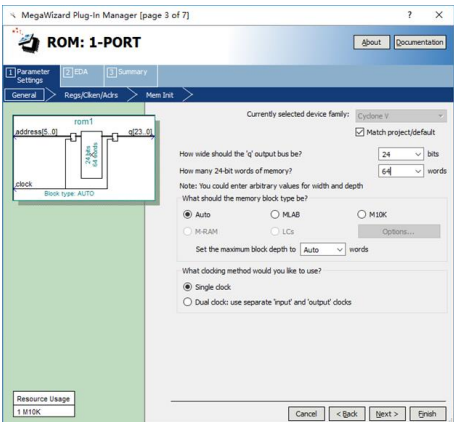
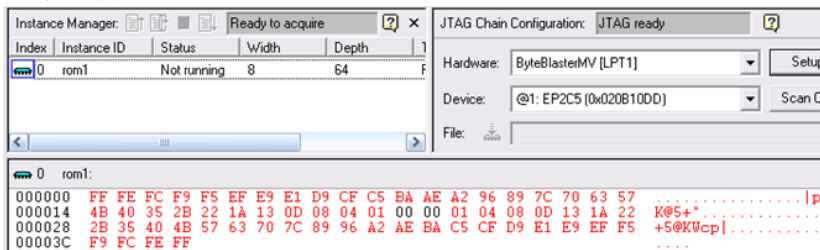


计算机组成与设计 课程实验报告

学号：202000130143	姓名： 郑凯饶	班级：2020 级 1 班
实验题目： LPM_RAM 实验		
实验学时：2	实验日期： 2022-5-10	
实验目的： 【1】 了解 FPGA 中 RAMlpm_ram_dp 的功能； 【2】 掌握 lpm_ram_dp 的参数设置和使用方法； 【3】 掌握 lpm_ram_dp 作为随机存储器 RAM 的仿真测试方法，工作特性和读写方法。		
实验软件和硬件环境： 软件环境： QuartusII 软件 硬件环境： 1. 实验室台式机 2. 计算机组成与设计实验箱		
实验原理和方法： 定制 RAM 元件符号： → 打开 MegaWizard Plug-In Manager 初始对话框 → 选择 create..... → 选择 Memory Compiler 项下的 RAM: 1-PORT → 选择 Cyclone 器件和 VHDL 语言方式 → 选择 RAM 控制线、地址线 and 数据线  → 单击 NEXT 后，Do you want to specify the initial content of the memory?下选择指定路径上的 mif 文件初始化存储器 利用 Quartus 的 EAB（嵌入式阵列）在系统（In-System）读写编辑器（In-System Memory Content Editor）： → 在菜单 tool 中选择 In-System Memory Content Editor 项 → 点击 Setup，选择 Hardware settings 页		

→ 选中 USB-laster

→ 设置完成后，点击数据文件即可观察到通过 FPGA 的 JTAG 口从其内部 EAB ROM 中读取的波形数据

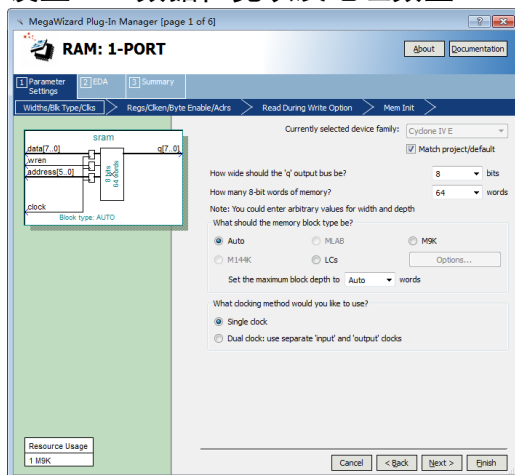


实验步骤：

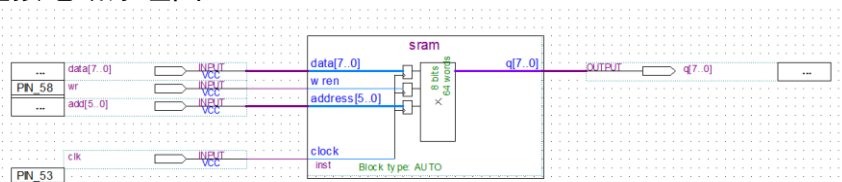
创建并初始化 mif 数据文件：

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
00	22	26	2A	2E	32	36	3A	3E	"&*.26:>
08	42	46	4A	4E	52	56	5A	5E	BFJNRVZ^
10	62	66	6A	6E	72	76	7A	7E	bfjnrvz~
18	82	86	8A	8E	92	96	9A	9E
20	A2	A6	AA	AE	B2	B6	BA	BE
28	C2	C6	CA	CE	D2	D6	DA	DE
30	E2	E6	EA	EE	F2	F6	FA	FE
38	02	06	0A	0E	12	16	1A	1E

设置 RAM 数据位宽以及地址数量：



连接电路原理图：



引脚分配：

add[5]	Input	PN_75	5	B5_N0	PN_75	2.5 V (default)	8mA (default)			
add[4]	Input	PN_67	4	B4_N0	PN_67	2.5 V (default)	8mA (default)			
add[3]	Input	PN_66	4	B4_N0	PN_66	2.5 V (default)	8mA (default)			
add[2]	Input	PN_64	4	B4_N0	PN_64	2.5 V (default)	8mA (default)			
add[1]	Input	PN_59	4	B4_N0	PN_59	2.5 V (default)	8mA (default)			
add[0]	Input	PN_52	3	B3_N0	PN_52	2.5 V (default)	8mA (default)			
clk	Input	PN_53	3	B3_N0	PN_53	2.5 V (default)	8mA (default)			
data[7]	Input	PN_39	3	B3_N0	PN_39	2.5 V (default)	8mA (default)			
data[6]	Input	PN_42	3	B3_N0	PN_42	2.5 V (default)	8mA (default)			
data[5]	Input	PN_83	5	B5_N0	PN_83	2.5 V (default)	8mA (default)			
data[4]	Input	PN_77	5	B5_N0	PN_77	2.5 V (default)	8mA (default)			
data[3]	Input	PN_74	5	B5_N0	PN_74	2.5 V (default)	8mA (default)			
data[2]	Input	PN_70	4	B4_N0	PN_70	2.5 V (default)	8mA (default)			
data[1]	Input	PN_65	4	B4_N0	PN_65	2.5 V (default)	8mA (default)			
data[0]	Input	PN_60	4	B4_N0	PN_60	2.5 V (default)	8mA (default)			
q[7]	Output	PN_80	5	B5_N0	PN_80	2.5 V (default)	8mA (default)	2 (default)		
q[6]	Output	PN_85	5	B5_N0	PN_85	2.5 V (default)	8mA (default)	2 (default)		
q[5]	Output	PN_73	5	B5_N0	PN_73	2.5 V (default)	8mA (default)	2 (default)		
q[4]	Output	PN_76	5	B5_N0	PN_76	2.5 V (default)	8mA (default)	2 (default)		
q[3]	Output	PN_71	4	B4_N0	PN_71	2.5 V (default)	8mA (default)	2 (default)		
q[2]	Output	PN_72	4	B4_N0	PN_72	2.5 V (default)	8mA (default)	2 (default)		
q[1]	Output	PN_68	4	B4_N0	PN_68	2.5 V (default)	8mA (default)	2 (default)		
q[0]	Output	PN_69	4	B4_N0	PN_69	2.5 V (default)	8mA (default)	2 (default)		
wr	Input	PN_58	4	B4_N0	PN_58	2.5 V (default)	8mA (default)			

测试、调试：

读验证：



读取存储单元 00 为 22



读取存储单元 01 为 26



读取存储单元 03 为 2E



读取存储单元 04 为 32



读取存储单元 14 为 72

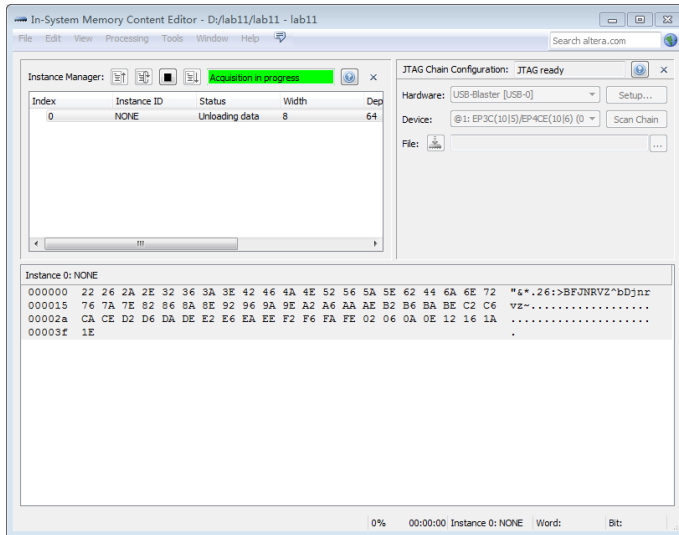


读取存储单元 24 为 B2



读取存储单元 34 为 F2

也可以通过 In-System Memory Content Editor 验证:



写验证:



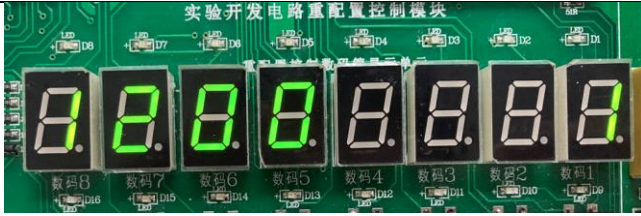
将 11 写入 00 单元



读取 00 单元为 11



将 12 写入 01 单元

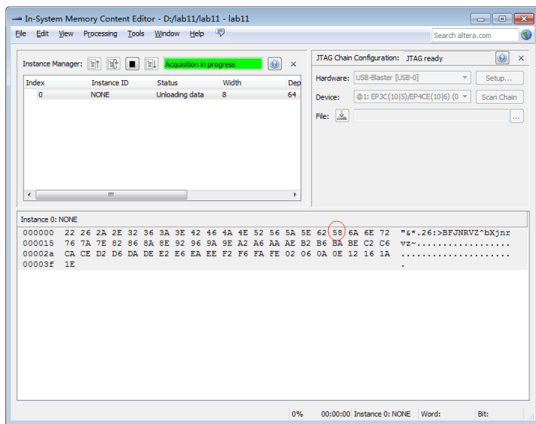


读取 01 单元为 12



将 58 写入 11 单元

通过系统编辑器查看：



结论分析与体会：

这次实验在 ROM 的使用基础上学习了 RAM 的用法，相比 ROM，RAM 增加了读功能，可以作为程序动态数据存储区。这次也学习了 In-System Memory Content Editor 的使用，更加快捷地获取硬件设备（如 RAM）的运行情况。