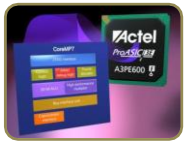
FPGA开发基础公益培训（第4讲）

感受SOPC的强大魅力

个人新浪微博：lucky_mao

本文档所述内容仅代表个人观点，仅供学习交流使用，请勿用于商业用途

本文档所涉及参考资料均源于互联网和个人总结，如有侵权请及时与我联系，以做更正

主要内容

- ➢SOPC的基本概念与解决方案
- ➢SOPC的软硬件协同设计流程
- ➢SOPC的系统级设计实例讲解

SOPC的基本概念与解决方案

SOC: System On Chip

从系统的角度进行功能的设计，将目标系统的处理机制、模型算法、芯片结构、各层次电路、直至器件的设计紧密结合起来，在单个芯片上完成整个系统的功能。

尽管传统ASIC面临发展的困境，但是SOC仍然是最主流的嵌入式系统解决方案的实现平台哦~

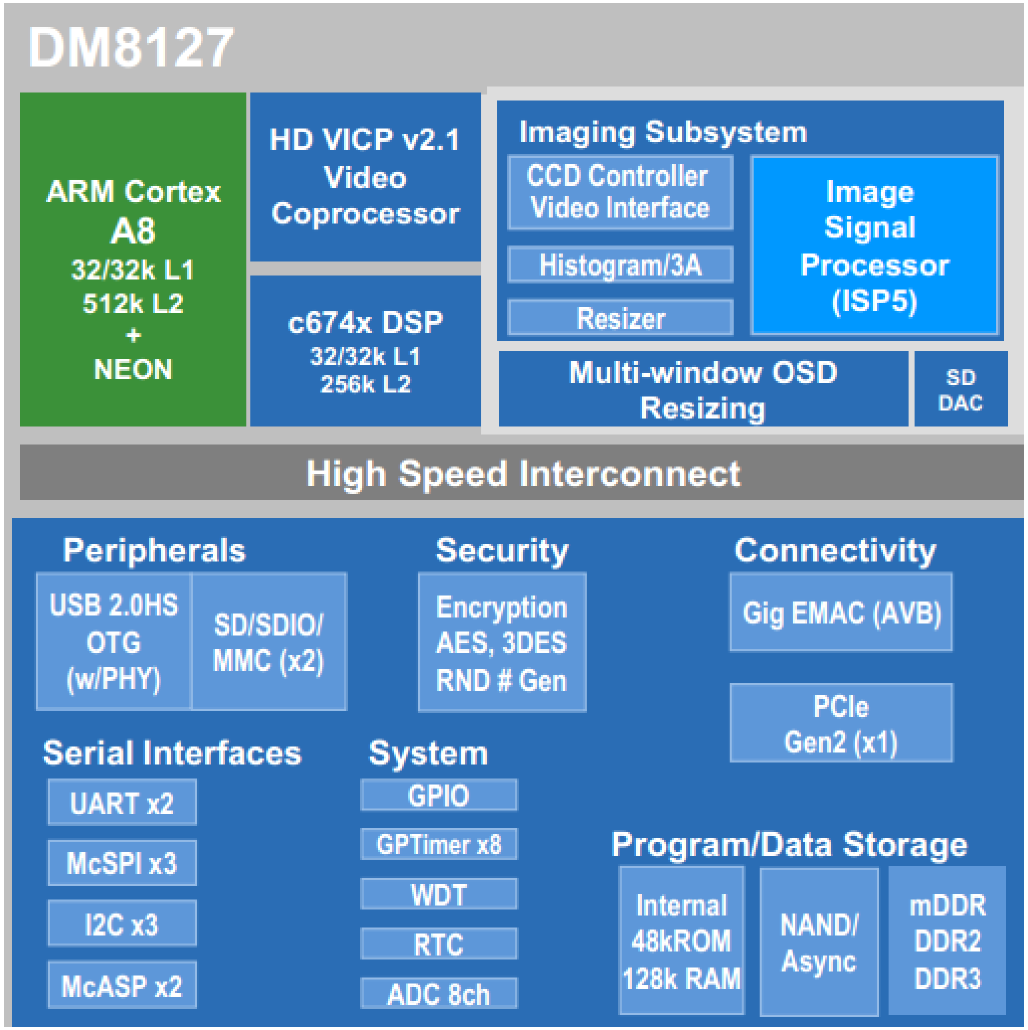
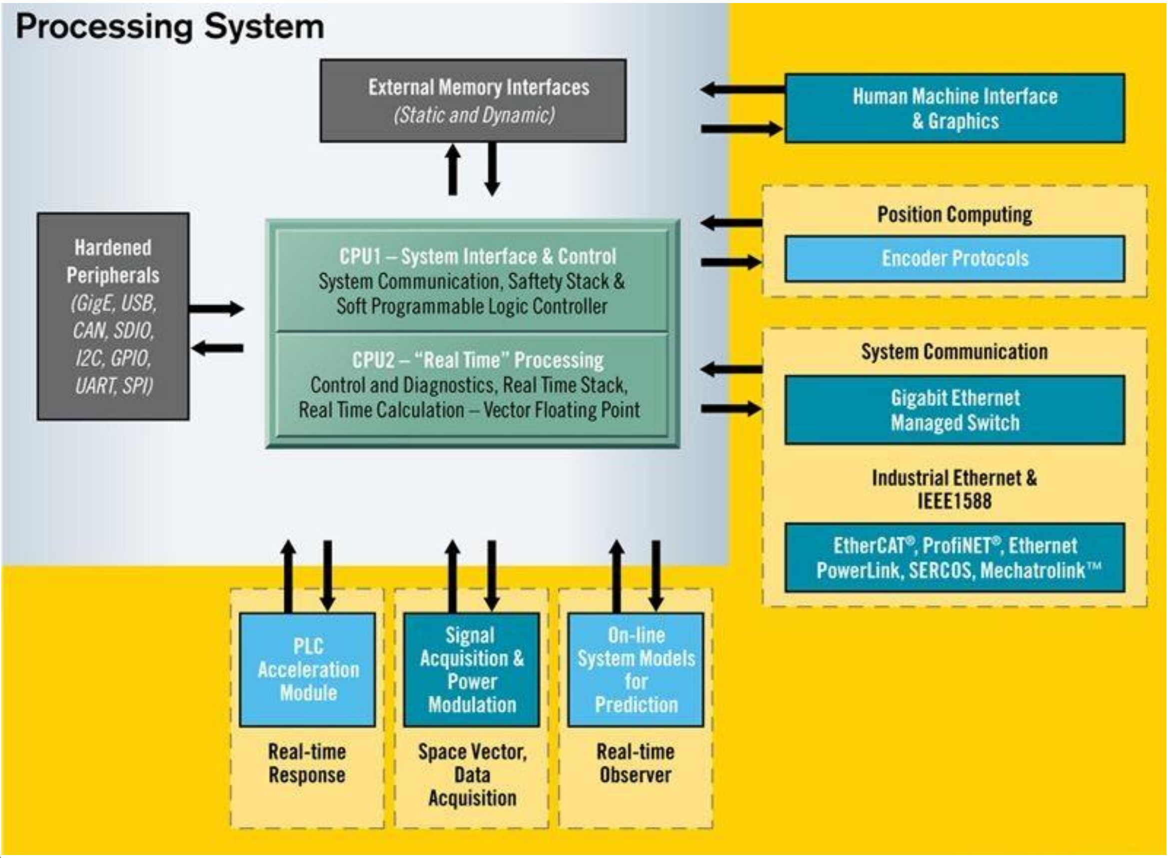


图 TI DM8127 SOC内部结构示意图

SOPC的基本概念与解决方案

SOPC: System on a Programmable Chip

SOPC 是PLD 和ASIC 技术融合的结果。集成了硬核或软核CPU、DSP、存储器、外围I/O 及可编程逻辑的SOPC 芯片在应用的灵活性和价格上有极大的优势。

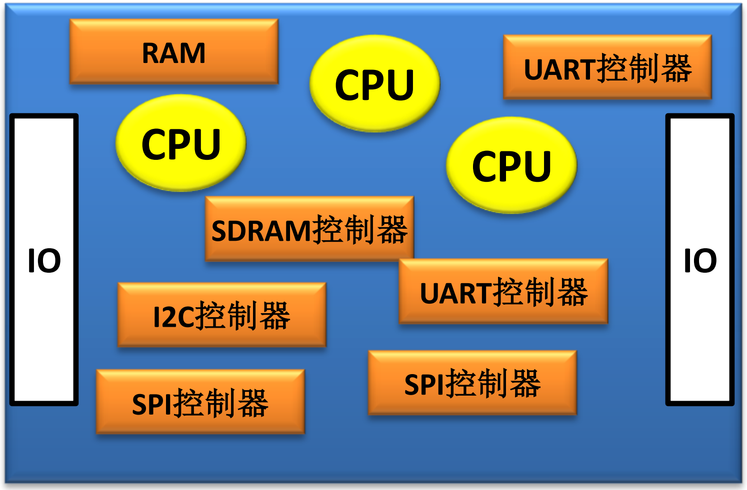


有人认为SOPC 代表了半导体产业未来发展的方向!

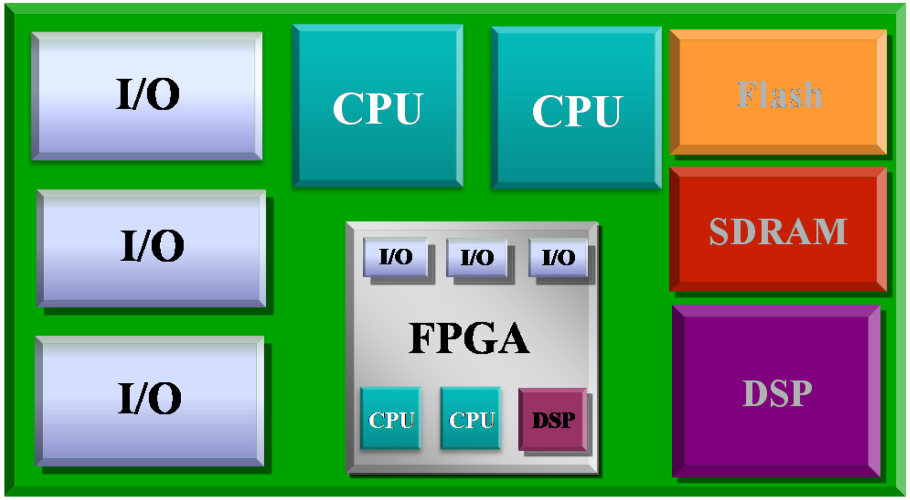
SOPC = FPGA + CPU + DSP

SOPC的基本概念与解决方案

SOPC的优势

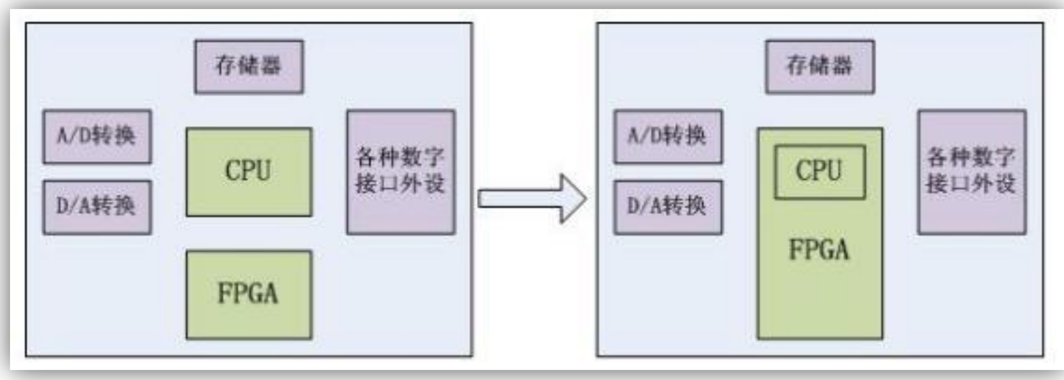


实现按需定制的片上系统

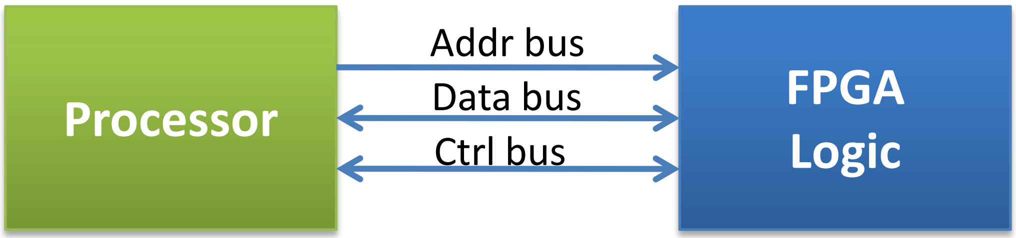


实现灵活精简的板上系统

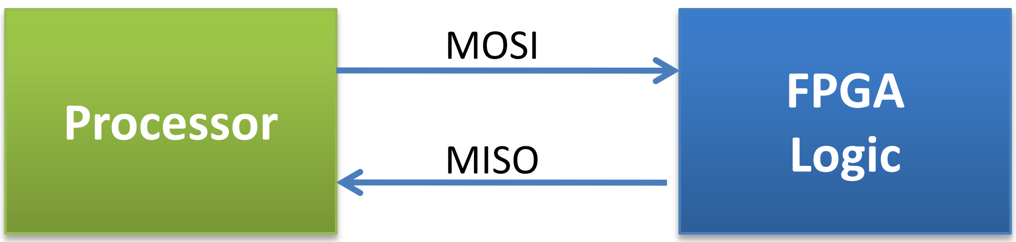
SOPC让复杂灵活的系统设计变得游刃有余，使系统简约而不简单。



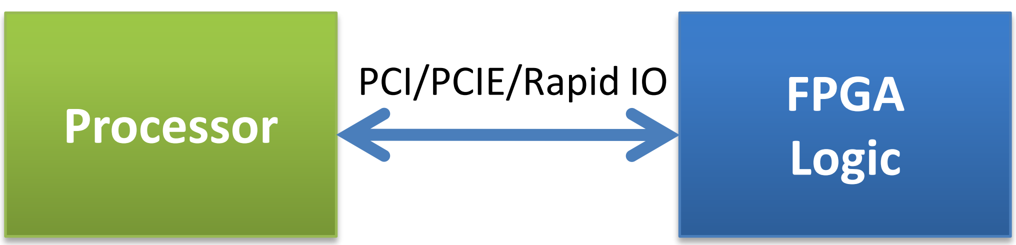
FPGA的互联解决方案



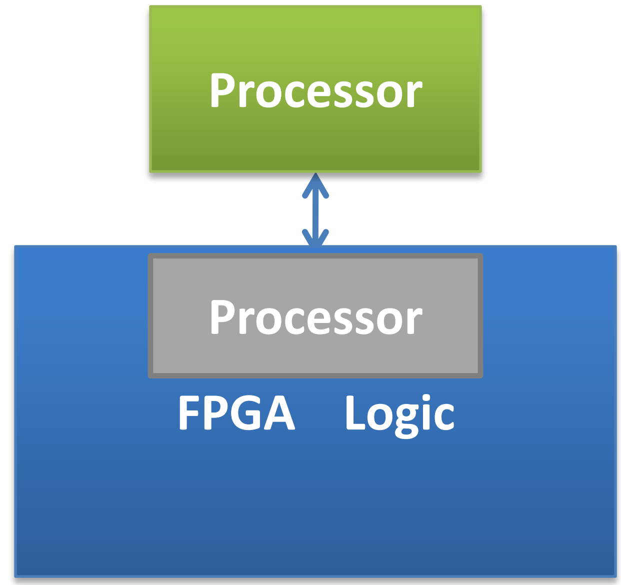
处理器并行总线互联



低速串行总线互联



高速串行总线互联



灵活且低成本的片内互联

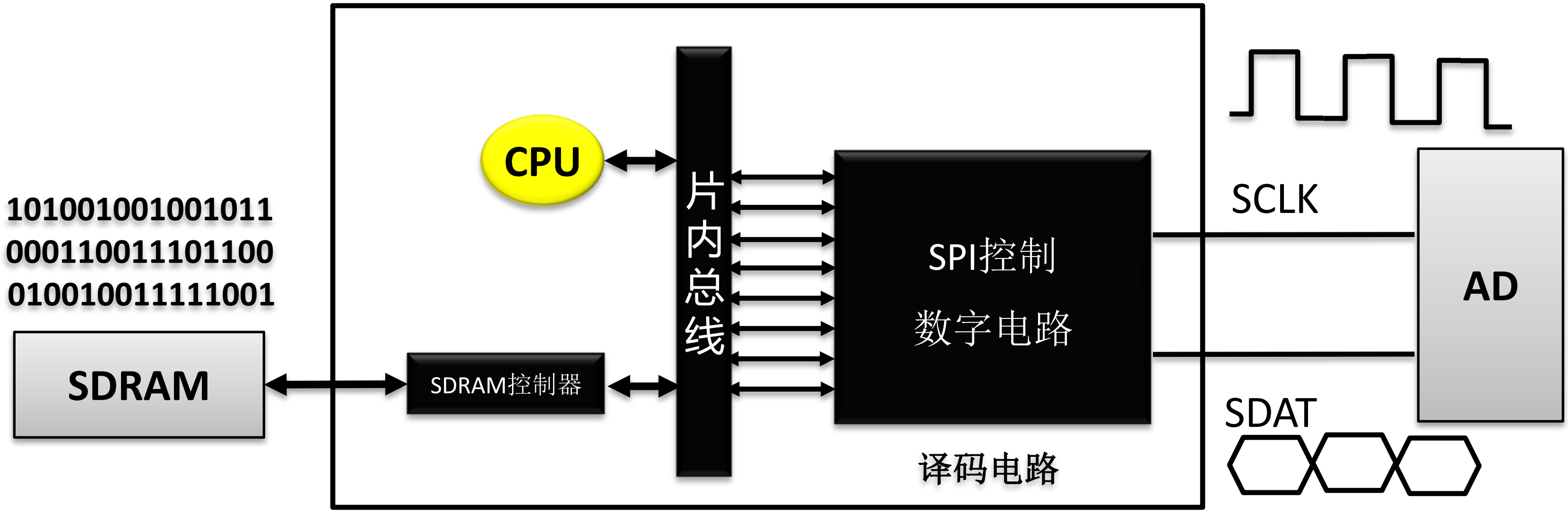
SOPC的基本概念与解决方案

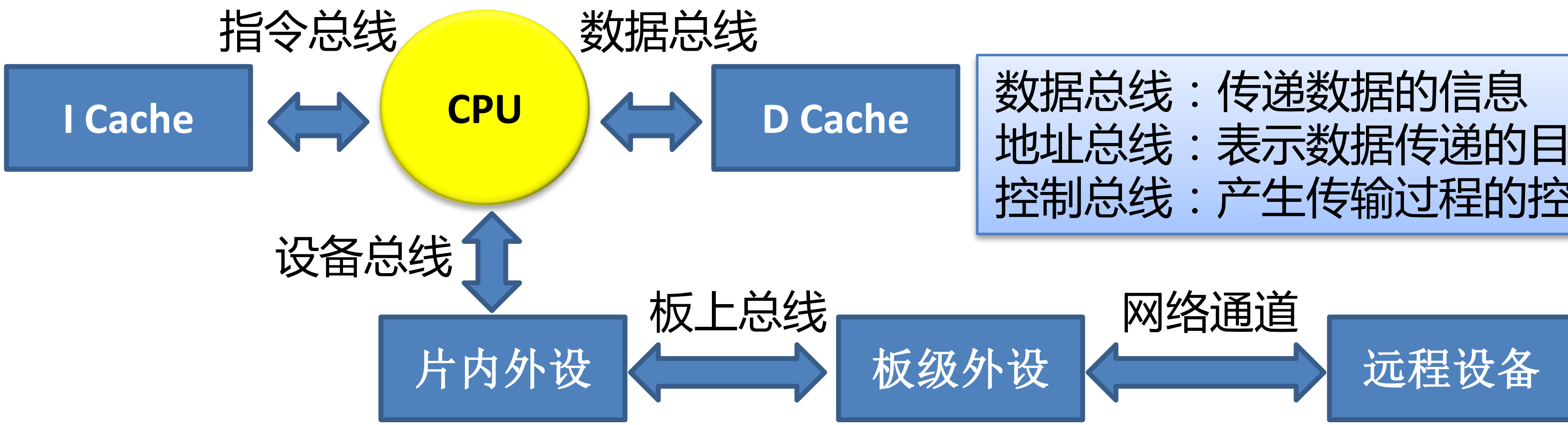
FPGA厂商的解决方案

- ◆Altera：NIOS II, ARM, MIPS, …
- ◆Xilinx：PicoBlaze, MicroBlaze,ARM, Power PC,…
- ◆Lattice：Mico32,…
- ◆Actel：LEON3, CoreABC, ARM,…

软件可编程的实现方法

编写软件指令驱动AD设备

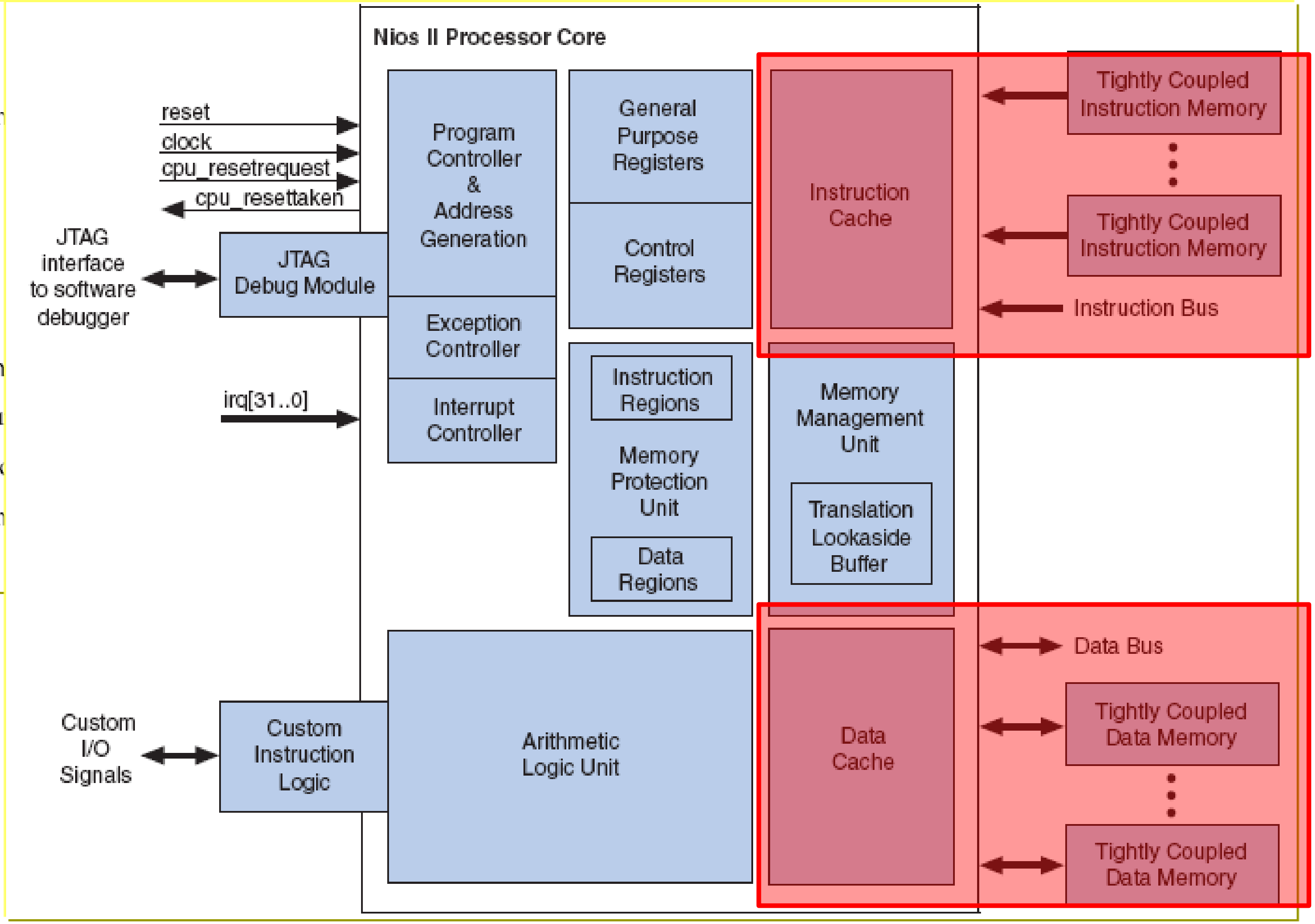




数据总线：传递数据的信息
地址总线：表示数据传递的目的地
控制总线：产生传输过程的控制协议

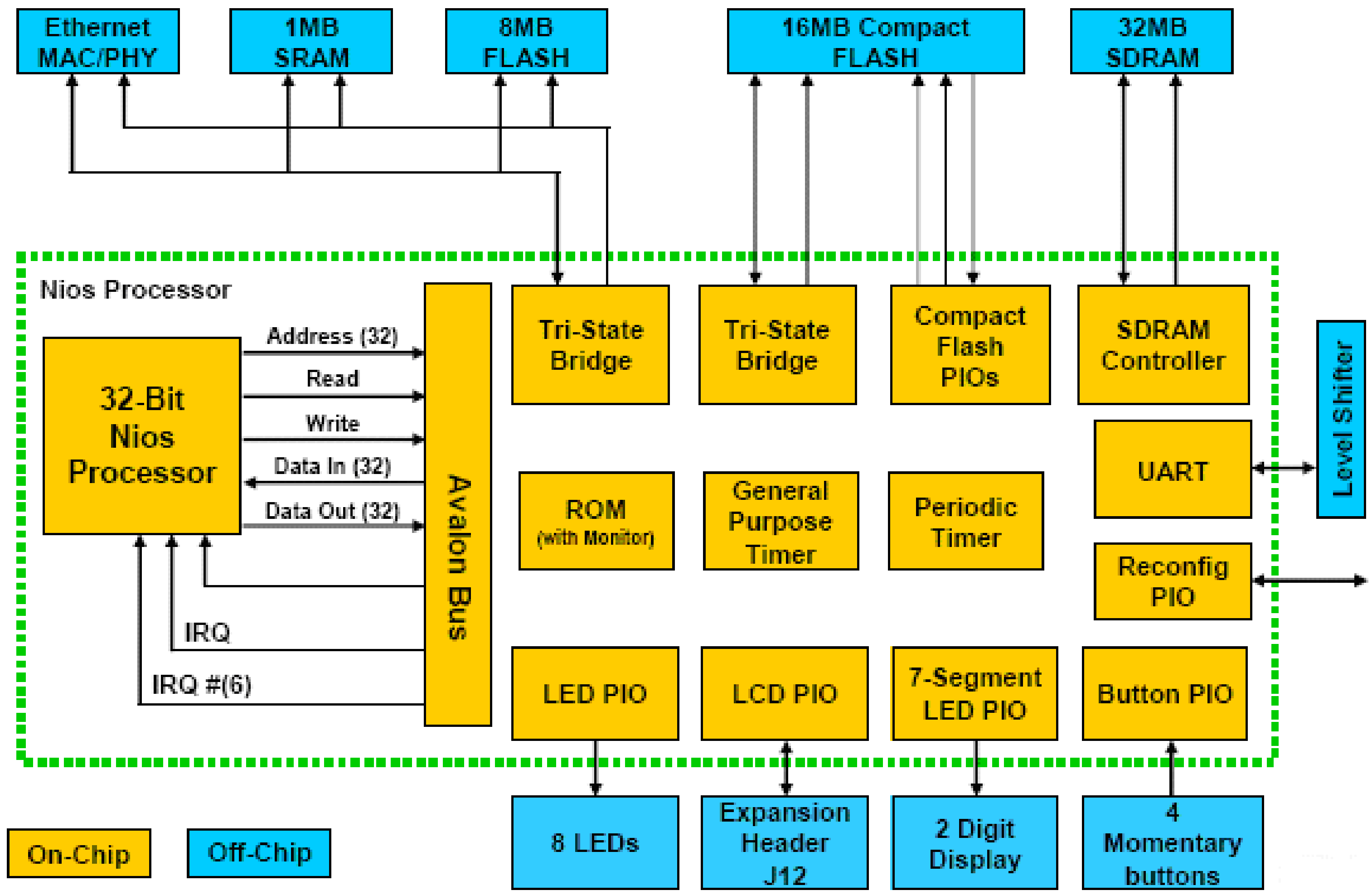
SOPC的基本概念与解决方案

- Register file
- Arithmetic logic unit
- Interface to custom ir
- Exception controller
- Interrupt controller
- Instruction bus
- Data bus
- Memory managemen
- Memory protection u
- Instruction and data c
- Tightly-coupled mem
- JTAG debug module



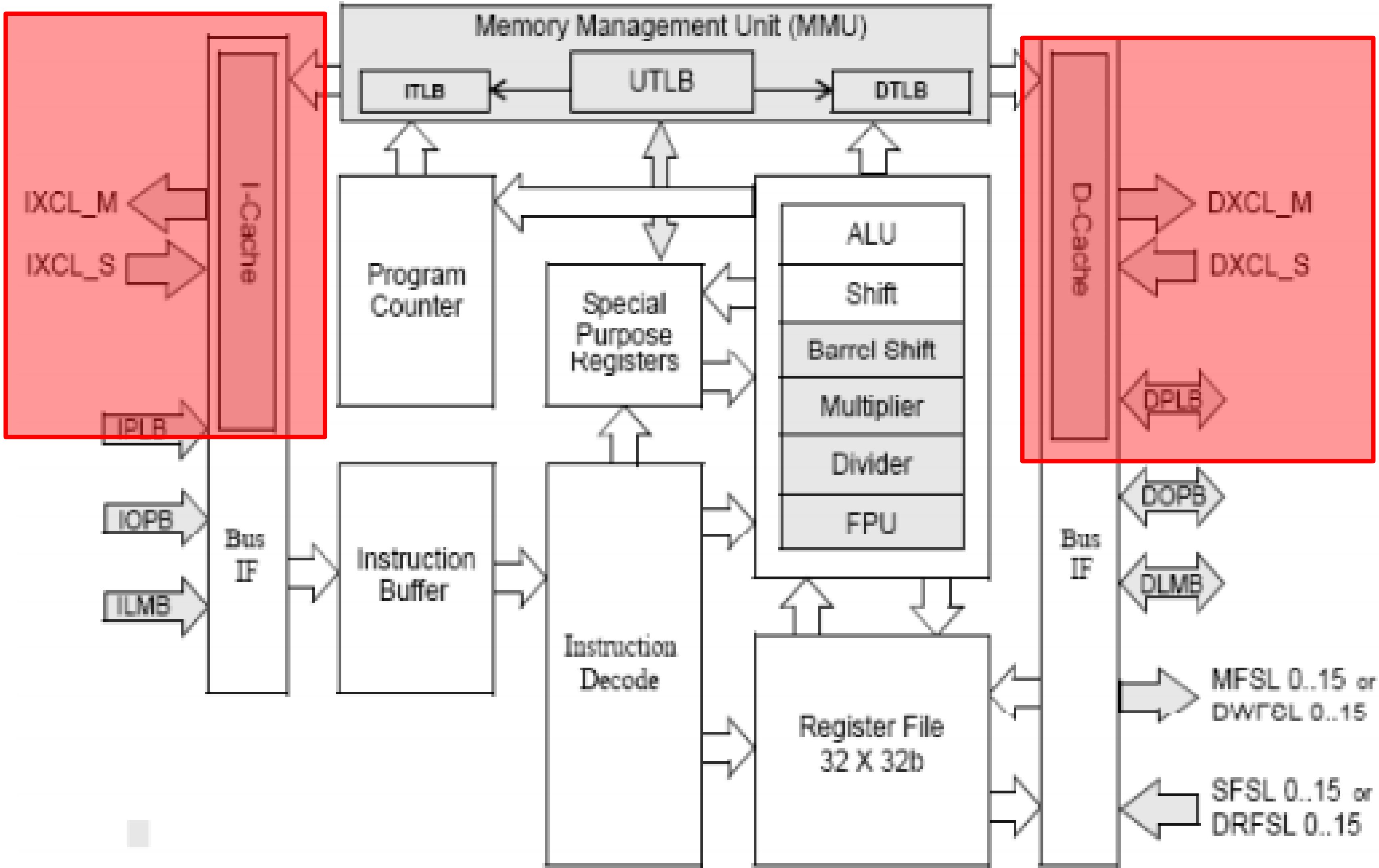
NIOS II处理器结构示意图

SOPC的基本概念与解决方案



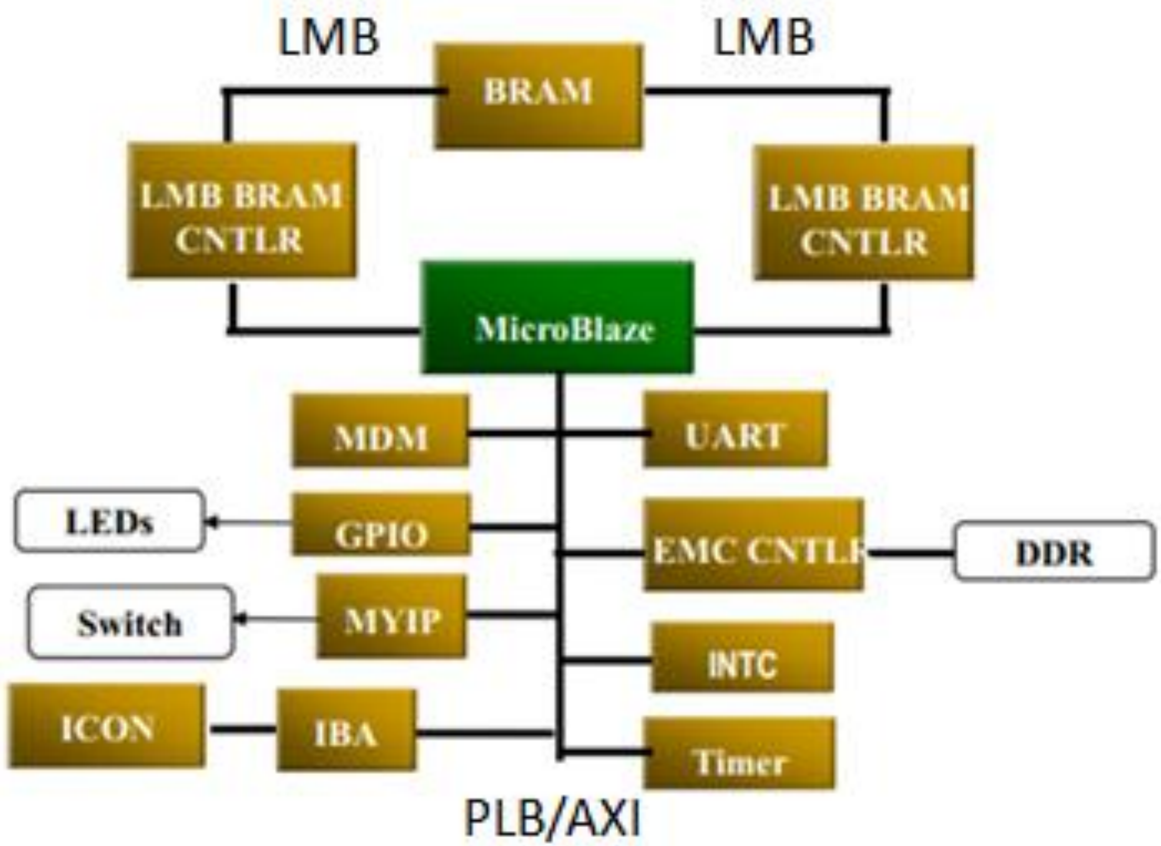
基于NIOS II的SOPC设计实例示意图

SOPC的基本概念与解决方案



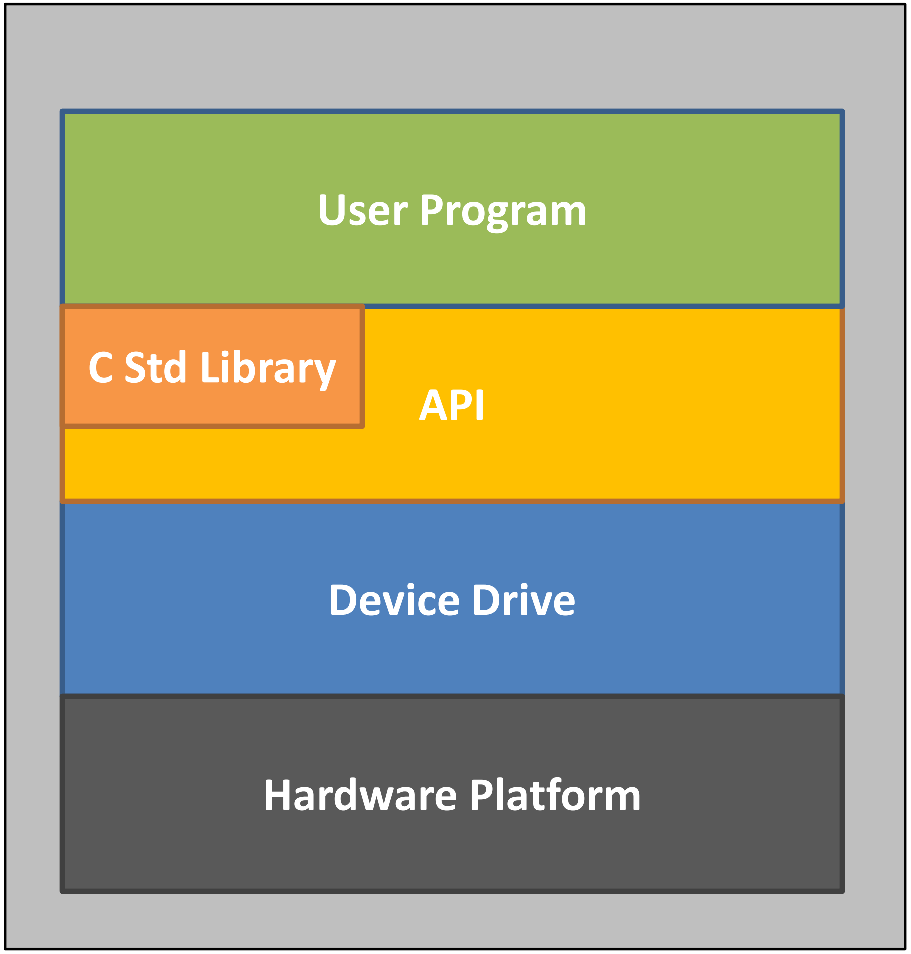
MicroBlaze处理器结构示意图

SOPC的基本概念与解决方案

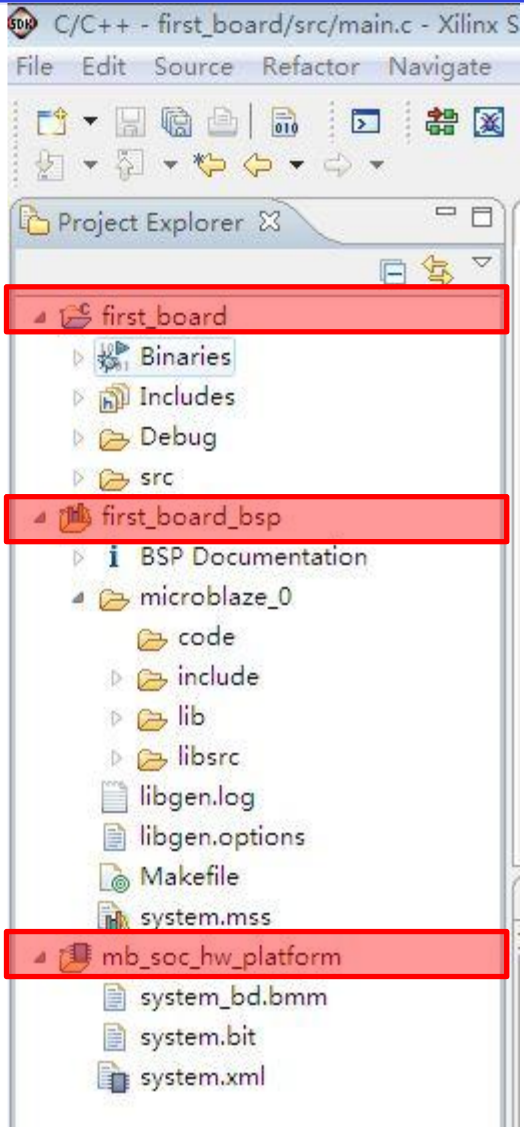


基于MicroBlaze的SOPC设计实例示意图

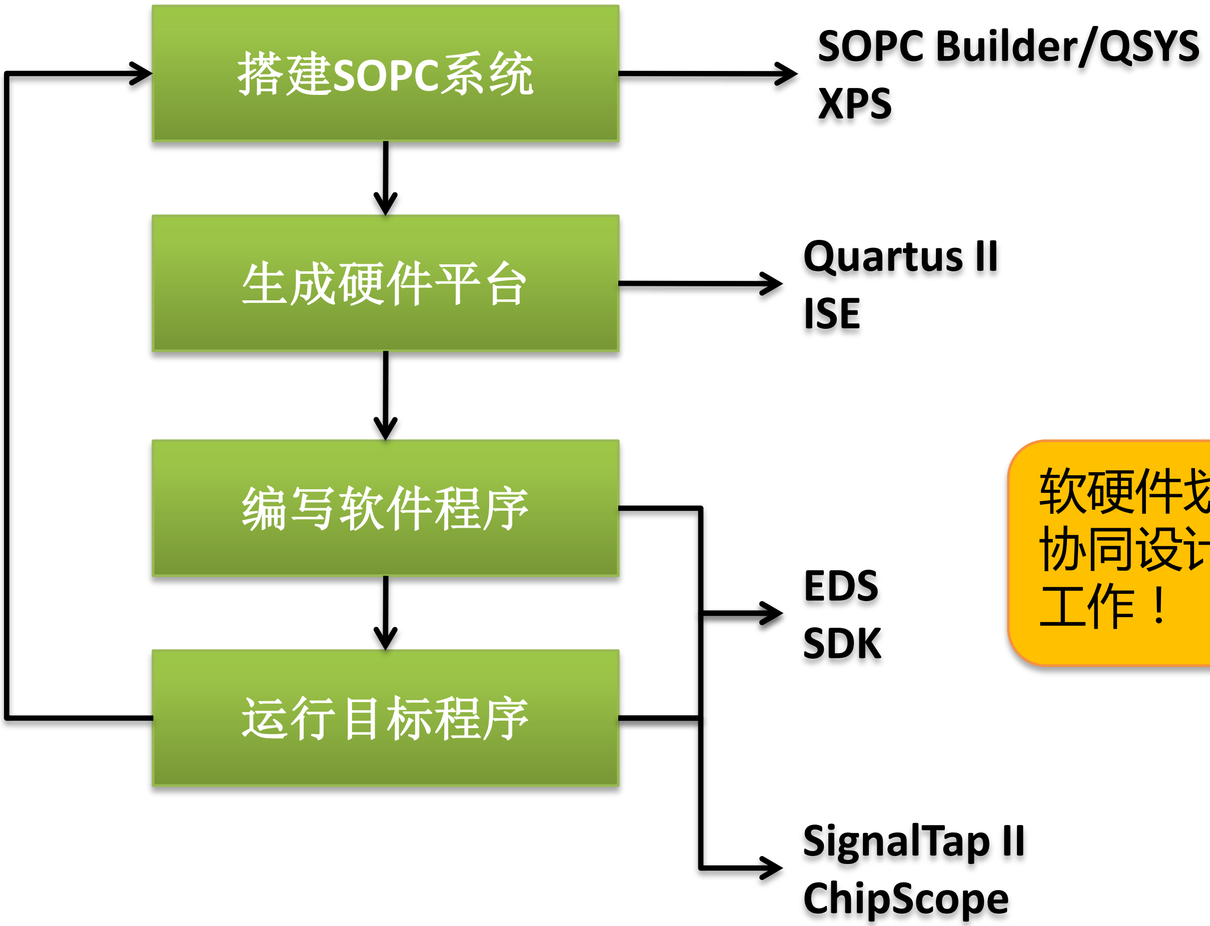
SOPC的基本概念与解决方案



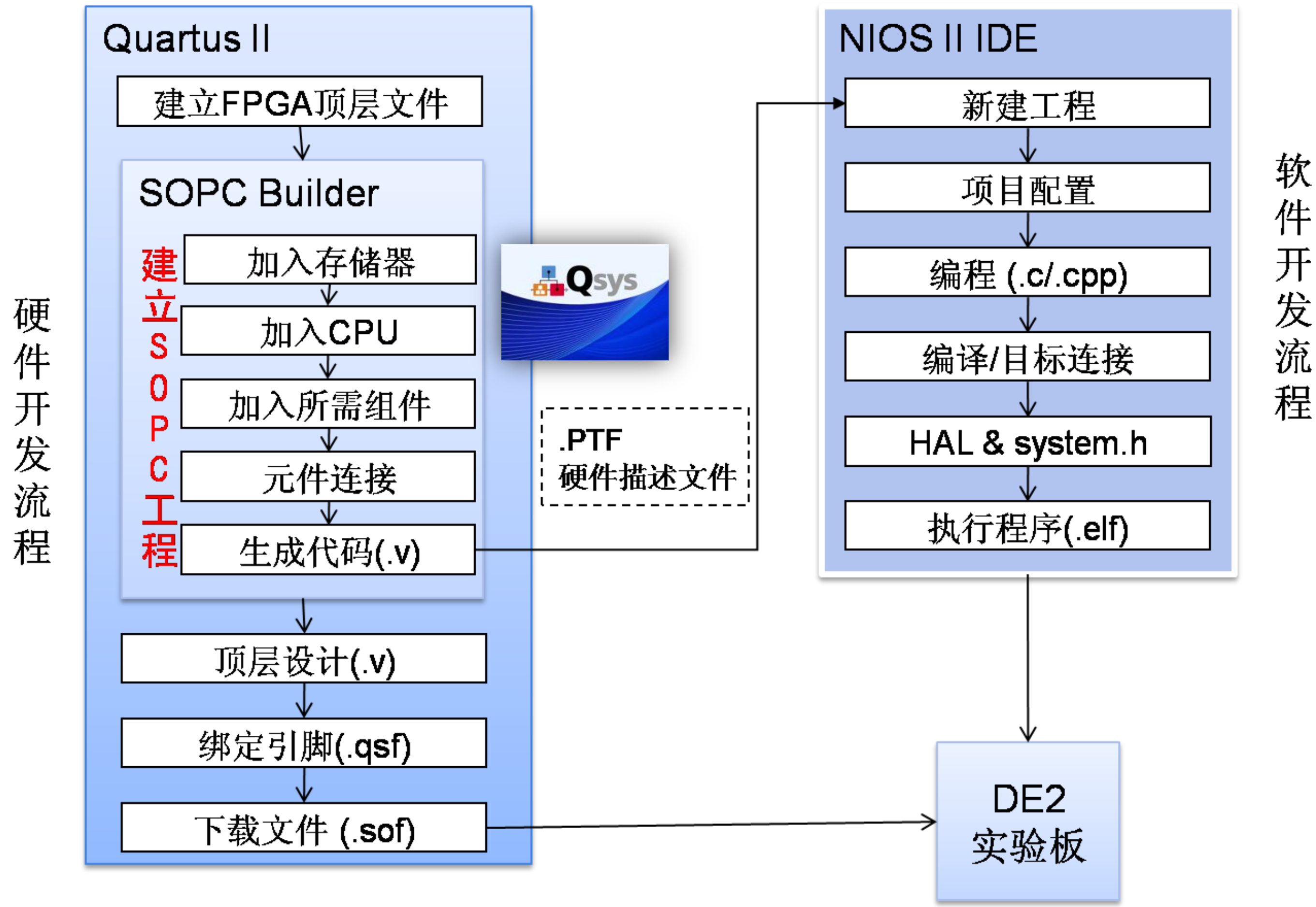
基于Nios II / MicroBlaze的软件架构层次示意图



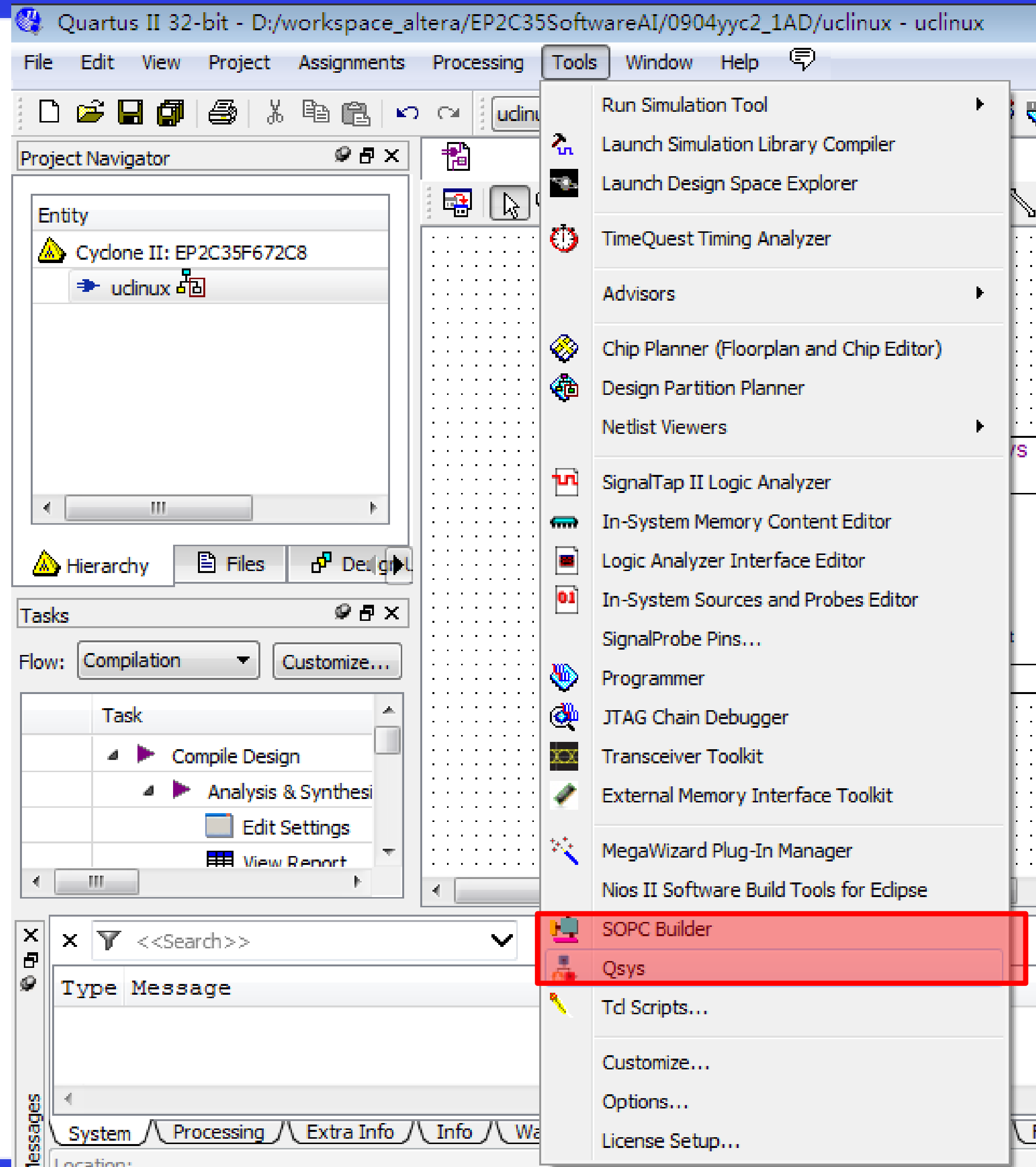
SOPC的软硬件协同设计流程



Altera的SOPC开发流程

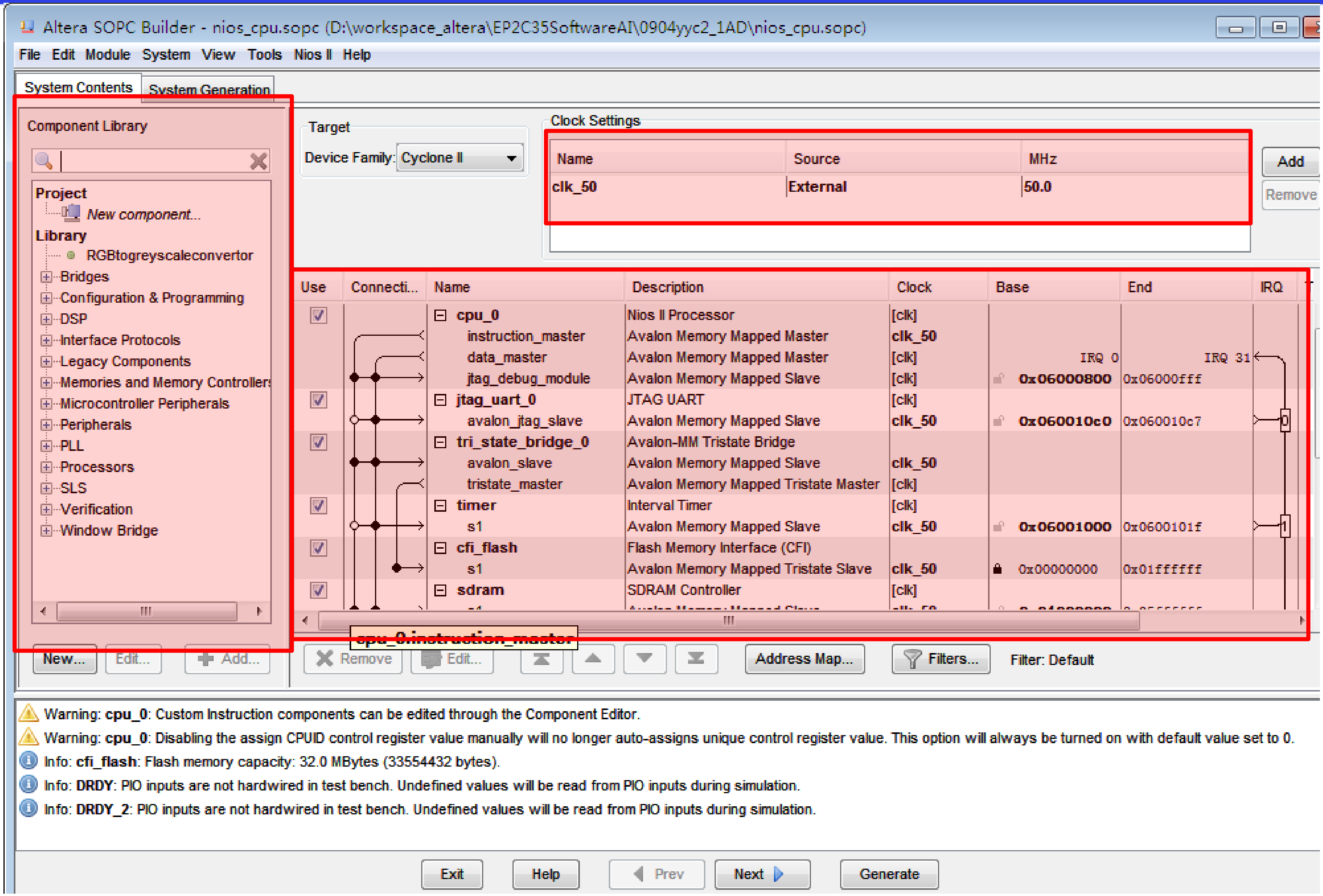


工具使用流程

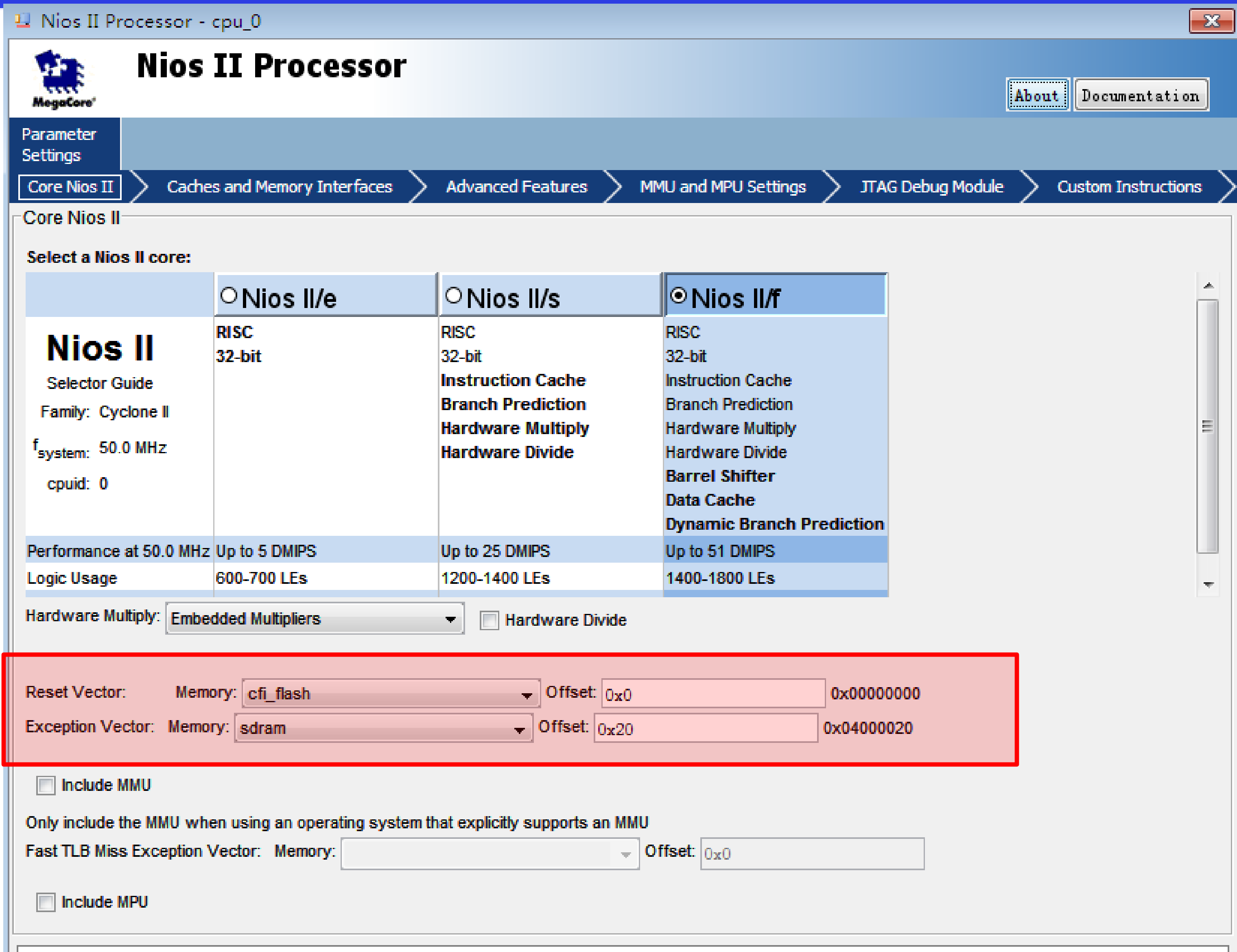


1、打开SOPC系统搭建工具

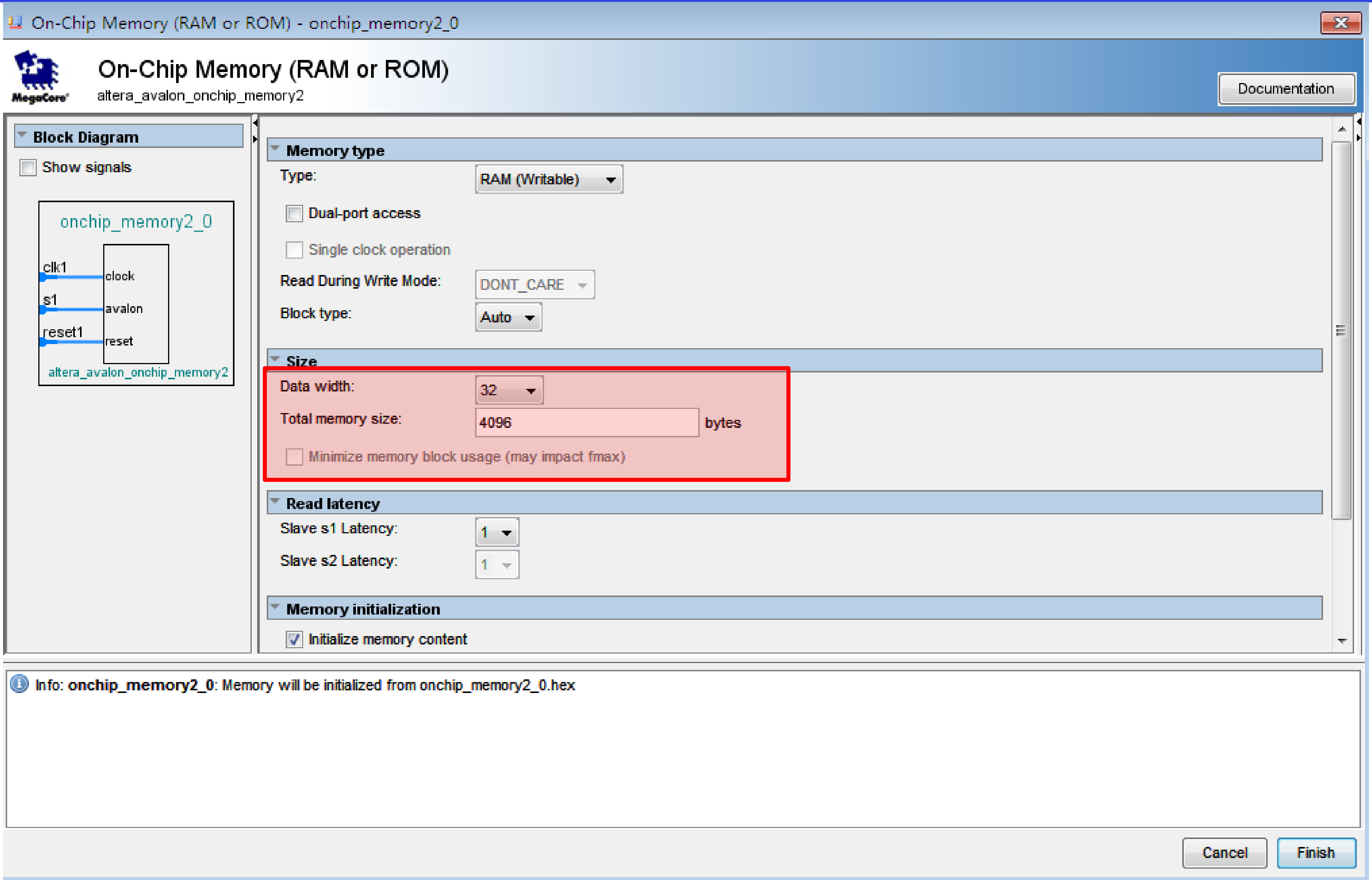
2、SOPC Builder中采用IP搭建SOPC系统并互联



设置处理器参数

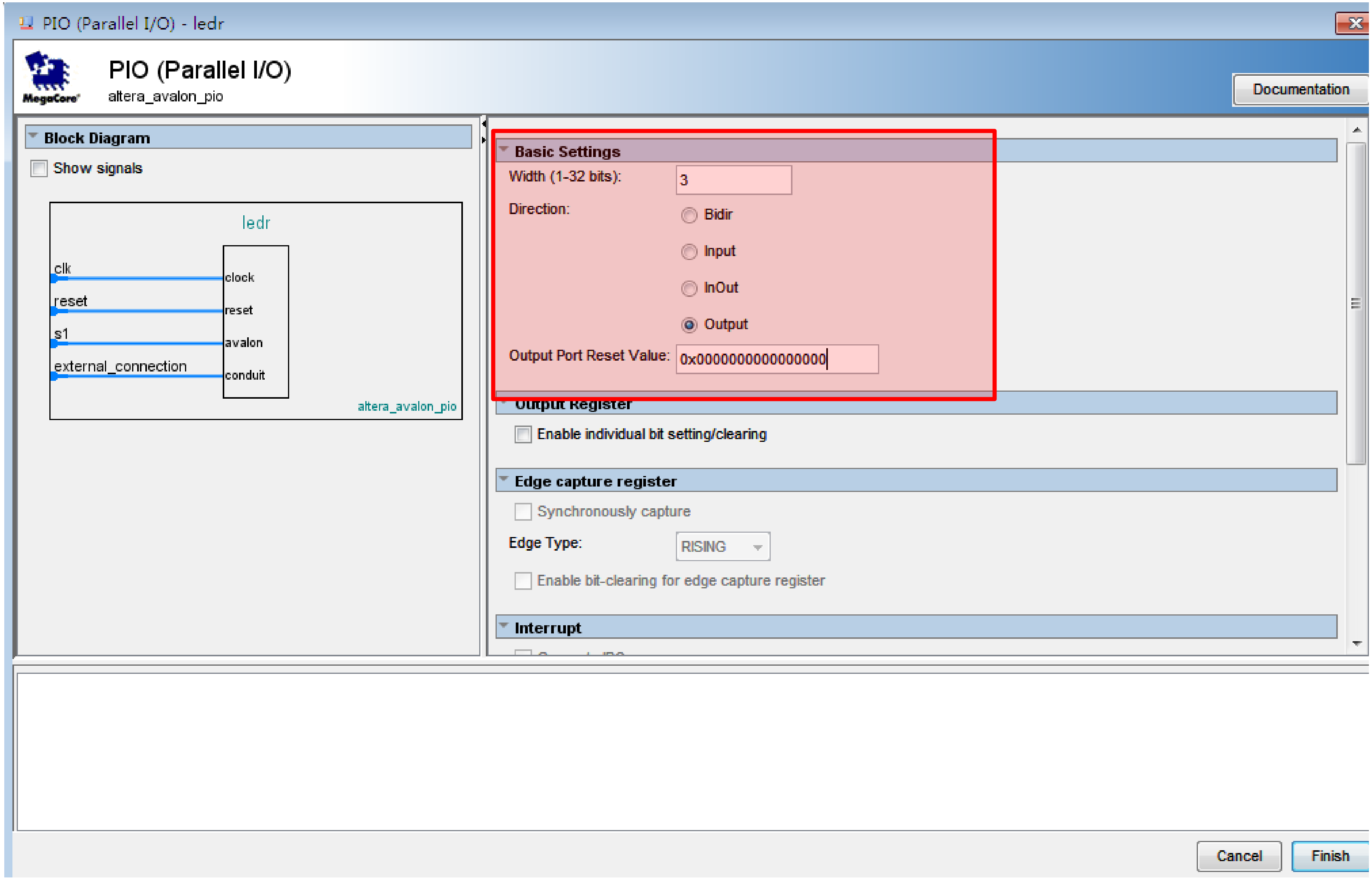

Nios II Processor - cpu_0
MegaCore
Nios II Processor
About
Documentation
Parameter Settings
Core Nios II
Caches and Memory Interfaces
Advanced Features
MMU and MPU Settings
JTAG Debug Module
Custom Instructions
Core Nios II
Select a Nios II core:
Nios II/e
Nios II/s
Nios II/f
Nios II
Selector Guide
Family: Cyclone II
fsystem: 50.0 MHz
cpuid: 0
RISC
32-bit
RISC
32-bit
Instruction Cache
Branch Prediction
Hardware Multiply
Hardware Divide
RISC
32-bit
Instruction Cache
Branch Prediction
Hardware Multiply
Hardware Divide
Barrel Shifter
Data Cache
Dynamic Branch Prediction
Performance at 50.0 MHz
Up to 5 DMIPS
Up to 25 DMIPS
Up to 51 DMIPS
Logic Usage
600-700 LEs
1200-1400 LEs
1400-1800 LEs
Hardware Multiply:
Embedded Multipliers
Hardware Divide
Reset Vector:
Memory:
cfi_flash
Offset:
0x0
0x00000000
Exception Vector:
Memory:
sdram
Offset:
0x20
0x04000020
Include MMU
Only include the MMU when using an operating system that explicitly supports an MMU
Fast TLB Miss Exception Vector:
Memory:
Offset:
0x0
Include MPU

设置片上存储器参数



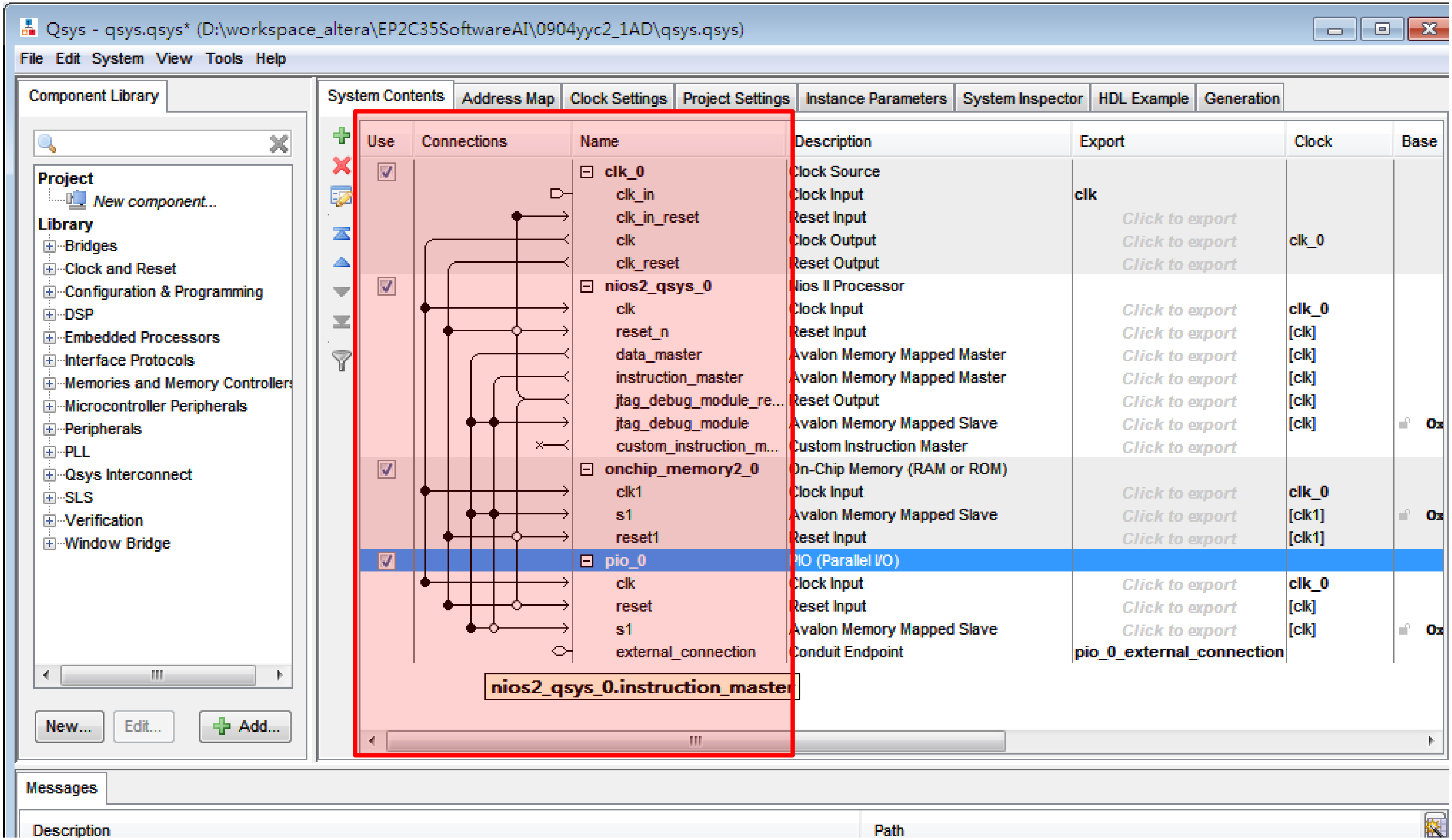
On-Chip Memory (RAM or ROM) - onchip_memory2_0
On-Chip Memory (RAM or ROM)
altera_avalon_onchip_memory2
Documentation
Block Diagram
Show signals
onchip_memory2_0
clk1
clock
s1
avalon
reset1
reset
altera_avalon_onchip_memory2
Memory type
Type:
RAM (Writable)
Dual-port access
Single clock operation
Read During Write Mode:
DONT_CARE
Block type:
Auto
Size
Data width:
32
Total memory size:
4096
bytes
Minimize memory block usage (may impact fmax)
Read latency
Slave s1 Latency:
1
Slave s2 Latency:
1
Memory initialization
Initialize memory content
Info: onchip_memory2_0: Memory will be initialized from onchip_memory2_0.hex
Cancel
Finish

设置片内外设的参数

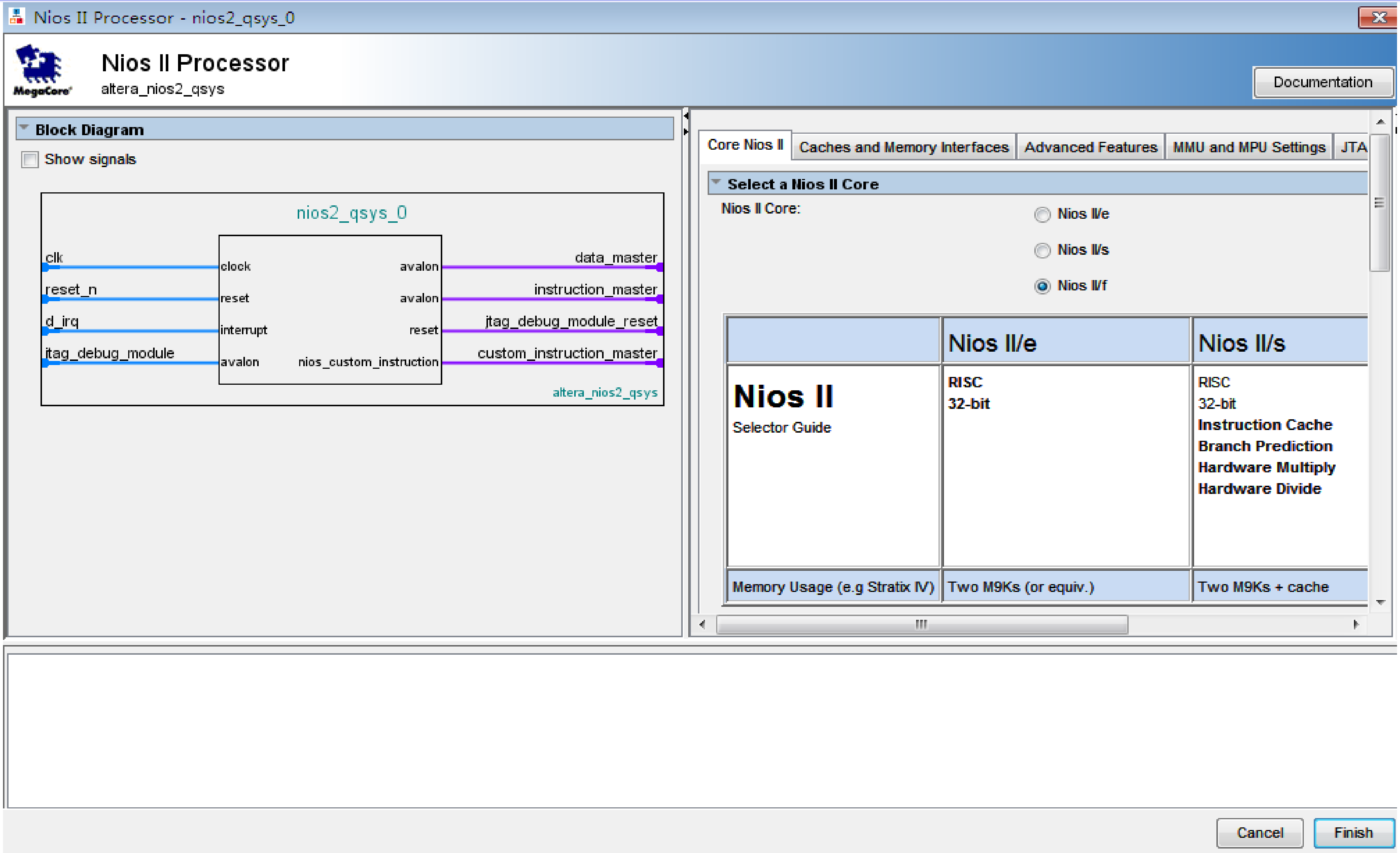


PIO (Parallel I/O) - ledr
PIO (Parallel I/O)
altera_avalon_pio
MegaCore
Documentation
Block Diagram
Show signals
ledr
clk
reset
s1
external_connection
clock
reset
avalon
conduit
altera_avalon_pio
Basic Settings
Width (1-32 bits):
3
Direction:
Bidir
Input
InOut
Output
Output Port Reset Value:
0x0000000000000000
Output Register
Enable individual bit setting/clearing
Edge capture register
Synchronously capture
Edge Type:
RISING
Enable bit-clearing for edge capture register
Interrupt
Cancel
Finish

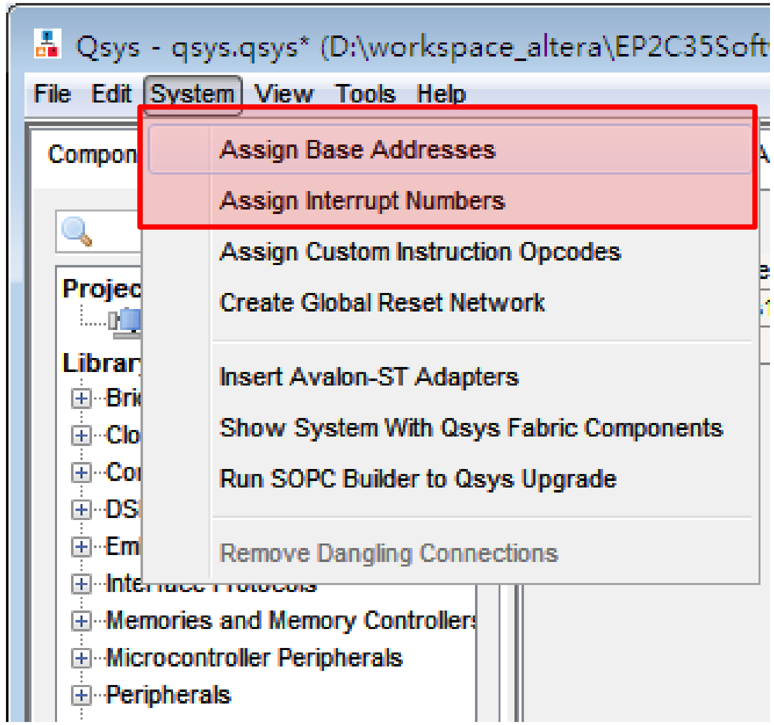
2、Qsys中采用IP搭建SOPC系统并互联



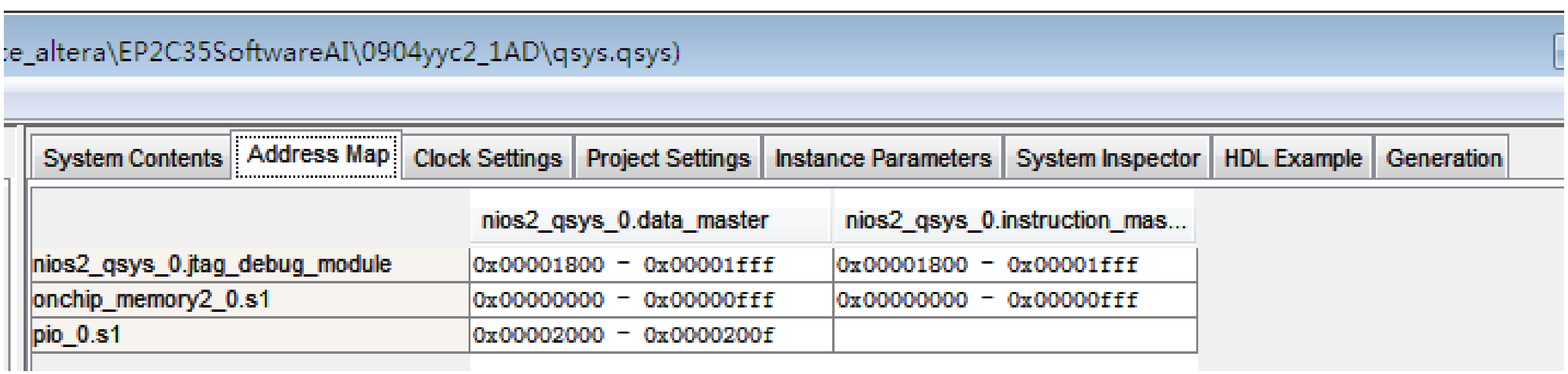
设置处理器参数


Nios II Processor - nios2_qsys_0
Nios II Processor
altera_nios2_qsys
Documentation
Block Diagram
Show signals
nios2_qsys_0
clk
reset_n
d_irq
jtag_debug_module
clock
reset
interrupt
avalon
avalon
avalon
reset
nios_custom_instruction
data_master
instruction_master
jtag_debug_module_reset
custom_instruction_master
altera_nios2_qsys
Core Nios II
Caches and Memory Interfaces
Advanced Features
MMU and MPU Settings
JTA
Select a Nios II Core
Nios II Core:
Nios II/e
Nios II/s
Nios II/f
Nios II/e
Nios II/s
Nios II
Selector Guide
RISC
32-bit
RISC
32-bit
Instruction Cache
Branch Prediction
Hardware Multiply
Hardware Divide
Memory Usage (e.g Stratix IV)
Two M9Ks (or equiv.)
Two M9Ks + cache
Cancel
Finish

分配地址和中断优先级

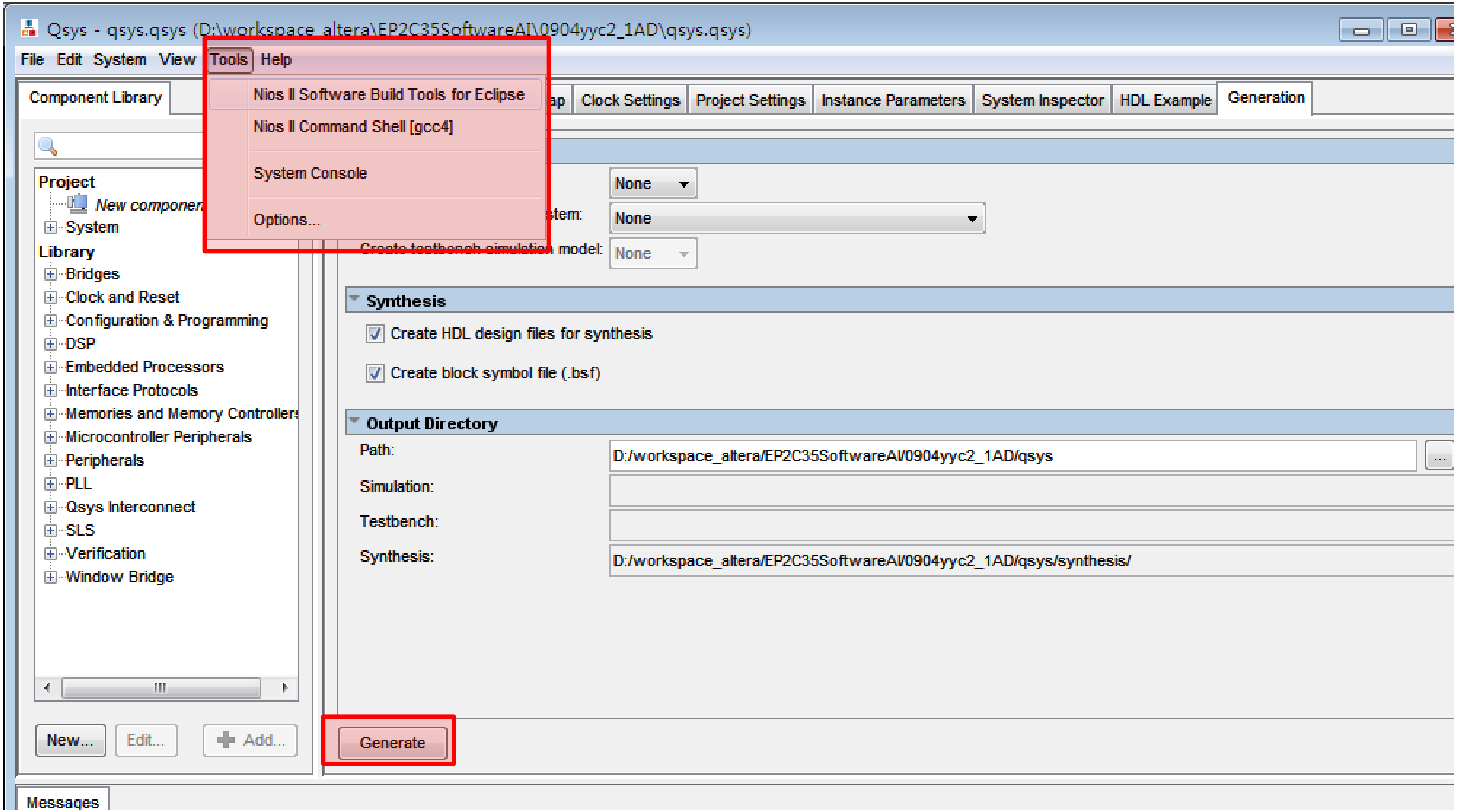


Qsys - qsys.qsys* (D:\workspace_altera\EP2C35Soft
File Edit System View Tools Help
Assign Base Addresses
Assign Interrupt Numbers
Assign Custom Instruction Opcodes
Create Global Reset Network
Insert Avalon-ST Adapters
Show System With Qsys Fabric Components
Run SOPC Builder to Qsys Upgrade
Remove Dangling Connections
Memories and Memory Controllers
Microcontroller Peripherals
Peripherals

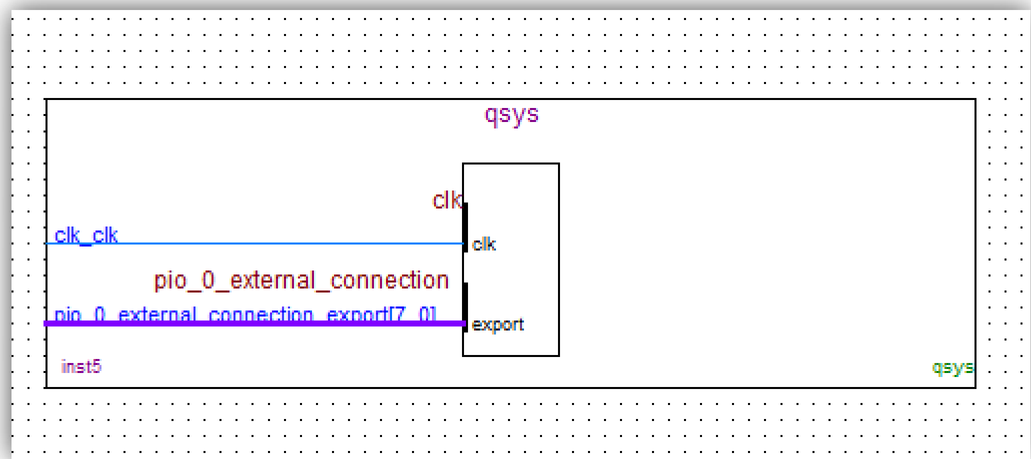


e_altera\EP2C35SoftwareAI\0904yyc2_1AD\qsys.qsys)
System Contents
Address Map
Clock Settings
Project Settings
Instance Parameters
System Inspector
HDL Example
Generation
nios2_qsys_0.data_master
nios2_qsys_0.instruction_mas...
nios2_qsys_0.jtag_debug_module
0x00001800 - 0x00001fff
0x00001800 - 0x00001fff
onchip_memory2_0.s1
0x00000000 - 0x00000fff
0x00000000 - 0x00000fff
pio_0.s1
0x00002000 - 0x0000200f

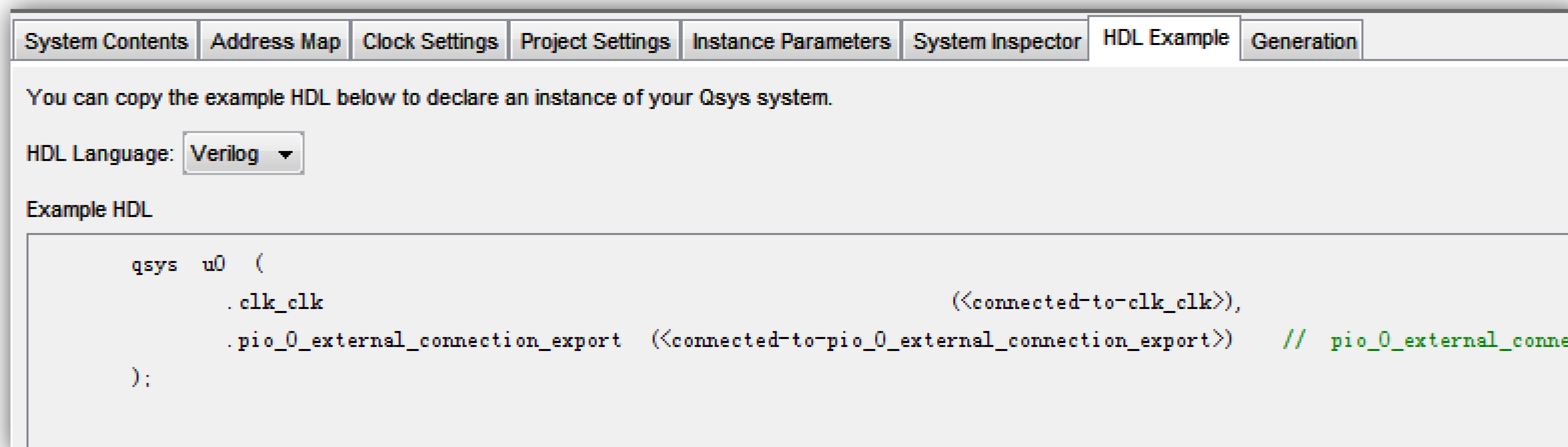
生成SOPC系统并打开Nios II EDS工具



SOPC系统添加到顶层设计中

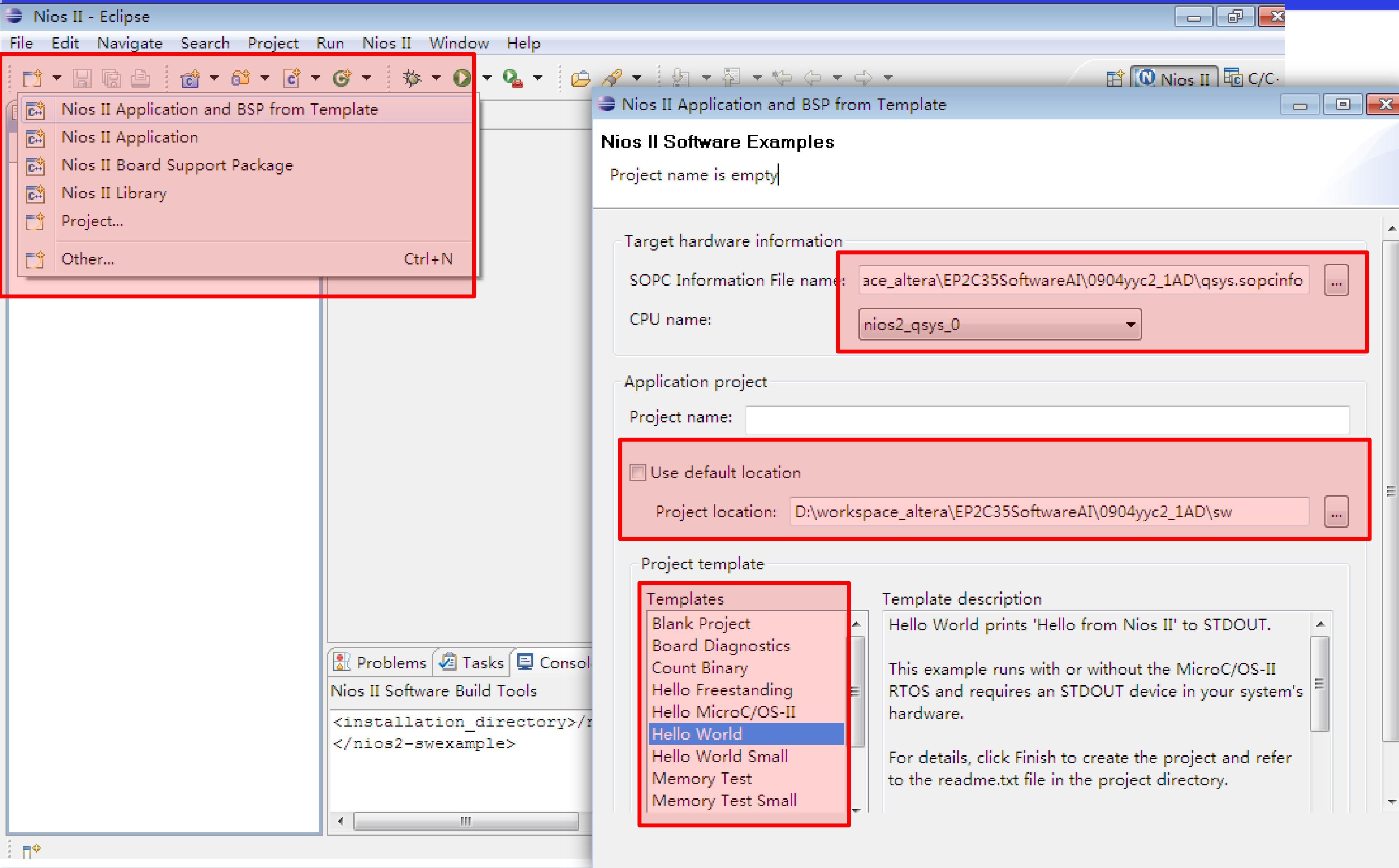


原理图方式

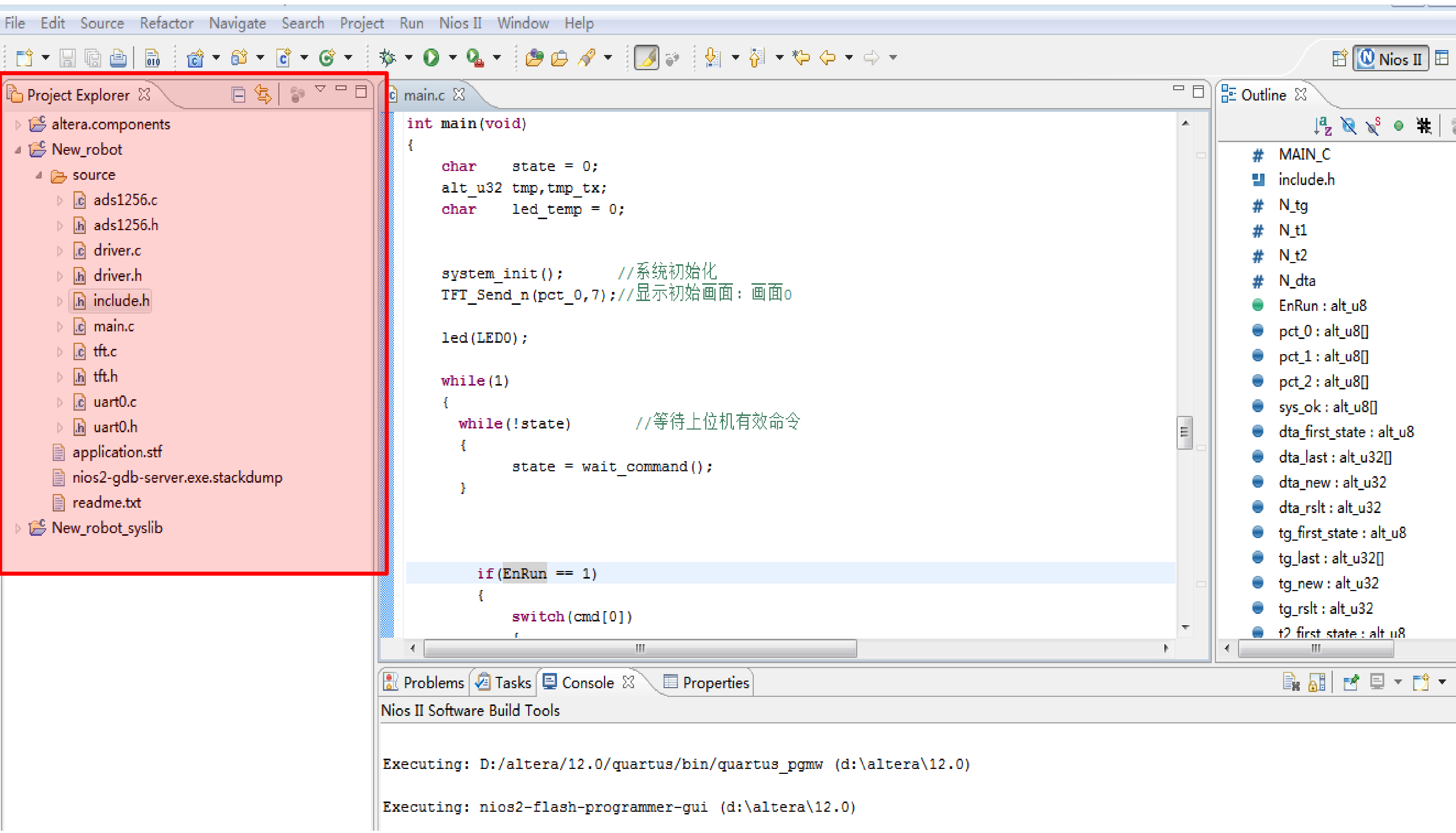


代码设计方式

创建应用工程

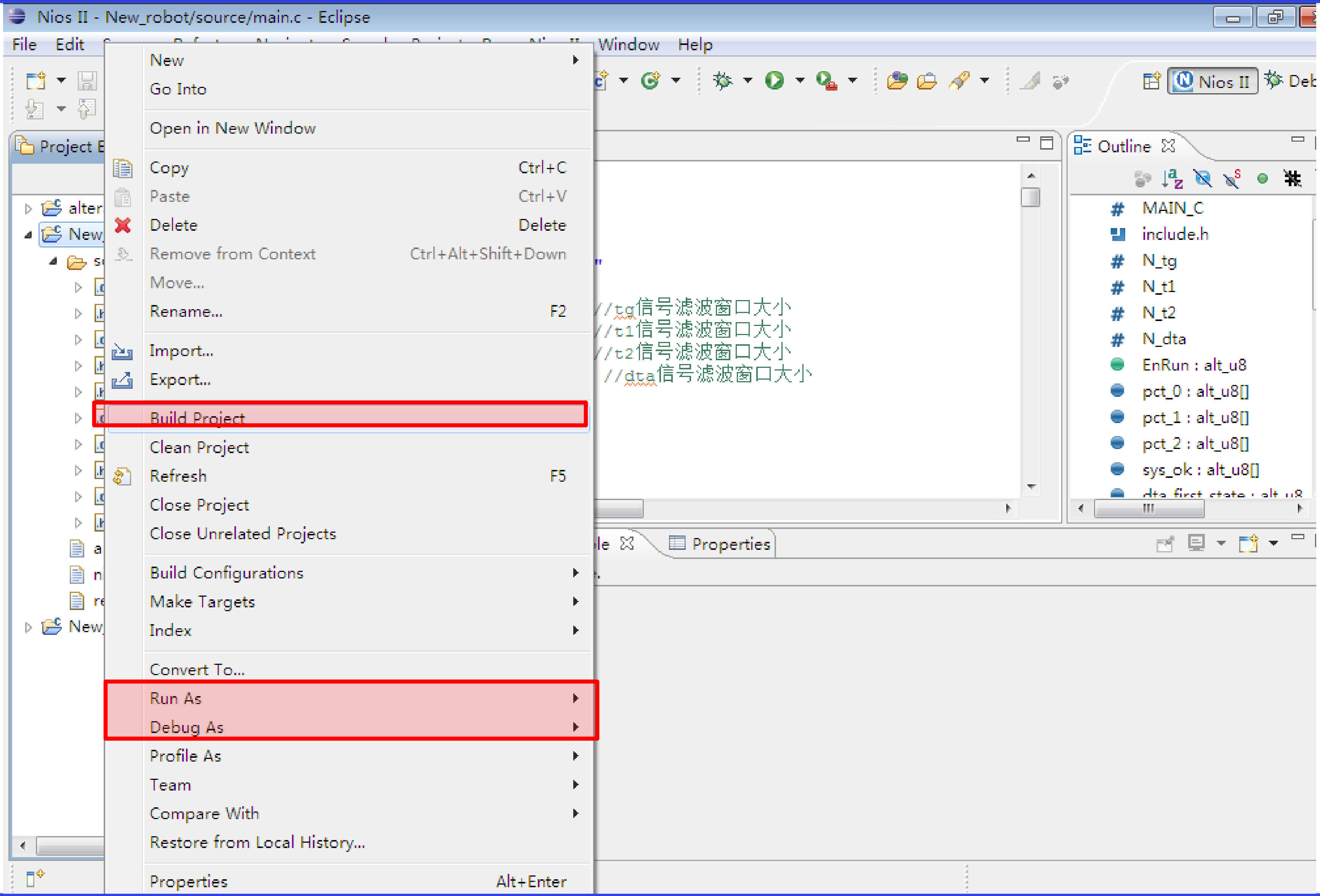

Nios II - Eclipse
File Edit Navigate Search Project Run Nios II Window Help
Nios II Application and BSP from Template
Nios II Application
Nios II Board Support Package
Nios II Library
Project...
Other... Ctrl+N
Problems Tasks Consol
Nios II Software Build Tools
<installation_directory>/r
</nios2-swexample>
Nios II Application and BSP from Template
Nios II Software Examples
Project name is empty
Target hardware information
SOPC Information File name: ace_altera\EP2C35SoftwareAI\0904yyc2_1AD\qsys.sopcinfo
CPU name: nios2_qsys_0
Application project
Project name:
Use default location
Project location: D:\workspace_altera\EP2C35SoftwareAI\0904yyc2_1AD\sw
Project template
Templates
Blank Project
Board Diagnostics
Count Binary
Hello Freestanding
Hello MicroC/OS-II
Hello World
Hello World Small
Memory Test
Memory Test Small
Template description
Hello World prints 'Hello from Nios II' to STDOUT.
This example runs with or without the MicroC/OS-II RTOS and requires an STDOUT device in your system's hardware.
For details, click Finish to create the project and refer to the readme.txt file in the project directory.

编写软件程序



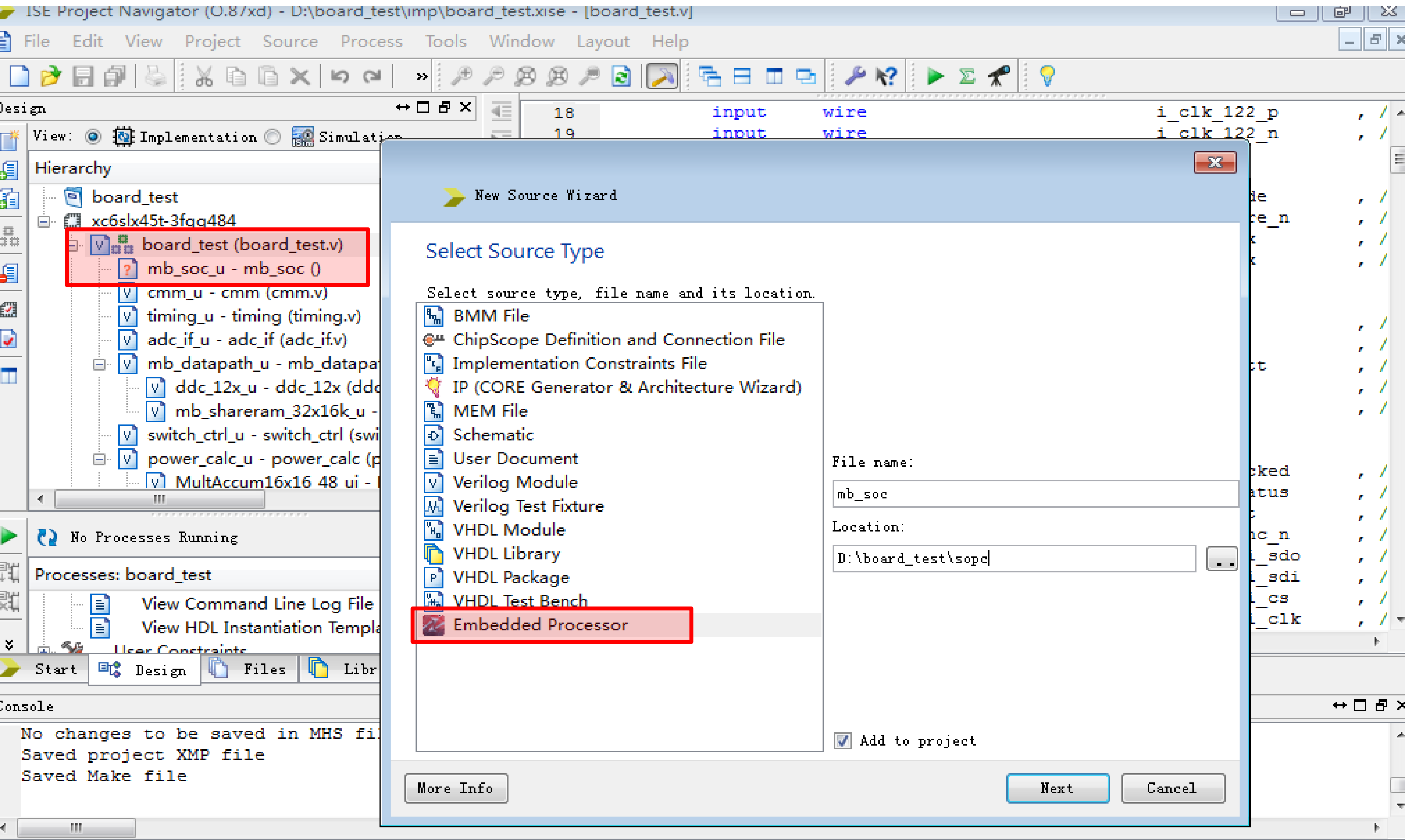
File Edit Source Refactor Navigate Search Project Run Nios II Window Help
Project Explorer
altera.components
New_robot
source
ads1256.c
ads1256.h
driver.c
driver.h
include.h
main.c
tft.c
tft.h
uart0.c
uart0.h
application.stf
nios2-gdb-server.exe.stackdump
readme.txt
New_robot_syslib
main.c
int main(void)
{
char state = 0;
alt_u32 tmp,tmp_tx;
char led_temp = 0;
system_init(); //系统初始化
TFT_Send_n(pct_0,7);//显示初始画面：画面0
led(LED0);
while(1)
{
while(!state) //等待上位机有效命令
{
state = wait_command();
}
if(EnRun == 1)
{
switch(cmd[0])
Outline
MAIN_C
include.h
N_tg
N_t1
N_t2
N_dta
EnRun : alt_u8
pct_0 : alt_u8[]
pct_1 : alt_u8[]
pct_2 : alt_u8[]
sys_ok : alt_u8[]
dta_first_state : alt_u8
dta_last : alt_u32[]
dta_new : alt_u32
dta_rslt : alt_u32
tg_first_state : alt_u8
tg_last : alt_u32[]
tg_new : alt_u32
tg_rslt : alt_u32
Problems
Tasks
Console
Properties
Nios II Software Build Tools
Executing: D:/altera/12.0/quartus/bin/quartus_pgmw (d:\altera\12.0)
Executing: nios2-flash-programmer-gui (d:\altera\12.0)

调试和下载目标程序

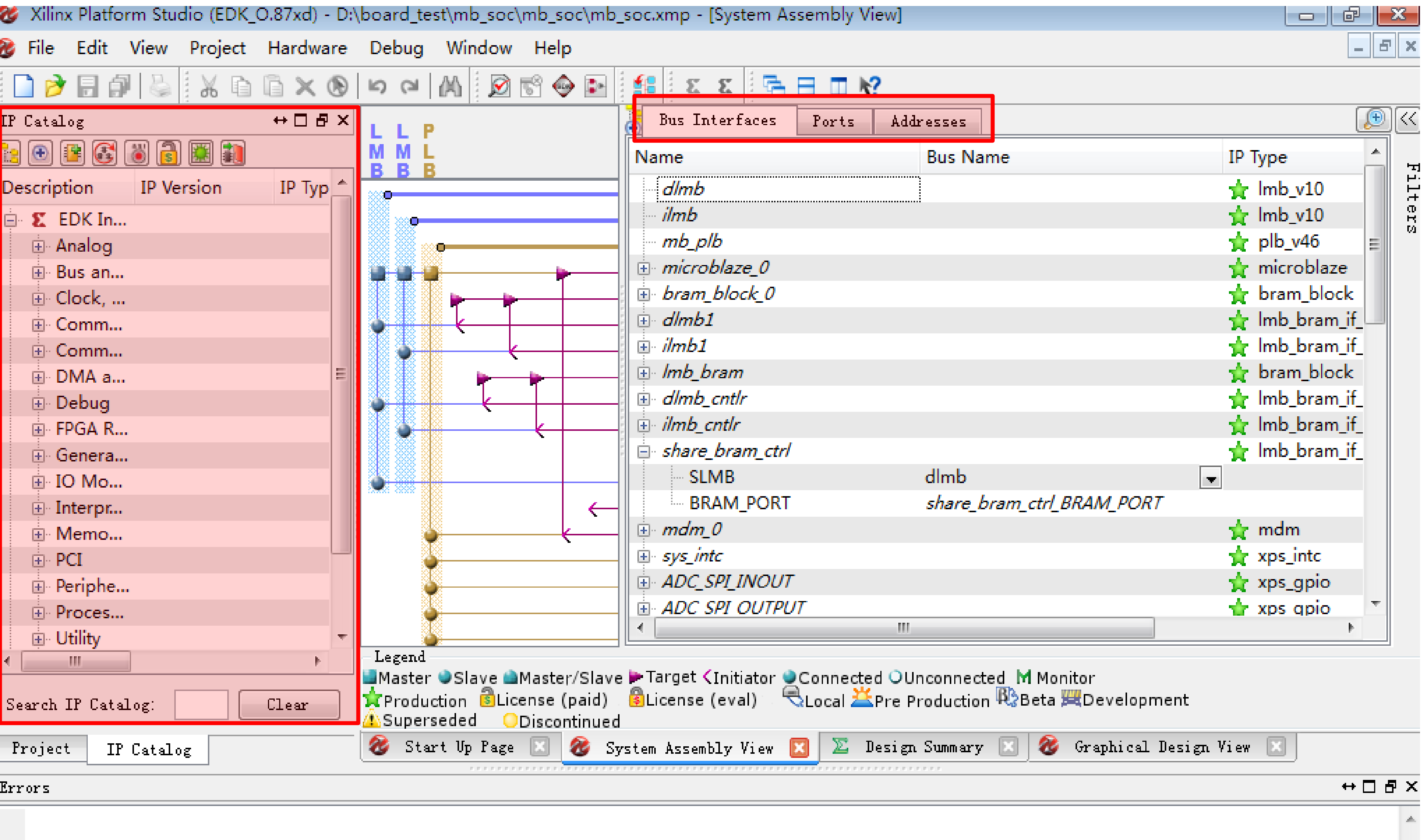


Nios II - New_robot/source/main.c - Eclipse
File
Edit
Window
Help
New
Go Into
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Build Configurations
Make Targets
Index
Convert To...
Run As
Debug As
Profile As
Team
Compare With
Restore from Local History...
Properties Alt+Enter
//tg信号滤波窗口大小
//t1信号滤波窗口大小
//t2信号滤波窗口大小
//dta信号滤波窗口大小
Outline
MAIN_C
include.h
N_tg
N_t1
N_t2
N_dta
EnRun : alt_u8
pct_0 : alt_u8[]
pct_1 : alt_u8[]
pct_2 : alt_u8[]
sys_ok : alt_u8[]
Properties
Nios II
Project E

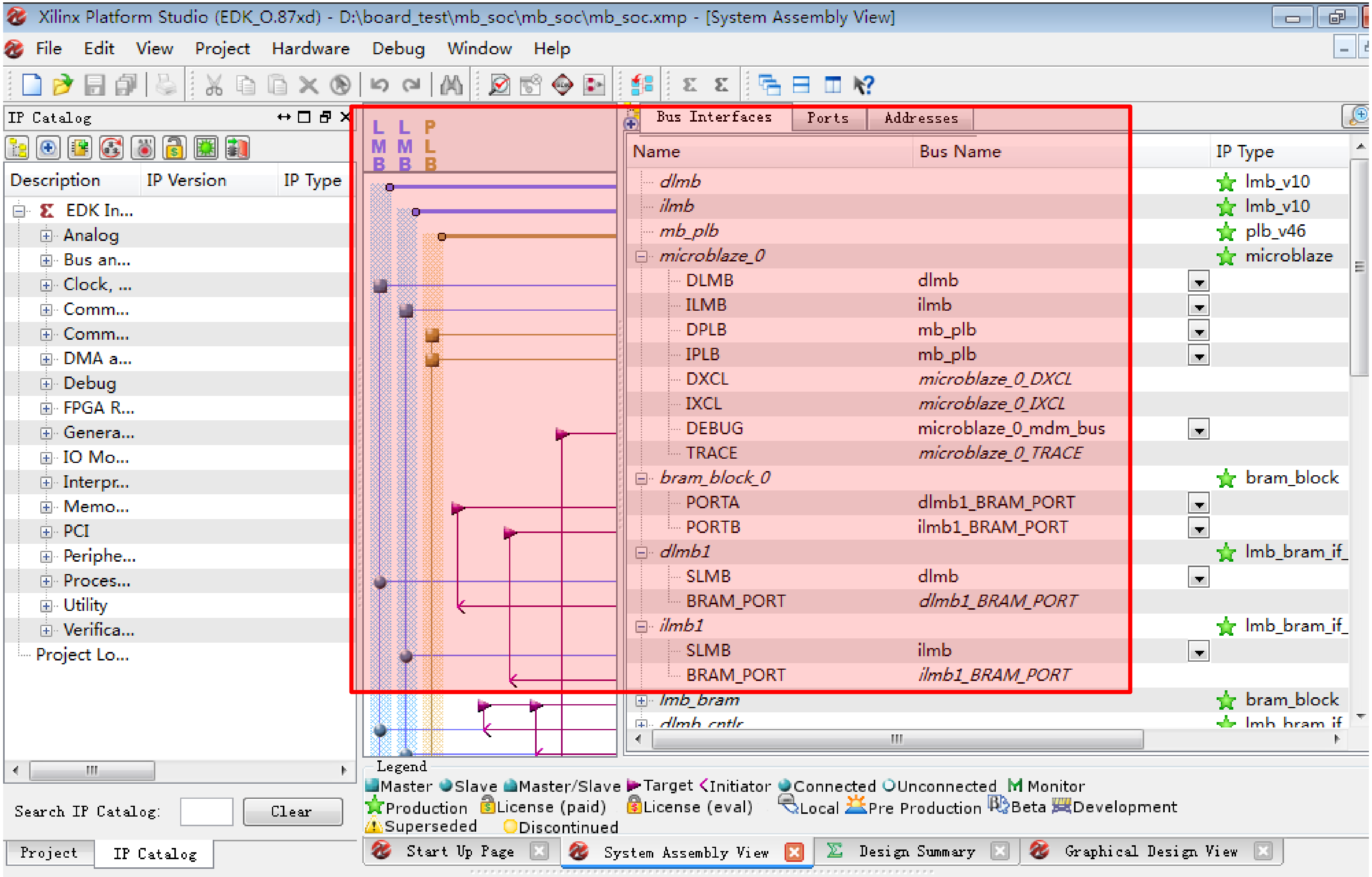
Xilinx的SOPC开发流程



采用IP搭建SOPC系统并互联

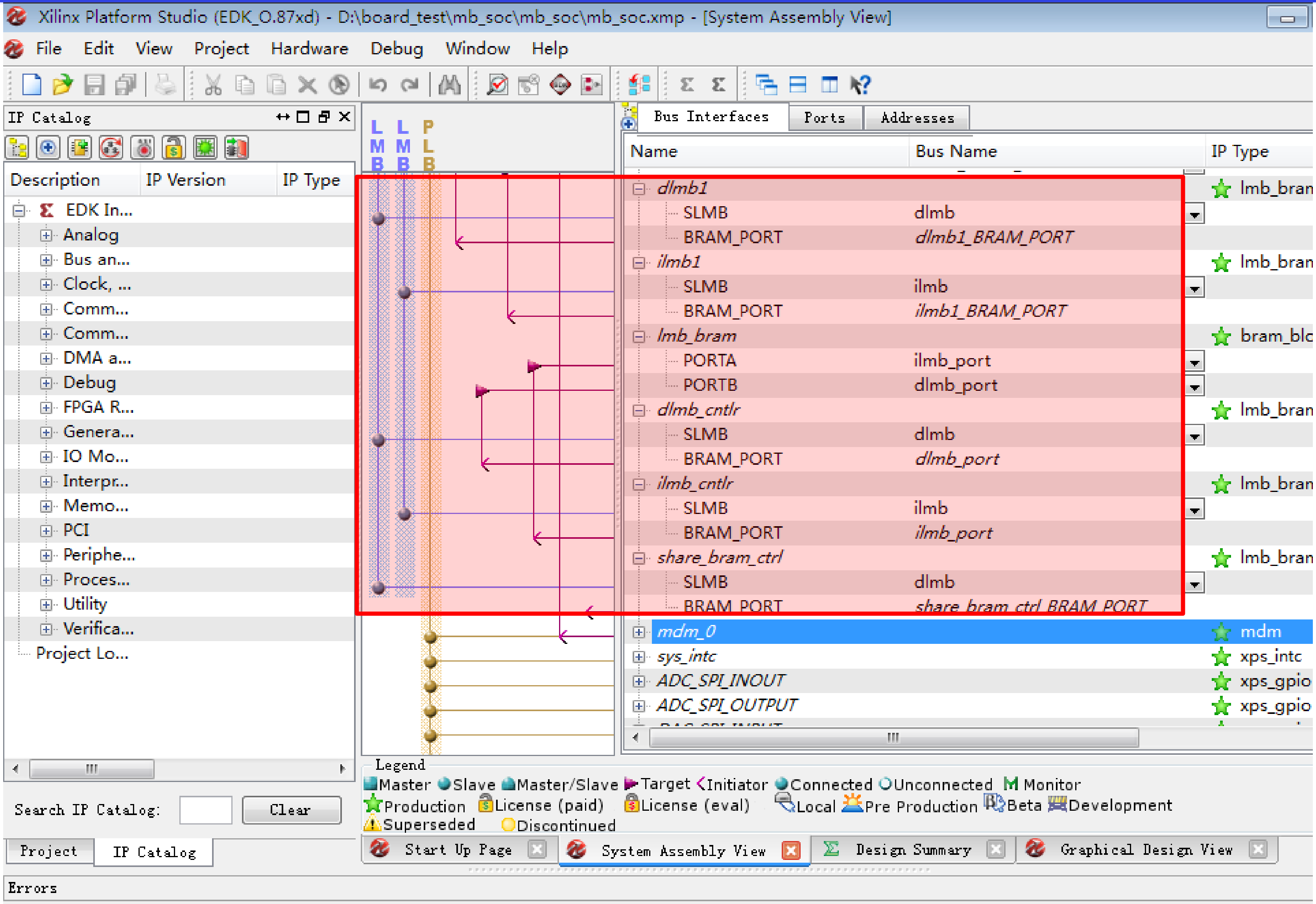


了解处理器的互联结构

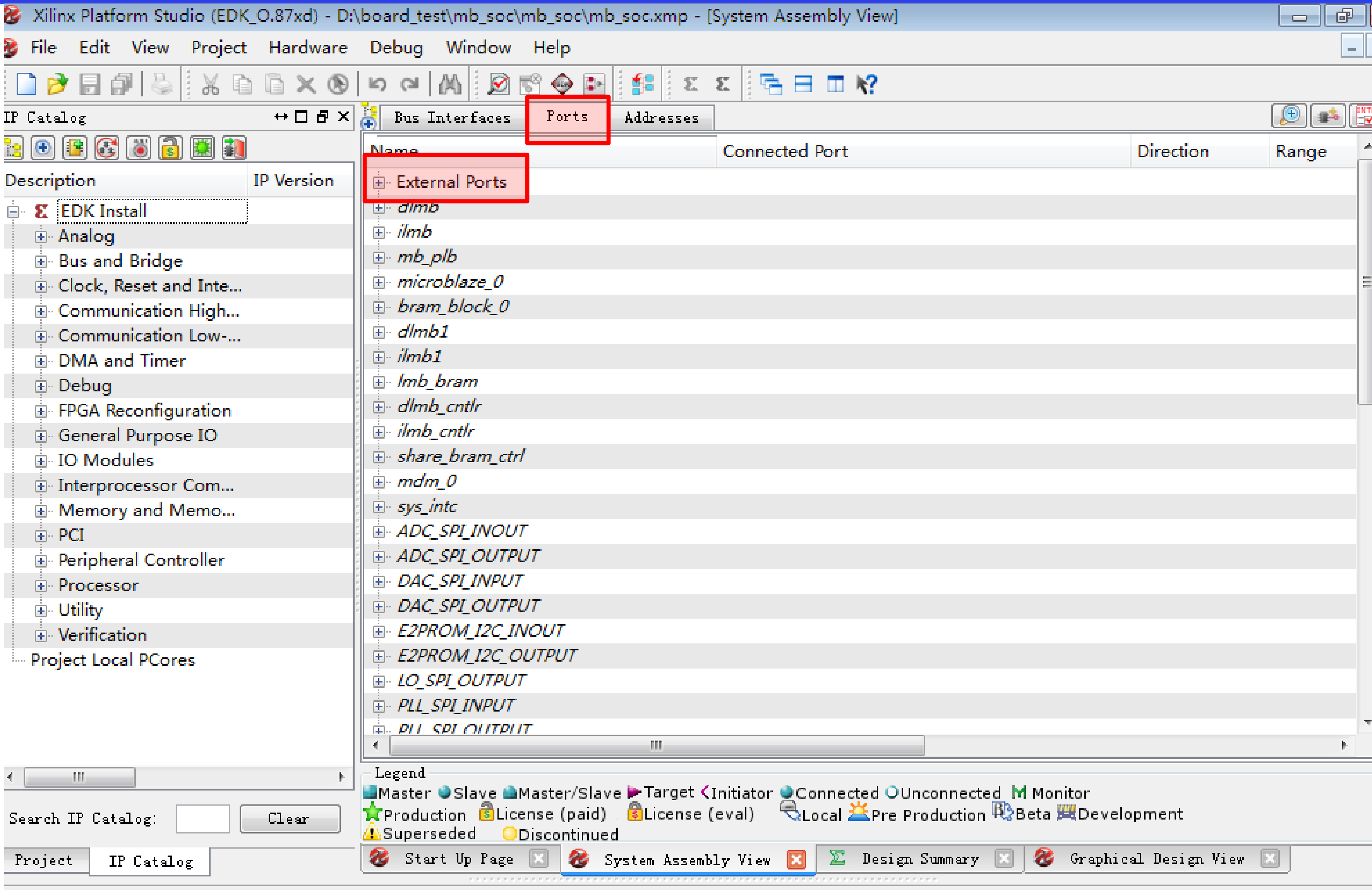


Xilinx Platform Studio (EDK_O.87xd) - D:\board_test\mb_soc\mb_soc\mb_soc.xmp - [System Assembly View]
File Edit View Project Hardware Debug Window Help
IP Catalog
Description IP Version IP Type
EDK In...
Analog
Bus an...
Clock, ...
Comm...
Comm...
DMA a...
Debug
FPGA R...
Genera...
IO Mo...
Interpr...
Memo...
PCI
Periphe...
Proces...
Utility
Verifica...
Project Lo...
Search IP Catalog:
Clear
Project
IP Catalog
LMB LMB PLB
Bus Interfaces Ports Addresses
Name Bus Name IP Type
dlmb lmb_v10
ilmb lmb_v10
mb_plb plb_v46
microblaze_0 microblaze
DLMB dlmb
ILMB ilmb
DPLB mb_plb
IPLB mb_plb
DXCL microblaze_0_DXCL
IXCL microblaze_0_IXCL
DEBUG microblaze_0_mdm_bus
TRACE microblaze_0_TRACE
bram_block_0 bram_block
PORTA dlmb1_BRAM_PORT
PORTB ilmb1_BRAM_PORT
dlmb1 lmb_bram_if_
SLMB dlmb
BRAM_PORT dlmb1_BRAM_PORT
ilmb1 lmb_bram_if_
SLMB ilmb
BRAM_PORT ilmb1_BRAM_PORT
lmb_bram bram_block
Legend
Master Slave Master/Slave Target Initiator Connected Unconnected Monitor
Production License (paid) License (eval) Local Pre Production Beta Development
Superseded Discontinued
Start Up Page
System Assembly View
Design Summary
Graphical Design View

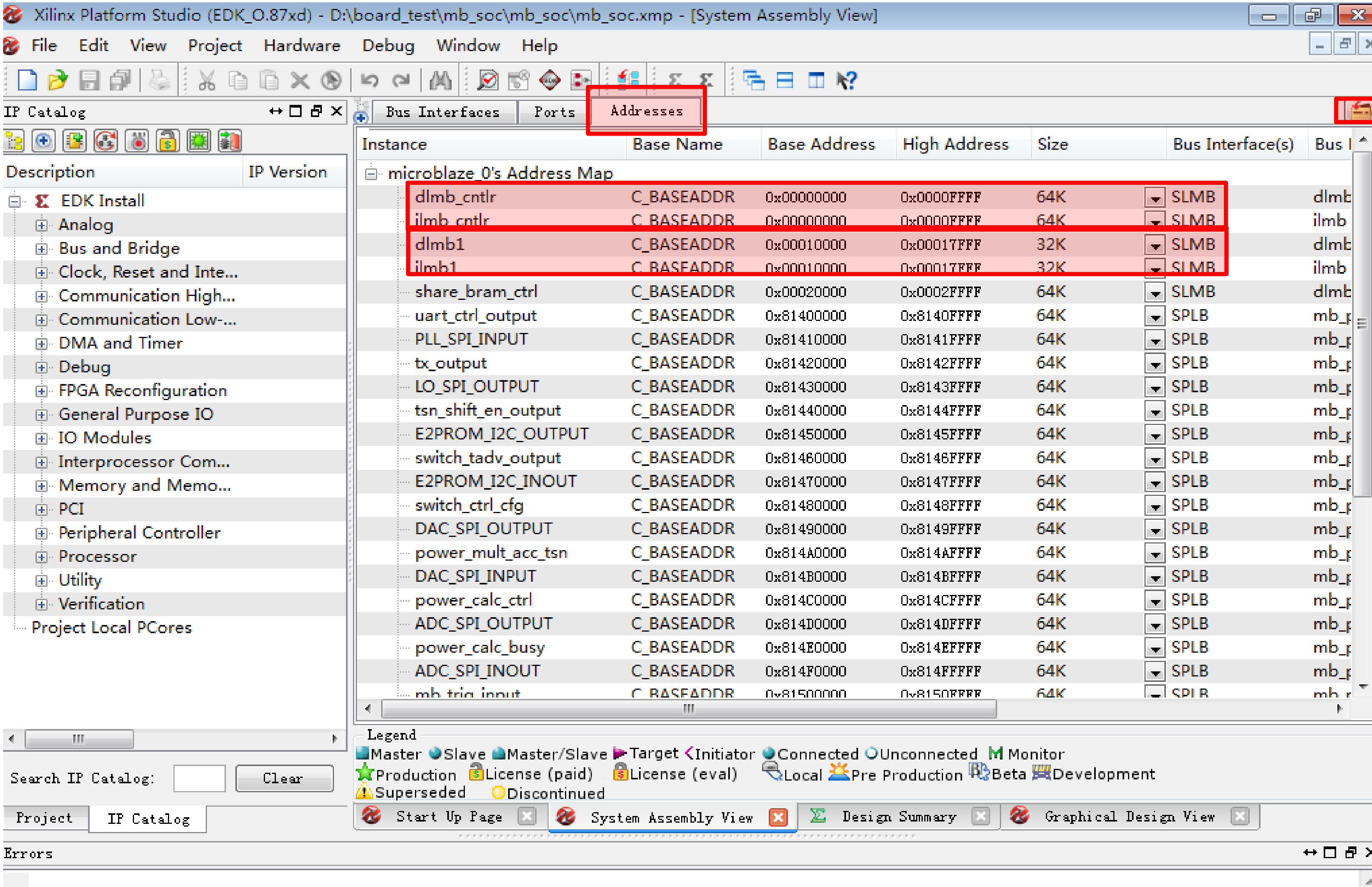
如何实现逻辑与软件的内存共享—非常实用的设计方法



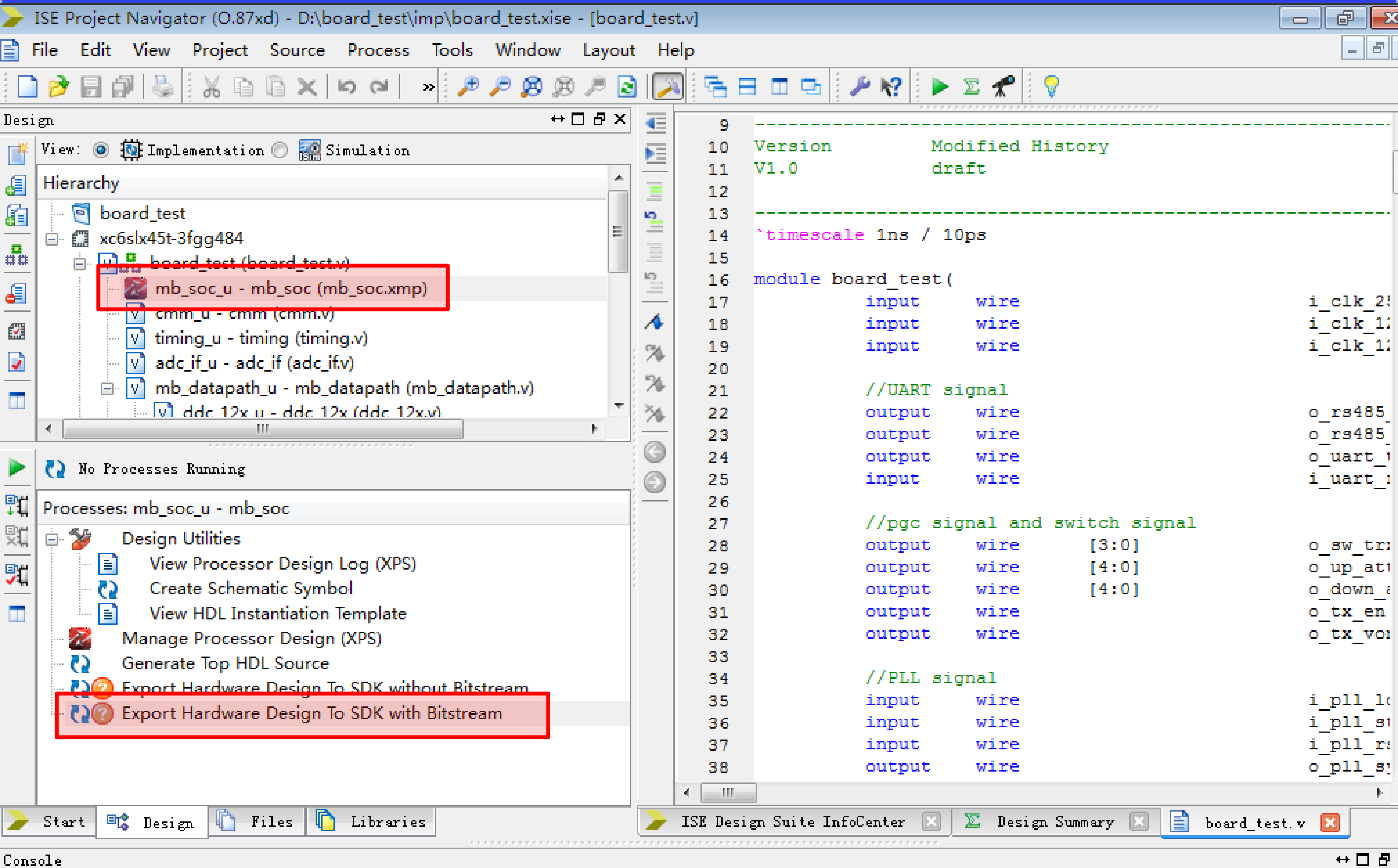
端口设置


Xilinx Platform Studio (EDK_O.87xd) - D:\board_test\mb_soc\mb_soc\mb_soc.xmp - [System Assembly View]
File Edit View Project Hardware Debug Window Help
IP Catalog
Description
IP Version
EDK Install
Analog
Bus and Bridge
Clock, Reset and Inte...
Communication High...
Communication Low-...
DMA and Timer
Debug
FPGA Reconfiguration
General Purpose IO
IO Modules
Interprocessor Com...
Memory and Memo...
PCI
Peripheral Controller
Processor
Utility
Verification
Project Local PCores
Search IP Catalog:
Clear
Project
IP Catalog
Bus Interfaces
Ports
Addresses
Name
Connected Port
Direction
Range
External Ports
dlmb
ilmb
mb_plb
microblaze_0
bram_block_0
dlmb1
ilmb1
lmb_bram
dlmb_cntlr
ilmb_cntlr
share_bram_ctrl
mdm_0
sys_intc
ADC_SPI_INOUT
ADC_SPI_OUTPUT
DAC_SPI_INPUT
DAC_SPI_OUTPUT
E2PROM_I2C_INOUT
E2PROM_I2C_OUTPUT
LO_SPI_OUTPUT
PLL_SPI_INPUT
Legend
Master Slave Master/Slave Target Initiator Connected Unconnected Monitor
Production License (paid) License (eval) Local Pre Production Beta Development
Superseded Discontinued
Start Up Page
System Assembly View
Design Summary
Graphical Design View

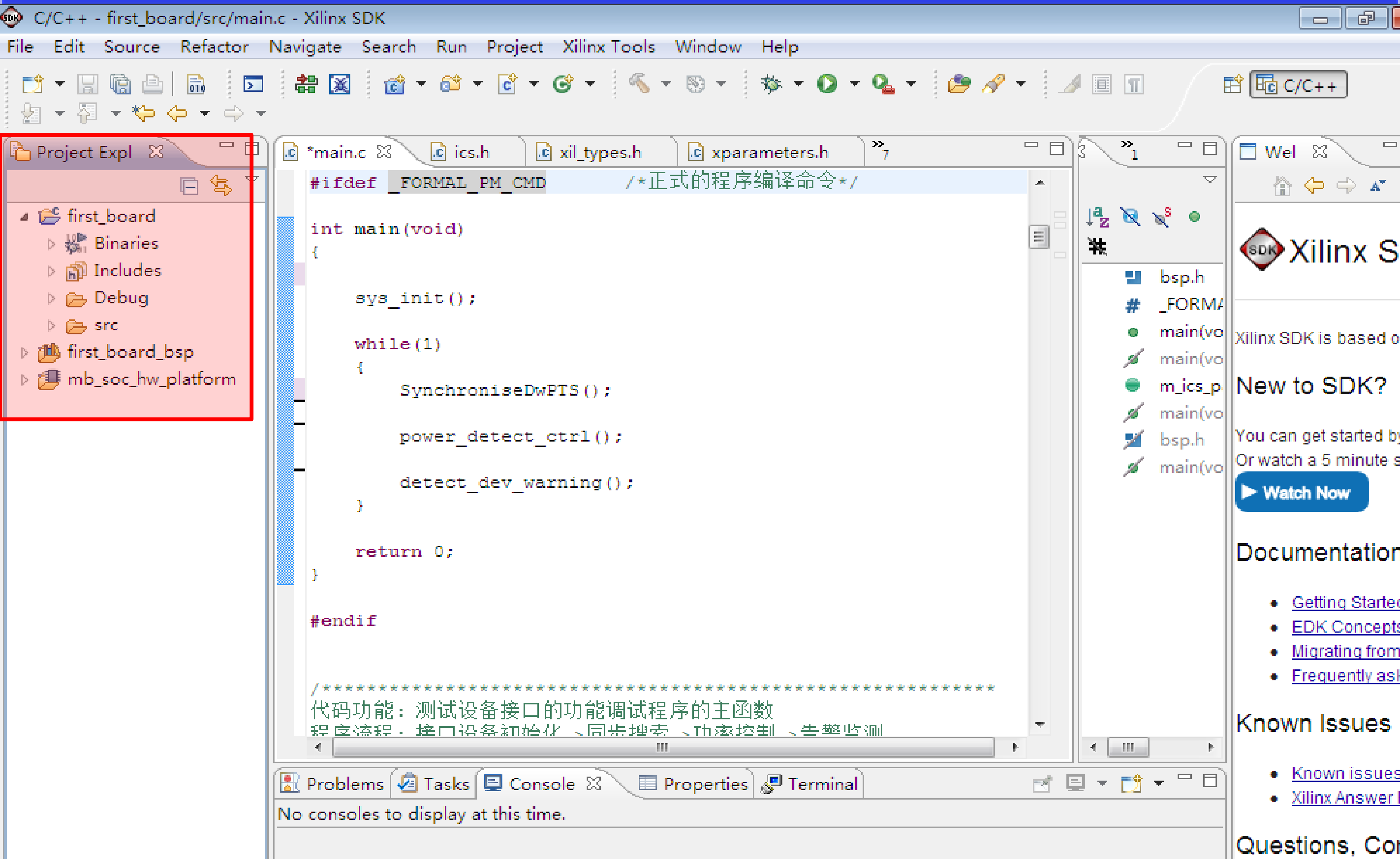
地址空间的管理


Xilinx Platform Studio (EDK_O.87xd) - D:\board_test\mb_soc\mb_soc\mb_soc.xmp - [System Assembly View]
File Edit View Project Hardware Debug Window Help
IP Catalog
Description IP Version
EDK Install
Analog
Bus and Bridge
Clock, Reset and Inte...
Communication High...
Communication Low-...
DMA and Timer
Debug
FPGA Reconfiguration
General Purpose IO
IO Modules
Interprocessor Com...
Memory and Memo...
PCI
Peripheral Controller
Processor
Utility
Verification
Project Local PCores
Search IP Catalog:
Clear
Project
IP Catalog
Bus Interfaces
Ports
Addresses
Instance
Base Name
Base Address
High Address
Size
Bus Interface(s)
microblaze_0's Address Map
dlmb_cntlr C_BASEADDR 0x00000000 0x0000FFFF 64K SLMB
dlmb1 C_BASEADDR 0x00010000 0x00017FFF 32K SLMB
share_bram_ctrl C_BASEADDR 0x00020000 0x0002FFFF 64K SLMB
uart_ctrl_output C_BASEADDR 0x81400000 0x8140FFFF 64K SPLB
PLL_SPI_INPUT C_BASEADDR 0x81410000 0x8141FFFF 64K SPLB
tx_output C_BASEADDR 0x81420000 0x8142FFFF 64K SPLB
LO_SPI_OUTPUT C_BASEADDR 0x81430000 0x8143FFFF 64K SPLB
tsn_shift_en_output C_BASEADDR 0x81440000 0x8144FFFF 64K SPLB
E2PROM_I2C_OUTPUT C_BASEADDR 0x81450000 0x8145FFFF 64K SPLB
switch_tadv_output C_BASEADDR 0x81460000 0x8146FFFF 64K SPLB
E2PROM_I2C_INOUT C_BASEADDR 0x81470000 0x8147FFFF 64K SPLB
switch_ctrl_cfg C_BASEADDR 0x81480000 0x8148FFFF 64K SPLB
DAC_SPI_OUTPUT C_BASEADDR 0x81490000 0x8149FFFF 64K SPLB
power_mult_acc_tsn C_BASEADDR 0x814A0000 0x814AFFFF 64K SPLB
DAC_SPI_INPUT C_BASEADDR 0x814B0000 0x814BFFFF 64K SPLB
power_calc_ctrl C_BASEADDR 0x814C0000 0x814CFFFF 64K SPLB
ADC_SPI_OUTPUT C_BASEADDR 0x814D0000 0x814DFFFF 64K SPLB
power_calc_busy C_BASEADDR 0x814E0000 0x814EFFFF 64K SPLB
ADC_SPI_INOUT C_BASEADDR 0x814F0000 0x814FFFFF 64K SPLB
Legend
Master Slave Master/Slave Target Initiator Connected Unconnected Monitor
Production License (paid) License (eval) Local Pre Production Beta Development
Superseded Discontinued
Start Up Page
System Assembly View
Design Summary
Graphical Design View
Errors

生成硬件平台并打开SDK

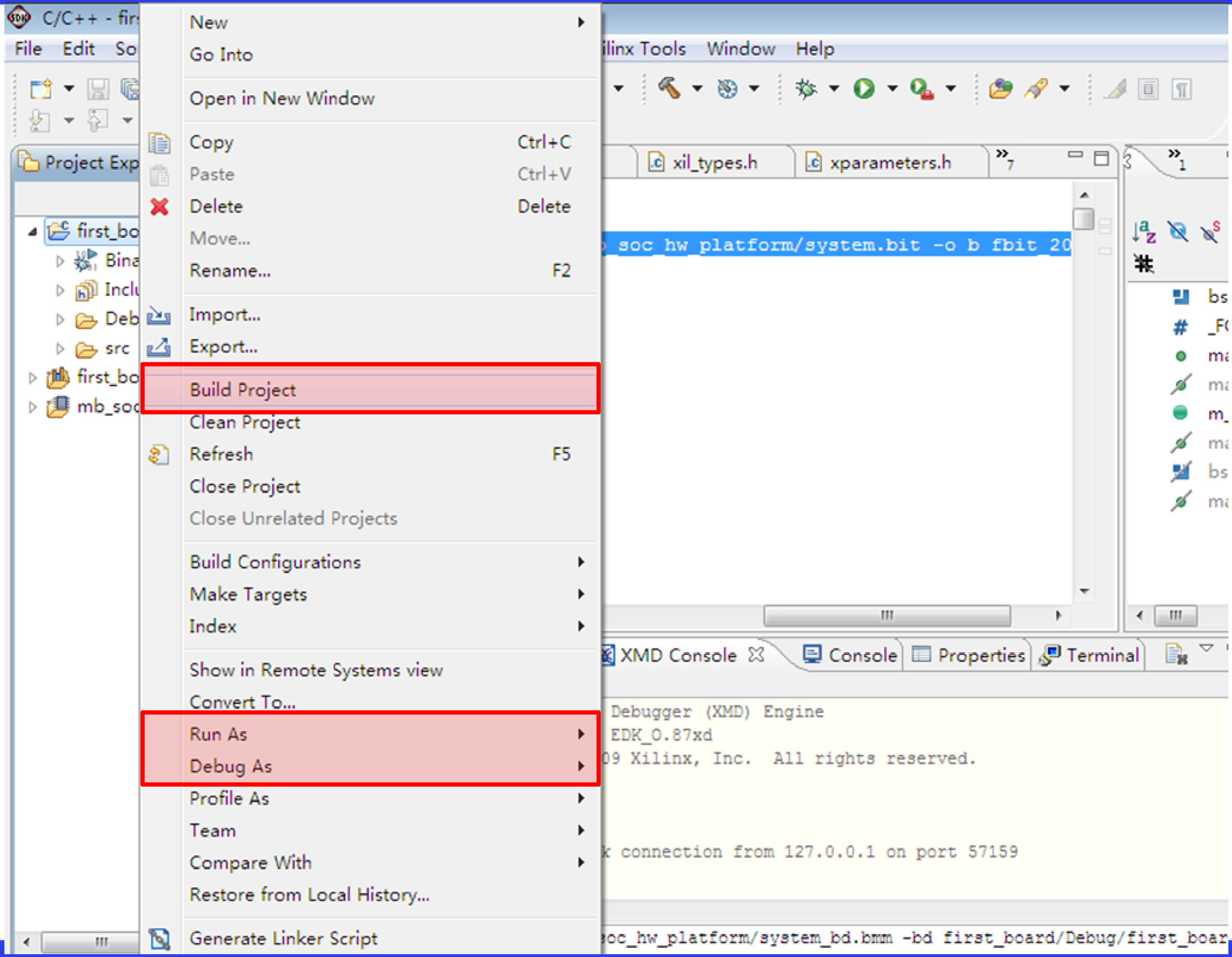


创建软件工程



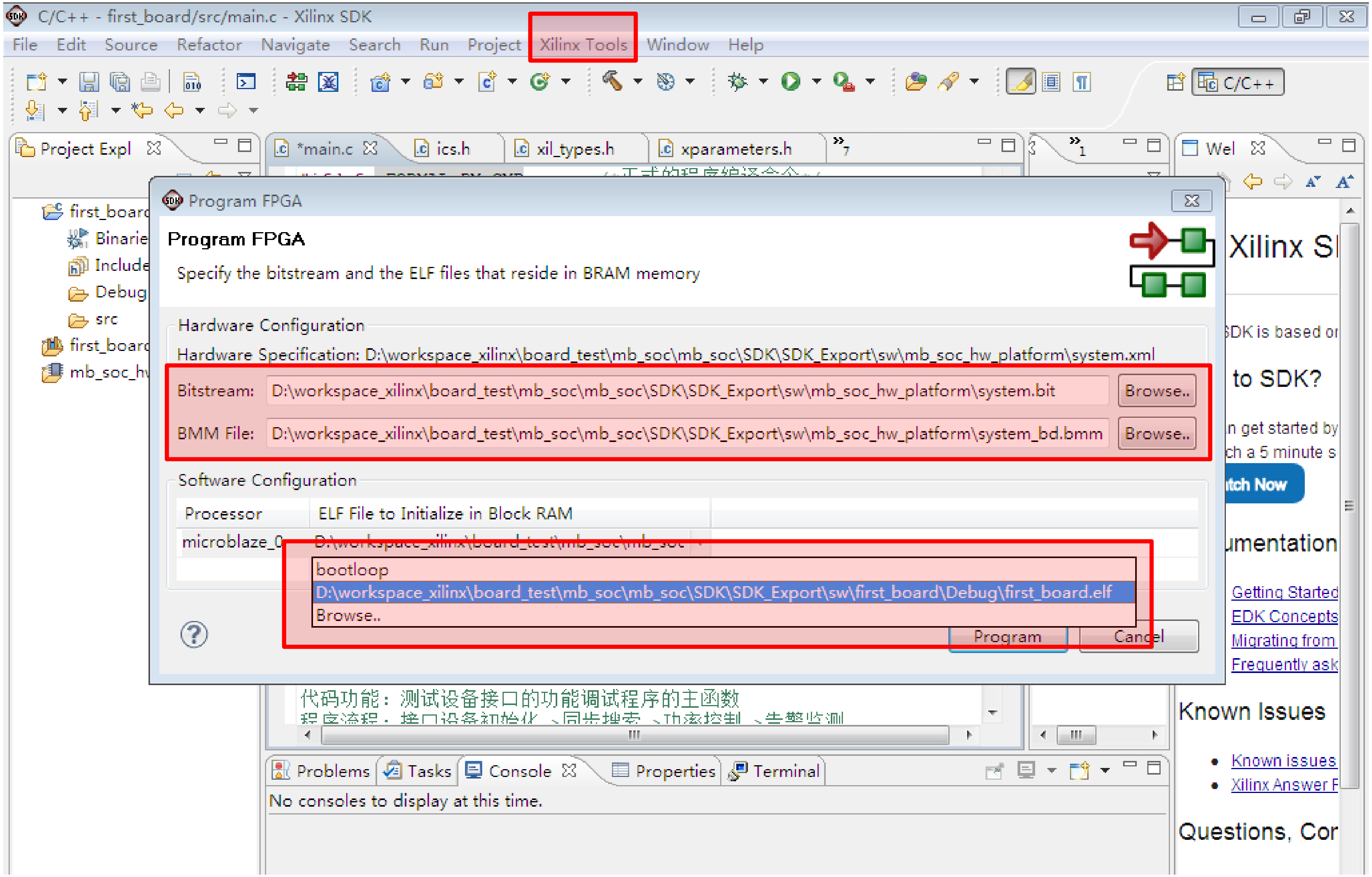
C/C++ - first_board/src/main.c - Xilinx SDK
File Edit Source Refactor Navigate Search Run Project Xilinx Tools Window Help
C/C++
Project Expl
first_board
Binaries
Includes
Debug
src
first_board_bsp
mb_soc_hw_platform
*main.c
ics.h
xil_types.h
xparameters.h
#ifdef _FORMAL_PM_CMD /*正式的程序编译命令*/
int main(void)
{
sys_init();
while(1)
{
SynchroniseDwPTS();
power_detect_ctrl();
detect_dev_warning();
}
return 0;
}
#endif
/***
代码功能：测试设备接口的功能调试程序的主函数
bsp.h
_FORMA
main(vo
m_ics_p
Wel
Xilinx S
Xilinx SDK is based o
New to SDK?
You can get started b
Or watch a 5 minute s
Watch Now
Documentation
Getting Started
EDK Concepts
Migrating from
Frequently ask
Known Issues
Known issues
Xilinx Answer
Questions, Co
Problems
Tasks
Console
Properties
Terminal
No consoles to display at this time.

编译和调试



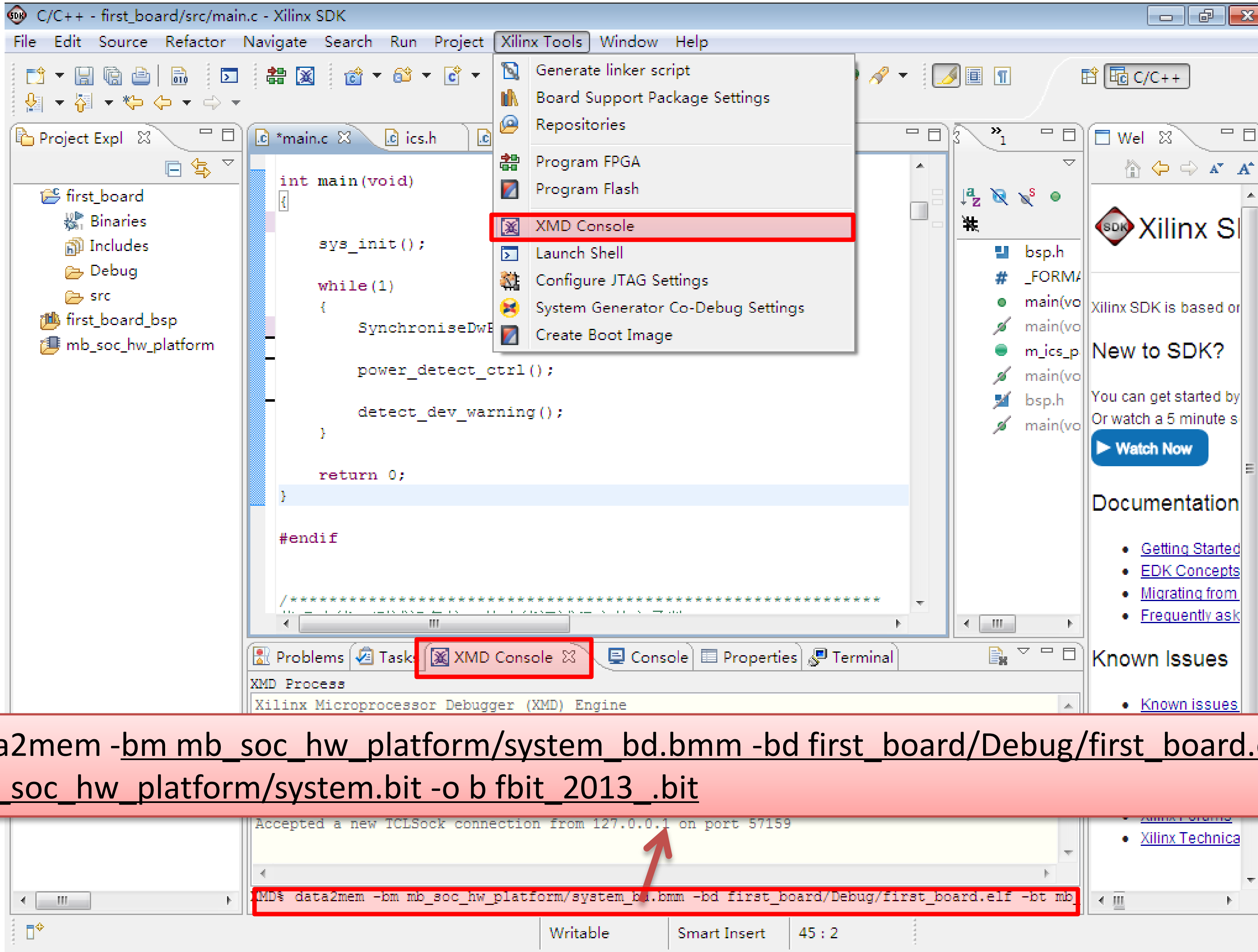
C/C++ - fir
File Edit So
ilinx Tools Window Help
Project Exp
first_bo
Bina
Inclu
Deb
src
first_bo
mb_soc
New
Go Into
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Build Configurations
Make Targets
Index
Show in Remote Systems view
Convert To...
Run As
Debug As
Profile As
Team
Compare With
Restore from Local History...
Generate Linker Script
xil_types.h
xparameters.h
soc hw platform/system.bit -o b fbit 20
XMD Console
Console
Properties
Terminal
Debugger (XMD) Engine
EDK_0.87xd
09 Xilinx, Inc. All rights reserved.
connection from 127.0.0.1 on port 57159
oc_hw_platform/system_bd.bmm -bd first_board/Debug/first_boar

下载目标程序



C/C++ - first_board/src/main.c - Xilinx SDK
File Edit Source Refactor Navigate Search Run Project Xilinx Tools Window Help
Program FPGA
Program FPGA
Specify the bitstream and the ELF files that reside in BRAM memory
Hardware Configuration
Hardware Specification: D:\workspace_xilinx\board_test\mb_soc\mb_soc\SDK\SDK_Export\sw\mb_soc_hw_platform\system.xml
Bitstream: D:\workspace_xilinx\board_test\mb_soc\mb_soc\SDK\SDK_Export\sw\mb_soc_hw_platform\system.bit
BMM File: D:\workspace_xilinx\board_test\mb_soc\mb_soc\SDK\SDK_Export\sw\mb_soc_hw_platform\system_bd.bmm
Browse..
Software Configuration
Processor
ELF File to Initialize in Block RAM
microblaze_0
bootloop
D:\workspace_xilinx\board_test\mb_soc\mb_soc\SDK\SDK_Export\sw\first_board\Debug\first_board.elf
Browse..
Program
Cancel
代码功能：测试设备接口的功能调试程序的主函数
Problems Tasks Console Properties Terminal
No consoles to display at this time.
Known Issues
Known issues

采用XMD工具生成带有ELF的bit文件

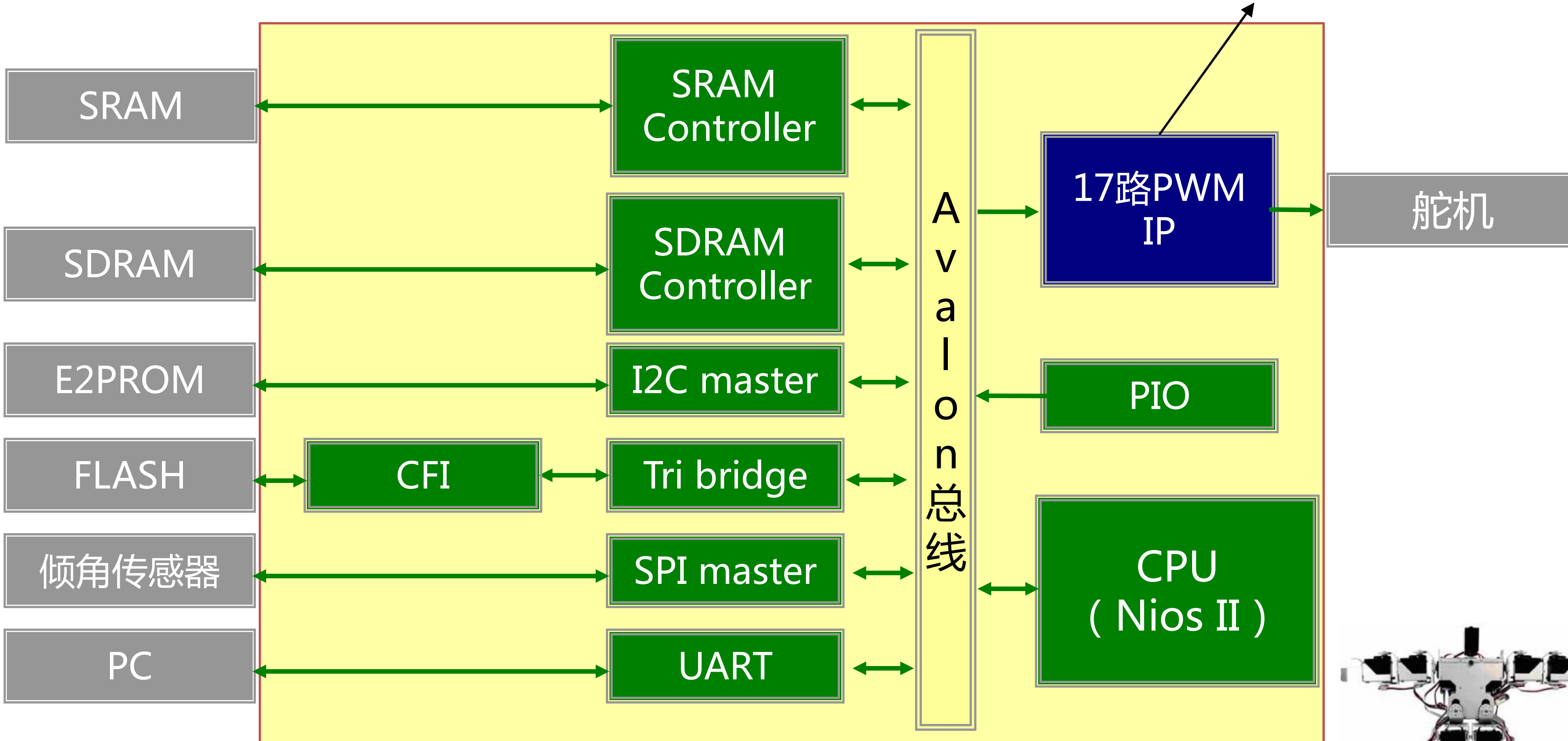


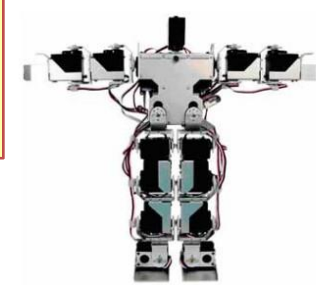
data2mem -bm mb_soc_hw_platform/system_bd.bmm -bd first_board/Debug/first_board.elf -bt mb_soc_hw_platform/system.bit -o b fbit_2013_.bit

SOPC的系统级设计实例讲解（1）

多路PWM实现机器人控制

自定义组件挂到总线上

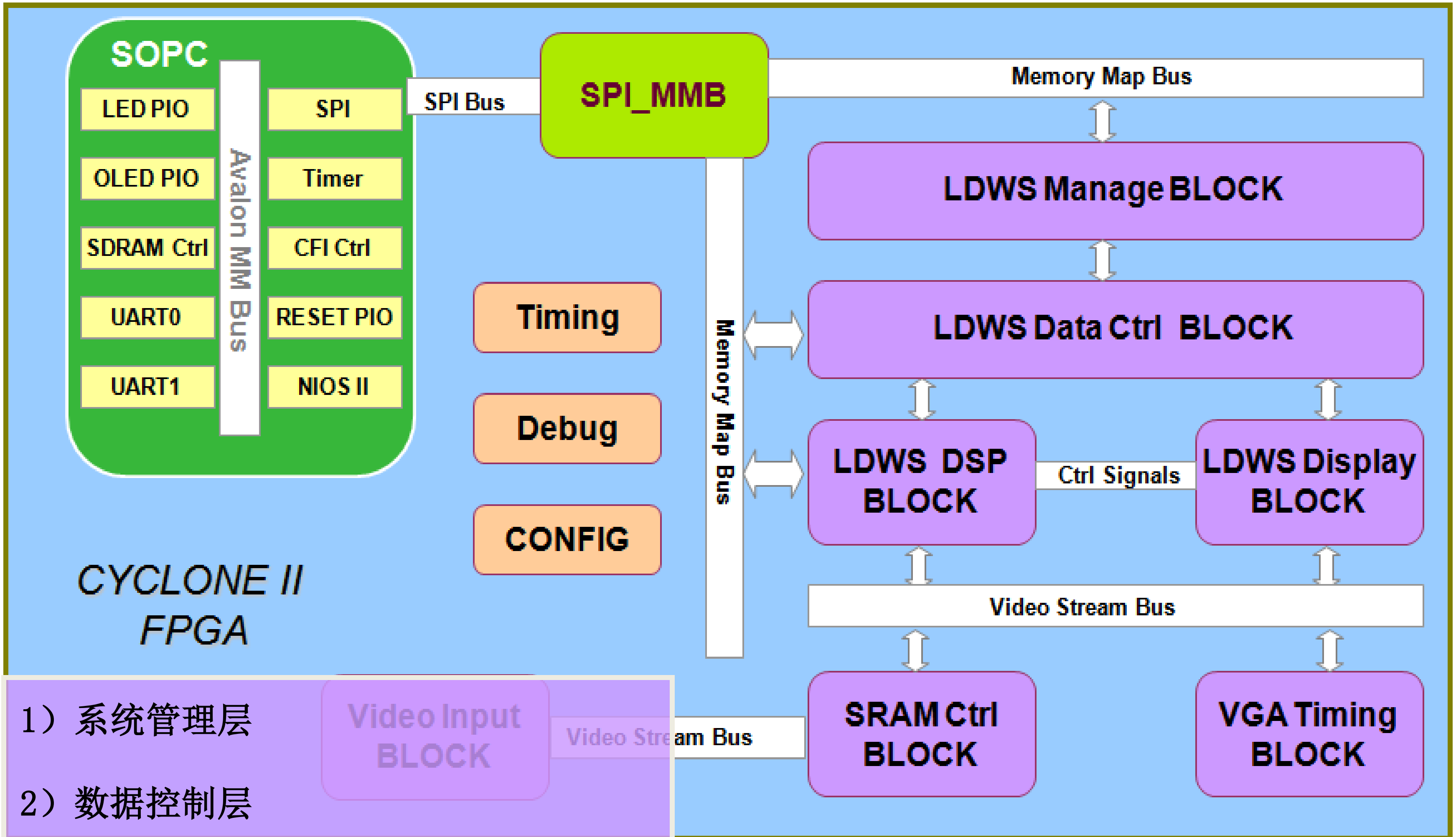




基于NIOS II的智能双足机器人

SOPC的系统级设计实例讲解（2）

车辆安全驾驶辅助系统



1）系统管理层

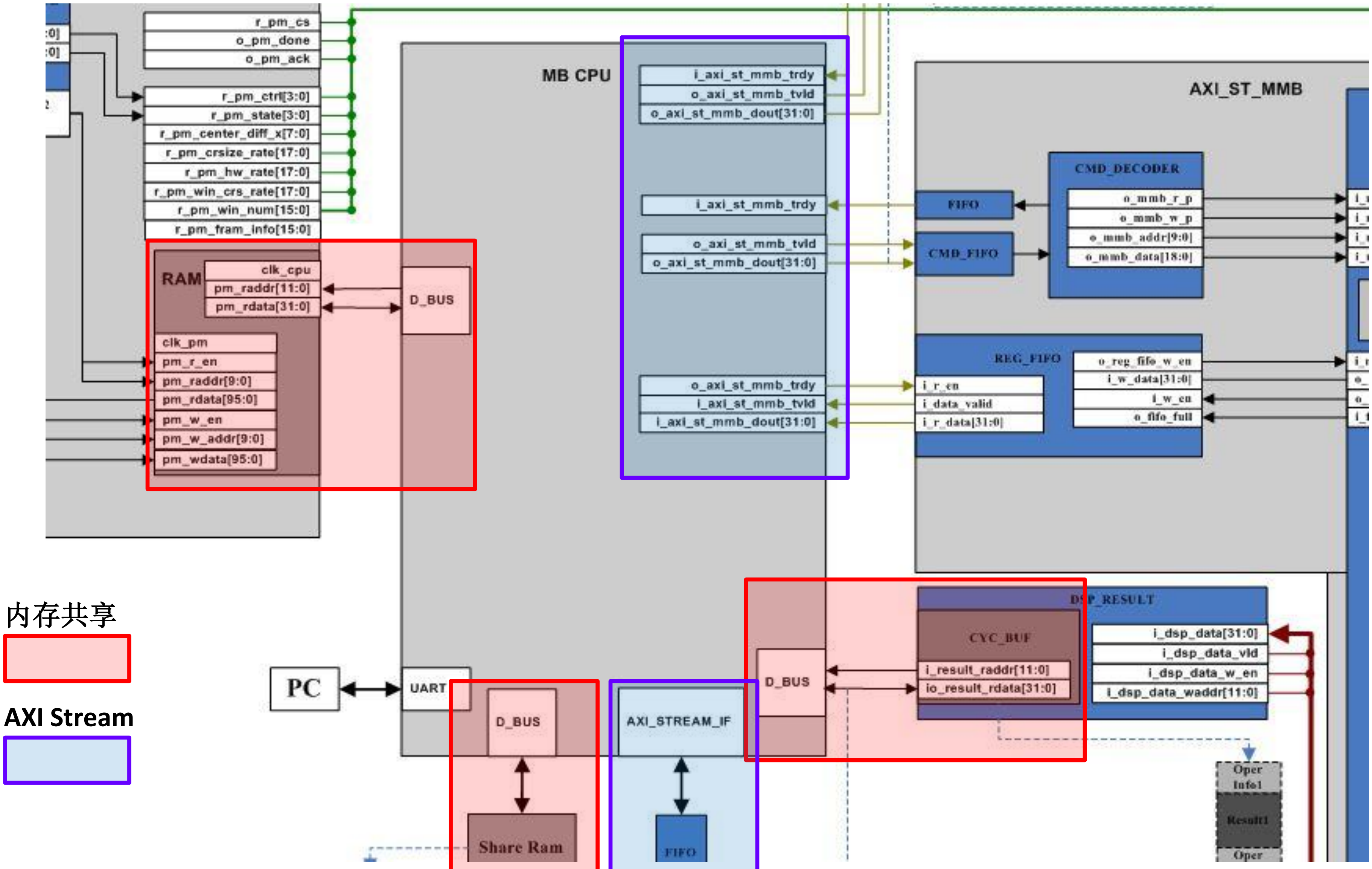
2）数据控制层

3）数字图像处理与显示方案层

4）存储器接口和显示接口层

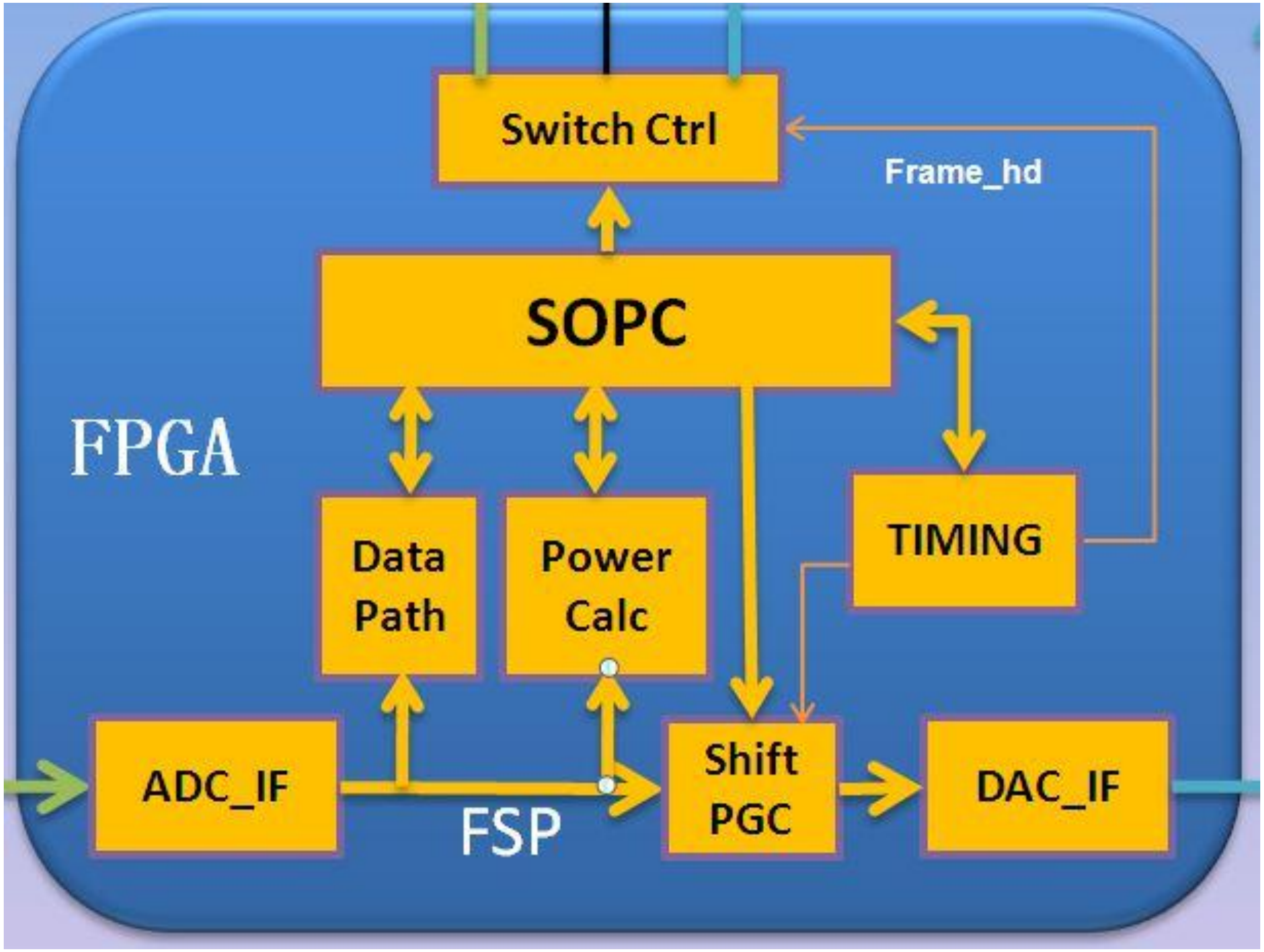
可搜索论文《基于FPGA的车道偏移告警系统》

SOPC的系统级设计实例讲解（3）



一种基于MB的图像处理系统设计

SOPC的系统级设计实例讲解（4）



一种基于MB的功率检测与实时控制系统

对于学习电子技术的一些感受：

◆对感兴趣的领域做深入的了解

◆逐步摸索积累良好的开发习惯

◆敢于做方案的提出者和改进者

在此祝愿大家早日成为一名优秀的工程师哦！

谢谢！

本期的FPGA开发基础公益培训结束，感谢大家的参与

由于个人知识和精力有限，内容难免有不足之处，还请大家多提宝贵意见！