

IP 核使用实例

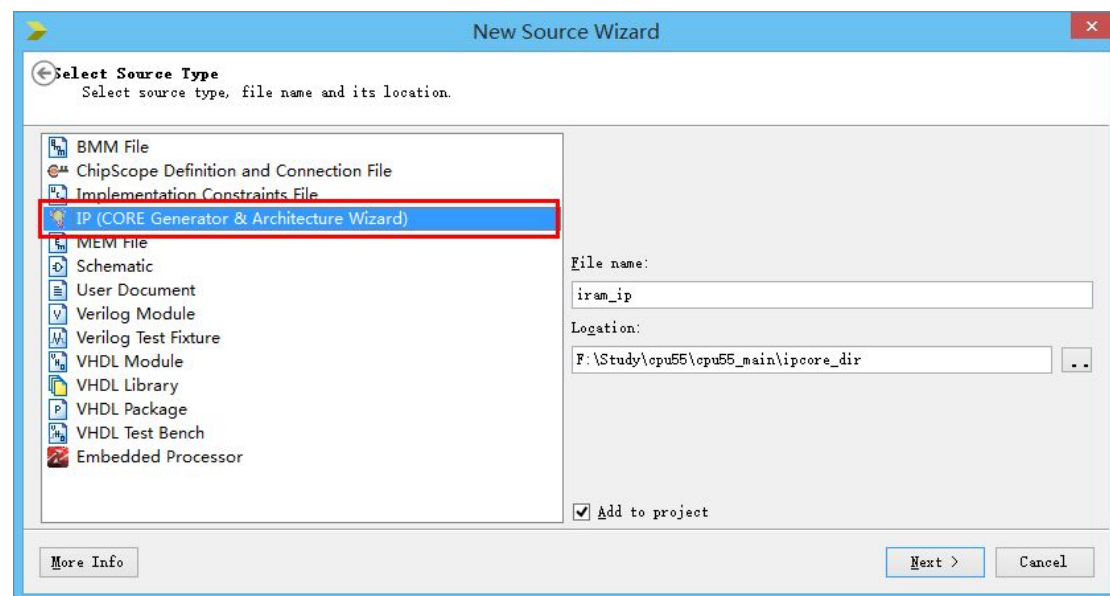
由于我们自行编写的指令 RAM 用来初始化内存的 initial 指令是不可综合的，无法在开发板上运行，所以，我们可以使用 ISE 提供的 ip 核来替换我们的 ram，其可以使用一个 coe 文件来初始化内存

Coe 为初始化 ROM 的配置文件，以下为 coe 文件格式实例

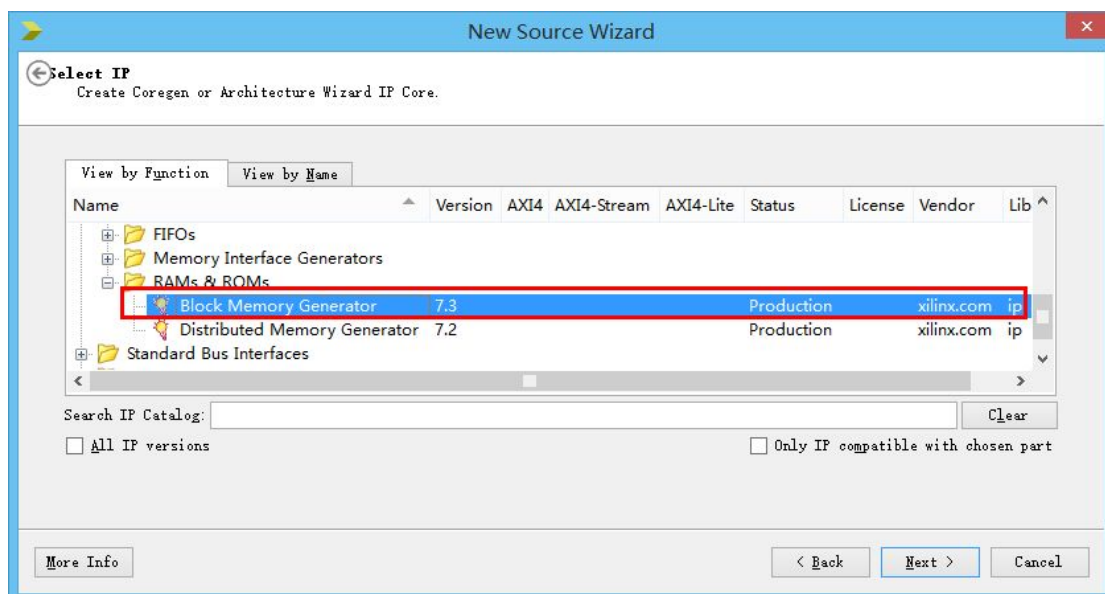
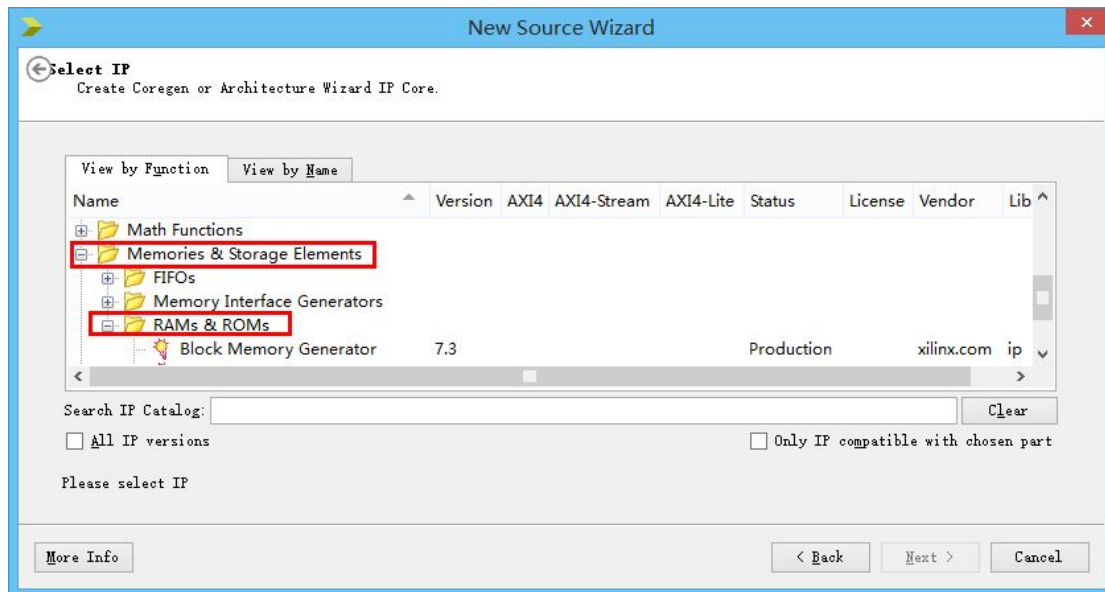
```
memory_initialization_radix=16; #16 表示指令以 16 进制表示，可以按需求更改
memory_initialization_vector=    #由此开始为指令序列，以 “,” 隔开 “;” 结束
00000000,
241d03fc,
0800001a;
```

跟随以下步骤来添加 IP 核：

- 1、右键单击工程，选择 new source
- 2、如图选择 IP 核，填写模块名称



- 3、选择 Memories & Storage Elements 中的 RAMs & ROMs



4、根据下图配置 ip 核，下图均为推荐配置，可根据自己的需求来更改

Block Memory Generator

Documents View

IP Symbol

logiCORE

Block Memory Generator

xilinx.com:ip:blk_mem_gen:7.3

Component Name

Interface Type

☒ Native ☐ AXI4

Mode

Native Interface Block Memory Generator (BMG) are the original standard BMG functions delivered by the previous versions of the LogiCORE Block Memory Generator (prior to v6.x). They are optimized for data storage, width conversion, and clock domain de-coupling functions..

Native Interface BMG cores can be customized to utilize Single Port RAM (SP), Simple Dual Port RAM (SDP), True Dual Port RAM (TDP) and Single Port ROM (SP ROM) configurations. In addition, Native Interface BMG core also support features such as SoftECC/ECC, Pipeline Stages and file based Memory initialization.

IP Symbol

Power Estimation

Datasheet

< Back

Page 1 of 6

Next >

Generate

Cancel

Help

Block Memory Generator

Documents View

IP Symbol

logiCORE

Block Memory Generator

xilinx.com:ip:blk_mem_gen:7.3

Memory Type

Clocking Options

☐ Common Clock

Addressing Options

☐ Enable 32-bit Address

ECC Options

ECC Type
☐ Use Error Injection Pins

Write Enable

☐ Use Byte Write Enable
Byte Size bits

Algorithm

Defines the algorithm used to concatenate the block RAM primitives. See the datasheet for more information.
☒ Minimum Area
☐ Low Power
☐ Fixed Primitives
Primitive (Write Port A) :
Actual Primitive(s) Used : 16kx2, 8kx2

IP Symbol

Power Estimation

Datasheet

< Back

Page 2 of 6

Next >

Generate

Cancel

Help

DocumentsView

IP Symbol

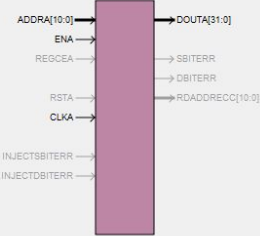


Diagram showing the Block Memory Generator IP Symbol. Inputs: ADDR[10:0], ENA, REGCEA, RSTA, CLKA, INJECTSBITERR, INJECTDBITERR. Outputs: DOUTA[31:0], SBITERR, DBITERR, RDADRECC[10:0].

Block Memory Generator

xilinx.com:ip:blk_mem_gen:7.3

Port A Options

Memory Size

Read Width32Range: 1..4608

Read Depth2048Range: 2..9011200

Operating Mode

☒ Write First

☐ Read First

☐ No Change

Enable

☐ Always Enabled

☒ Use ENA Pin

Datasheet

< Back

Page 3 of 6

Next >

Generate

Cancel

Help

DocumentsView

IP Symbol

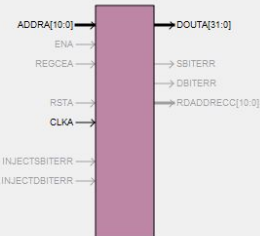


Diagram showing the Block Memory Generator IP Symbol. Inputs: ADDR[10:0], ENA, REGCEA, RSTA, CLKA, INJECTSBITERR, INJECTDBITERR. Outputs: DOUTA[31:0], SBITERR, DBITERR, RDADRECC[10:0].

Block Memory Generator

xilinx.com:ip:blk_mem_gen:7.3

Optional Output Registers

Port A

☐ Register Port A Output of Memory Primitives

☐ Register Port A Output of Memory Core

☐ Register Port A Input of SoftECC logic

☐ Use REGCEA Pin (separate enable pin for Port A output registers)

Pipeline Stages within Mux0Mux Size: 1x1

Memory Initialization

☒ Load Init File

Coe FileF:\Study\cpu55\cpu55_copy\instructions.coe

Browse

Show

☐ Fill Remaining Memory Locations

Remaining Memory Locations (Hex)0

Datasheet

< Back

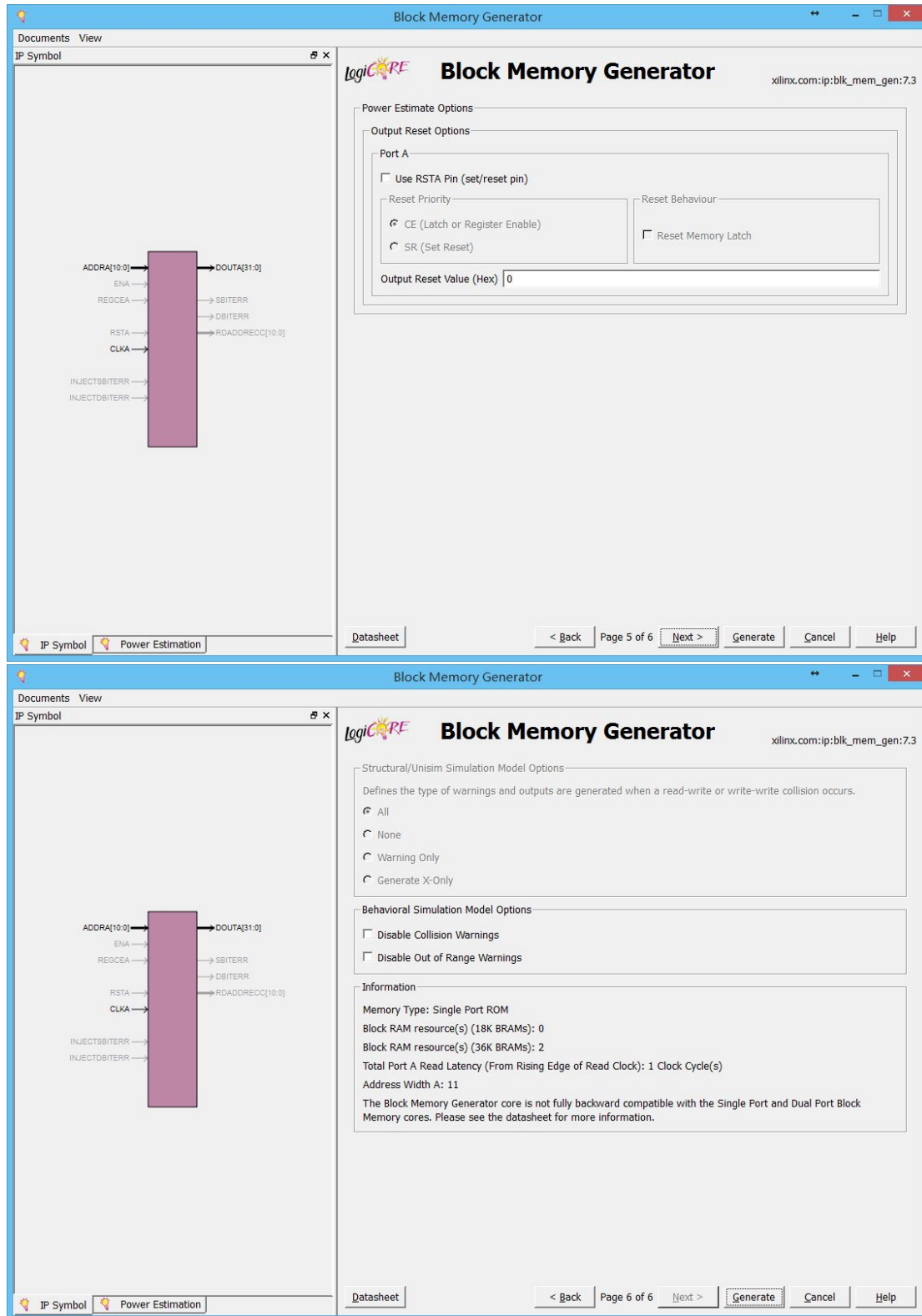
Page 4 of 6

Next >

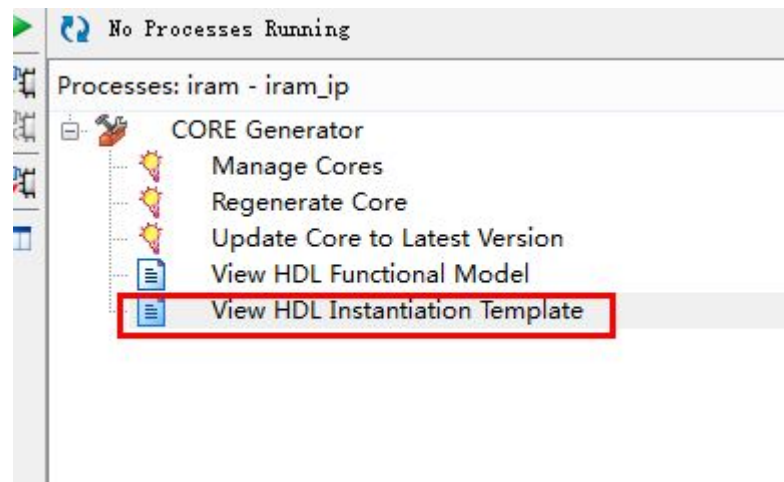
Generate

Cancel

Help



5、点击生成的 ip 核模块，可以看到如下选项，可以更具 ip 核的接口来使用生成的模块（双击 ip 核模块，可以对 ip 核进行修改）



```
//----- Begin Cut here for INSTANTIATION Template ---// INST_TAG
iram_ip your_instance_name (
  .clka(clka), // input clka
  .ena(ena), // input ena
  .addra(addra), // input [10 : 0] addra
  .douta(douta) // output [31 : 0] douta
);
// INST_TAG_END ----- End INSTANTIATION Template -----
```