

第五章

存储器

5.1 存储器的分类和性能指标





5.1 存储器的分类和性能指标

5.1.1 存储器分类方法

1. 根据存储元件的构成材料分类

2. 根据存储器的存取方式分类

① 随机存储器**RAM (Random Access Memory)**

② 只读存储器 **ROM (Read-Only Memory)**

3. 根据制造工艺分类

① 双极型存储器。

② **MOS**型。

5.1 存储器的分类和性能指标

5.1.1 存储器分类方法

4. 根据存储器在计算机中的用途分类

① 主存储器 (**Main Memory**)

② 辅助存储器 (**External Memory**)

三级存储器中的各级存储器的用途和特点

① 高速缓冲存储器**Cache**。

② 主存储器。

③ 辅助主存储器。

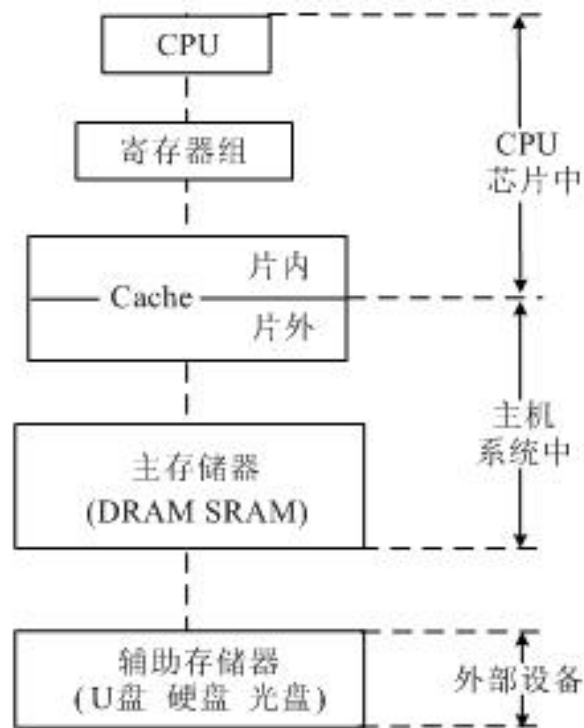


图 5-1 三级存储器构成



5.1 存储器的分类和性能指标

5.1.2 存储器的性能指标

1. 存储器容量

2. 存取速度

3. 功耗

4. 存储器可靠性

5. 性价比

5.2 半导体存储器

5.2.1 只读存储器ROM

1. 掩膜ROM

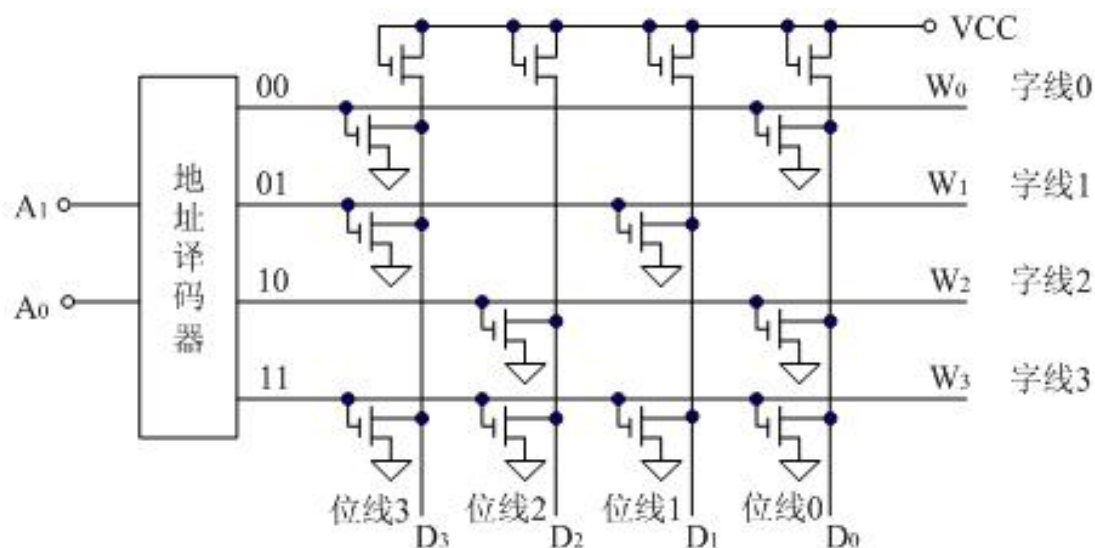


图 5-2 4×4 位 MOS ROM 存储阵列

5.2 半导体存储器

5.2.1 只读存储器ROM

2. 可编程的ROM

3. 可擦除可编程的ROM (EPROM)

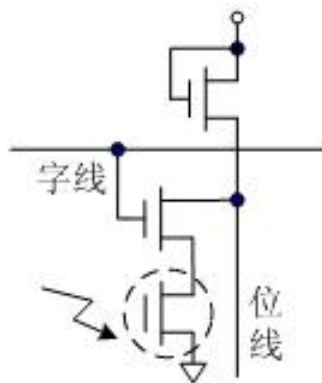


图 5-3 可编程单元电路原理图



5.2 半导体存储器

5.2.1 只读存储器ROM

4. 电可擦型可编程ROM (EEPROM)

5. 闪速存储器(Flash Memory)

5.2 半导体存储器

5.2.2 随机存取存储器RAM



5.2 半导体存储器

5.2.2 随机存取存储器RAM

1. 静态随机存储器SRAM

(1) SRAM存储单元的内部结构。

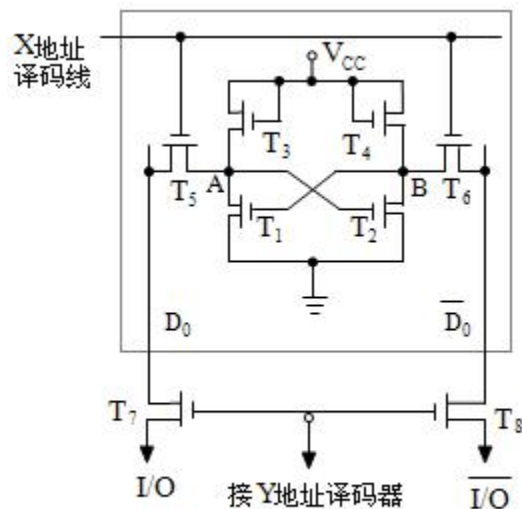


图 5-4 SRAM 存储单元内部结构

5.2 半导体存储器

5.2.2 随机存取存储器RAM

(2) 2114的结构与时序特点。

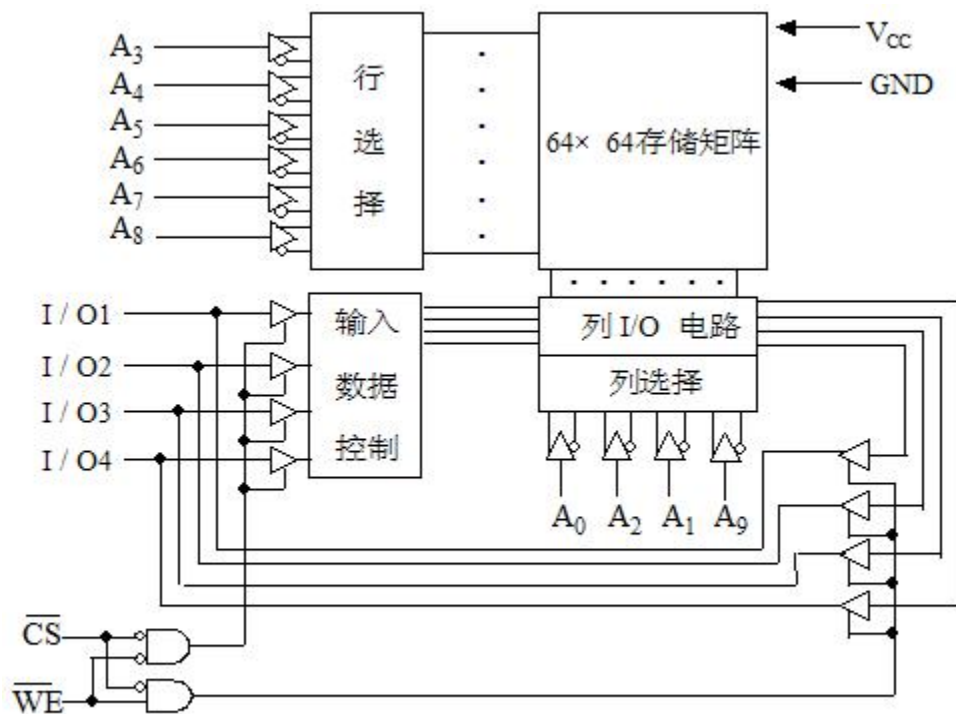


图 5-5 2114 框图和管脚配置

5.2 半导体存储器

5.2.2 随机存取存储器RAM

(2) 2114的结构与时序特点。

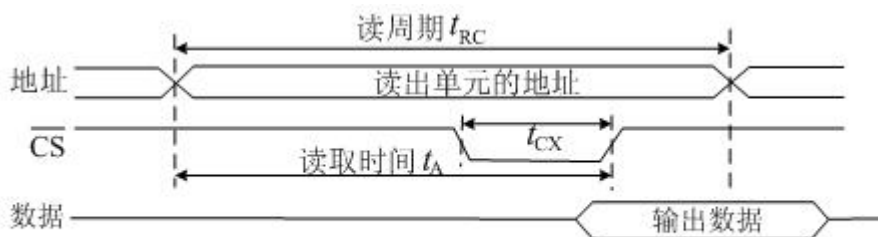


图 5-6 SRAM 读操作时序

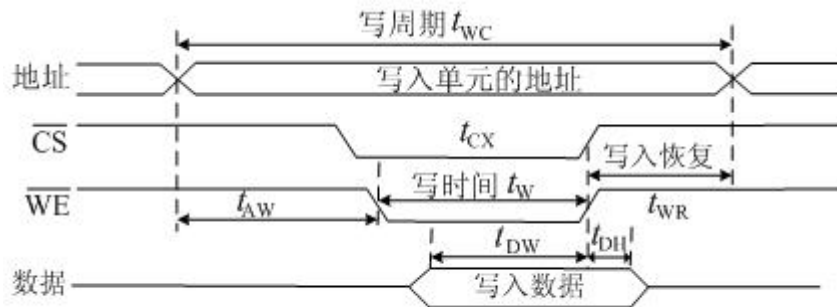


图 5-7 SRAM 写操作时序

5.2 半导体存储器

5.2.2 随机存取存储器RAM

(3) 6264芯片及其工作方式。

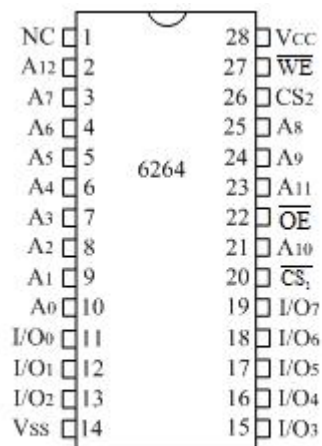


图 5-8 6264 芯片引脚图

表5-1 6264的工作方式

\overline{CS}_2	\overline{CS}_1	\overline{OE}	\overline{WE}	工作方式
1	0	0	1	读
1	0	1	0	写
0	1	x	x	未选

5.2 半导体存储器

5.2.2 随机存取存储器RAM

2. 动态随机存储器DRAM

(1) DRAM存储单元的内部结构。

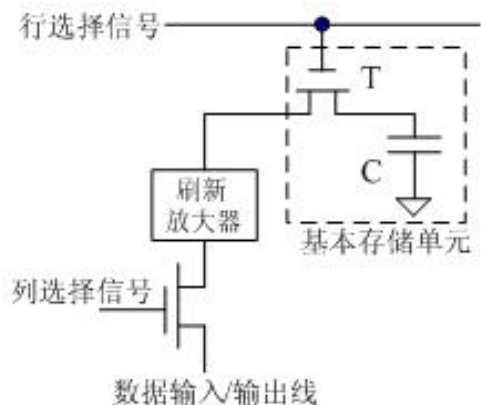


图 5-9 单管 DRAM 基本存储单元电路

5.2 半导体存储器

5.2.2 随机存取存储器RAM

2. 动态随机存储器DRAM

(2) DRAM芯片2164A。

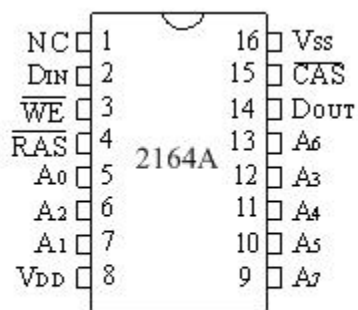


图 5-10 2164A 引脚



5.2 半导体存储器

5.2.3 新型的DRAM

1. EDO DRAM

2. SDRAM

3. DDR SDRAM

4. RDRAM

5.3 存储器容量扩展

5.3.1 存储器片选控制方法

1. 线选法

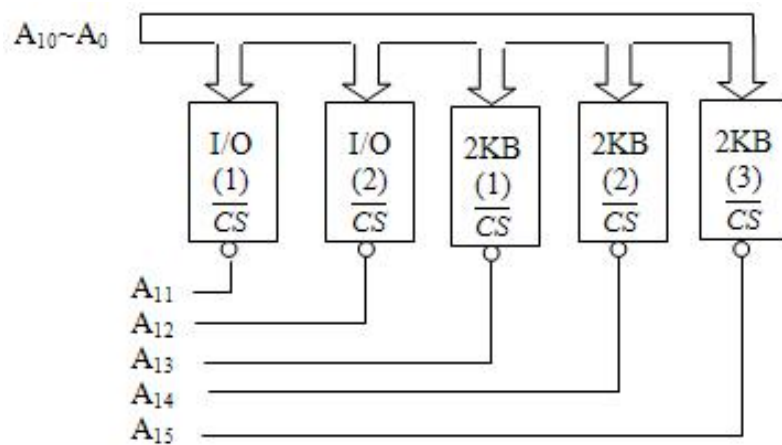


图 5-11 线选法译码

5.3 存储器容量扩展

5.3.1 存储器片选控制方法

2. 全译码法

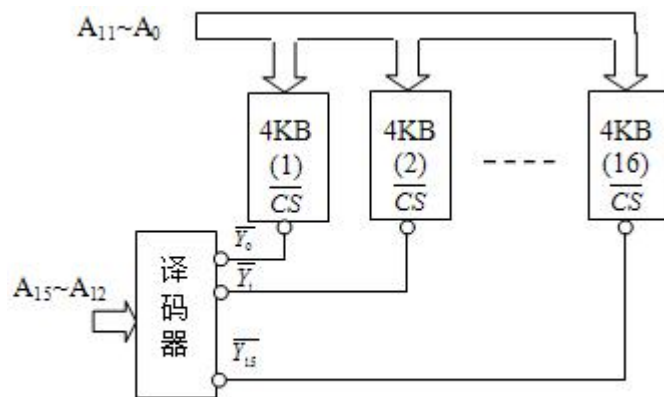


图 5-12 全译码电路模型图

5.3 存储器容量扩展

5.3.1 存储器片选控制方法

3. 部分译码法

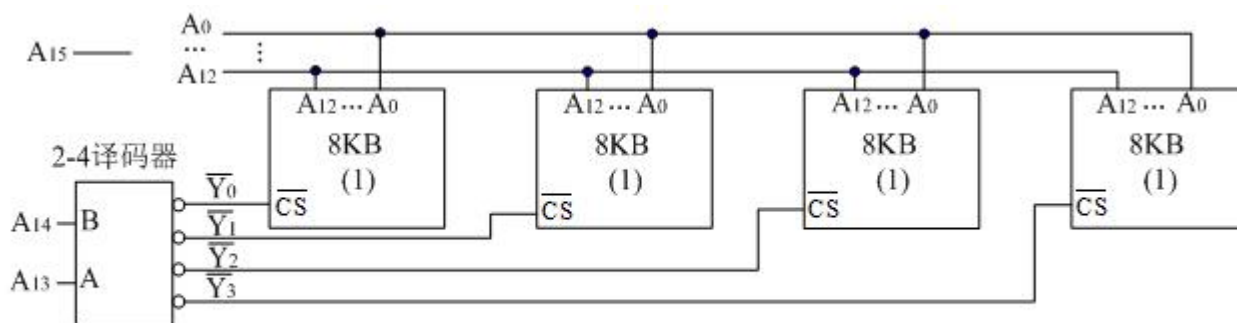


图 5-13 部分译码法的电路示意图

4. 混合译码法

5.3 存储器容量扩展

5.3.2 存储芯片的扩展

1. 位扩展

【例 5-1】 用 $1\text{K} \times 4$ 位的 2114 芯片构成 $1\text{K} \times 8$ 位的存储器系统。

由于每个芯片只能提供 4 位数据，故需用 2 片这样的芯片，它们分别提供 4 位数据至系统的数据总线高 4 位和低 4 位，以满足存储器系统的字长要求。

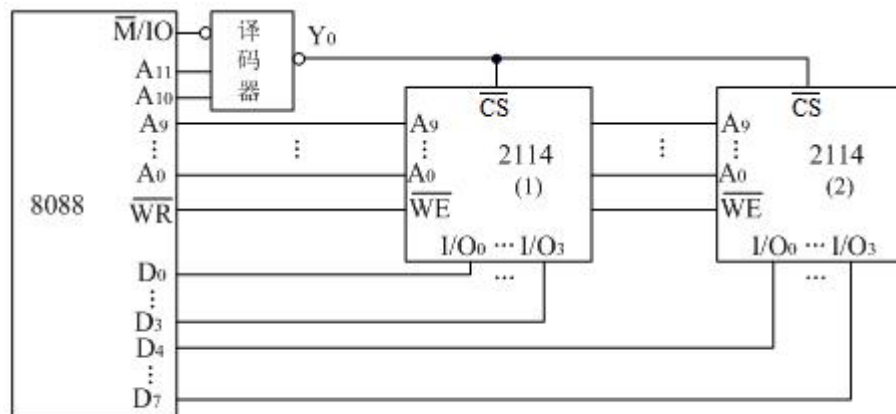


图 5-14 2 片 2114 完成存储器的位扩展电路模型

5.3 存储器容量扩展

5.3.2 存储芯片的扩展

2. 字扩展

【例 5-2】用 $8K \times 8$ 位的 6264 存储器芯片组成 $32K \times 8$ 位的存储器系统。

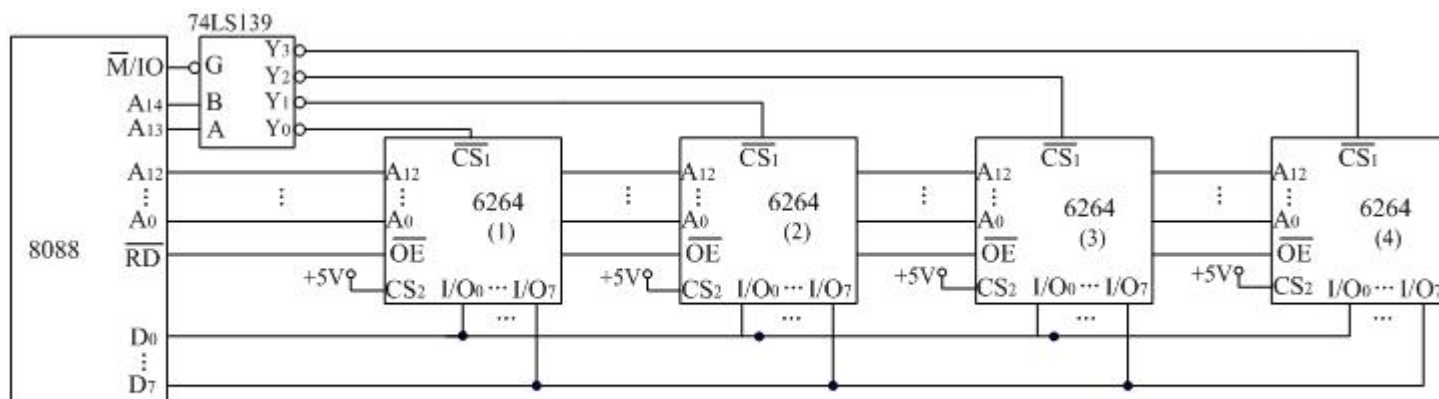


图 5-15 4 片 2764 实现存储器的字扩展电路模型

5.3 存储器容量扩展

5.3.2 存储芯片的扩展

2. 字扩展

表 5-2 32K×8 位存储器系统的地址空间分布

6264 序号	译码 器输 出选 择端	译码 器 输 入 端 A ₁₄ A ₁₃	6264 地址输入端												芯片地址 范围
			A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	
(1)	Y0	00	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0000H 1FFFH
(2)	Y1	01	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	2000H 3FFFH
(3)	Y2	10	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	4000H 5FFFH
(4)	Y3	11	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	0 · 1	6000H 7FFFH

5.3 存储器容量扩展

5.3.3 存储器系统建立举例

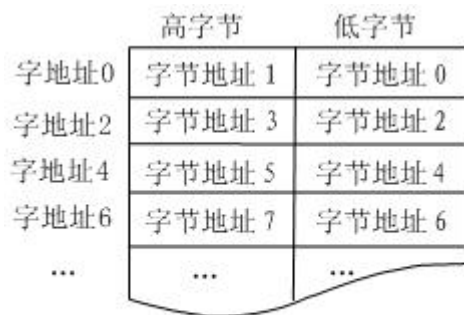


图 5-16 16 位存储器结构图

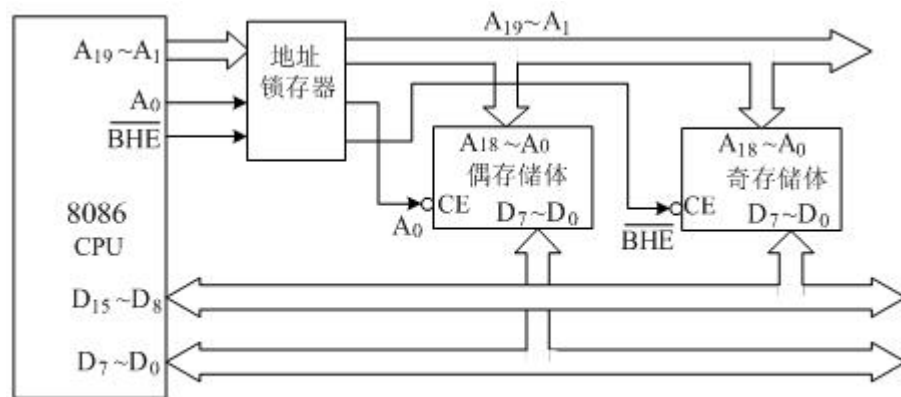


图 5-17 8086 与存储器的接口电路原理图

5.3 存储器容量扩展

5.3.4 IBM PC/XT的存储器子系统

1. PC/XT的存储空间的分配

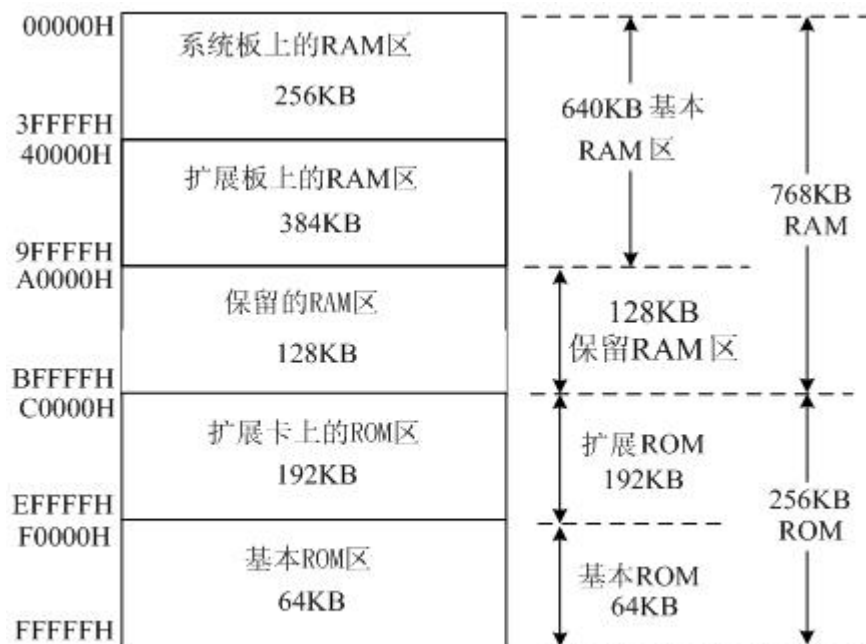


图 5-18 PC/XT 存储空间的分配



5.3 存储器容量扩展

5.3.4 IBM PC/XT的存储器子系统

2. RAM子系统

- (1) 基本RAM区 (00000H~9FFFFH)。
- (2) 保留RAM区 (A0000H~BFFFFH)。

3.ROM子系统

- (1) 扩展ROM区 (C0000H~EFFFFH)。
- (2) 基本ROM区 (F0000H~FFFFFFH)。

5.3 存储器容量扩展

5.3.5 微机存储器层次结构

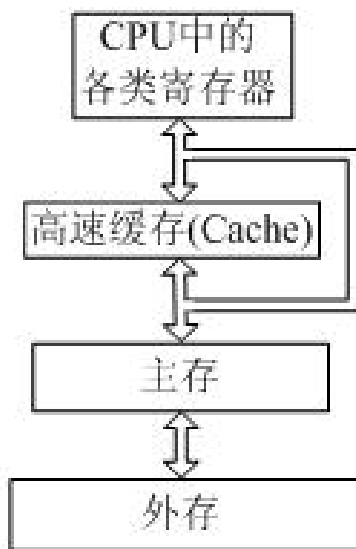


图 5-19 存储器层次结构



5.3 存储器容量扩展

5.3.5 微机存储器层次结构

1. 存储器系统的概念

2. CPU内的寄存器组

3. 高速缓存

4. 主存储器

5. 外部存储器



5.3 存储器容量扩展

5.3.6 Cache工作原理

5.4 FPGA中的嵌入式存储器

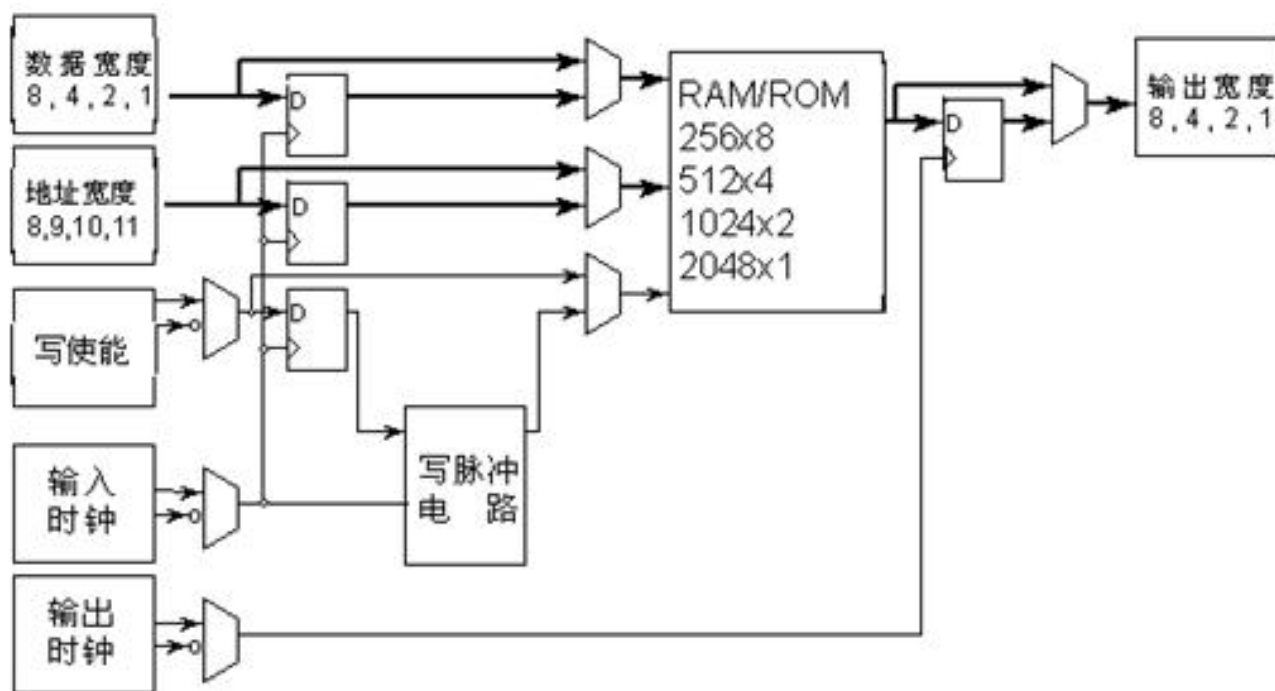


图 5-20 用 M4K 构成不同结构的 RAM 和 ROM

5.5 基于8088 IP最简SOC构建与测试

5.5.1 建立工作库文件夹和存盘原理图空文件

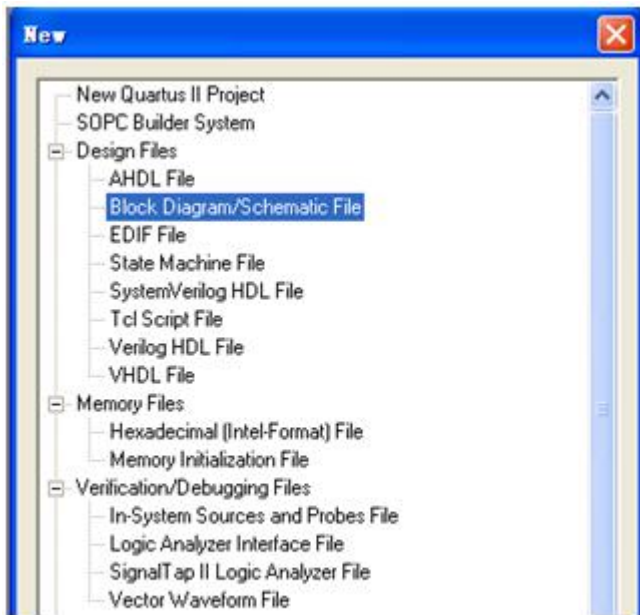


图 5-21 选择原理图编辑文件类型

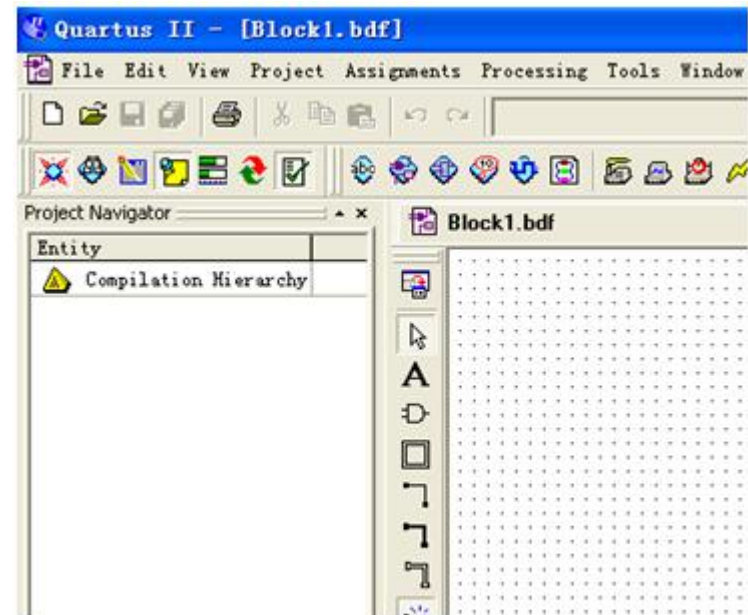


图 5-22 打开原理图编辑窗

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

1. 打开建立新工程管理窗



图 5-23 利用“New Project Wizard”创建工程 Core8088SYS

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

2. 将设计文件加入工程中



图 5-24 将所有相关的文件都加入进此工程

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

3. 选择目标芯片

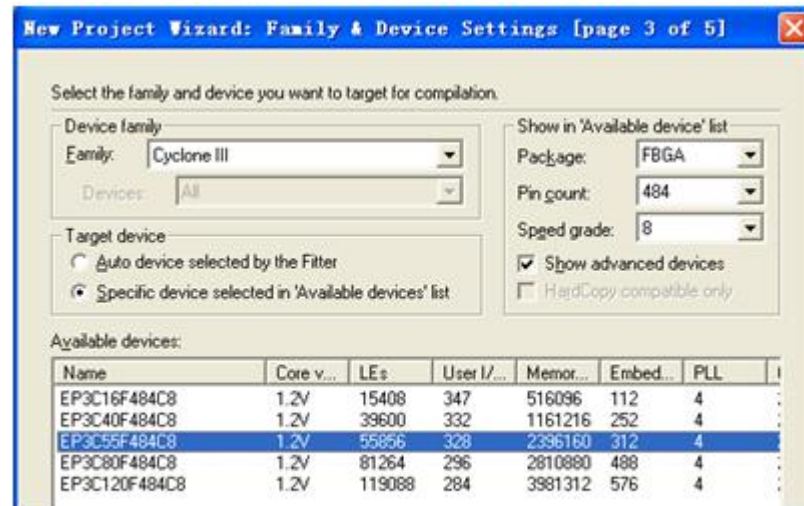


图 5-25 选择目标器件 EP3C55F484C8 型 FPGA

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

4. 结束设置



图 5-26 Core8088SYS 工程管理窗

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

5. 向编辑窗调入逻辑元件

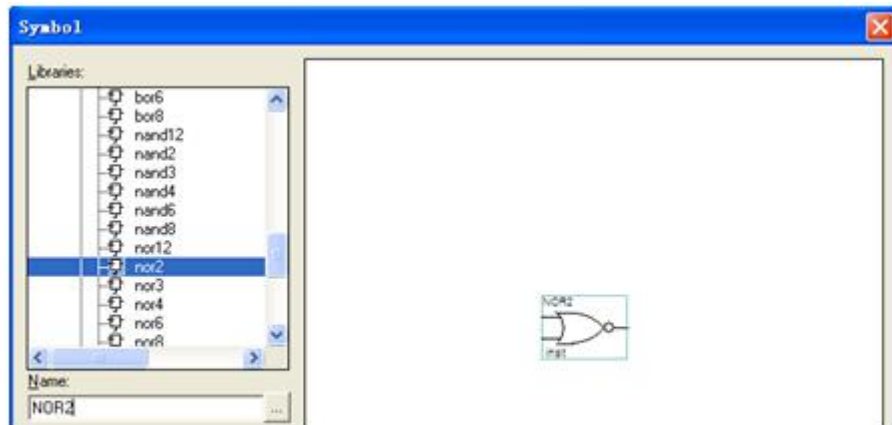


图 5-27 在元件调用对话框调出需要的逻辑元件：二输入或非门 NOR2

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

5. 向编辑窗调入逻辑元件

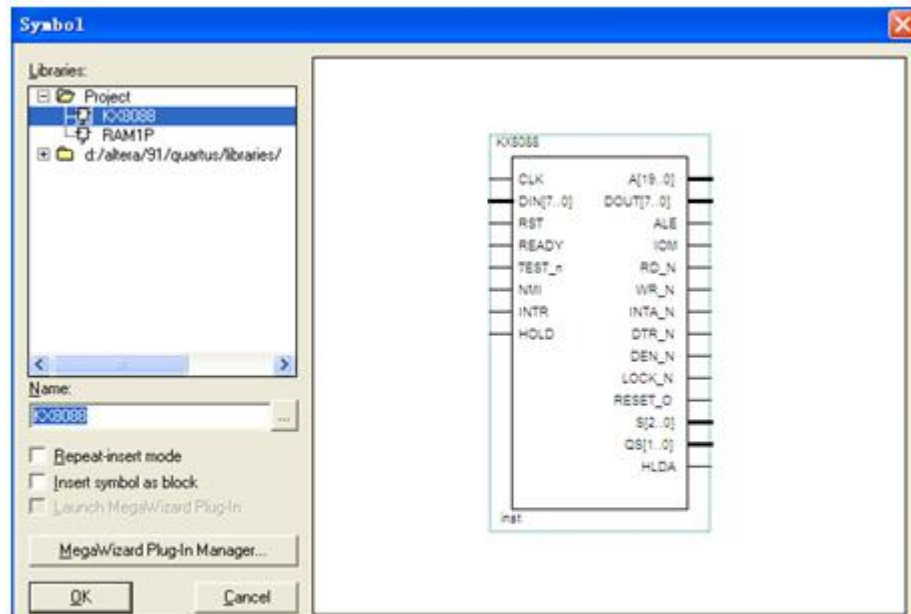


图 5-28 调出 8088 CPU 核 KX8088

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

6. 编辑测试系统的汇编程序

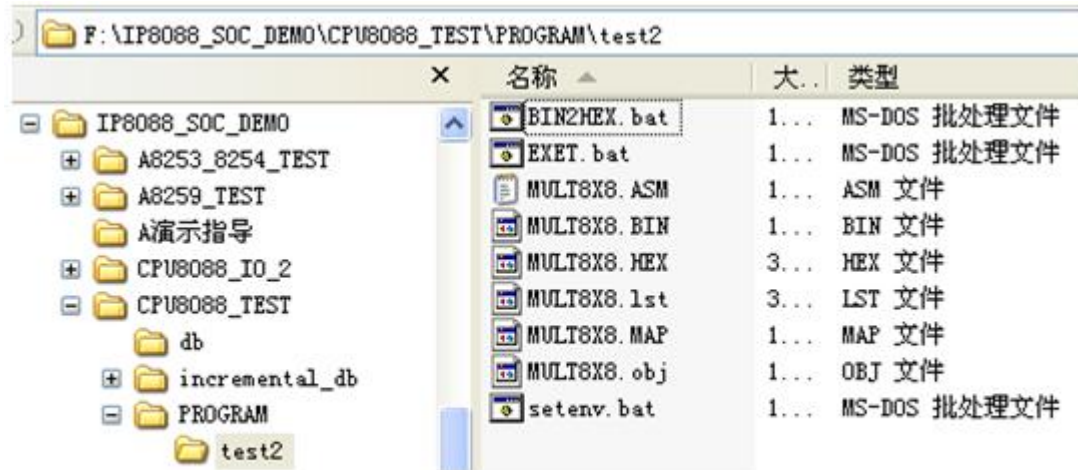


图 5-29 汇编程序编译文件文件夹

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

6. 编辑测试系统的汇编程序

表 5-3 8088 最小系统测试程序

指令地址	指令代码	汇编程序: MULT8X8.ASM, 完整程序见例 4-13
0010H	56H	DAT1 DB 56H
0011H	E7H	DAT2 DB 0E7H
0012H	4567H	RESULT DW 4567H ; 注意 45H 放在高位地址 0013H 中
0014H	A0 0010H	START : MOV AL, DAT1
0017H	8A 1E 0011H	MOV BL, DAT2
001BH	F6 E3H	MUL BL
001DH	A3 0012H	MOV RESULT, AX
0020H	EB FEH	RND : JMP RND
00F0H	EAH	ORG 00F0H ; BOOT CODE
00F1H	0014H	DB 0EAH ; JMP FAR PTR start
00F3H	00 00H	DW START
		DB 00H,00H ; JMP 0014 0000

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

7. 对LPM_RAM进行设置和调用

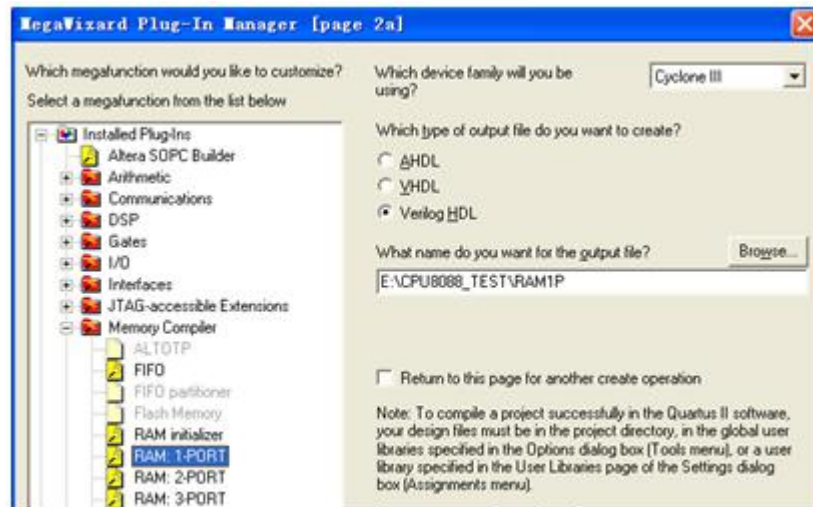


图 5-30 调用单口 LPM RAM

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

7. 对LPM_RAM进行设置和调用

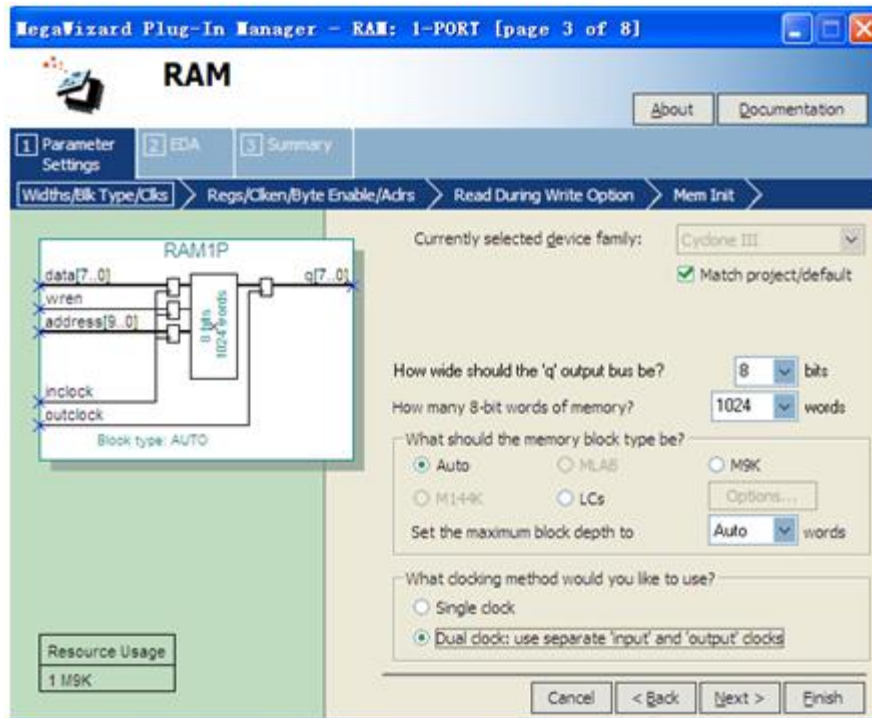


图 5-31 设定 RAM 参数

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

7. 对LPM_RAM进行设置和调用

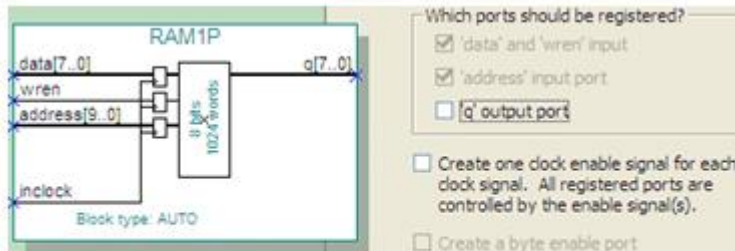


图 5-32 设定 RAM 仅输入时钟控制

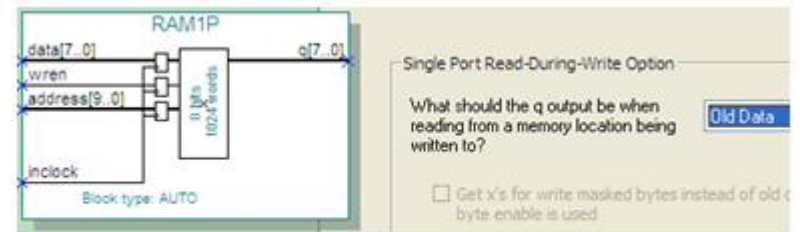


图 5-33 设定在写入同时读出原数据: Old Data

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

7. 对LPM_RAM进行设置和调用

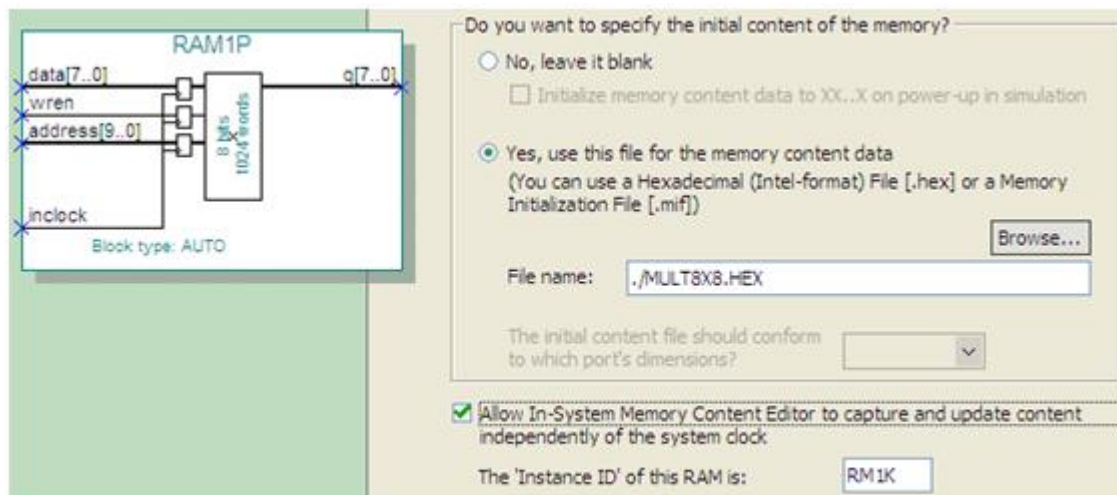


图 5-34 设定初始化文件和允许在系统编辑

5.5 基于8088 IP最简SOC构建与测试

5.5.2 创建工程

8. 编辑构建基于8088软核的最小系统电路图

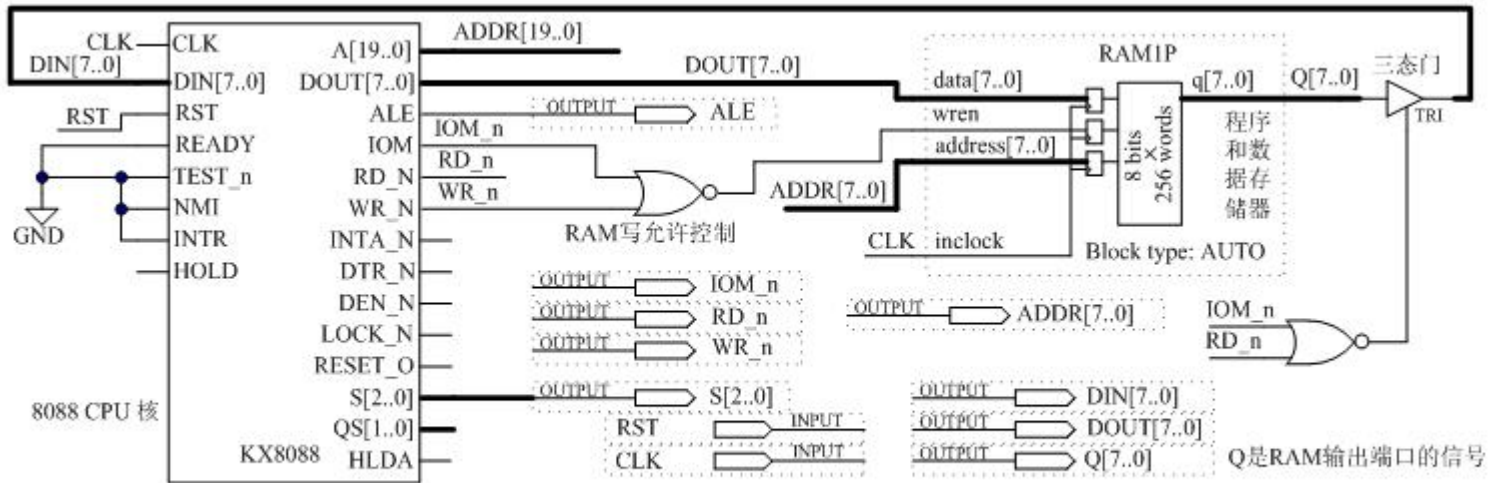


图 5-35 基于 8088 核的最简 SOC 微机系统电路原理图。

5.5 基于8088 IP最简SOC构建与测试

5.5.3 全程编译

5.5.4 时序仿真

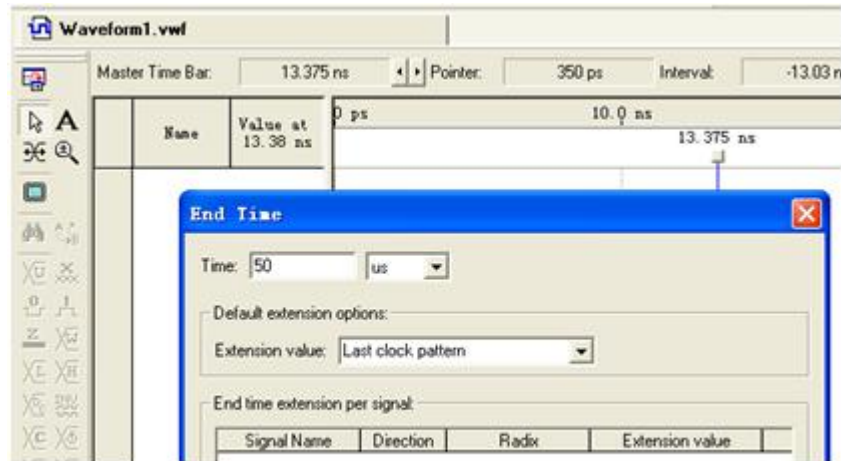


图 5-36 设置仿真时间长度

5.5 基于8088 IP最简SOC构建与测试

5.5.3 全程编译

5.5.4 时序仿真

- ① 打开波形编辑器。
- ② 设置仿真时间区域。

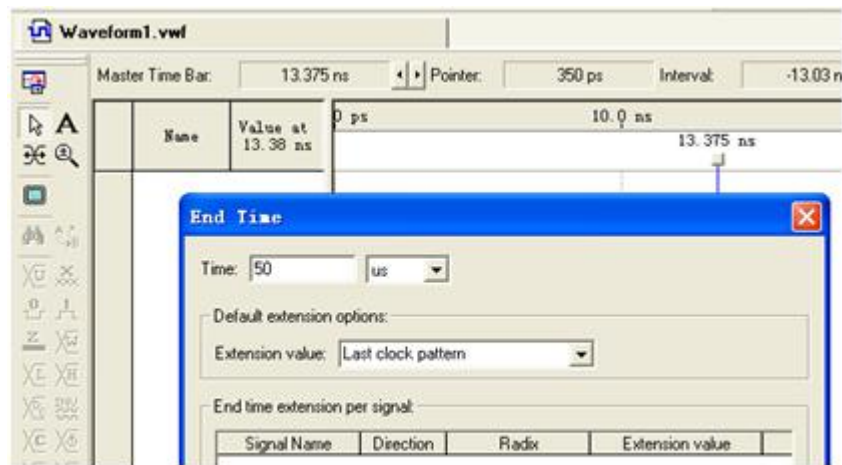


图 5-36 设置仿真时间长度

5.5 基于8088 IP最简SOC构建与测试

5.5.4 时序仿真

- ③ 波形文件存盘。
- ④ 将工程**Core8088SYS**的端口信号名选入波形编辑器中。

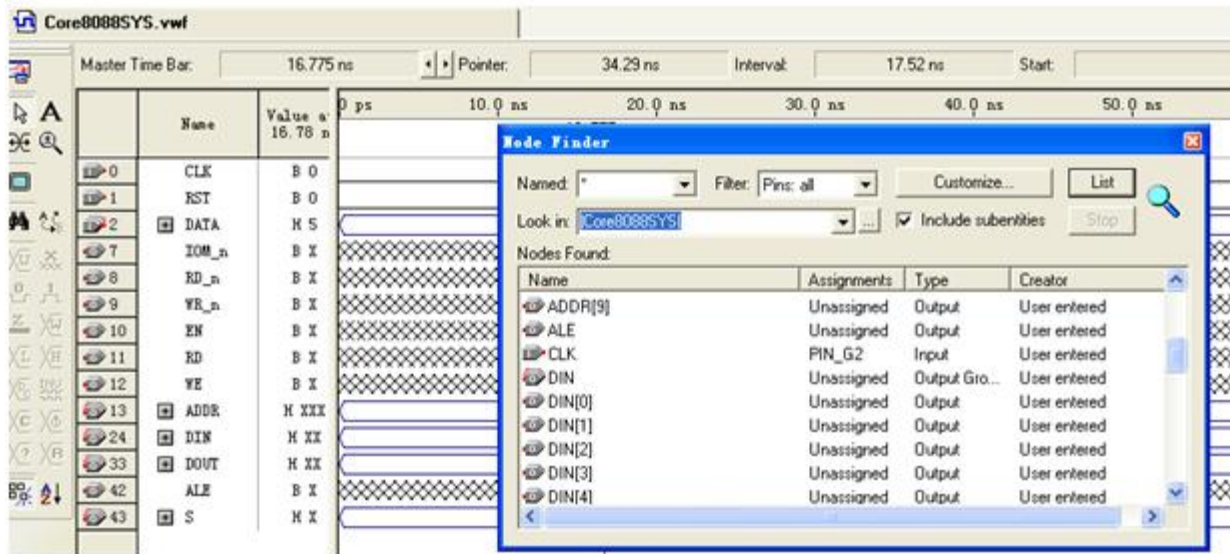


图 5-37 从 Node Finder 窗向波形编辑器拖入信号节点

5.5 基于8088 IP最简SOC构建与测试

5.5.4 时序仿真

⑤ 编辑输入波形（输入激励信号）。

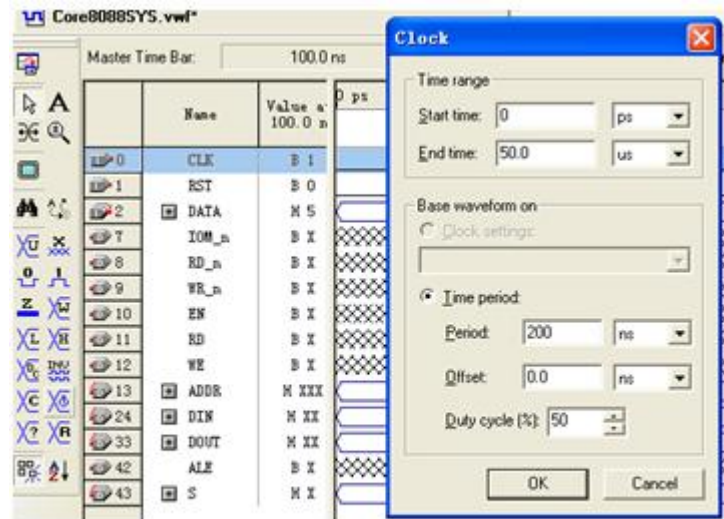


图 5-38 为 CLK 设置周期

5.5 基于8088 IP最简SOC构建与测试

5.5.4 时序仿真

⑥ 总线数据格式设置。

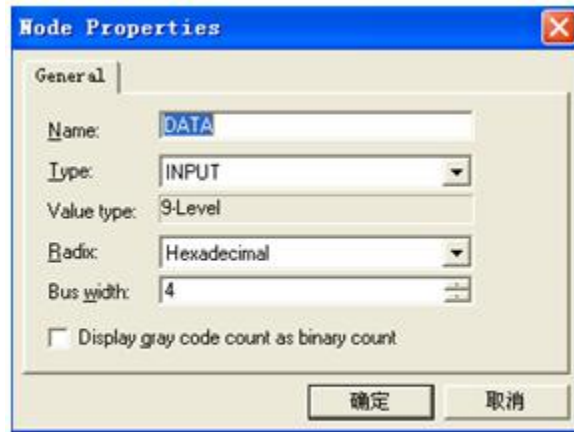


图 5-39 为总线 DATA 设置数制 Radix

5.5 基于8088 IP最简SOC构建与测试

5.5.4 时序仿真

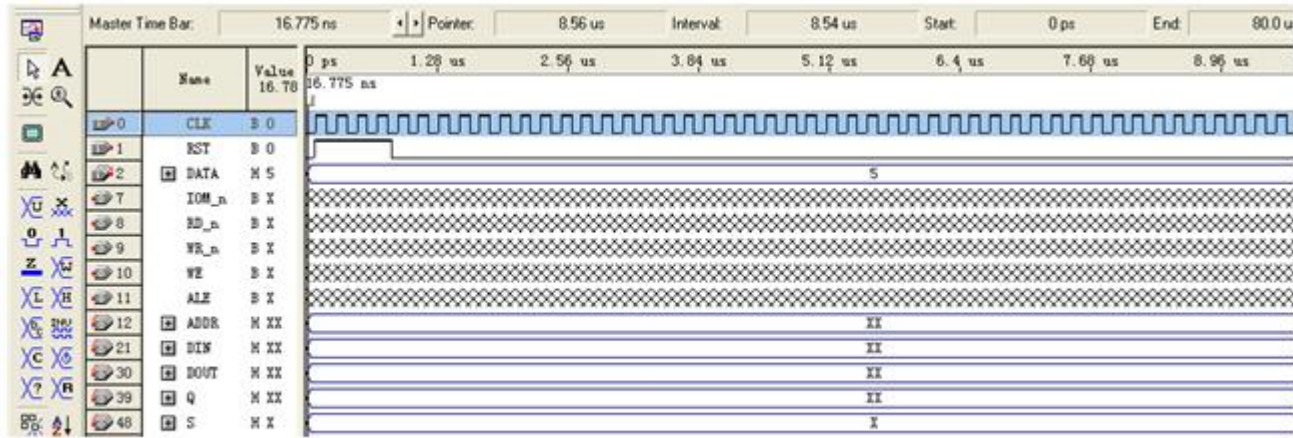


图 5-40 设置好的激励波形图

⑦ 仿真器参数设置。

⑧ 启动仿真器。现在所有设置进行完毕。

5.5 基于8088 IP最简SOC构建与测试

5.5.5 分析时序仿真结果

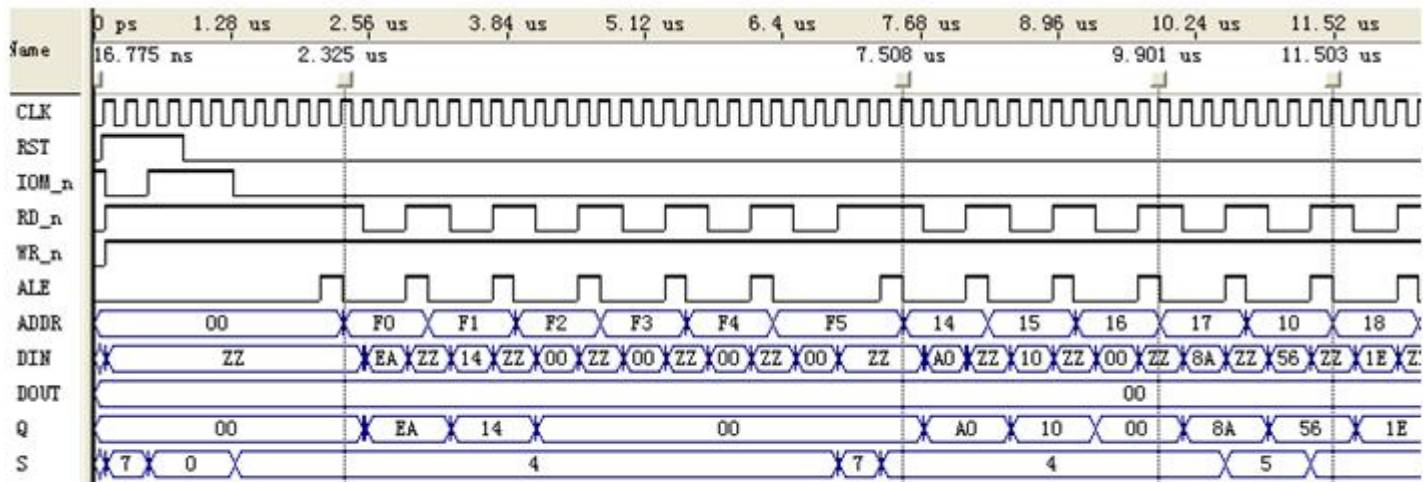


图5-41 8088系统执行指令的时序波形图前半部

5.5 基于8088 IP最简SOC构建与测试

5.5.5 分析时序仿真结果

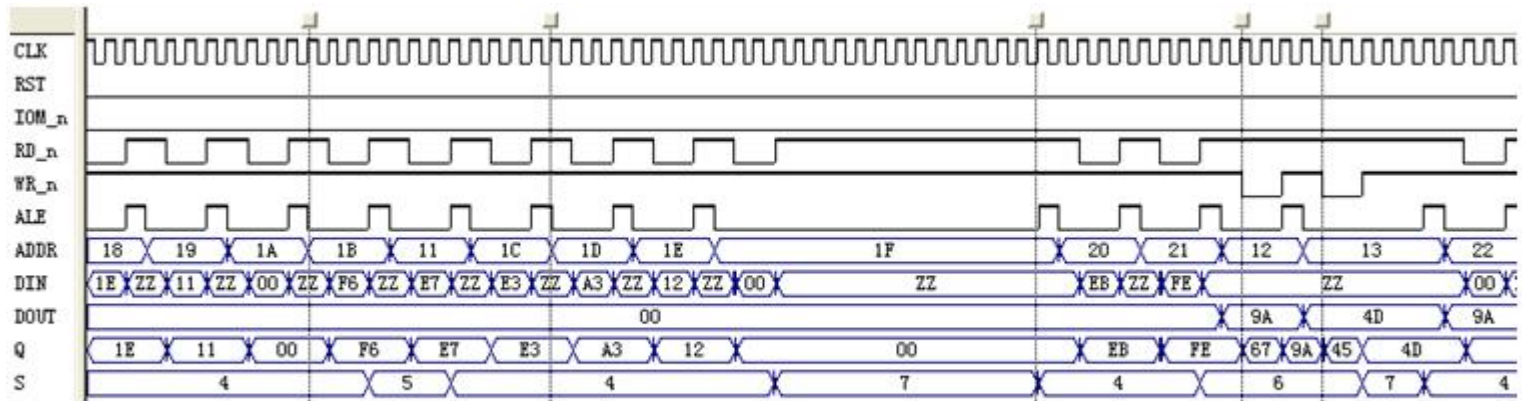


图 5-42 8088 系统执行指令的时序波形图后半部

5.5 基于8088 IP最简SOC构建与测试

5.5.6 引脚锁定

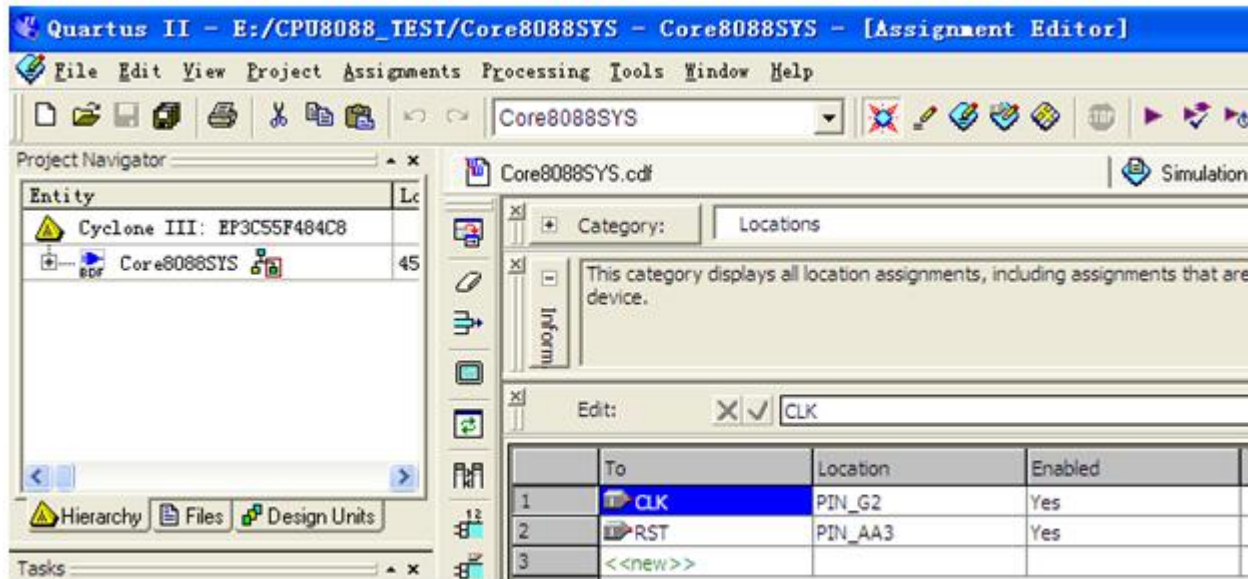


图 5-43 利用 Assignment Editor 编辑器锁定 FPGA 引脚

5.5 基于8088 IP最简SOC构建与测试

5.5.7 编译文件下载

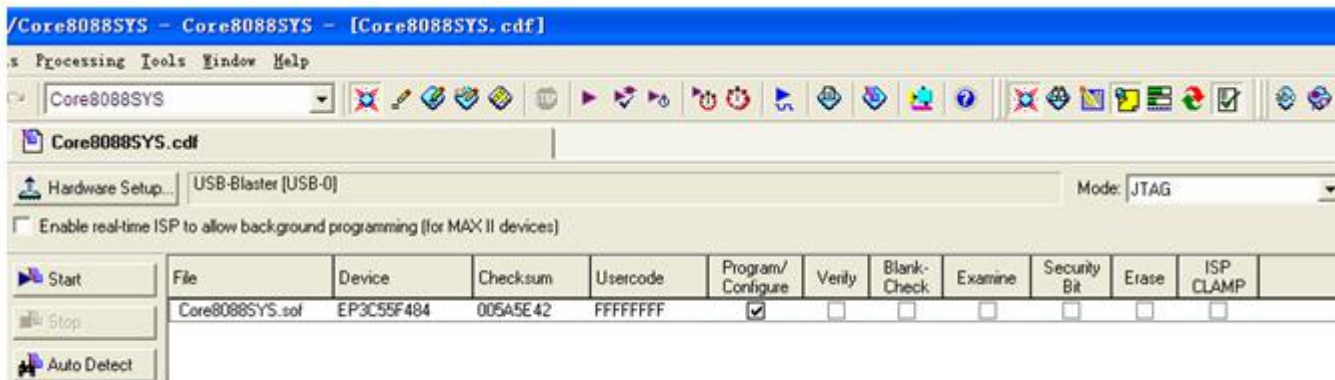


图 5-44 选择 JTAG 编程模式准备下载

5.5.8 USB-Blaster驱动程序安装方法

5.5 基于8088 IP最简SOC构建与测试

5.5.9 8088系统主存在线读写与软件调试

① 打开在系统存储单元编辑窗口。

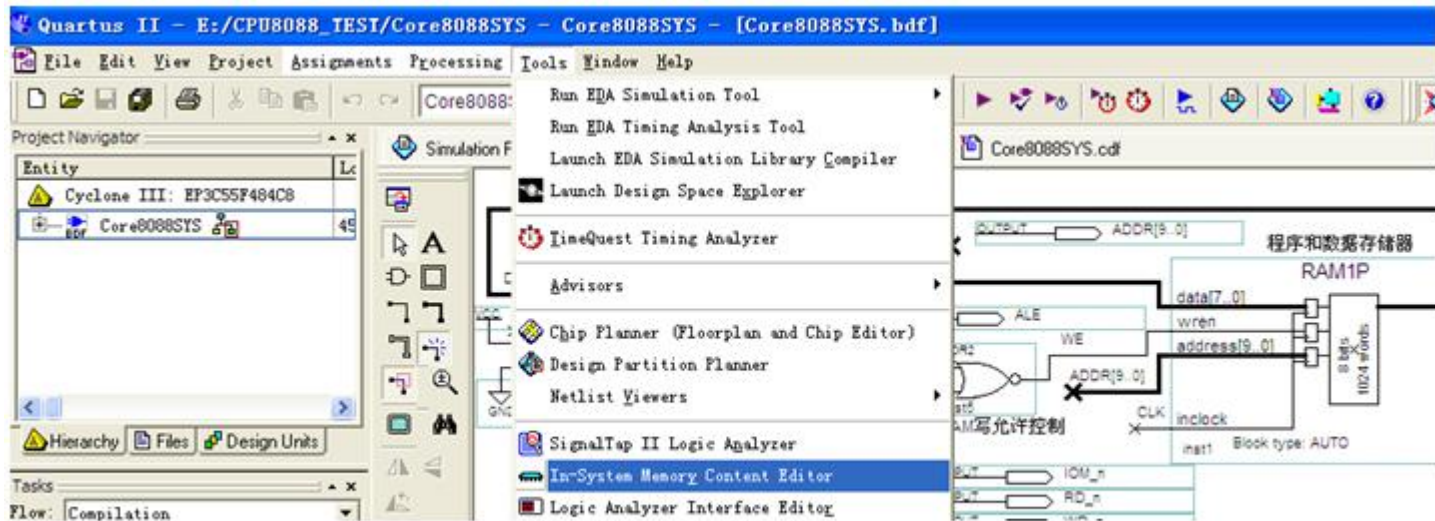


图 5-45 选择存储器内容在系统编辑器 In-System Memory Content Editor

5.5 基于8088 IP最简SOC构建与测试

5.5.9 8088系统主存在线读写与软件调试

① 打开在系统存储单元编辑窗口。

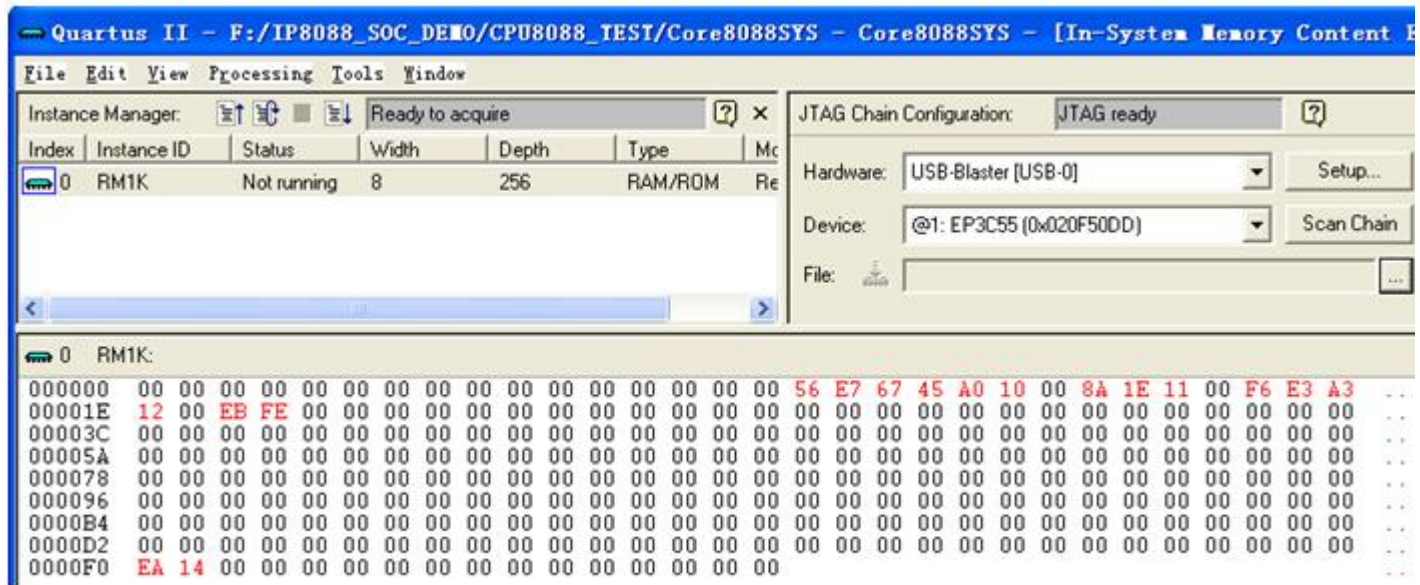


图 5-46 In-System Memory Content Editor 编辑窗（系统复位前的数据情况）

5.5 基于8088 IP最简SOC构建与测试

5.5.9 8088系统主存在线读写与软件调试

- ② 读取RAM/ROM中的数据。
- ③ 向RAM/ROM写数据。
- ④ 硬件系统测试。

0 RM1K:																																	
000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	56	E7	9A	4D	A0	10	00	8A	1E	11	00	F6	E3	A3	
00001E	12	00	EB	FE	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00003C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00005A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000078	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000096	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000B4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000D2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000F0	EA	14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

图 5-47 按复位键，CPU 执行程序后的数据情况

5.5 基于8088 IP最简SOC构建与测试

5.5.9 8088系统主存在线读写与软件调试

⑤ 代码文件的载入。

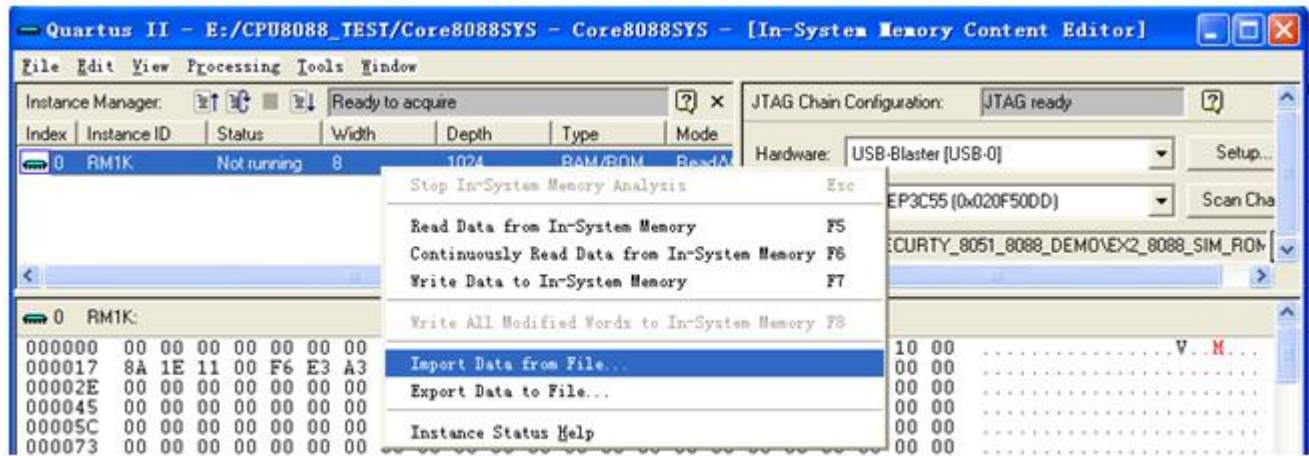


图 5-48 选择 JTAG 编程模式准备下载