计算机组成与结构专题实验

# 实验报告

# 第二次 LPM\_ROM及LPM\_RAM

## 实验目的

### LPM\_ROM实验

1、 掌握 FPGA 中 lpm\_ROM 的设置，作为只读存储器 ROM 的工作特性和配置方法；

2、 用文本编辑器编辑 mif 文件配置 ROM，学习将程序代码以 mif 格式文件加载于 lpm\_ROM中；

3、 在初始化存储器编辑窗口编辑 mif 文件配置 ROM；

4、 验证 FPGA 中 LPM\_ROM 的功能。

### LPM\_RAM实验

1、 了解 FPGA 中 lpm\_ram\_dq 的功能；

2、 掌握 lpm\_ram\_dq 的参数设置和使用方法；

3、 掌握 lpm\_ram\_dq 作为随机存储器 RAM 的仿真测试方法，工作特性和读写方法。

## 实验原理

### LPM\_ROM实验

LPM\_ROM 是 Quartus II 软件中的一个 IP 核，用于实现 ROM（只读存储器）功能。利用 Quartus II 软件提供的 IP 核，将用户定义的数据加载到 FPGA（现场可编程门阵列）中，实现在 FPGA 内部实现的只读存储器功能。

选择实验电路模式为0， 24 位数据输出由数码 8 至数码 3 显示(PIO47~PIO24)， 6 位地址由键 2(PIO12~PIO15)、键 1(PIO8~PIO11)输入，键 1 负责低 4 位，地址锁存时钟 CLK 由键 8(PIO7)控制，每一次上升沿，将地址锁入，数码管 8/7/6/5/4/3 将显示 ROM 中输出的数据。发光管 8 至 1 显示输入的 6 位地址值。

### LPM\_RAM实验

随机存取存储器（RAM）是一种计算机主存储器，它允许数据随机读写，而不受数据存取顺序的限制。

在 FPGA 中利用嵌入式阵列块 EAB 可以构成存储器，数据从 ram\_dp0的左边 D[7..0]输入，从右边 Q[7..0]输出， R/W 为读/写控制信号端。当输入数据和地址准备好以后，在 inclock是地址锁存时钟，当信号上升沿到来时，地址被锁存，数据写入存储单元。数据的读出控制是从 A[7..0]输入存储单元地址，在 CLK 信号上升沿到来时，该单元数据从 Q[7..0]输出。 R/W 是读/写控制端，低电平时进行读操作，高电平时进行写操作； CLK 是读/写时钟脉冲信号； DATA[7..0]是 RAM\_dq0 的 8 位数据输入端； A[7..0]是 RAM 的读出和写入地址； Q[7..0]是 RAM\_dq0 的 8 位数据输出端。

## 实验任务

### LPM\_ROM实验

1、要求用 LPM 元件库设计 LPM\_ROM，地址总线宽度 address[]和数据总线宽度 q[]分别为 6 位和 24位。

2、建立相应的工程文件，设置 lpm\_rom 数据参数， lpm\_ROM 配置文件的路径（ROM\_A.mif），并设置在系统 ROM/RAM 读写允许，以便能对 FPGA 中的 ROM 在系统读写。

3、锁定输入输出引脚并完成编译。

4、下载 SOF 文件至 FPGA，改变键 1、 2 输入地址 lpm\_ROM 的地址 a[5..0]， 每改变一次， 外加一次键 8CLK 读脉冲，通过实验台上的数码管 8~3 比较读出的数据是否与初始化数据(rom4.mif 中的数据)一致。

5、记录实验数据，写出实验报告，给出仿真波形图。

6、 打开 QuartusII 的在系统存储模块读写工具图 4-10，了解 FPGA 中 ROM 中的数据，并对其进行在系统写操作， 实时修改 ROM 数据，载到系统上看是否对应实时写的数据。

### LPM\_RAM实验

1、 按电路图，进行编译、引脚锁定、下载至FPGA；

2、 锁定引脚： 通过键 1（PIO0~PIO3)、键 2（PIO4~PIO7)输入 RAM 的 8 位数据（选择实验电路模式 1），键 3（PIO8~PIO11)、键 4（PIO12~PIO15)输入存储器的 8 位地址。键 8(PIO49)控制读/写允许，低电平时读允许，高电平时写允许；键 7(PIO48)CLK0 产生读/写时钟脉冲，即生成写地址锁存脉冲，对lpm\_ram\_dq 进行写/读操作； Q 输出显示与数码管 7、 8（PIO24~PIO31)。

3、 lpm\_ram\_dq 也能加入初始化文件（这里是 5\_ram.mif 是后面将要用到的模型 CPU 执行微程序文件），注意此文件加入的路径表达和文件表达方式： 5\_ram.mif（注意后缀 mif 要小写）；选择RAM 的 ID 名取为： ram1。

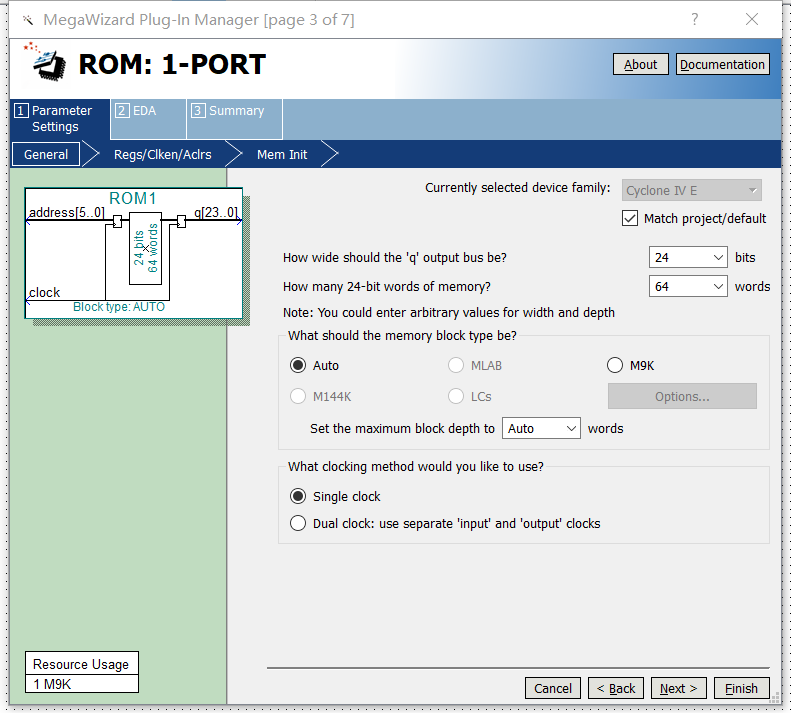
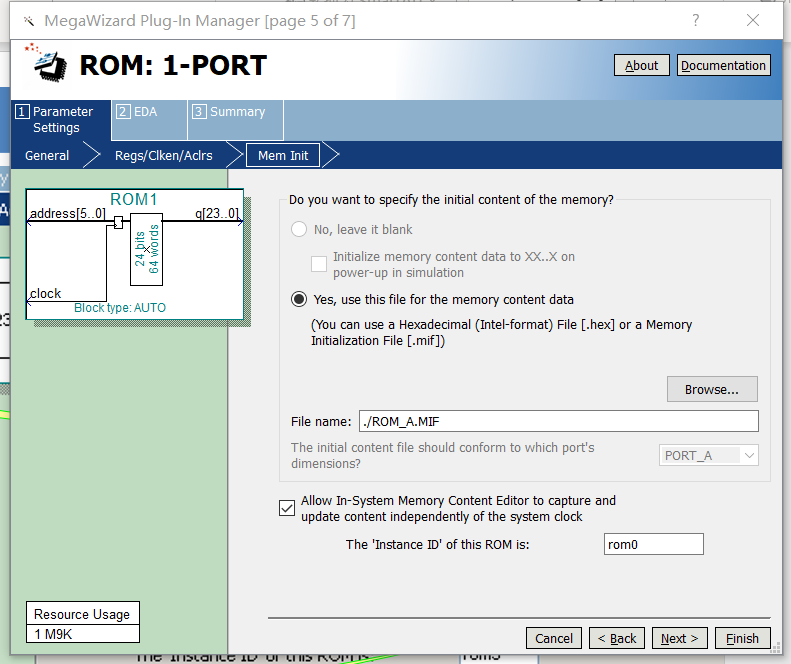
4、 实验中选择实验电路模式为 NO.1，按以上方式进行验证实验。 键 8 置‘0’，键 7CLK 每 0-1-0按动按键， 控制读出初始化数据，与载入的初始化文件 ram\_dp1.mif 中的数据进行比较，然后键 8 置‘1’每键 3、 4 输入一次地址，键 1、 2 输入一次数据，键 7 时钟控制写入，记录写入的一些数据， 再键 8 置‘1’。根据输入的地址的数据，键 7 时钟控制读出的数据与记录的数据比较。

5、 使用在系统读写 RAM 的工具对其中的数据进行读写操作，设置成连续读模式，将在系统读写工具窗口的数据与实验箱上数码管 7、 8 上显示的数据进行对比。

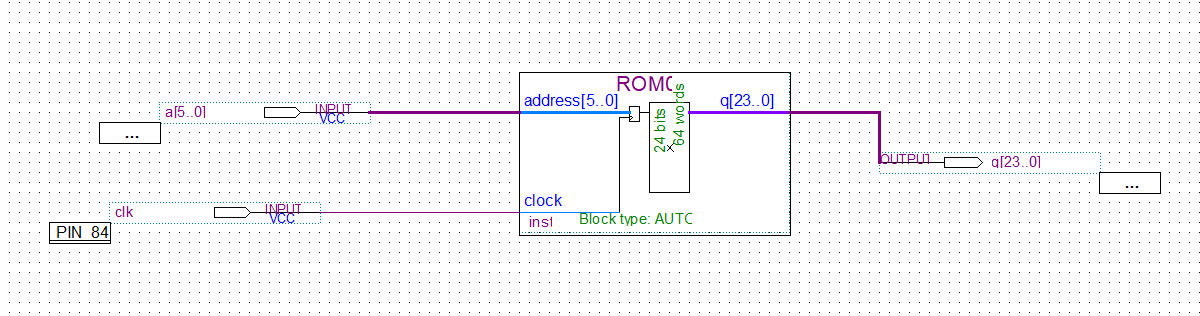
## 实验步骤及结果

### LPM\_ROM实验

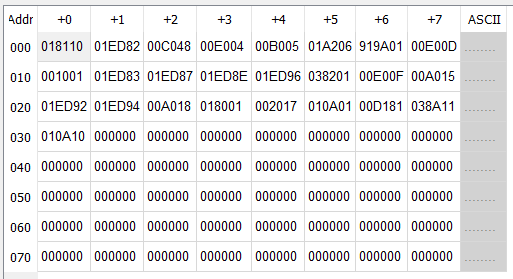
设计ROM

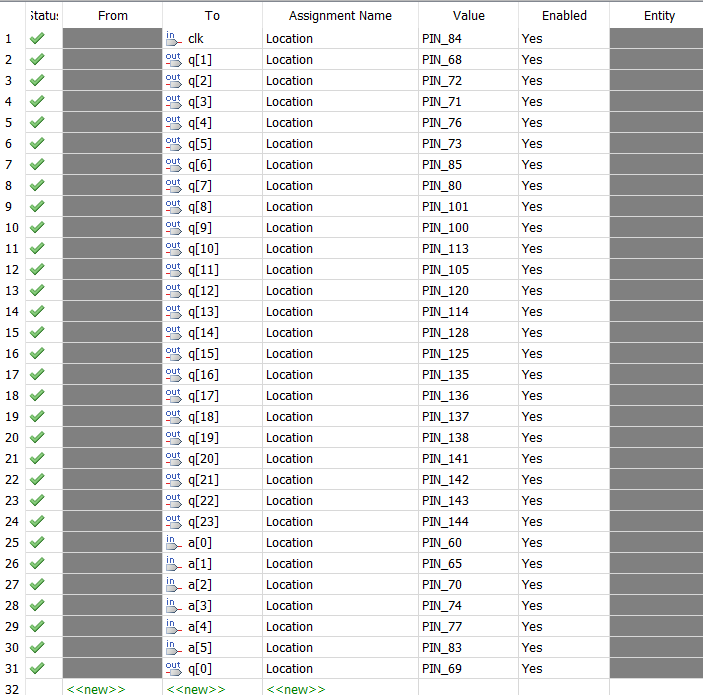
绘制电路图



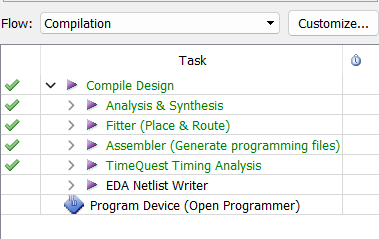
编辑ROM的初始数据

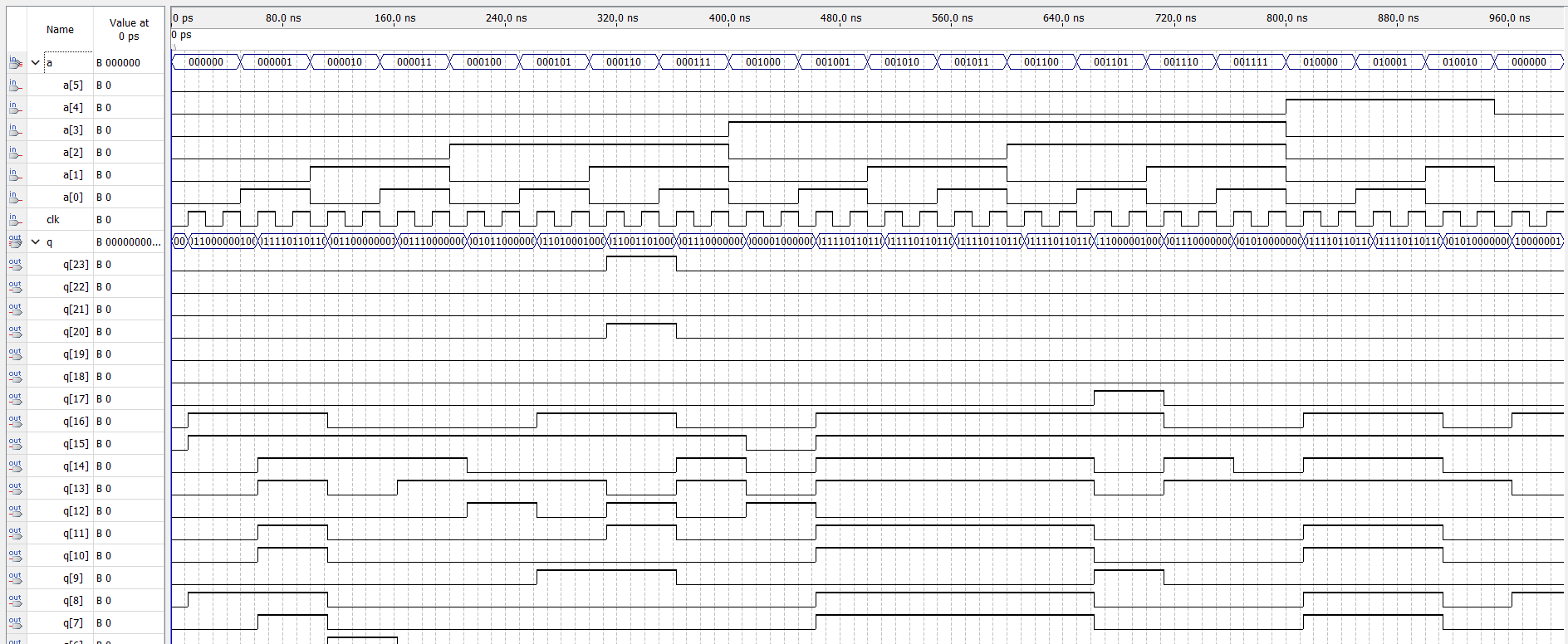


绑定引脚

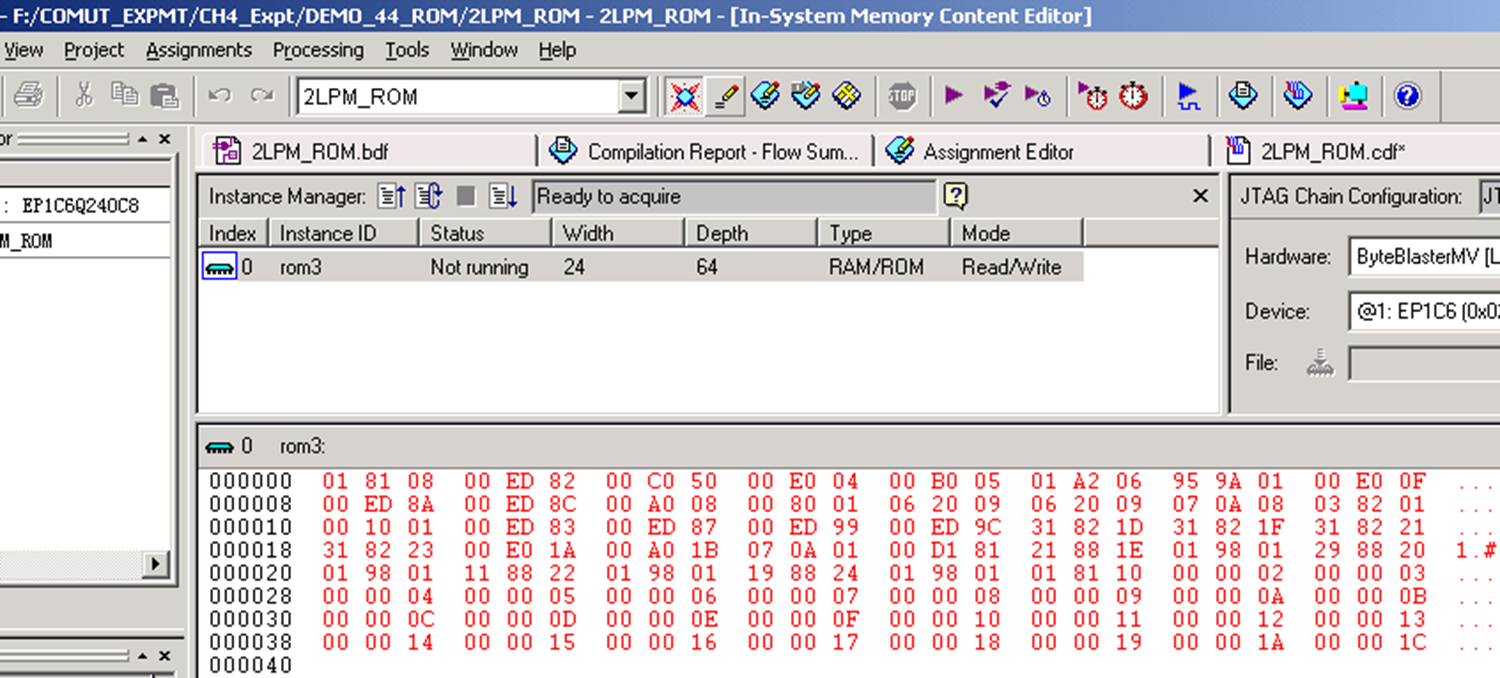


编译和仿真



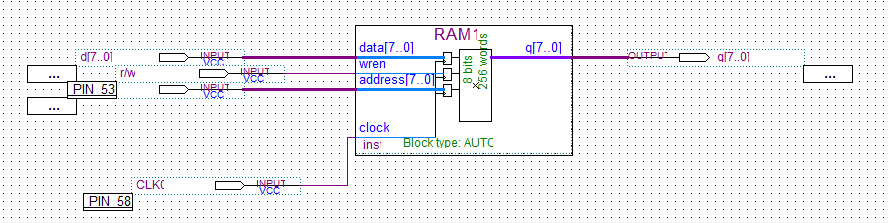


下载至FPGA，根据实验任务对开发板进行操作，实验结果与预期结果一致。利用In-System Memory Content Editor从FPGA中的ROM中读取微程序数据。

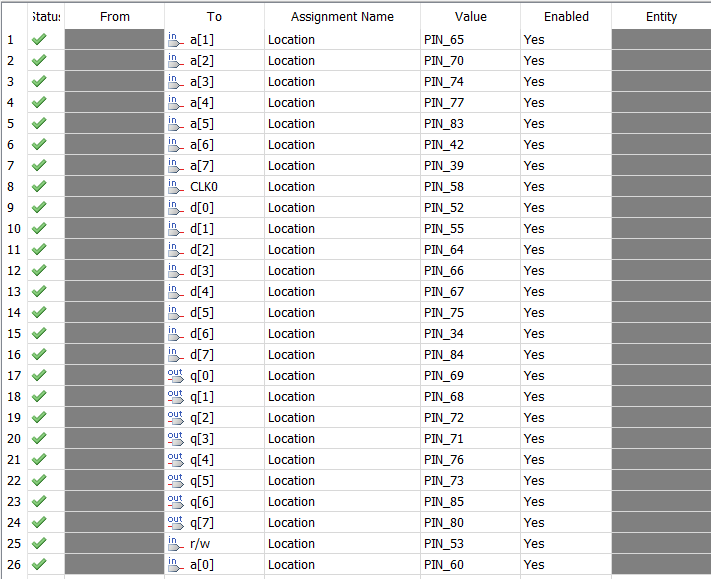


**LPM\_RAM实验**

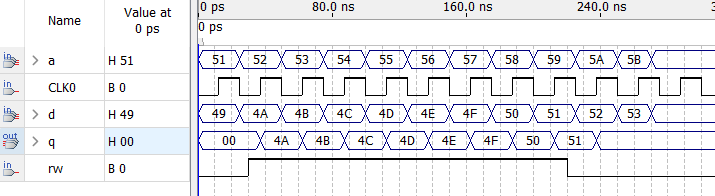
设计和绘制电路图

****

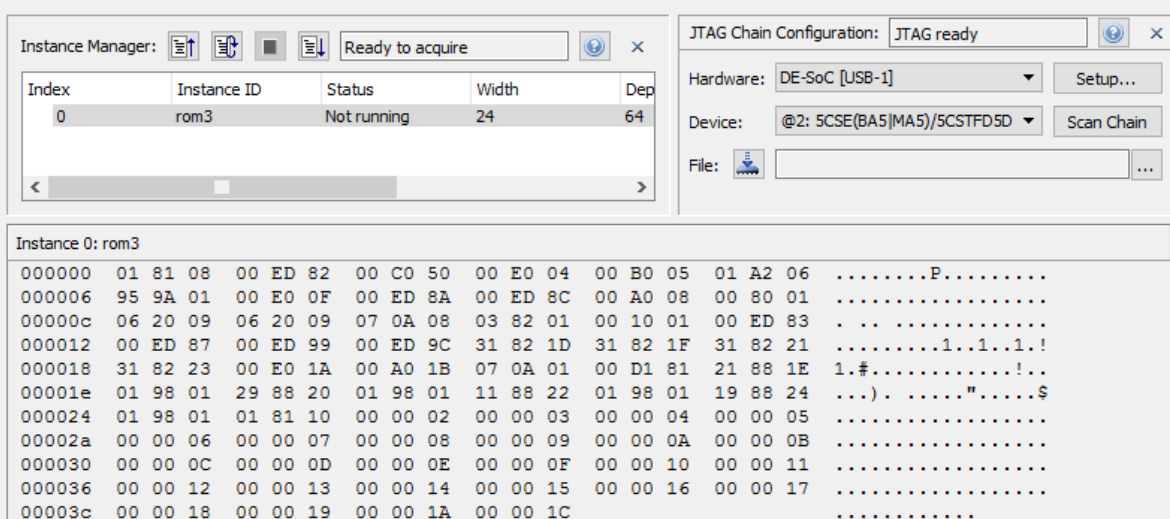
绑定引脚

****

编译和仿真

****

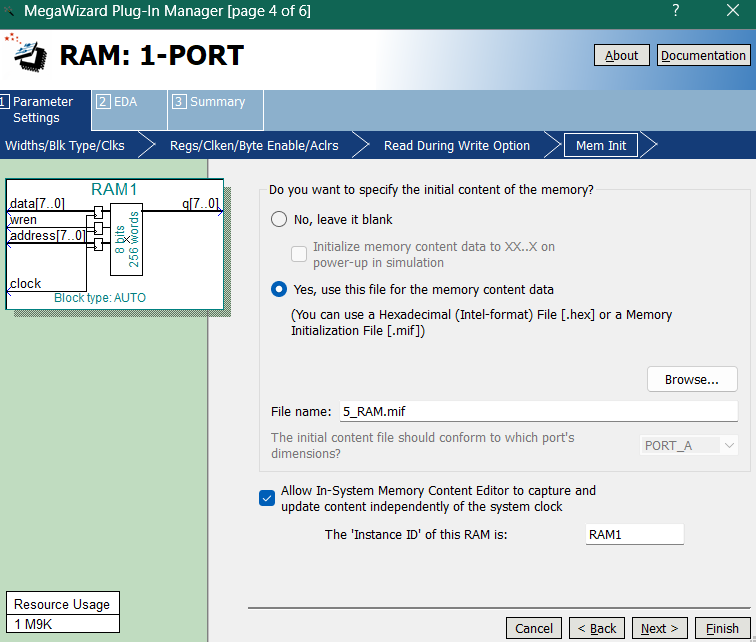
下载至FPGA，根据实验任务对开发板进行操作，实验结果与预期结果一致。利用In-System Memory Content Editor从FPGA中的RAM中读取数据。



## 实验总结及问题分析

### 如何建立LPM\_RAM\_DQ的数据初始化，如何导入和存储LPM\_RAM\_DQ参数文件？

创建需要初始化的数据的mif或hex等文件，在创建RAM的mem init选项时选择为LPM\_RAM\_DQ的初始化文件，如图所示。



亦可以在勾选允许系统读写后，在In-System Memory Content Editor中导入和写入文件。