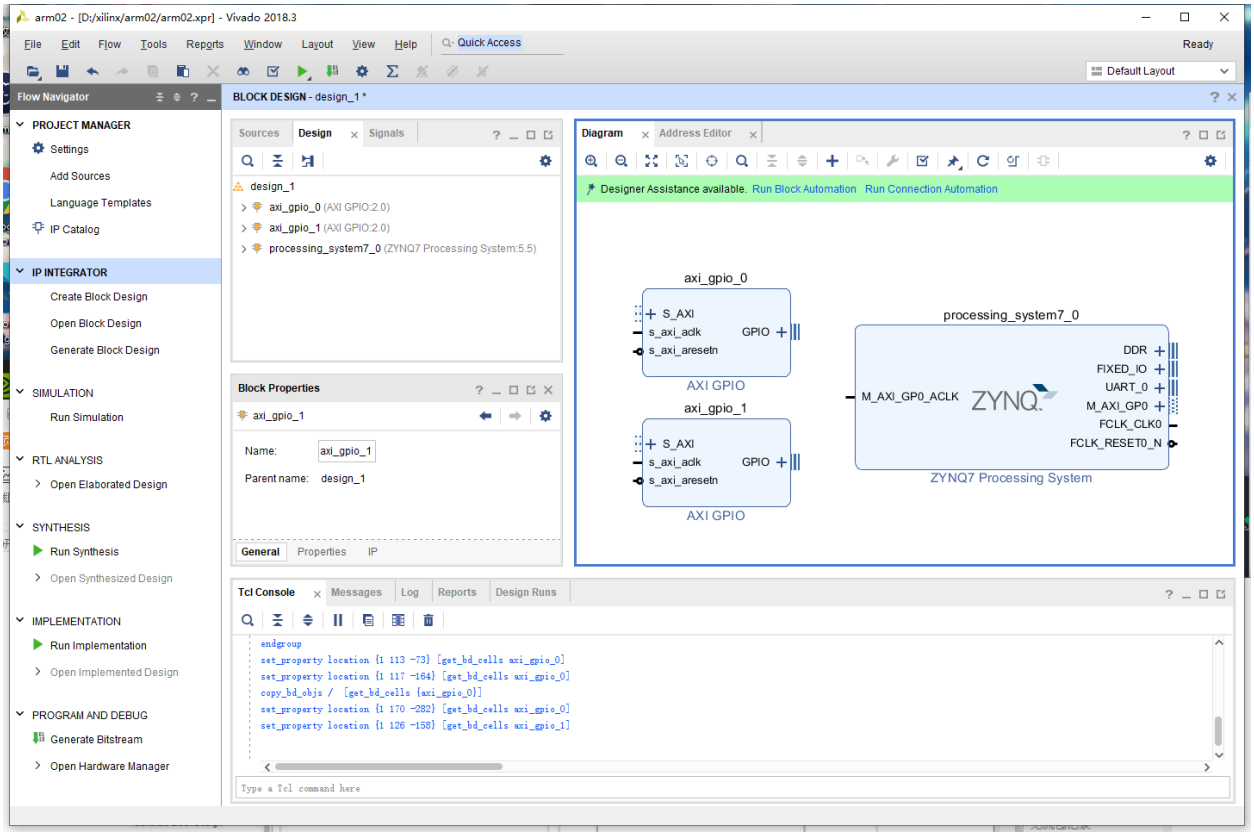
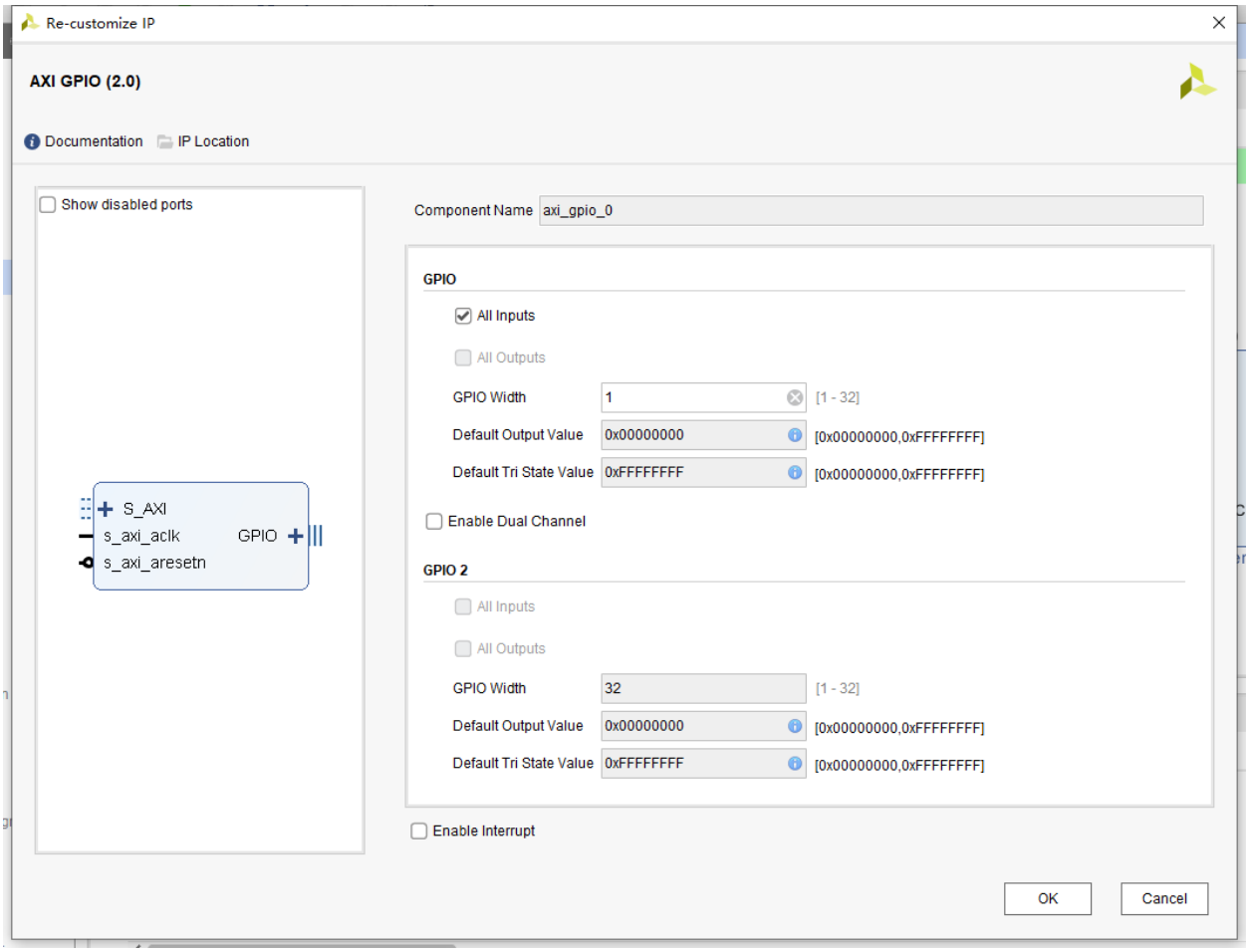
[L08]ARM GPIO使用

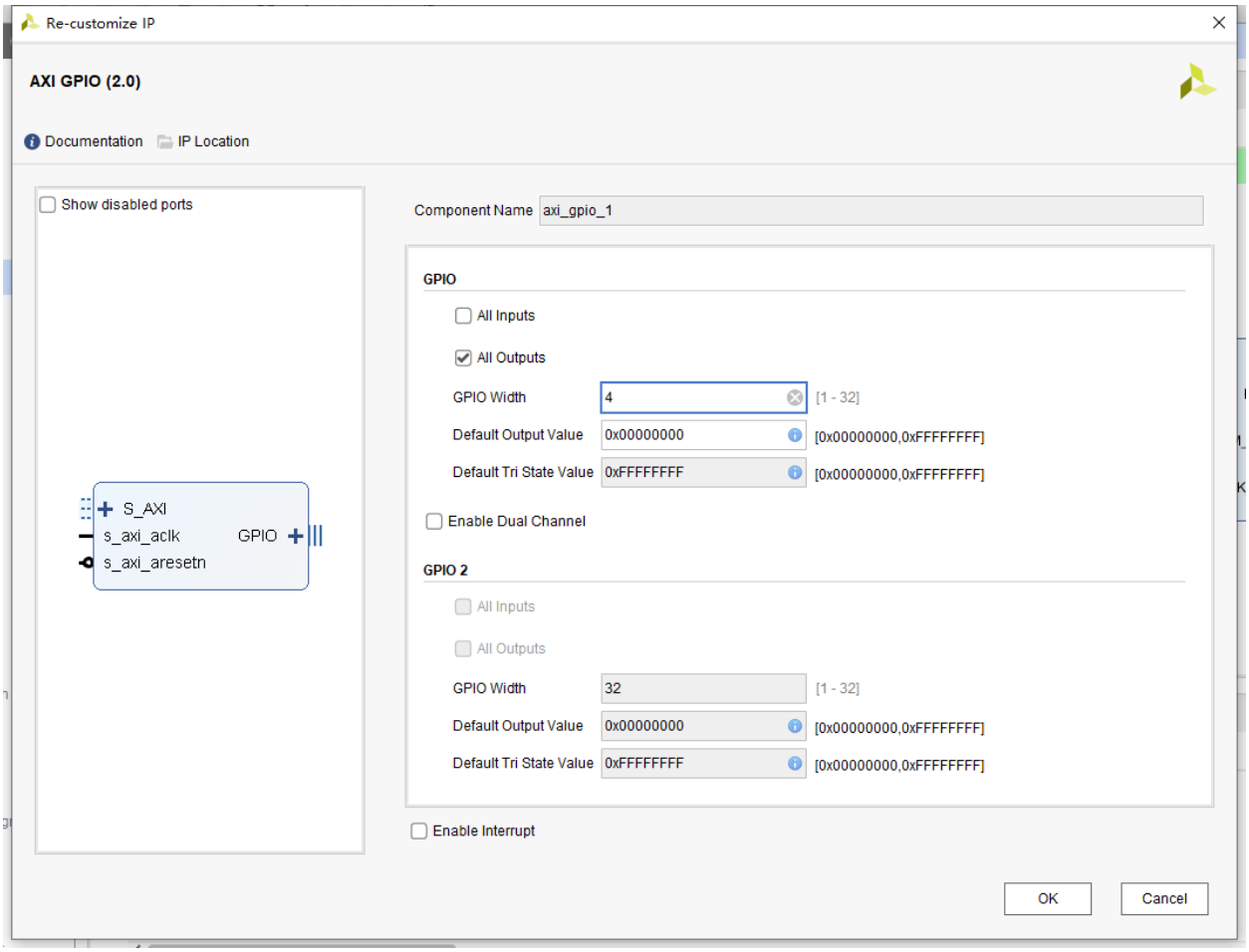
新建ZYNQ Processing System并配置好串口和DDR,不需要关闭PL部分的时钟,不需要关闭AXI各种,然后添加两个AXI GPIO模块.



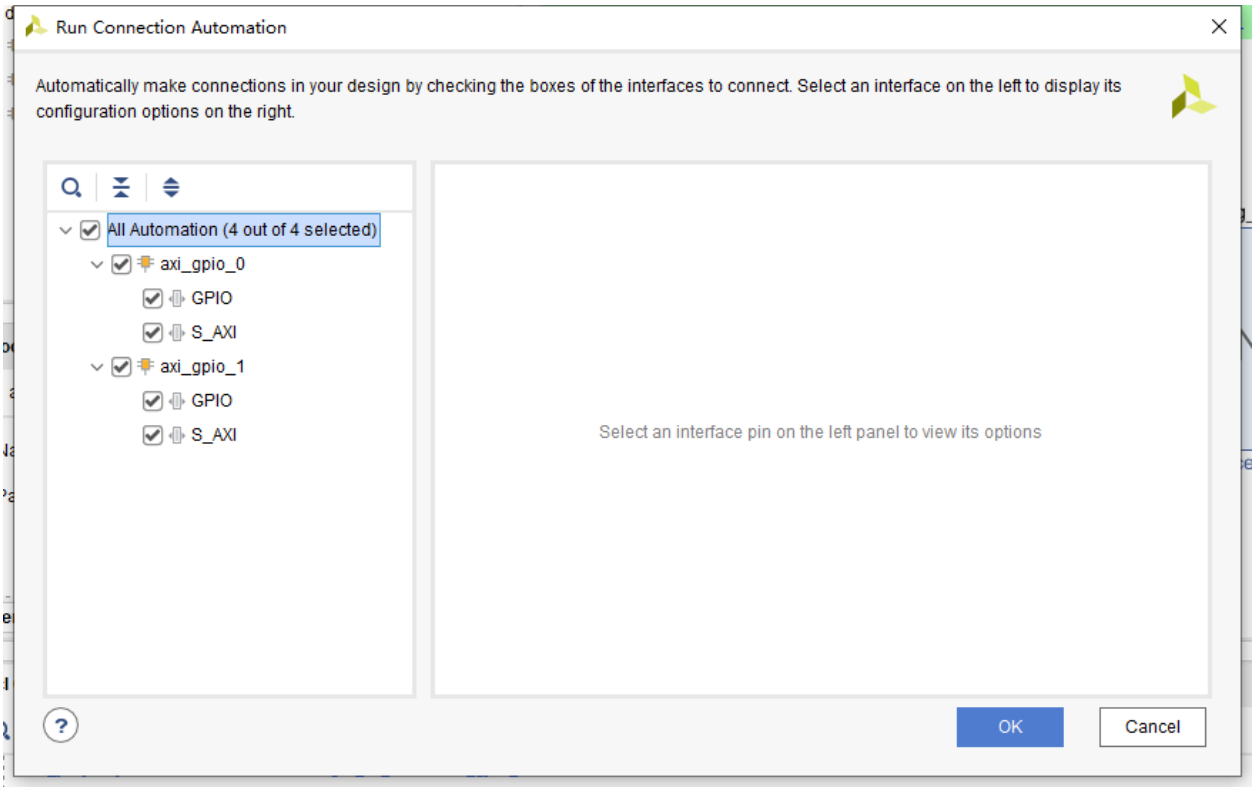
设置第一个IO是输入,位宽1.



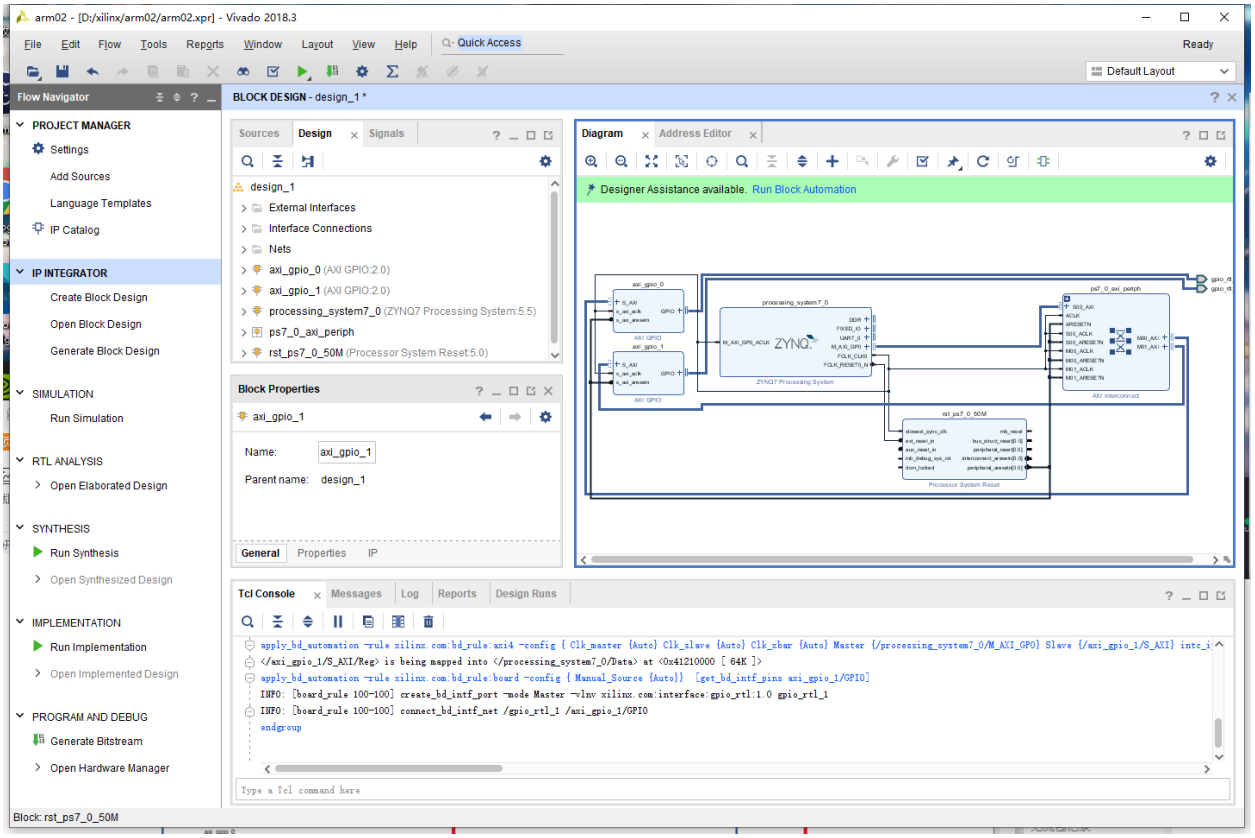
第二个设置全部输出,位宽4.



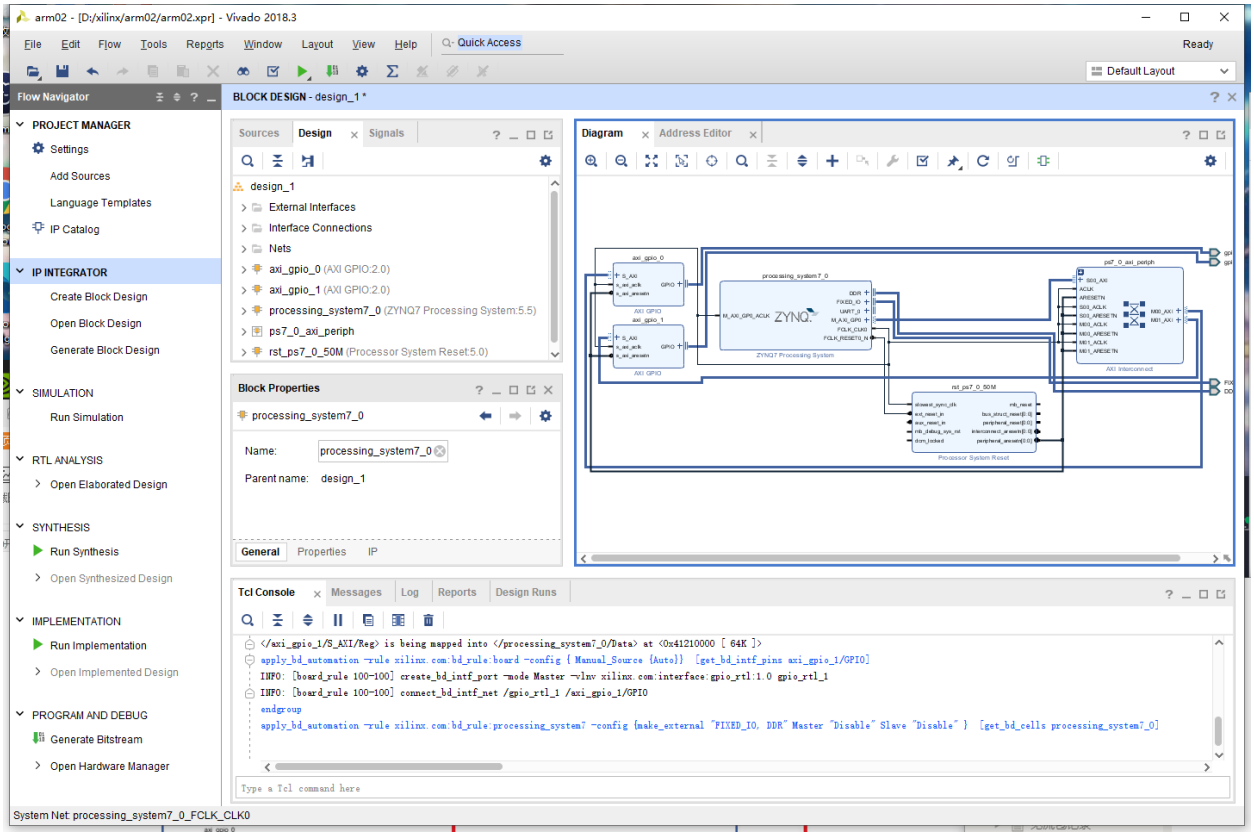
然后Run Connect Automation,全选并全部链接.


Run Connection Automation
Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.
All Automation (4 out of 4 selected)
axi_gpio_0
GPIO
S_AXI
axi_gpio_1
GPIO
S_AXI
Select an interface pin on the left panel to view its options
OK
Cancel

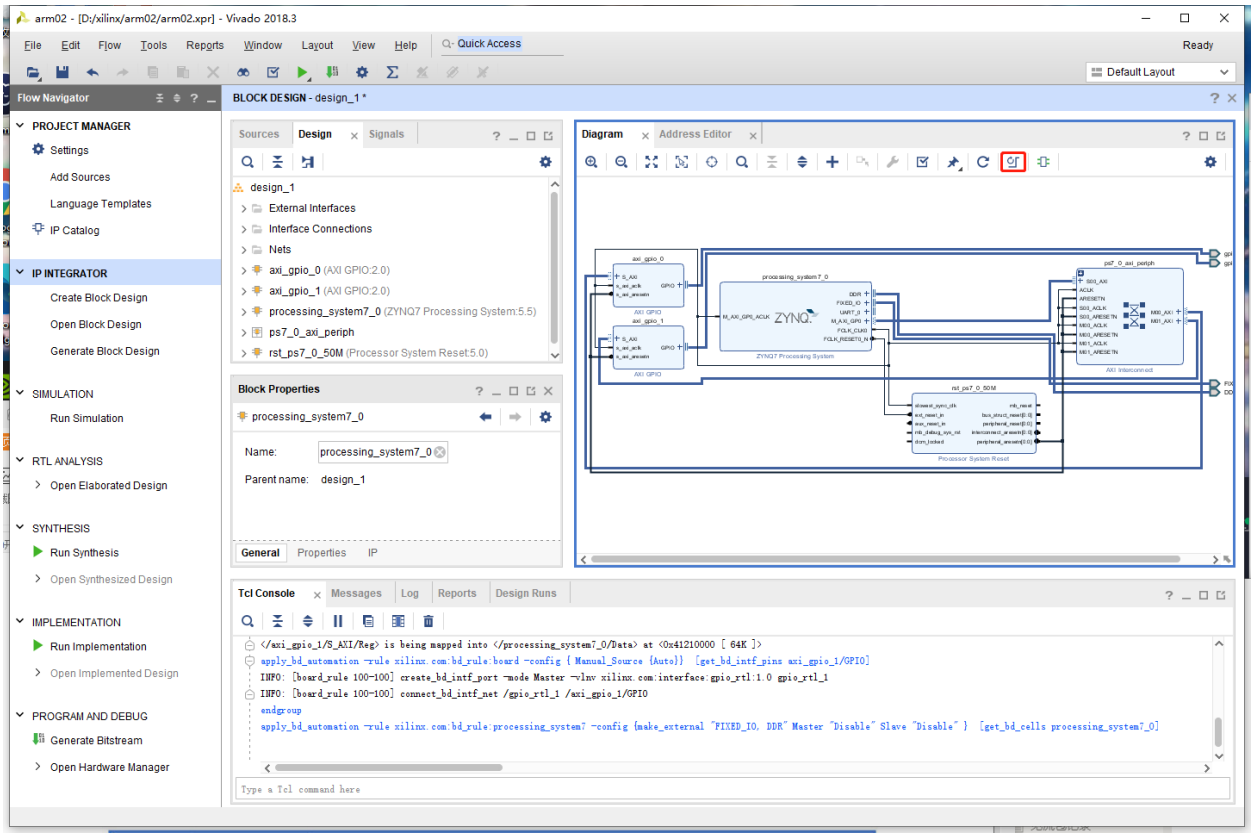
然后多了两个总线块.


arm02 - [D:/xilinx/arm02/arm02.xpr] - Vivado 2018.3
Flow Navigator
BLOCK DESIGN - design_1 *
Sources
Design
Signals
Diagram
Address Editor
Designer Assistance available. Run Block Automation
Block Properties
Tcl Console
Messages
Log
Reports
Design Runs
Block: rst_ps7_0_50M

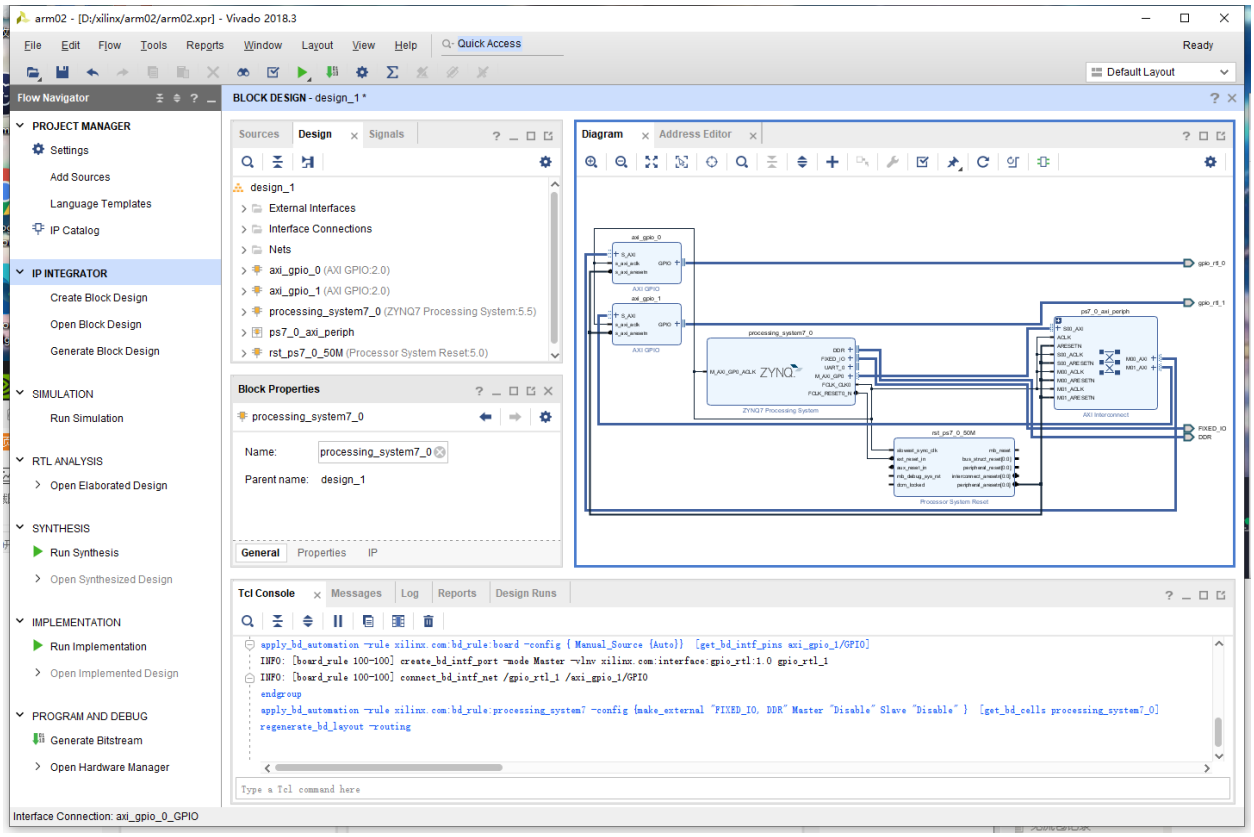
接着用Run Block Automation,导出引脚.



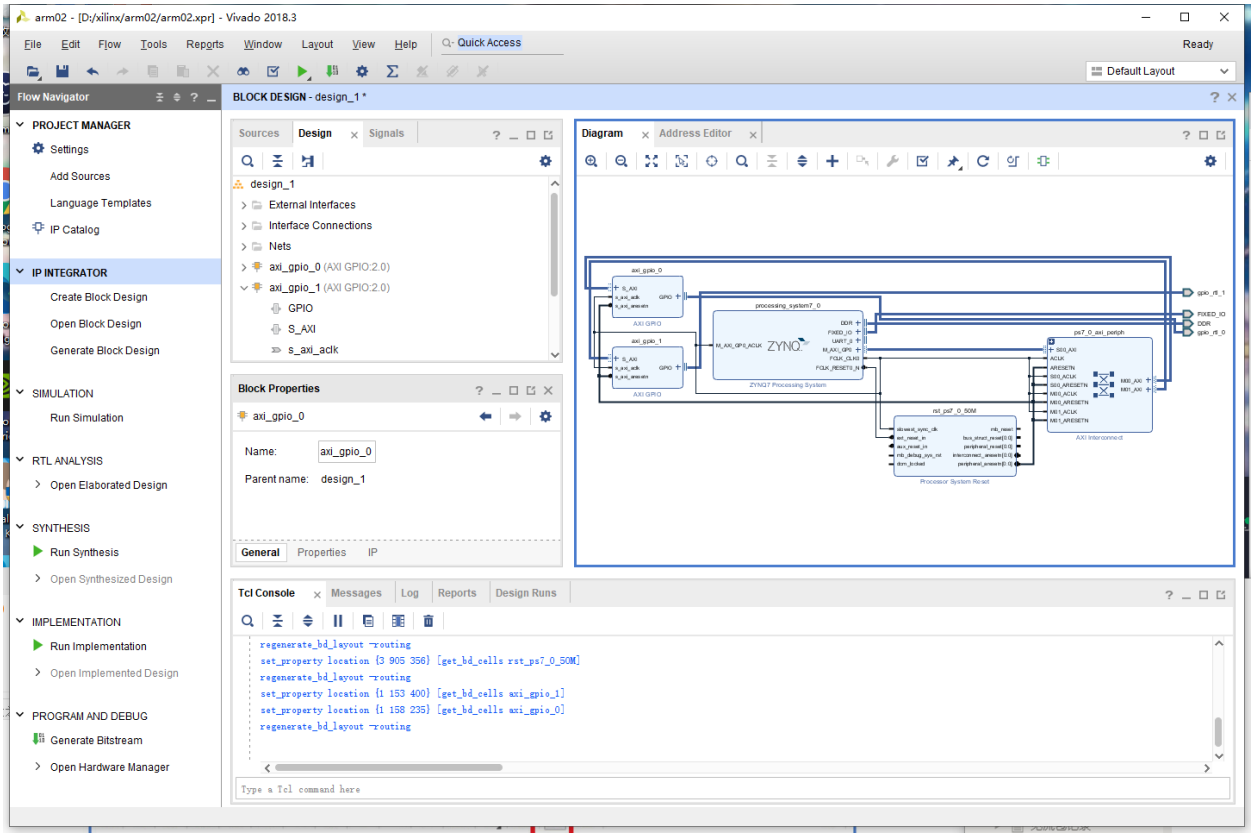
由于现在很乱,所以可以按整理按钮.


arm02 - [D:/xilinx/arm02/arm02.xpr] - Vivado 2018.3
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
BLOCK DESIGN - design_1 *
Sources
Design
Signals
design_1
External Interfaces
Interface Connections
Nets
axi_gpio_0 (AXI GPIO:2.0)
axi_gpio_1 (AXI GPIO:2.0)
processing_system7_0 (ZYNQ7 Processing System:5.5)
ps7_0_axi_periph
rst_ps7_0_50M (Processor System Reset:5.0)
Block Properties
processing_system7_0
Name: processing_system7_0
Parent name: design_1
General
Properties
IP
Diagram
Address Editor
Tcl Console
Messages
Log
Reports
Design Runs
</axi_gpio_1/S_AXI/Reg> is being mapped into </processing_system7_0/Data> at <0x41210000 [64K]>
apply_bd_automation -rule xilinx.com:bd_rule:board -config { Manual_Source {Auto}} [get_bd_intf_pins axi_gpio_1/GPIO]
INFO: [board_rule 100-100] create_bd_intf_port -mode Master -vlnv xilinx.com:interface:gpio_rtl:1.0 gpio_rtl_1
INFO: [board_rule 100-100] connect_bd_intf_net /gpio_rtl_1 /axi_gpio_1/GPIO
endgroup
apply_bd_automation -rule xilinx.com:bd_rule:processing_system7 -config {make_external "FIXED_IO, DDR" Master "Disable" Slave "Disable" } [get_bd_cells processing_system7_0]
Type a Tcl command here

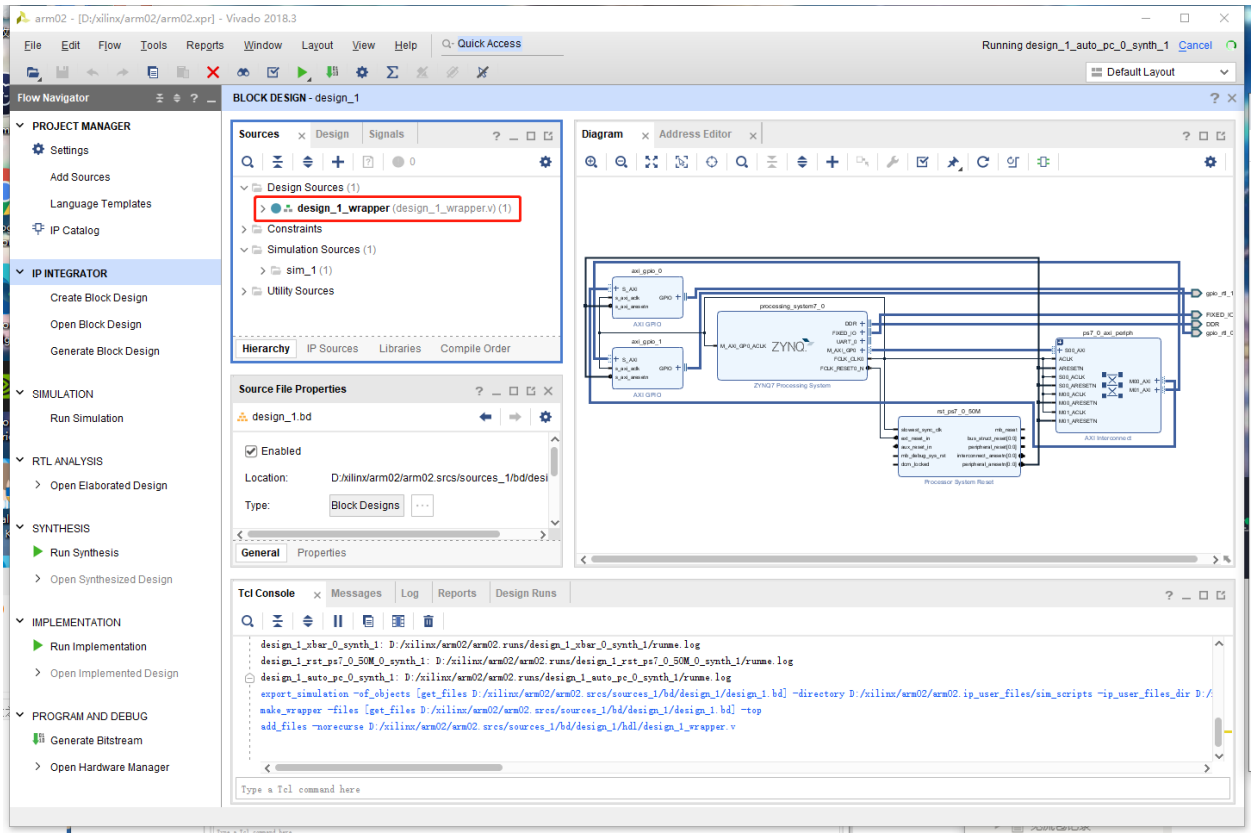
看起来是不是舒服很多.



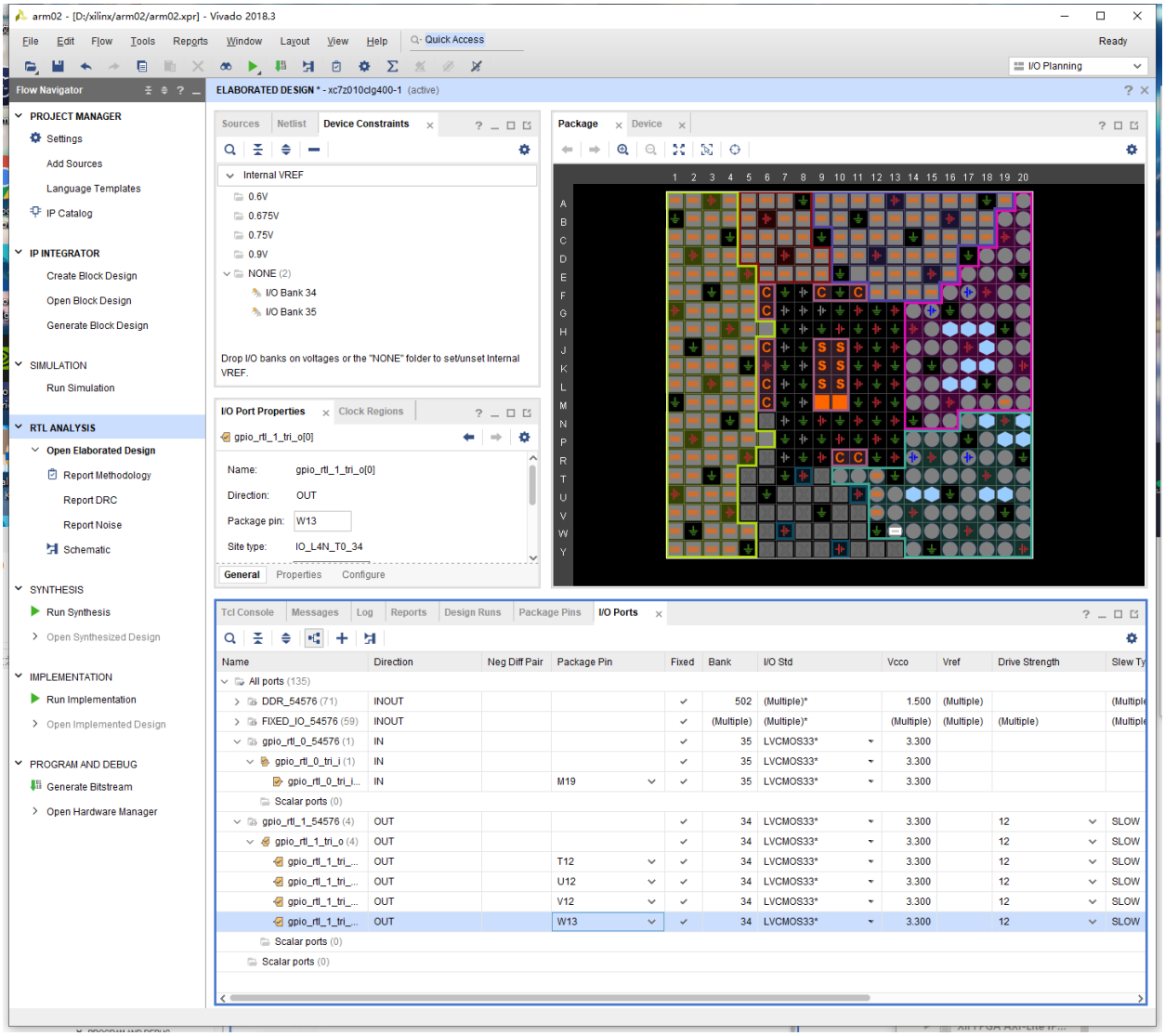
不舒服还可以继续人工整理.



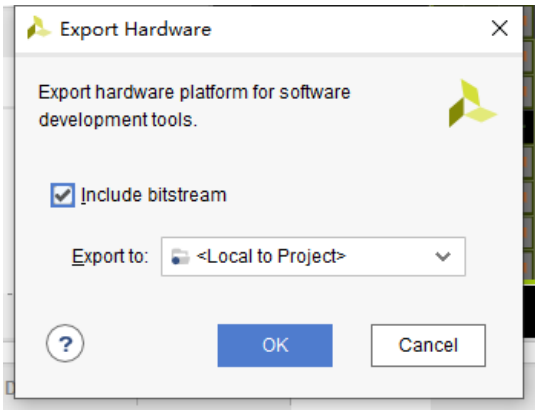
这部分完成了,那么开始Generate Outpu Products,接着Create HDL Wrapper,接着综合.


arm02 - [D:/xilinx/arm02/arm02.xpr] - Vivado 2018.3
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Running design_1_auto_pc_0_synth_1 Cancel
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
BLOCK DESIGN - design_1
Sources
Design
Signals
Design Sources (1)
design_1_wrapper (design_1_wrapper.v) (1)
Constraints
Simulation Sources (1)
sim_1 (1)
Utility Sources
Hierarchy
IP Sources
Libraries
Compile Order
Source File Properties
design_1.bd
Enabled
Location:
Type:
Block Designs
General
Properties
Diagram
Address Editor
Tcl Console
Messages
Log
Reports
Design Runs
design_1_xbar_0_synth_1: D:/xilinx/arm02/arm02.runs/design_1_xbar_0_synth_1/runme.log
design_1_rst_ps7_0_50M_0_synth_1: D:/xilinx/arm02/arm02.runs/design_1_rst_ps7_0_50M_0_synth_1/runme.log
design_1_auto_pc_0_synth_1: D:/xilinx/arm02/arm02.runs/design_1_auto_pc_0_synth_1/runme.log
export_simulation -of_objects [get_files D:/xilinx/arm02/arm02.srcs/sources_1/bd/design_1/design_1.bd] -directory D:/xilinx/arm02/arm02.ip_user_files/sim_scripts -ip_user_files_dir D:/
make_wrapper -files [get_files D:/xilinx/arm02/arm02.srcs/sources_1/bd/design_1/design_1.bd] -top
add_files -norecurse D:/xilinx/arm02/arm02.srcs/sources_1/bd/design_1/hdl/design_1_wrapper.v
Type a Tcl command here

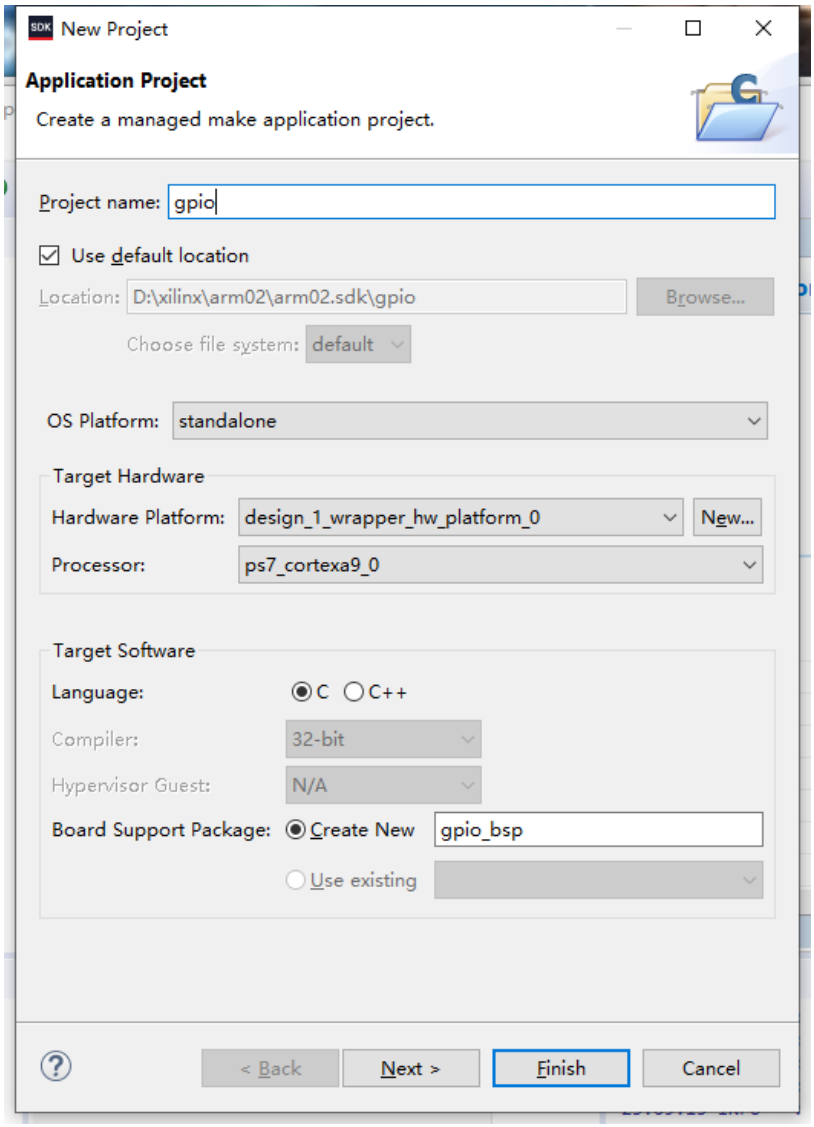
设置IO.



这下要生成bitstream了,然后导出,最后启动SDK.



创建软件工程并且继续沿用Hello World.



默认代码:

```
/******************************************************************************
*
* Copyright (C) 2009 - 2014 Xilinx, Inc.  All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* Use of the Software is limited solely to applications:
* (a) running on a Xilinx device, or
* (b) that interact with a Xilinx device through a bus or interconnect.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
```

```
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
* XILINX  BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*
* Except as contained in this notice, the name of the Xilinx shall not be used
* in advertising or otherwise to promote the sale, use or other dealings in
* this Software without prior written authorization from Xilinx.
*
******************************************************************************/

/*
 * helloworld.c: simple test application
 *
 * This application configures UART 16550 to baud rate 9600.
 * PS7 UART (Zynq) is not initialized by this application, since
 * bootrom/bsp configures it to baud rate 115200
 *
 * ------------------------------------------------
 * | UART TYPE   BAUD RATE                        |
 * ------------------------------------------------
 *   uartns550   9600
 *   uartlite    Configurable only in HW design
 *   ps7_uart    115200 (configured by bootrom/bsp)
 */

#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"


int main()
{
    init_platform();

    print("Hello World\n\r");

    cleanup_platform();
    return 0;
}
```

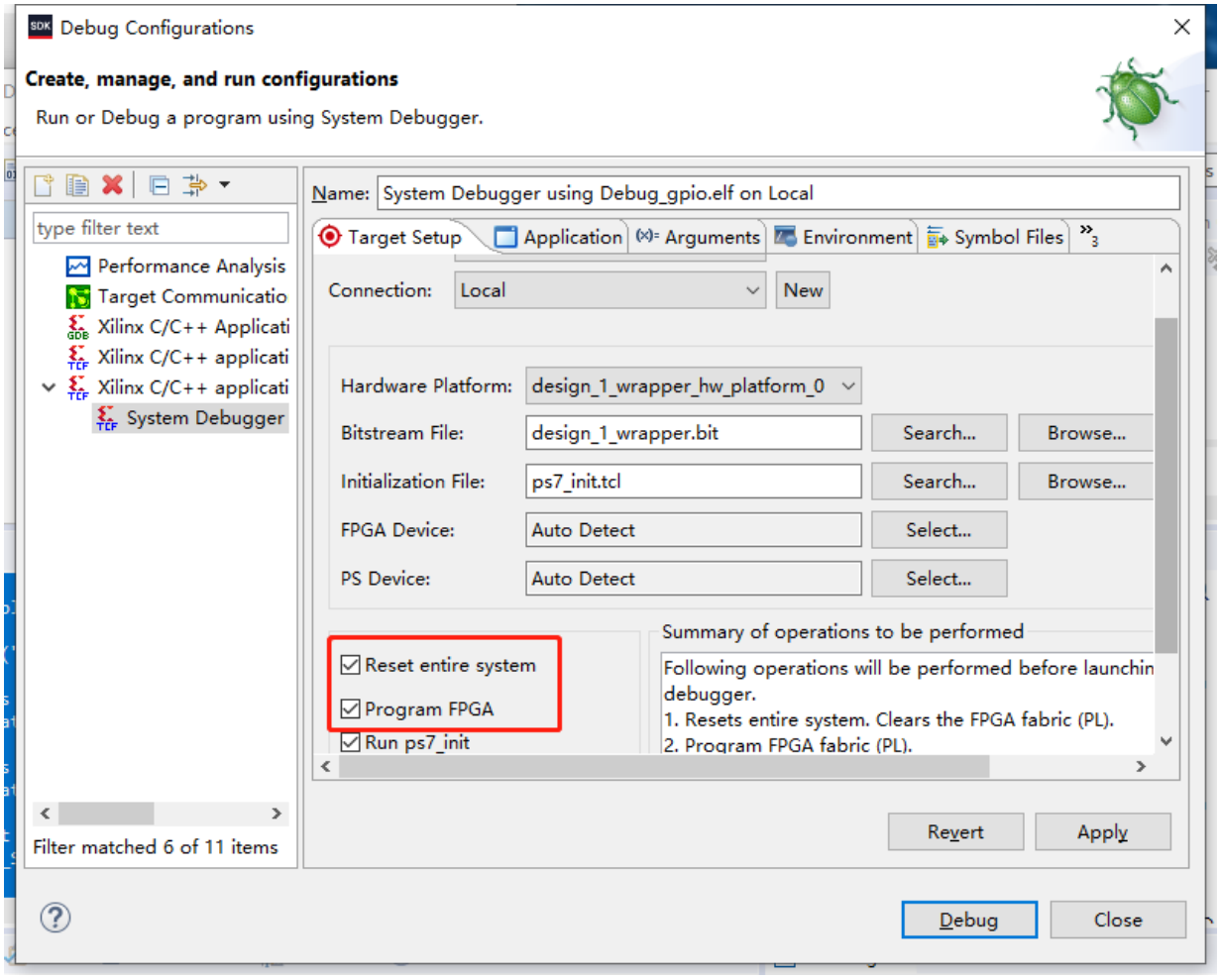
写入GPIO相关操作代码.

```
/******************************************************************************
*
* Copyright (C) 2009 - 2014 Xilinx, Inc.  All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* Use of the Software is limited solely to applications:
* (a) running on a Xilinx device, or
* (b) that interact with a Xilinx device through a bus or interconnect.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
* XILINX  BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
```

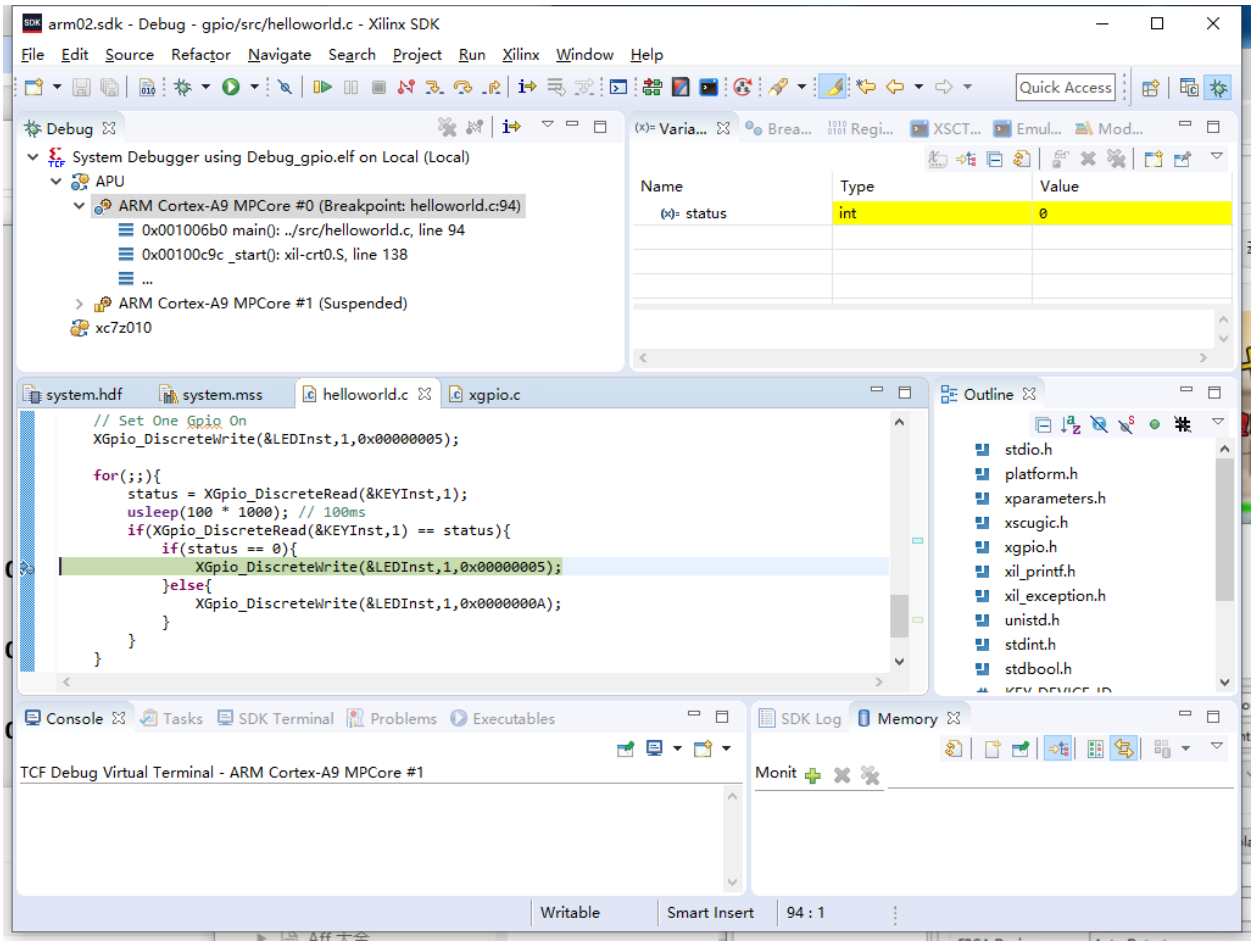
```
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*
* Except as contained in this notice, the name of the Xilinx shall not be used
* in advertising or otherwise to promote the sale, use or other dealings in
* this Software without prior written authorization from Xilinx.
*
******************************************************************************/

/*
 * helloworld.c: simple test application
 *
 * This application configures UART 16550 to baud rate 9600.
 * PS7 UART (Zynq) is not initialized by this application, since
 * bootrom/bsp configures it to baud rate 115200
 *
 * ------------------------------------------------
 * | UART TYPE   BAUD RATE                        |
 * ------------------------------------------------
 *   uartns550   9600
 *   uartlite    Configurable only in HW design
 *   ps7_uart    115200 (configured by bootrom/bsp)
 */

#include <stdio.h>
#include "platform.h"
#include "xparameters.h"
#include "xscugic.h"
#include "xgpio.h"
#include "xil_printf.h"
#include "xil_exception.h"

#include <unistd.h>
#include <stdint.h>
#include <stdbool.h>

#define KEY_DEVICE_ID XPAR_AXI_GPIO_0_DEVICE_ID
#define LED_DEVICE_ID XPAR_AXI_GPIO_1_DEVICE_ID

XGpio KEYInst;
XGpio LEDInst;

int main()
{
  int status;

    init_platform();

    print("Hello World\n\r");

    status = XGpio_Initialize(&KEYInst,KEY_DEVICE_ID);
    if(status != XST_SUCCESS) return XST_FAILURE;

    status = XGpio_Initialize(&LEDInst,LED_DEVICE_ID);
    if(status != XST_SUCCESS) return XST_FAILURE;

    // Set as Input
    XGpio_SetDataDirection(&KEYInst,1,0x00000000);

    // Set as Output
    XGpio_SetDataDirection(&LEDInst,1,0x0000000F);

    // Set One Gpio On
    XGpio_DiscreteWrite(&LEDInst,1,0x00000005);

    for(;;){
      status = XGpio_DiscreteRead(&KEYInst,1);
      usleep(100 * 1000); // 100ms
      if(XGpio_DiscreteRead(&KEYInst,1) == status){
```

```
        if(status == 0){
            XGpio_DiscreteWrite(&LEDInst,1,0x00000005);
        }else{
            XGpio_DiscreteWrite(&LEDInst,1,0x0000000A);
        }
      }
    }

    cleanup_platform();
    return 0;
}
```

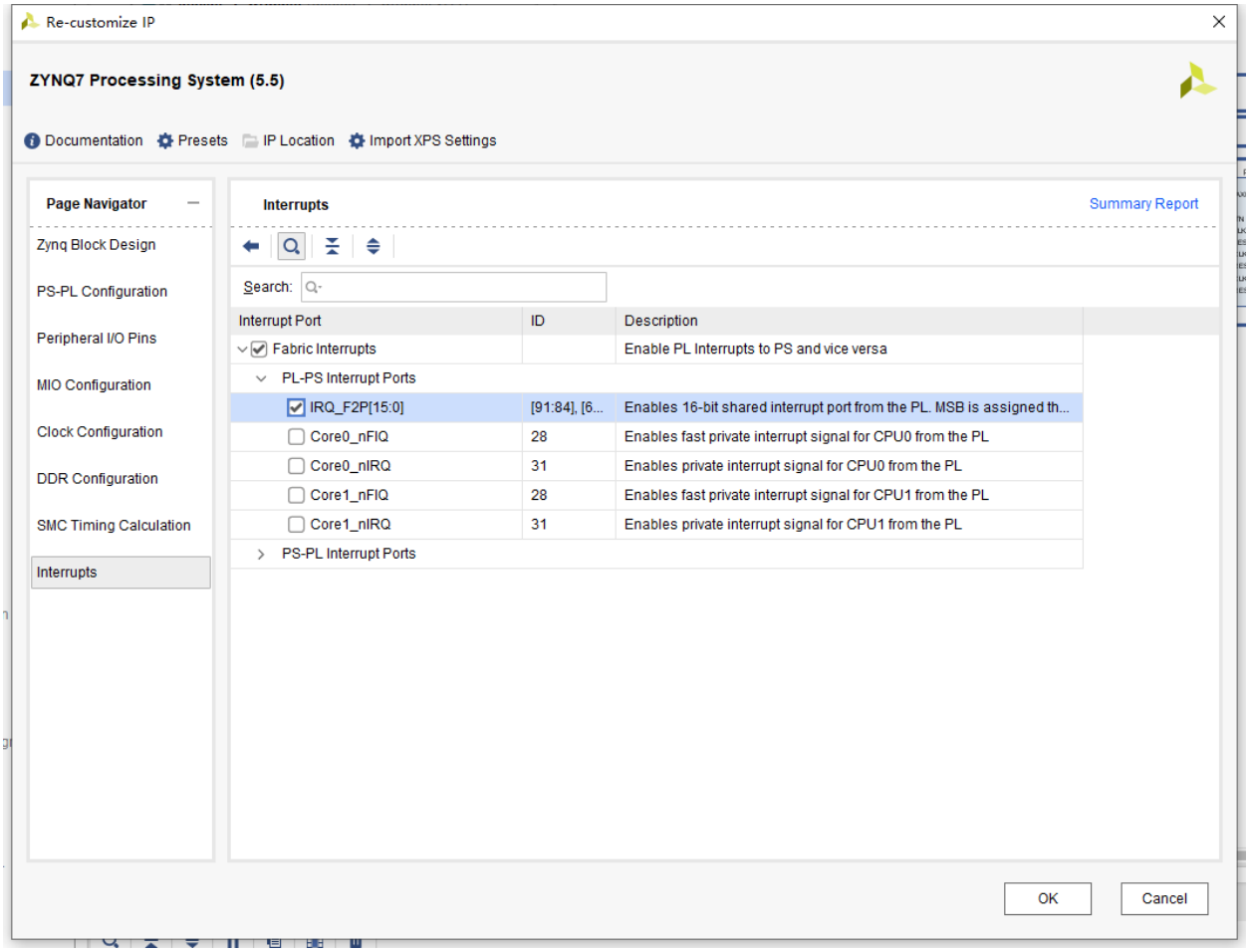
由于要下载bitstream所以不能直接调试,我们可以在SDK直接下载,调试配置里把编程FPGA勾上.



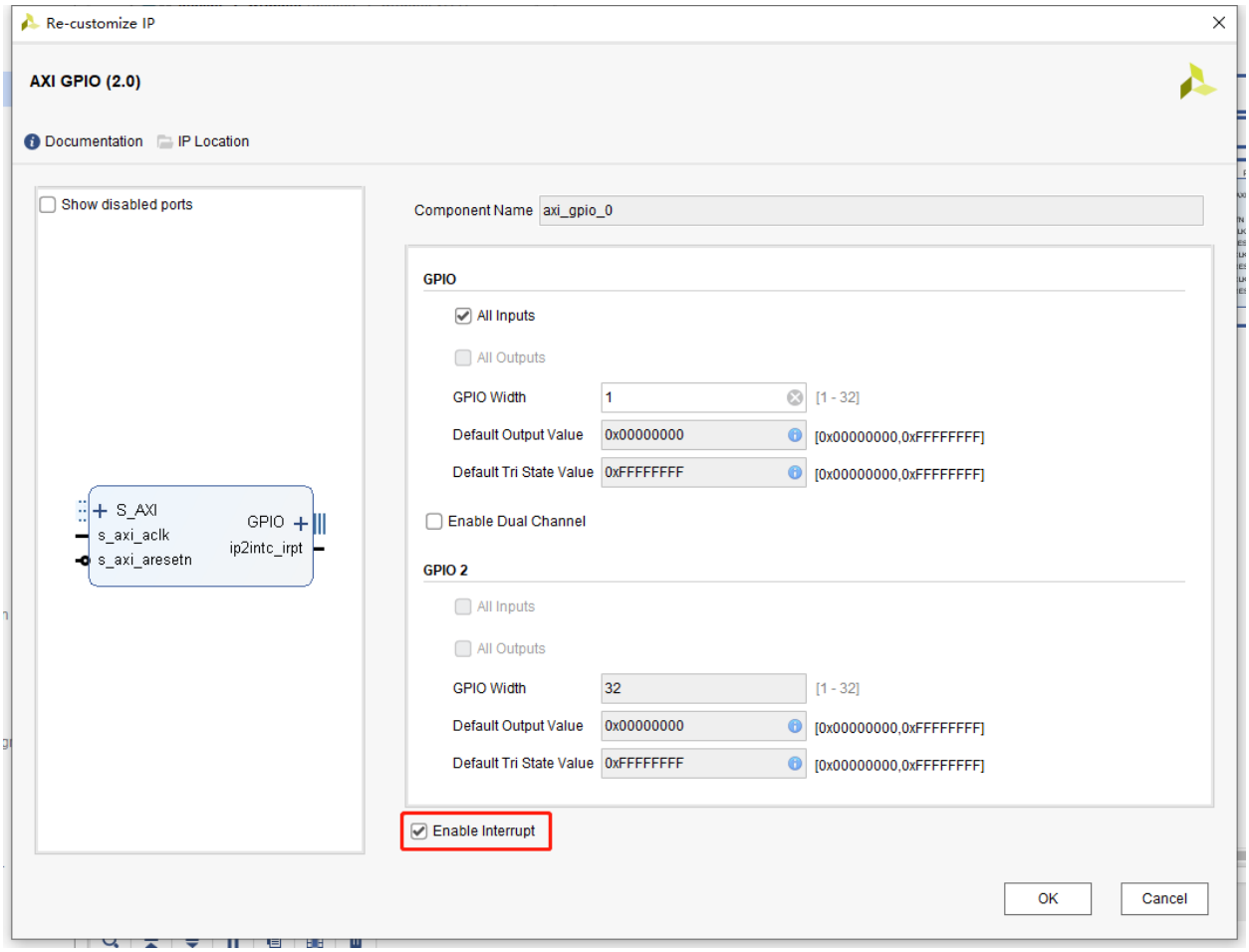
下载调试后,按一下按键,切换LED.



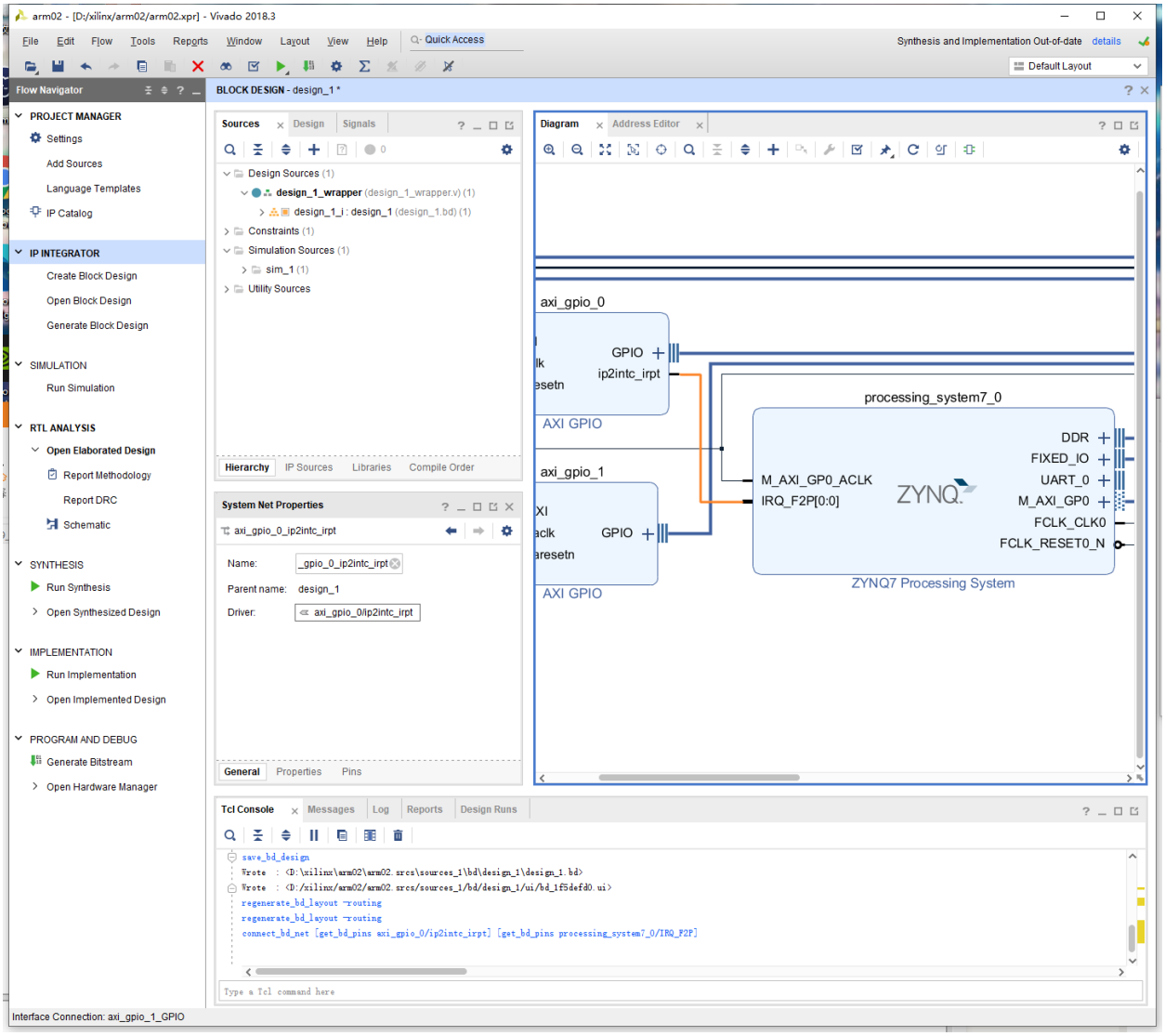
现在尝试做一个中断,设置Block Design中的ZYNQ Processing System的中断.



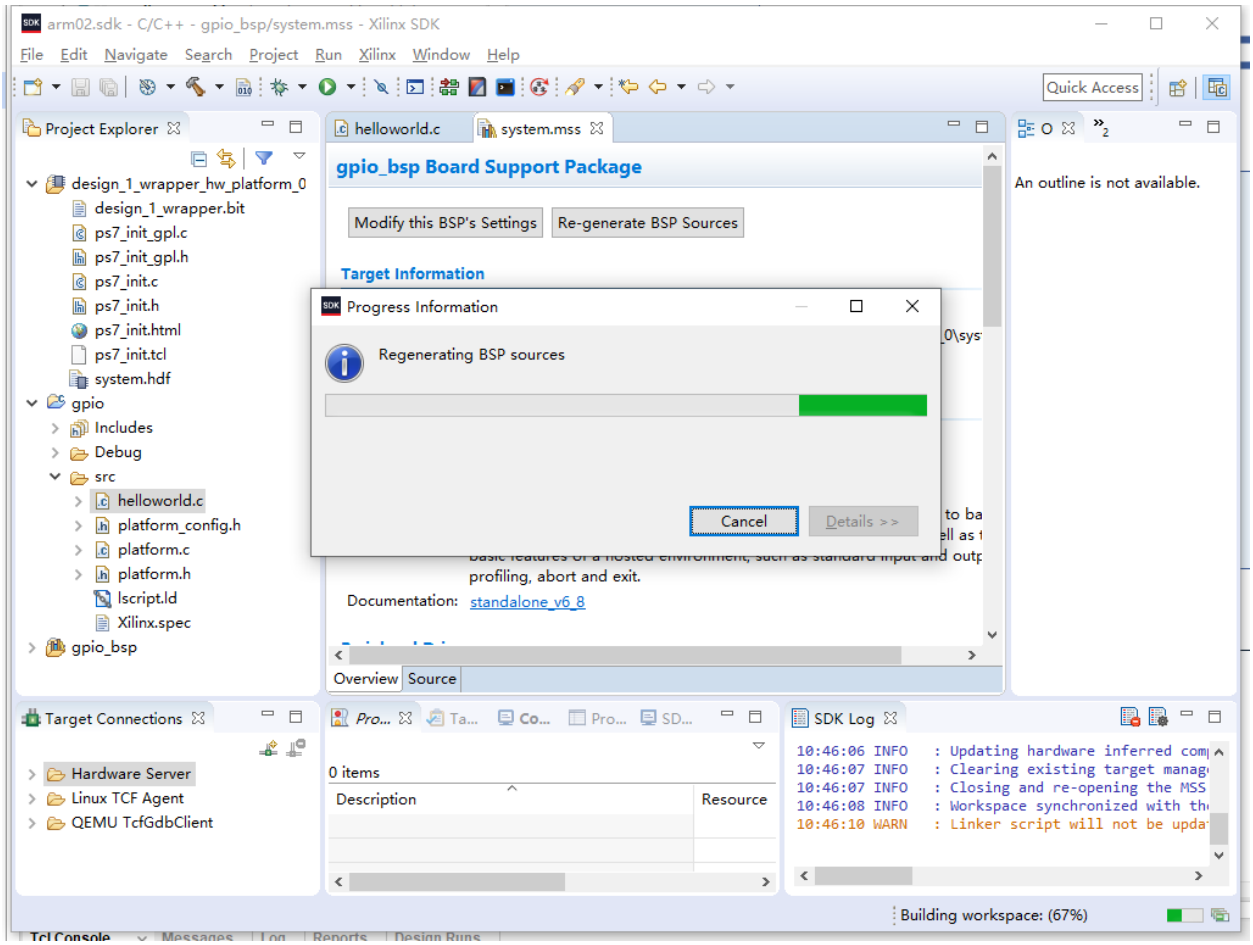
给输入IP设置一个中断允许.


Re-customize IP
AXI GPIO (2.0)
Documentation
IP Location
Show disabled ports
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
ip2intc_irpt
Component Name
axi_gpio_0
GPIO
All Inputs
All Outputs
GPIO Width
1
[1 - 32]
Default Output Value
0x00000000
[0x00000000,0xFFFFFFFF]
Default Tri State Value
0xFFFFFFFF
[0x00000000,0xFFFFFFFF]
Enable Dual Channel
GPIO 2
All Inputs
All Outputs
GPIO Width
32
[1 - 32]
Default Output Value
0x00000000
[0x00000000,0xFFFFFFFF]
Default Tri State Value
0xFFFFFFFF
[0x00000000,0xFFFFFFFF]
Enable Interrupt
OK
Cancel

人工连接一下IRQ.



然后重新Generate Outpu Product,Create HDL Wrapper,Run Synthesis,Generate Birstream,Export SDK,Lauch SDK,这些应该很熟悉了就不多说了,可能还需要重新生成BSP文件.



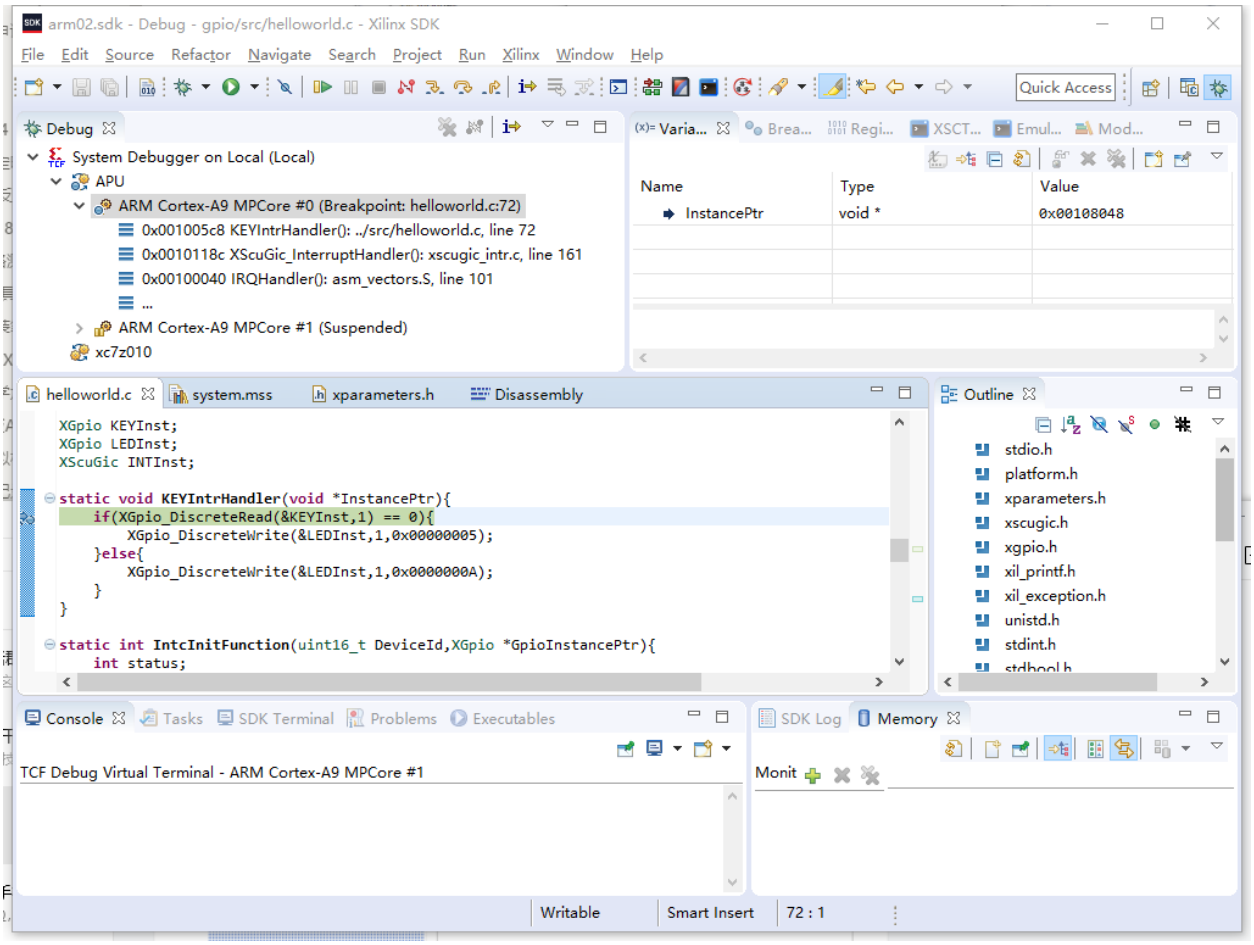
然后修改代码:

```
/******************************************************************************
*
* Copyright (C) 2009 - 2014 Xilinx, Inc.  All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* Use of the Software is limited solely to applications:
* (a) running on a Xilinx device, or
* (b) that interact with a Xilinx device through a bus or interconnect.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
* XILINX  BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*
* Except as contained in this notice, the name of the Xilinx shall not be used
```

```
* in advertising or otherwise to promote the sale, use or other dealings in
* this Software without prior written authorization from Xilinx.
*
******************************************************************************/

/*
 * helloworld.c: simple test application
 *
 * This application configures UART 16550 to baud rate 9600.
 * PS7 UART (Zynq) is not initialized by this application, since
 * bootrom/bsp configures it to baud rate 115200
 *
 * ------------------------------------------------
 * | UART TYPE   BAUD RATE                        |
 * ------------------------------------------------
 *   uartns550   9600
 *   uartlite    Configurable only in HW design
 *   ps7_uart    115200 (configured by bootrom/bsp)
 */

#include <stdio.h>
#include "platform.h"
#include "xparameters.h"
#include "xscugic.h"
#include "xgpio.h"
#include "xil_printf.h"
#include "xil_exception.h"

#include <unistd.h>
#include <stdint.h>
#include <stdbool.h>

#define INTC_DEVICE_ID XPAR_PS7_SCUGIC_0_DEVICE_ID
#define KEY_DEVICE_ID XPAR_AXI_GPIO_0_DEVICE_ID
#define LED_DEVICE_ID XPAR_AXI_GPIO_1_DEVICE_ID

#define INTC_GPIO_INTERRUPT_ID XPAR_FABRIC_AXI_GPIO_0_IP2INTC_IRPT_INTR
#define KEY_INT_MASK XGPIO_IR_CH1_MASK

XGpio KEYInst;
XGpio LEDInst;
XScuGic INTInst;

static void KEYIntrHandler(void *InstancePtr){
  if(XGpio_DiscreteRead(&KEYInst,1) == 0){
    XGpio_DiscreteWrite(&LEDInst,1,0x00000005);
  }else{
    XGpio_DiscreteWrite(&LEDInst,1,0x0000000A);
  }
}

static int IntcInitFunction(uint16_t DeviceId,XGpio *GpioInstancePtr){
  int status;
  XScuGic_Config *IntcConfig;

  IntcConfig = XScuGic_LookupConfig(DeviceId);
  status = XScuGic_CfgInitialize(&INTInst,IntcConfig,IntcConfig->CpuBaseAddress);

  if(status != XST_SUCCESS) return XST_FAILURE;

  Xil_ExceptionRegisterHandler(XIL_EXCEPTION_ID_INT,(Xil_ExceptionHandler)XScuGic_InterruptHandler,&INTInst);
  Xil_ExceptionEnable();

  status = XScuGic_Connect(&INTInst,INTC_GPIO_INTERRUPT_ID,(Xil_ExceptionHandler)KEYIntrHandler,(void *)GpioInstancePtr);

  if(status != XST_SUCCESS) return XST_FAILURE;

  XGpio_InterruptEnable(GpioInstancePtr,1);
  XGpio_InterruptGlobalEnable(GpioInstancePtr);
  XScuGic_Enable(&INTInst,INTC_GPIO_INTERRUPT_ID);
```

```
  return XST_SUCCESS;
}

int main()
{
  int status;

    init_platform();

    print("Hello World\n\r");

    status = XGpio_Initialize(&KEYInst,KEY_DEVICE_ID);
    if(status != XST_SUCCESS) return XST_FAILURE;

    status = XGpio_Initialize(&LEDInst,LED_DEVICE_ID);
    if(status != XST_SUCCESS) return XST_FAILURE;

    // Set as Input
    XGpio_SetDataDirection(&KEYInst,1,0x00000000);

    // Set as Output
    XGpio_SetDataDirection(&LEDInst,1,0x0000000F);

    // Set One Gpio On
    XGpio_DiscreteWrite(&LEDInst,1,0x00000005);

    status = IntcInitFunction(INTC_DEVICE_ID,&KEYInst);
    if(status != XST_SUCCESS) return XST_FAILURE;

    for(;;){
    }

    cleanup_platform();
    return 0;
}
```

尝试触发中断.


arm02.sdk - Debug - gpio/src/helloworld.c - Xilinx SDK
File Edit Source Refactor Navigate Search Project Run Xilinx Window Help
Quick Access
Debug
System Debugger on Local (Local)
APU
ARM Cortex-A9 MPCore #0 (Breakpoint: helloworld.c:72)
0x001005c8 KEYIntrHandler(): ../src/helloworld.c, line 72
0x0010118c XScuGic_InterruptHandler(): xscugic_intr.c, line 161
0x00100040 IRQHandler(): asm_vectors.S, line 101
...
ARM Cortex-A9 MPCore #1 (Suspended)
xc7z010
Name
Type
Value
InstancePtr
void *
0x00108048
helloworld.c
system.mss
xparameters.h
Disassembly
XGpio KEYInst;
XGpio LEDInst;
XScuGic INTInst;
static void KEYIntrHandler(void *InstancePtr){
if(XGpio_DiscreteRead(&KEYInst,1) == 0){
XGpio_DiscreteWrite(&LEDInst,1,0x00000005);
}else{
XGpio_DiscreteWrite(&LEDInst,1,0x0000000A);
}
}
static int IntcInitFunction(uint16_t DeviceId,XGpio *GpioInstancePtr){
int status;
Outline
stdio.h
platform.h
xparameters.h
xscugic.h
xgpio.h
xil_printf.h
xil_exception.h
unistd.h
stdint.h
Console
Tasks
SDK Terminal
Problems
Executables
TCF Debug Virtual Terminal - ARM Cortex-A9 MPCore #1
SDK Log
Memory
Monit
Writable
Smart Insert
72 : 1