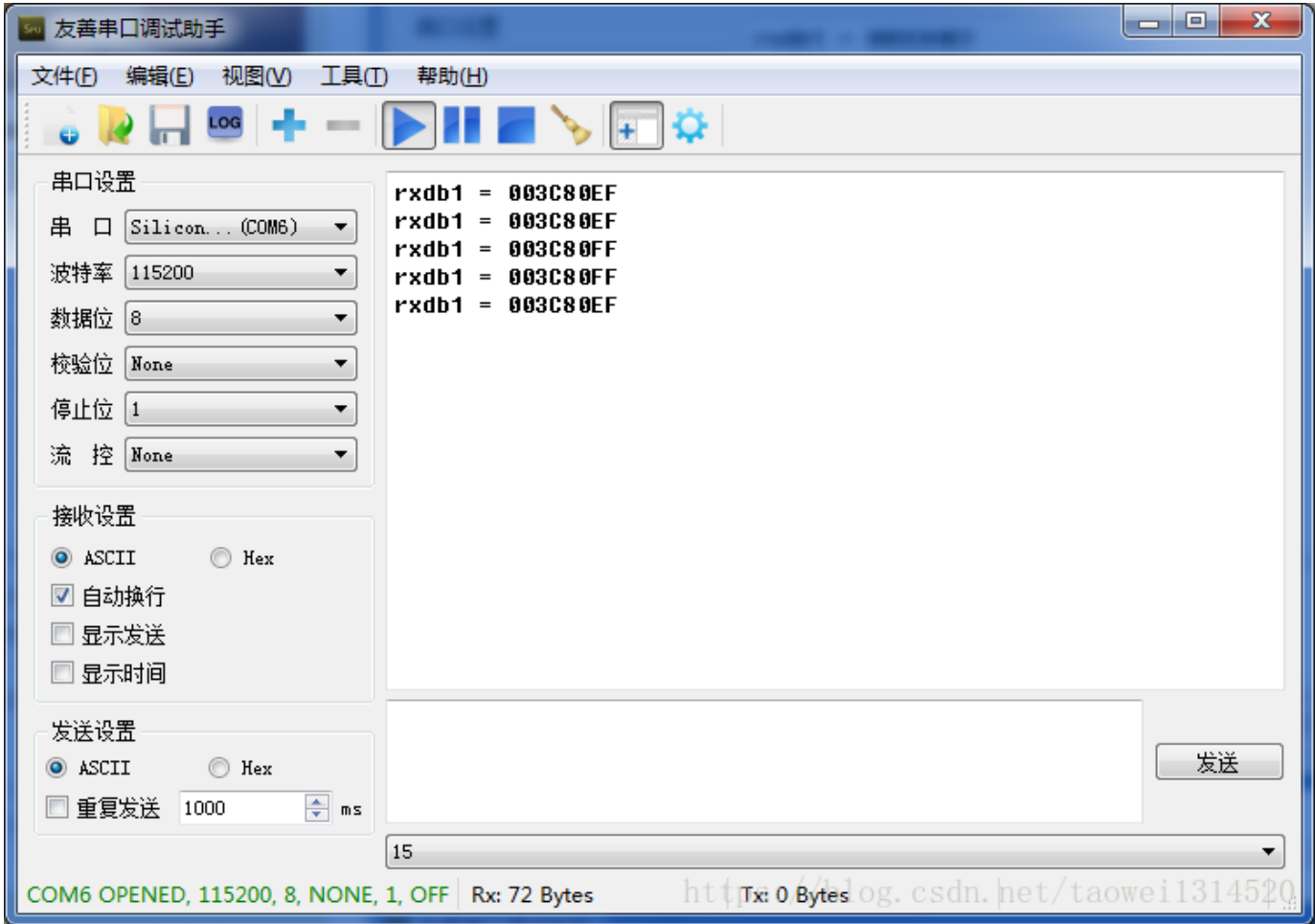






生成BOOT.bin文件后将BOOT.bin文件拷贝到SD卡运行，运行结果如下图所示

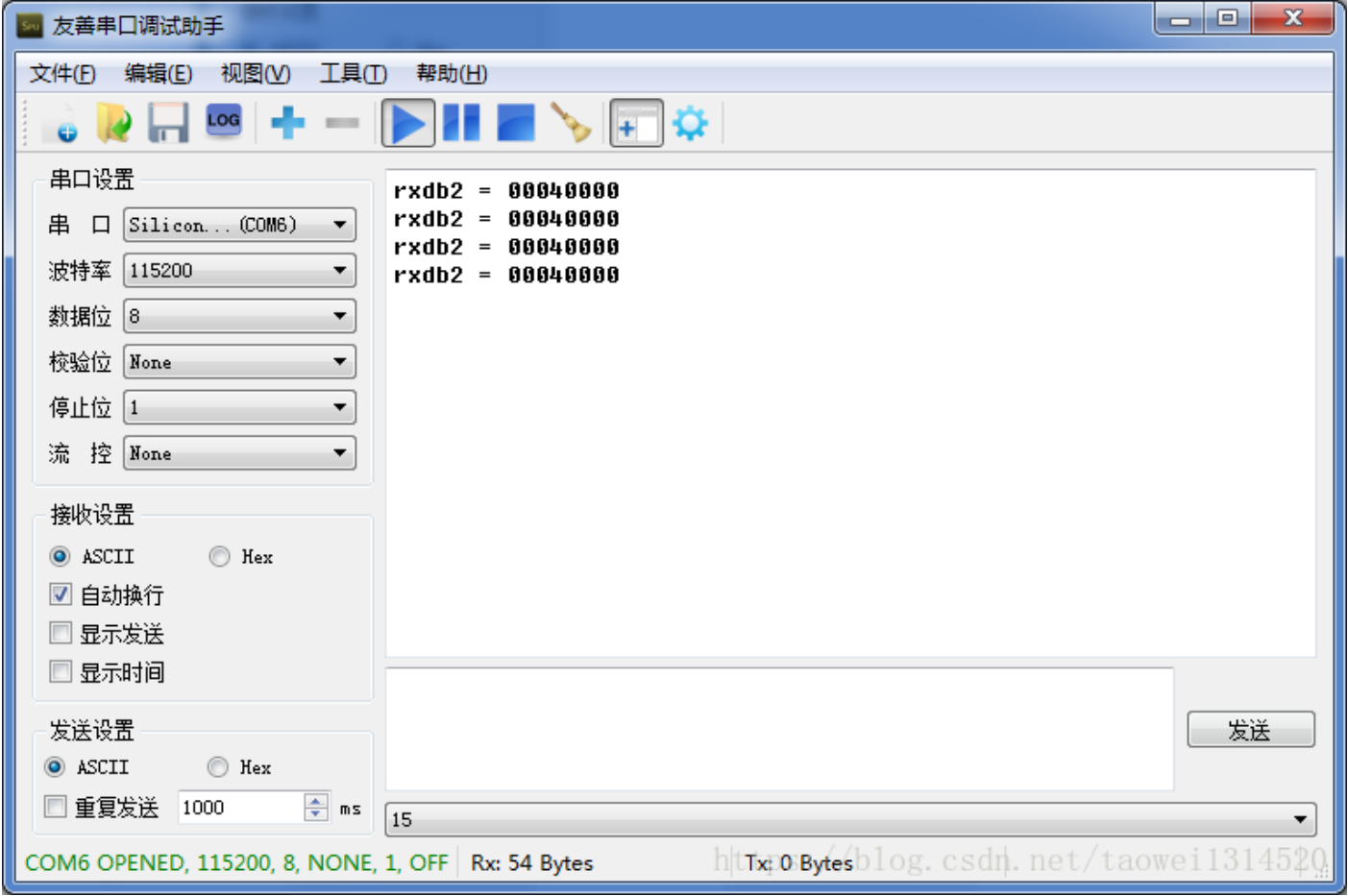
这里是rxdb1的输出数据，可能大家会奇怪这个mio32~mio53输入数据为什么不全为0，因开发板的ps端的mio都接入了不同的设备，而这些设备上电后有的设备是高电平、有的设备是低电平所以输入值不全为0



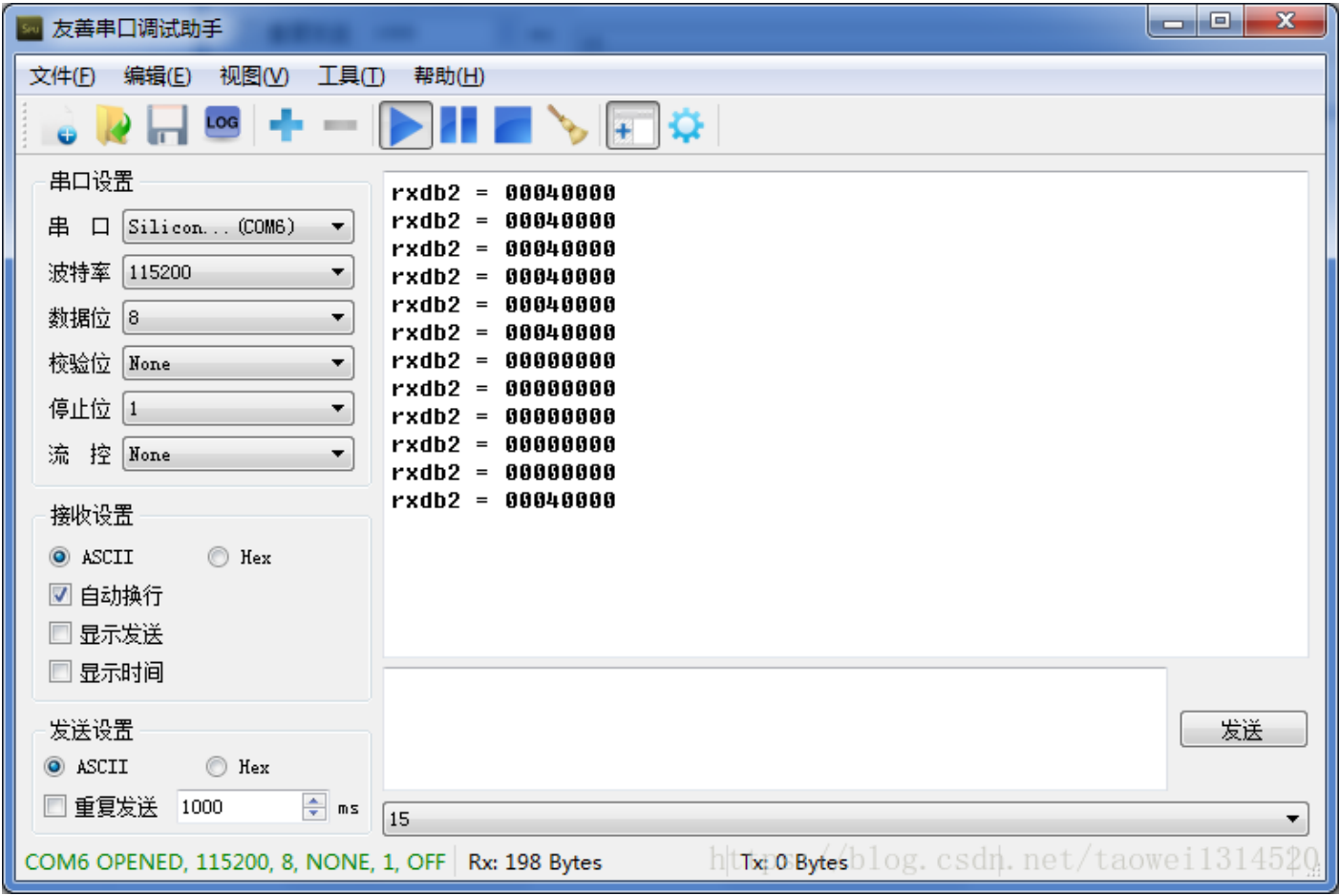


举报


我们再来看下rxdb2的输出，rxdb2是直接接收的mio50的值也就是按键的值，这个是没有按按键的输出



这个是按了按键的输出，按了按键输入变成0了，然后松开后又为高了



点赞 2 收藏 分享 ...

 虚无缥缈vs威武
发布了37 篇原创文章 · 获赞 59 · 访问量 15万+

私信 关注

 想对作者说点什么

ZYNQ SOC 入门基础（一）Hello World 实验

阅读数 3044

ZYNQ是一款SOC芯片，其最突出的功能就是其内部包含了一个双核的Cortex_A9内核。从本节开始，进行ZYNQ的S... 博文 来自： MaoChuangAn的...

Zynq视频教程

阅读数 2267

网址为：http://www.osrc.cn/forum.php?mod=forumdisplay&fid=333 博文 来自： ffdia的博客

zynq 图像处理教程fpga教程，（含例程）手把手教会你zynq图像处理。下载

12-13

zynq 图像处理教程fpga教程，（含例程）手把手教会你zynq图像处理。摄像头图像采集，图像处理，例程全，易学好上手... 论坛

Zynq7000的MIO和EMIO之区别

阅读数 3500

Zynq7000系列芯片有5 博文 来自： linyangspring的专栏

- 2
-
-
-
-
-
- 赏

举报

zynq 7000修炼秘籍fpga教程《经典》最新最全的教程，超过1000页的pdf电子书12-13

最新最全的FPGA开发教程，zynq7000系列fpga，ps和pl端的开发教程，超过1000页的pdf电子书。下载

基于Zynq的MIO与EMIO的区别和应用 阅读数 8378

MIO与EMIO的区别与应用1 MIO与EMIO概念MIO：多功能IO接口，属于Zynq的PS部分，在芯片外部有54个引脚。... 博文 来自： limoon1212的博客

黑金zynq7000教程 11-19

黑金Zynq7000开发平台配套教程 适合学些Xilinx ZYNQ系列SOC的朋友们学习，本文档主要讲解vivado的调试技巧，sdk... 下载

学会Zynq（4）GPIO中MIO的使用方法 阅读数 1679

本文将介绍PS部分GPIO中MIO的使用。本文先通过一个控制LED闪烁的实例体会MIO的用法，学习GPIO相关结构体... 博文 来自： FPGADesigner的...

MYIR-ZYNQ7000系列-zturn教程(2)：Hello_World 阅读数 3875

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1（工程末尾提供了工程源代码大家可以去网盘下载）ste... 博文 来自： taowei1314520的...

翻译：A Tutorial on the Device Tree (Zynq) -- Part IV 阅读数 15

获取资源信息内核模块驱动加载之后，就开始把硬件资源管理起来，如读写寄存器、接收中断。来看看设备树里的一... 博文 来自： weixin_30606669...

MYIR-ZYNQ7000系列-zturn教程(26)：自定义axi_lite IP点亮LED灯 阅读数 173

开发板环境：vivado 2017.4，开发板型号xc7z020clg400-1，这个工程主要功能是自定义一个axi_lite IP然后在SDK... 博文 来自： taowei1314520的...



为中国IC之崛起而读书
85篇文章
排名:千里之外

关注



ffdia
65篇文章
排名:千里之外

关注



眼镜333
129篇文章
排名:千里之外

关注



linyangspring
52篇文章
排名:千里之外

关注

MYIR-ZYNQ7000系列-zturn教程(8)-PS给PL时钟点亮LED 阅读数 3614

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这里用的这个工程是“从新建工程到下载bit”这个工... 博文 来自： taowei1314520的...

zynq中三种实现GPIO的方式 阅读数 9125

本文介绍在zynq中三种实现GPIO的方式，分别为MIO、EMIO和IP方式。MIO和EMIO方式是使用PS部分的GPIO模... 博文 来自： husipeng86的专栏

翻译：A Tutorial on the Device Tree (Zynq) -- Part I 阅读数 23

A Tutorial on the Device Tree (Zynq) -- Part I此教程的目的本教程是针对Xilinx' Zynq-7000 EPP设备(一个集成了... 博文 来自： weixin_30678349...

《Xilinx ZYNQ-7000 AP SoC开发实战指南》（符晓）高清扫描版下载 07-01

《Xilinx ZYNQ-7000 AP SoC开发实战指南》（符晓）高清扫描版 比工大“he滨”的书好多了哦！ 相关下载链接：//dow... 论坛

ALINX黑金Zynq7000开发平台配套教程 03-19

当我们拿刚开収板是否应该测试一下收刚乧开収板是否功能正常，以及各个接口工作是否工作正常。在出厂前我们已经在S... 下载

Zedboard学习（五）：MIO与EMIO操作 阅读数 4747

MIO： Zynq7000 系列芯片有 54 个 MIO(multiuse I/O)， 它们分配在 GPIO 的 Bank0 和Bank1 隶属于 PS 部分， ... 博文 来自： hongbin_xu的博客

Xilinx Vivado的使用详细介绍（3）：使用IP核 阅读数 7600

IP核（IPCore）Vivado中有很多IP核可以直接使用，例如数学运算（乘法器、除法器、浮点运算器等）、信号处理（... 博文 来自： jzj1993的专栏

Zynq 7000嵌入式设计官方教程.pdf 05-11

Zynq 7000嵌入式设计官方教程,Zynq 7000嵌入式设计官方教程 下载

学习Zynq-7000的入门书单 阅读数 215

根据选用的芯片型号和应用领域的不同，读者可以适当裁减。Entrance Readings:1. Zynq-7000 User Guides Zynq... 博文 来自： wangzh92429的博...

【JokerのZYNQ7020】AXI_DMA_PL_PS。 阅读数 450

软件环境：vivado 2017.4 硬件平台：XC7Z020这篇跟上一篇AXI_DMA_LOOP大体框架差不多，差别主要是体现在... 博文 来自： Joker_是小王的博客

zynq开发板教程 09-07


黑金开发板zynq手把手教程，黑金开发板zynq手把手教程，黑金开发板zynq手把手教程，黑金开发板zynq手把手教程，黑... 下载


【ZYNQ-7000开发之十】TGP+VDMA+HDMI搭建视频通路 阅读数 1万+


本编文章在ADI官方HDMI例程的基础上进行修改，实现视频通路，为使用ZYNQ视频处理做好必要准备。在【ZYN... 博文 来自： RZJMPB的博客


zynq-7000学习笔记(四)——Zedboard HDMI核的构建和输出显示测试（2017/6/9补充修改） 阅读数 1万+


参考：1、【ZYNQ-7000开发之三】ZYNQ平台的HDMI驱动测试2、 ADV7511 Xilinx Evaluation Boards Refere... 博文 来自： luotong86的专栏


2



















举报

Zynq 7000嵌入式设计官方教程.zip05-20

这本书是关于Zynq 7000嵌入式设计官方教程,可以熟读的可以快速的入门zynq 7000开发板. 下载

ZYNQ中断使用入门基础教程 阅读数 223

http://www.elecfans.com/emb/fpga/20171122583385.html任何一个嵌入式系统级的设计都离不开中断，对于拥... 博文 来自： 爱她就要努力

ALINX黑金Zynq7000开发平台配套教程(共2部)03-01

ALINX黑金Zynq7000开发平台配套教程，文档介绍详细，适合zynq7000新手开发帮助使用。 下载

MYIR-ZYNQ7000系列-zturn教程(19)：对axi_stream核进行仿真以及axi_stream总线的初步讲解 阅读数 1226

我这里一共调用了两个自定义的IP都是基于axi_stream的IP核，一个是主机master一个是从机slave，然后将这两个... 博文 来自： taowei1314520的...

zynq-7000系列基于zynq-zed的vivado初步设计之linux下控制PL扩展的UART 阅读数 5532

zynq-7000系列基于zynq-zed的vivado初步设计之linux下控制PL扩展的UART 作者： 卢浩 ... 博文 来自： luhao806的专栏

MYIR-ZYNQ7000系列-zturn教程(1)-从新建工程到下载bit文件 阅读数 1417

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1（工程末尾提供了工程源代码大家可以去网盘下载）St... 博文 来自： taowei1314520的...

MYIR-ZYNQ7000系列-zturn教程(24)：用vdma搭建hdmi显示通路 阅读数 920

开发板环境：vivado 2017.4，开发板型号xc7z010clg400-1，这个工程主要用vdma核搭建hdmi显示工程链接：ht... 博文 来自： taowei1314520的...

【黑金ZYNQ7000系列原创视频教程】02.视频接口——hdmi编码输出实验 阅读数 131

黑金论坛地址：http://www.heijin.org/forum.php?mod=viewthread&tid=36636&extra=page%3D1 爱奇艺地... 博文 来自： weixin_34245169...

在ZYNQ上裸机跑ARM程序演示 阅读数 130

今天给大家演示如何在ZYNQ上，裸机跑ARM程序，本测试用的是米尔Z-turn Board单板，测试代码用的XILINX官... 博文 来自： u011837423的专栏

Zynq 7000从零开始之一 -- HelloWord 阅读数 2万+

使用myir的z-turn开发板，做一个从uart打印hello world的实验，只用PS，不用PL部分,程序从SD卡启动,跑在PS的... 博文 来自： 青蛙@嘎嘎

zynq-7000 从零开始入门06-23

zynq-7000开发流程，快速入门，快速学习vivado和SDK开发软件 下载

学习zynq 阅读数 1353

学习Zynq-7000的入门书单排行榜收藏 打印 发给朋友 举报 发布者：jackzhang热度300票 浏览2187次 【共0条评... 博文 来自： xiaguangjwc的专栏

基于ZYNQ的嵌入式学习笔记三（独立按键中断） 阅读数 461

一、ZYNQ中断概述中断类型：私有外设中断（PPI）：全局定时器、私有看门狗、私有定时器、PL部分产生的FIQ/I... 博文 来自： 万万VV

学会Zynq（5）GPIO中EMIO的使用方法 阅读数 1547

之前的Hello World和MIO使用都算是纯PS部分，也就是把Zynq单纯地当作ARM使用。很多人都是因为FPGA+AR... 博文 来自： FPGADesigner的...

MYIR-ZYNQ7000系列-zturn教程(4)：gpio_emio 阅读数 1445

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要本章描述怎样通过EMIO引出GPIO，并... 博文 来自： taowei1314520的...

MYIR-ZYNQ7000系列-zturn教程(10)：debug调试 阅读数 1574

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要是用hello_world这个工程进行debug... 博文 来自： taowei1314520的...

python json java mysql pycharm android linux json格式



虚无缥缈vs威武

TA的个人主页 >

原创	粉丝	获赞	评论	访问
37	195	59	164	15万+

等级: 博客 4 周排名: 3万+

积分: 1254 总排名: 6万+

勋章:  

关注

私信



2



举报

專業汽車借款、天上

重視隱私，避免尷尬，安心

汽車借款業務，不限車種車齡，也可，最高可借車價的二倍，增運用
037593333.tw

開啟

最新文章

quartus II 12.1 使用教程（7） vga显示测试

MYIR-ZYNQ7000系列-zturn教程(27): lwip测试

quartus II 12.1 使用教程（6） ROM 测试

quartus II 12.1 使用教程（5） eeprom 读写测试

quartus II 12.1 使用教程（4） uart 测试

分类专栏

	VIVADO 安装教程	1篇
	quartus II	5篇
	三态门详解	
	quartus II 12.1 使用...	1篇
	ZYNQ7000	27篇

归档

2019年12月	1篇
2019年9月	1篇
2019年8月	5篇
2019年7月	2篇
2019年4月	1篇
2019年3月	2篇
2019年1月	1篇
2018年11月	1篇

展开

热门文章

VIVADO 安装教程
阅读数 84216

三态门详解
阅读数 15398

quartus II 12.1 使用教程（1） 怎样调用PLL 核
阅读数 7556

MYIR-ZYNQ7000系列-zturn教程(17): 用axi_uart发送数据
阅读数 4156

MYIR-ZYNQ7000系列-zturn教程(9): 将bit文件固化到QSPI_Flash
阅读数 4055

最新评论

VIVADO 安装教程
rq8866: 缺License的小伙伴 链接: https://pan.baidu.com/s/11mjkpyERdUH3q5C_TpfQxQ ...



2



举报

FT232H如何使用jtag接口

taowei1314520: [reply]qq_42662835[/reply]我是直接对eeprom里写数据进去的，数据我已经 ...

FT232H如何使用jtag接口

taowei1314520: [reply]sssshhhhhhhh[/reply]这个vivado有这个usb驱动也需要安装一下，你 ...

FT232H如何使用jtag接口

sssshhhhhhhh: 你好，插上电脑以后显示 USB S erial Conventor （仅配置了USB和EEPROM这 ...

MYIR-ZYNQ7000系列-z...

kuyunge: SPI一次是通信一个字节码？



👤 QQ客服 ✉ kefu@csdn.net

🗣 客服论坛 ☎ 400-660-0108

工作时间 8:30-22:00

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

京ICP备19004658号 经营性网站备案信息

🇨🇳 公安备案号 11010502030143

©1999-2020 北京创新乐知网络技术有限公司
网络110报警服务

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉

👍
2

🔗

💬

☆

📱

<

>

赏

🔊

举报