

19. ROM IP 核使用实验例程

19.1 实验简介

ROM 即只读存储器（Read Only Memory），在程序的运行过程中他只能被读取，无法被写入，因此我们应该在初始化的时候就给他配置初值，一般是在生成 IP 的时候通过导入后缀为. dat 的文件对其进行初值配置。

注意：PDS 的 IP 配置工具中提供两种不同的 ROM，一种是 Distributed ROM(分布式 ROM) 另一种是 DRM Based ROM，分布式 ROM 用的是 LUT(查找表) 资源去构成的 ROM，这种 ROM 会消耗大量 LUT 资源，因此通常在一些比较小的存储才会用到这种 RAM，以节省 DRAM 资源。而 DRM Based ROM 是利用片内的 DRAM 资源去构成的 ROM，不占用逻辑资源，而且速度快，通常设计中均使用 DRM Based ROM。

19.2 实验目的

掌握 ROM 对数据的读取并灵活根据需求更改

19.3 实验设计

使用 matlab 工具生成我们的.dat 文件作为我们 ROM IP 读取的数据源
利用 ROM IP 核读取预先存入数据的.dat 文件。

Matlab 代码如下：

```
clc;
clear all;
close all;
% 生成从 0 到 127 的数字
numbers = 0:127;
% 打开一个名为 'data.dat' 的文件以写入
fileID = fopen('e:/work/test/ROM/src/data.dat', 'w');
% 遍历数字数组，并将每个数字以十六进制格式写入文件
for i = 1:length(numbers)
    fprintf(fileID, '%02X\n', numbers(i));
end
% 关闭文件
fclose(fileID);
disp('数据已写入到 data.dat 文件中');
```

IP 配置

进入 IP 选择界面后

第一步选择 DRM ROM;

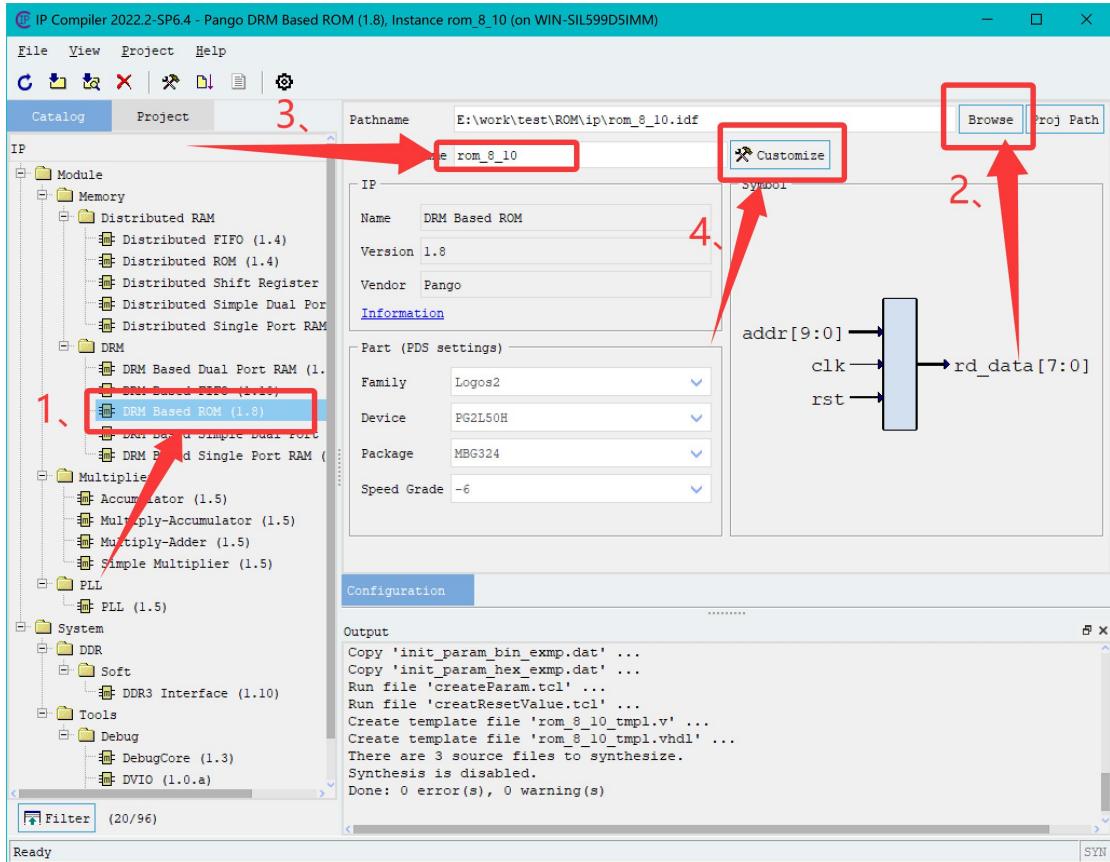
第二步选择我们 IP 的存放路径;

第三步编辑 IP 的名称，这里我们名称为 rom_8_10。从名称可以获取的信息为这个 IP 的类型，数据位宽为 8bit，地址位宽为 10bit。

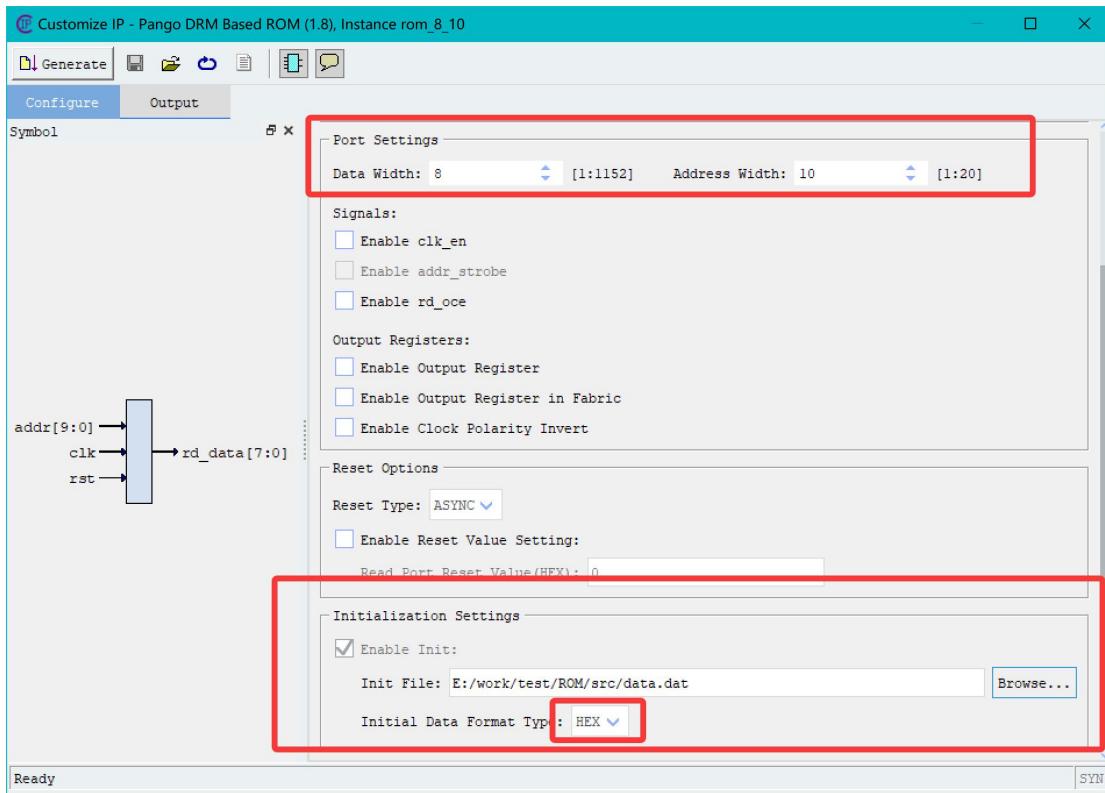
命名规则为：IP 类型_数据位宽_地址位宽；

这样命名可以直观的看到 IP 的重要配置信息，我们用户一般关心的参数一目了然

第四步进入 IP 配置页进行参数的配置



进入参数配置界面选择所需要的数据位宽以及地址位宽即可，其余可保持默认设置。最后选择我们.dat 文件的位置以及数据的进制，即可完成 ROM IP 的配置



所有参数配置完成之后我们点击 **Generate**,

例化我们的 IP 核

激励我们在仿真文件中编写

```
module ROM_test
(
    input    wire          i_clk      ,
    input    wire          i_RST      ,
    input    wire [9 :0]    i_ADDR     ,
    output   wire [7 :0]    o_DATA
);

    rom_8_10 rom_8_10_u0
    (
        .addr      (i_ADDR      ),
        .clk       (i_CLK       ),
        .rst       (i_RST       ),
        .rd_data  (o_DATA      )
    );
endmodule
```

仿真激励我们只需要让地址从零开始递增即可，我这里已知数据有 128 个，因此我的地址只需要从 0 到 127 即可

```
always @(posedge r_clk) begin
    if(r_RST)
        r_addr <= 'd0;
```

```

        else if(r_addr == 'd127)
            r_addr <= 'd0;
        else
            r_addr <= r_addr + 1'd1;
    end

```

需要注意要例化这一语句不然无法进入仿真

```

GTP_GRS GRS_INST
(
    .GRS_N(1'b1)
);

```

端口连接

```

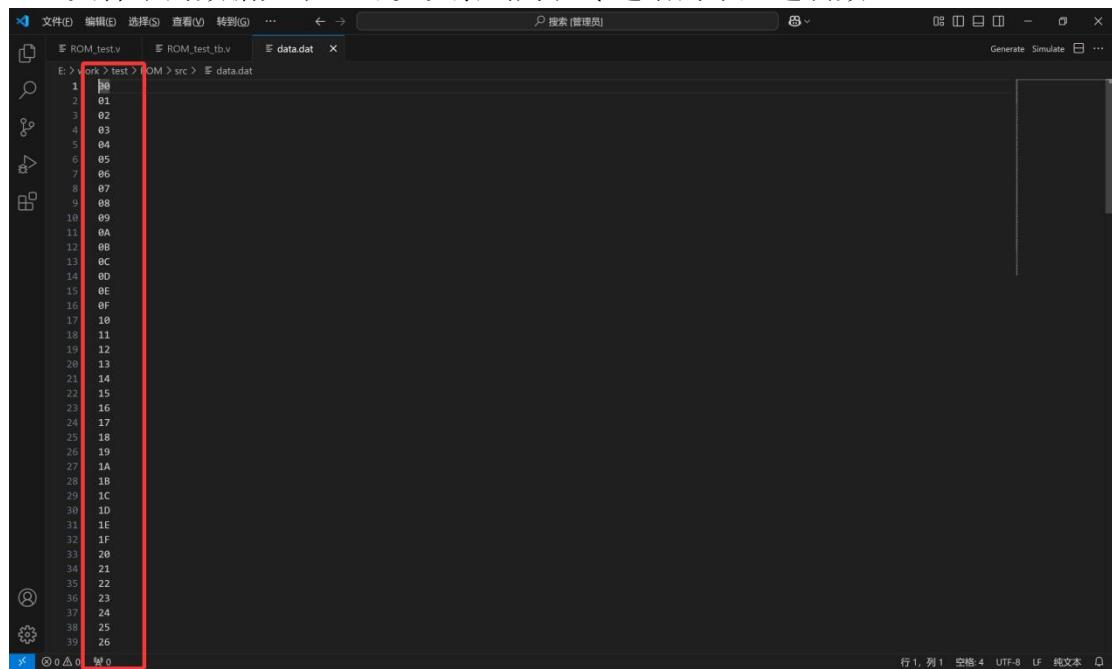
ROM_test ROM_test_u0
(
    .i_clk          (r_clk      ),
    .i_rst          (r_rst      ),
    .i_addr         (r_addr     ),
    .o_data         (w_data     )
);

```

19.4 仿真波形

可以看到和我们的设计一致：复位完成后开始读取.dat 文件中的数据

.dat 文件中的数据如下：可以观察到为从零递增的十六进制数



```

1 00
2 01
3 02
4 03
5 04
6 05
7 06
8 07
9 08
10 09
11 0A
12 0B
13 0C
14 0D
15 0E
16 0F
17 10
18 11
19 12
20 13
21 14
22 15
23 16
24 17
25 18
26 19
27 1A
28 1B
29 1C
30 1D
31 1E
32 1F
33 20
34 21
35 22
36 23
37 24
38 25
39 26

```

图 1.1 .dat 文件中的数据

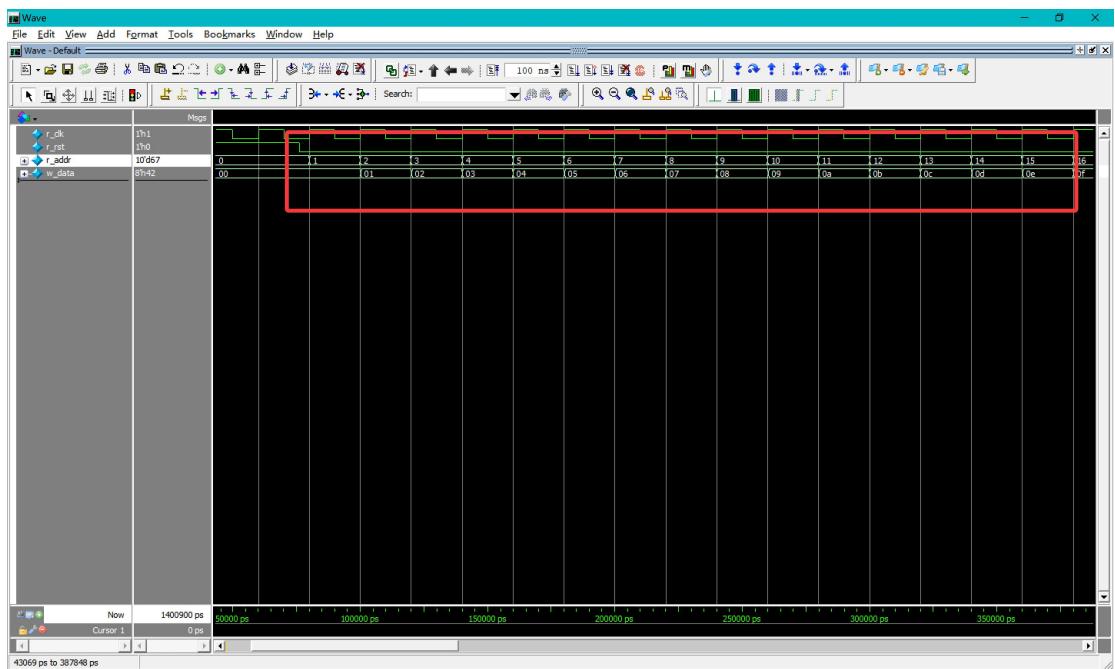


图 1.2 仿真波形

至此，我们 ROM 的实验已经成功完成！！！