

## FPGA 片内 ROM 读例程

黑金动力社区 2020-03-13

### 1 实验简介

本实验将为大家介绍如何使用 FPGA 内部的 ROM 以及程序对该 ROM 的数据读操作。

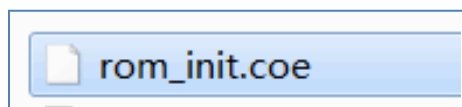
### 2 实验原理

Xilinx 在 VIVADO 里为我们已经提供了 ROM 的 IP 核, 我们只需通过 IP 核例化一个 ROM, 根据 ROM 的读时序来读取 ROM 中存储的数据。实验中会通过 VIVADO 集成的在线逻辑分析仪 ila, 我们可以观察 ROM 的读时序和从 ROM 中读取的数据。

### 3 程序设计

#### 3.1 创建 ROM 初始化文件

既然是 ROM, 那么我们就必须提前给它准备好数据, 然后在 FPGA 实际运行时, 我们直接读取这些 ROM 中预存储好的数据就行。Xilinx FPGA 的片内 ROM 支持初始化数据配置。如下图所示, 我们可以创建一个名为 rom\_init.coe 的文件, 注意后缀一定是“.coe”, 前面的名称当然你可以随意起。



ROM 初始化文件的内容格式很简单, 如下图所示。第一行为定义数据格式, 16 代表 ROM 的数据格式为 16 进制。从第 3 行开始到第 34 行, 是这个 32\*8bit 大小 ROM 的初始化数据。每行数字后面用逗号, 最后一行数字结束用分号。

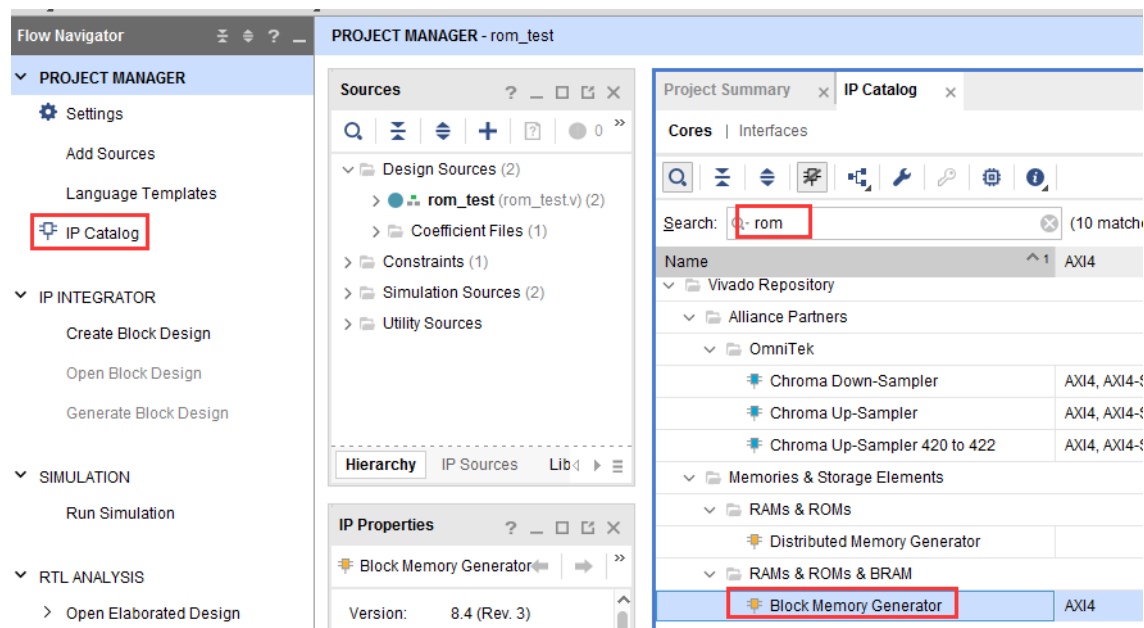
```
rom_test.v  rom_init.coe
1 MEMORY_INITIALIZATION_RADIX=16; //表示ROM内容的数据格式是16进制
2 MEMORY_INITIALIZATION_VECTOR=
3 11,
4 22,
5 33,
6 44,
7 55,
8 66,
9 77,
10 88,
11 99,
12 aa,
13 bb,
14 cc,
15 dd,
16 ee,
17 ff,
18 00,
19 a1,
20 a2,
21 a3,
22 a4,
23 a5,
24 a6,
25 a7,
26 a8,
27 b1,
28 b2,
29 b3,
30 b4,
31 b5,
32 b6,
33 b7,
34 b8; //每个数据后面用逗号或者空格或者换行符隔开，最后一个数据后面加分号
35
```

rom\_init.coe 编写完成后保存一下, 接下去我们开始设计和配置 ROM IP 核。

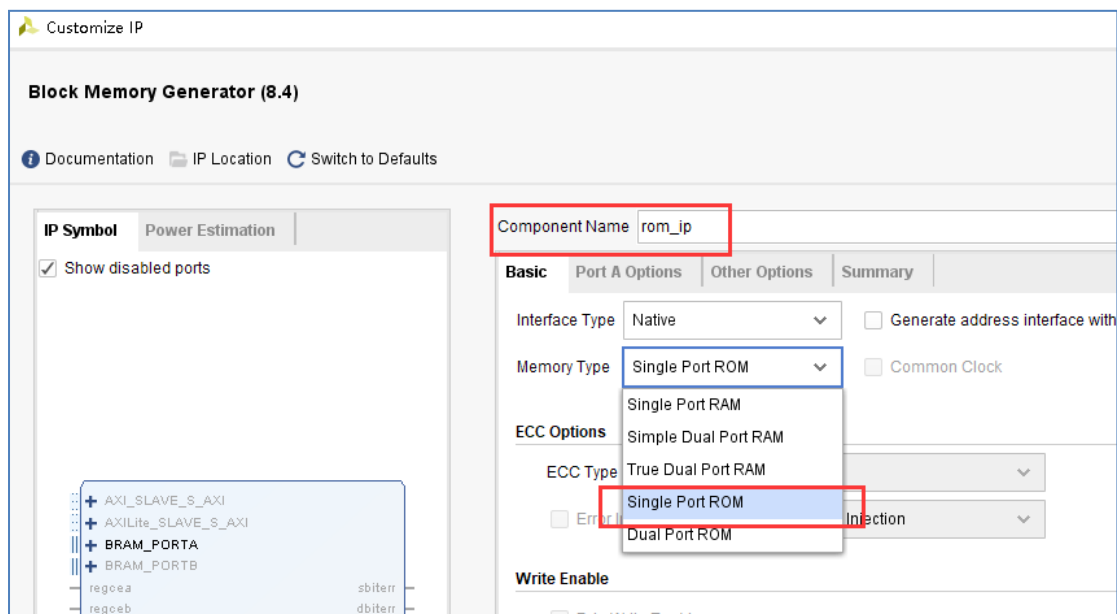
### 3.2 ROM IP 的添加和配置

在添加 ROM IP 之前先新建一个 rom\_test 的工程, 然后在工程中添加 ROM IP , 方法如下 :

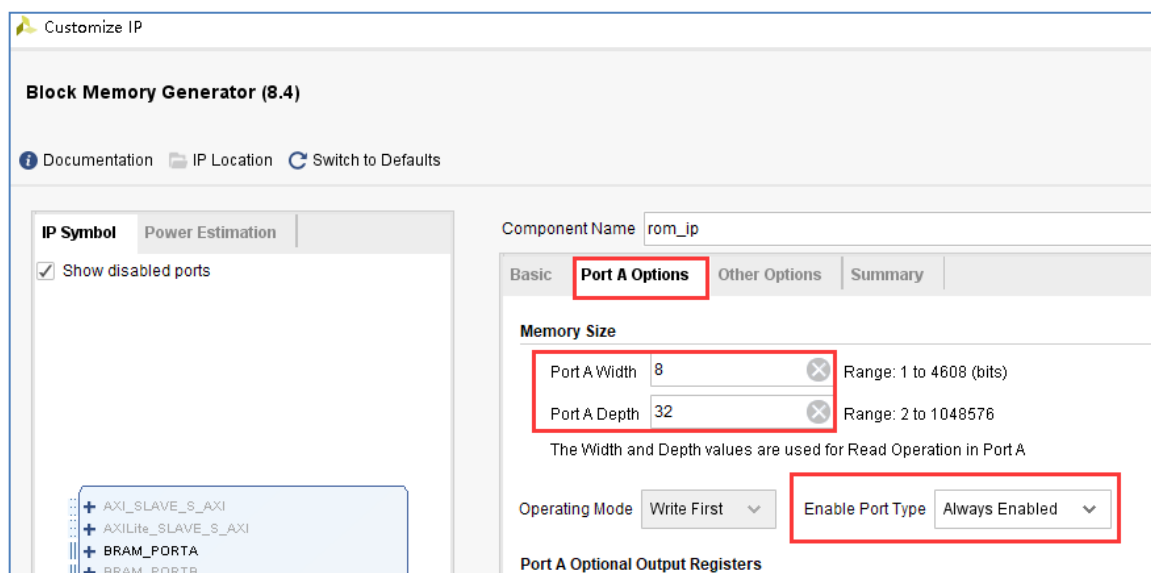
1. 点击下图中 IP Catalog , 在右侧弹出的界面中搜索 rom , 找到 Block Memory Generator, 双击打开。



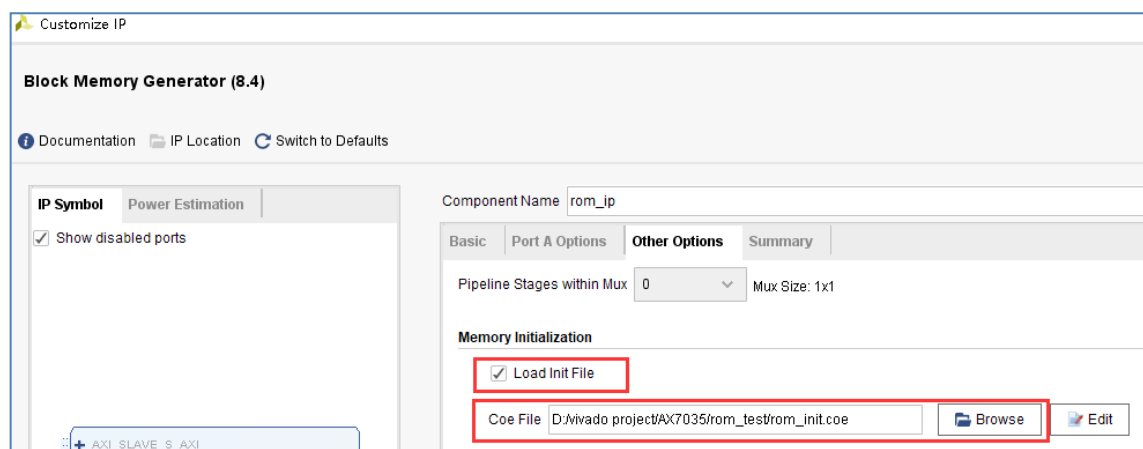
2. 将 Component Name 改为 rom\_ip, 在 Basic 栏目下, 将 Memory Type 改为 Single Port ROM.



3. 切换到 Port A Options 栏目下, 将 ROM 位宽 Port A Width 改为 8, 将 ROM 深度 Port A Depth 改为 32, 使能管脚 Enable Port Type 改为 Always Enable.



8. 切换到 Other Options 栏目下，勾选 Load Init File，点击 Browse，选中之前制作好的.coe 文件。



9. 点击 ok，点击 Generate 生成 ip 核。

### 3.3 ROM 测试程序编写

ROM 的程序设计非常简单, 在程序中我们只要每个时钟改变 ROM 的地址, ROM 就会输出当前地址的内部存储数据。ROM IP 的实例化及程序设计如下:

```

/*****
Generate ROM address
*****/
always @ (posedge sys_clk or negedge rst_n)
begin
    if(rst_n == 1'b0)
        rom_addr <= 10'd0;
    else
        rom_addr <= rom_addr+1'b1;
    end
/*****
calling xilinx rom IP
*****/
rom_ip rom_ip_inst
(
    .clka      (sys_clk      ),    //input clka
    .addra     (rom_addr     ),    //input [4:0] addra
    .douta     (rom_data     )    //output [7:0] douta
);
/*****

```

为了能实时看到 ROM 中的数据，我们这里添加了 ila 工具来观察 ROM 的数据信号。关于如何生成 ila 大家请参考"I2C 接口 EEPROM 实验.pdf"教程。

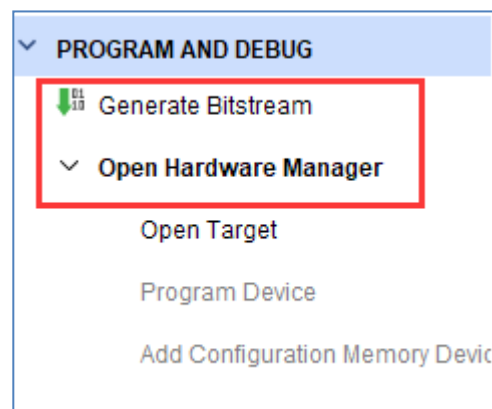
```

46 : ila_0 ila_m0
47 : (
48 :     .clk      (sys_clk),
49 :     .probe0    (rom_data)
50 : );

```

## 4 实验现象

添加 XDC 文件，生成 bit 文件后打开 Hardware Manager 将 bit 文件下载到 FPGA。



在 Waveform 的窗口我们可以看到 rom\_data 的数据正是我们存放在 ROM 中的 coe 文件。

