

# 第七章 存储器

7.1 存储器介绍

7.2 随机存取存储器RAM

7.3 只读存储器ROM

## 7.1 半导体存储器

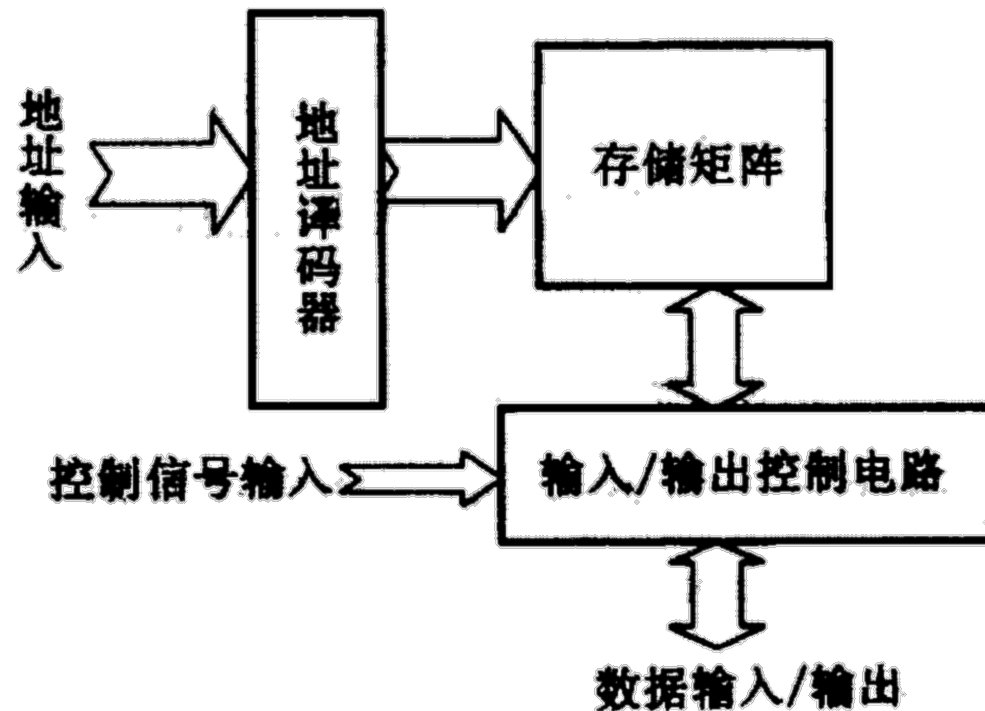
根据使用功能的不同，半导体存储器可分为随机存取存储器（**RAM**）和只读存储器（**ROM**）。

按照存储原理的不同，RAM可分为静态（**SRAM**）和动态（**DRAM**）。RAM使用灵活、方便，可以随时从其中任一指定地址读出或写入数据。RAM具有易失性，断电后存储的数据立即丢失。

ROM是存储固定信息的存储器，预先把信息写入存储器。在操作过程中，只能读，不能写，断电信息不丢失，是非易失性存储器。

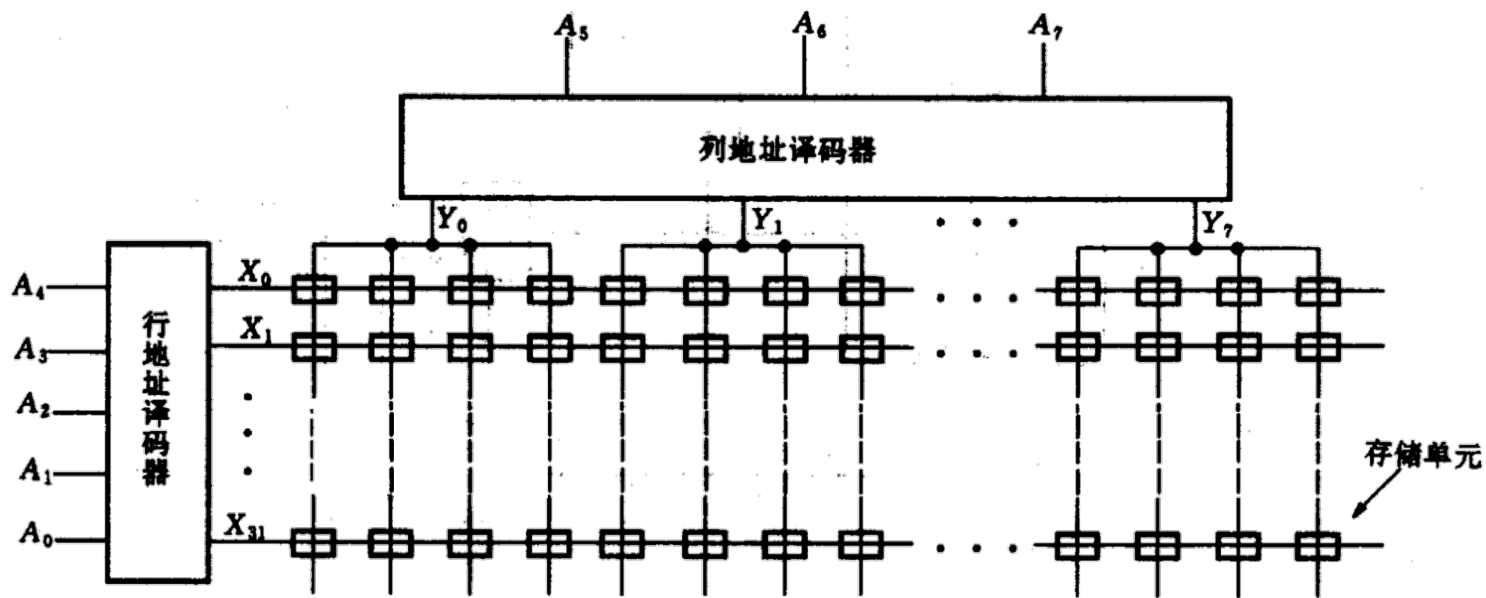
## 1、半导体存储器的基本结构

一般而言，存储器由存储矩阵、地址译码器和输入/输出控制电路3部分组成，有3类信号线，即地址线、数据线和控制线。



### (1) 存储矩阵

存储器由存储单元排列成矩阵形式组成，每个存储单元存放1位二进制数据。存储器以字为单位组织内部结构。1个字中所含的存储单元个数（即位宽或宽度）称为字长。字数（深度）与字长的乘积表示存储器的容量。



32行，每行32个存储单元，每4列存储单元连接在相同的列地址译码线上，组成一个行列。

每根行地址选择线选中一行，每根列地址选择线选中一个行列，因此有32根行地址选择线和8根列地址选择线，用以选择存储器256个字的地址。

容量为 $256 \times 4$ （256个字，每字4位）的存储器

## (2) 地址译码电路

实现地址选择。大容量存储器中，通常将输入地址分为行地址和列地址分别由行、列地址译码电路译码。

行、列地址译码电路的输出作为存储矩阵的行、列地址选择线，由它们**共同**确定欲选择的地址单元。字地址单元的个数**N**与二进制地址码的位数**n**满足关系式 **$N=2^n$** 。对于上图所示的存储矩阵，256个字需要8位二进制地址码( $A_7 \sim A_0$ )。

## (3) 输入/输出控制电路

存储单元的读、写操作由读/写控制信号R/W决定，存储单元是否工作可以由片选信号CS决定。

## 2、半导体存储器的技术指标

存储器有两个主要技术指标：存储容量和存取周期。

存储容量：存放二进制信息的多少。存储容量的公式为 $N$ （字数） $\times M$ （字长或位数）。存储器的字数通常采用K、M或G为单位。

存取周期：连续两次读（写）操作间隔的最短时间。存储器的性能基本上取决于从存储器读出信息或把信息写入存储器的速率。存储器的存取速度用存取周期或读写周期来表征。

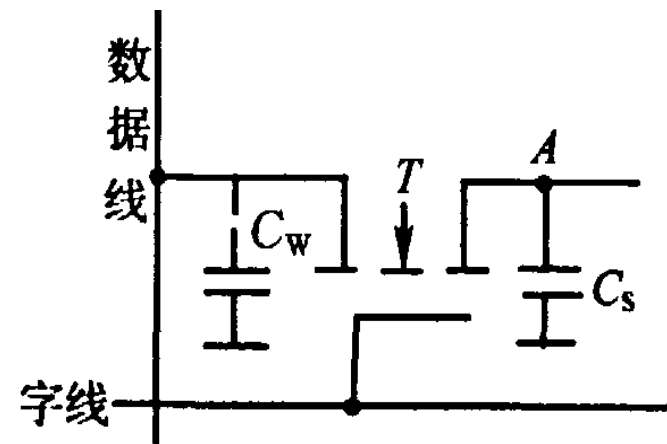
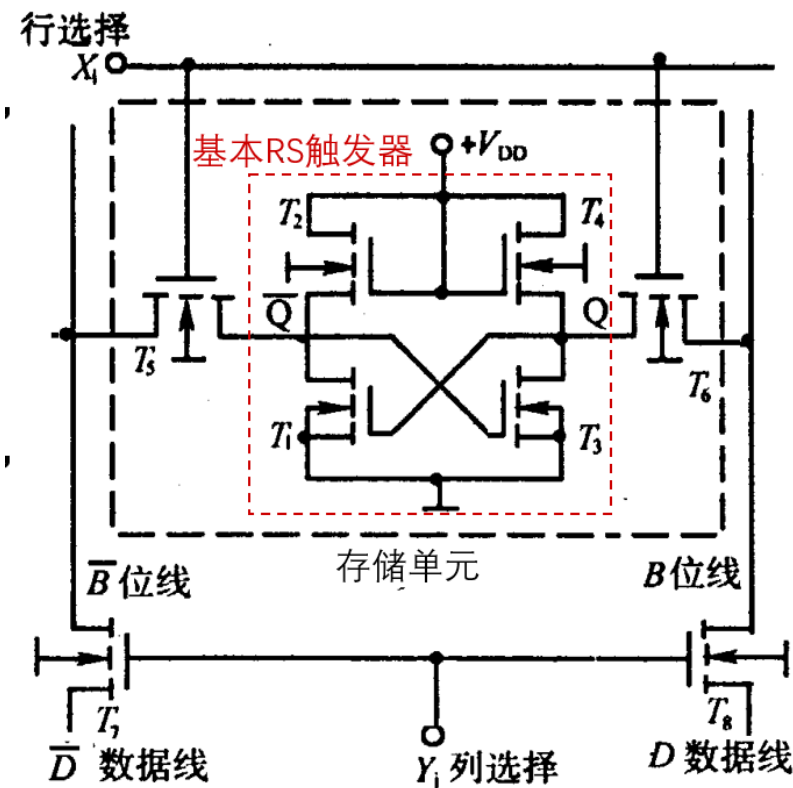
## 7.2 随机存取存储器RAM

用作内部存储器 and 高速缓冲存储器。

RAM存储单元：在存储器中，存储单元是最基本存储细胞，它可以存放1位二进制数据。

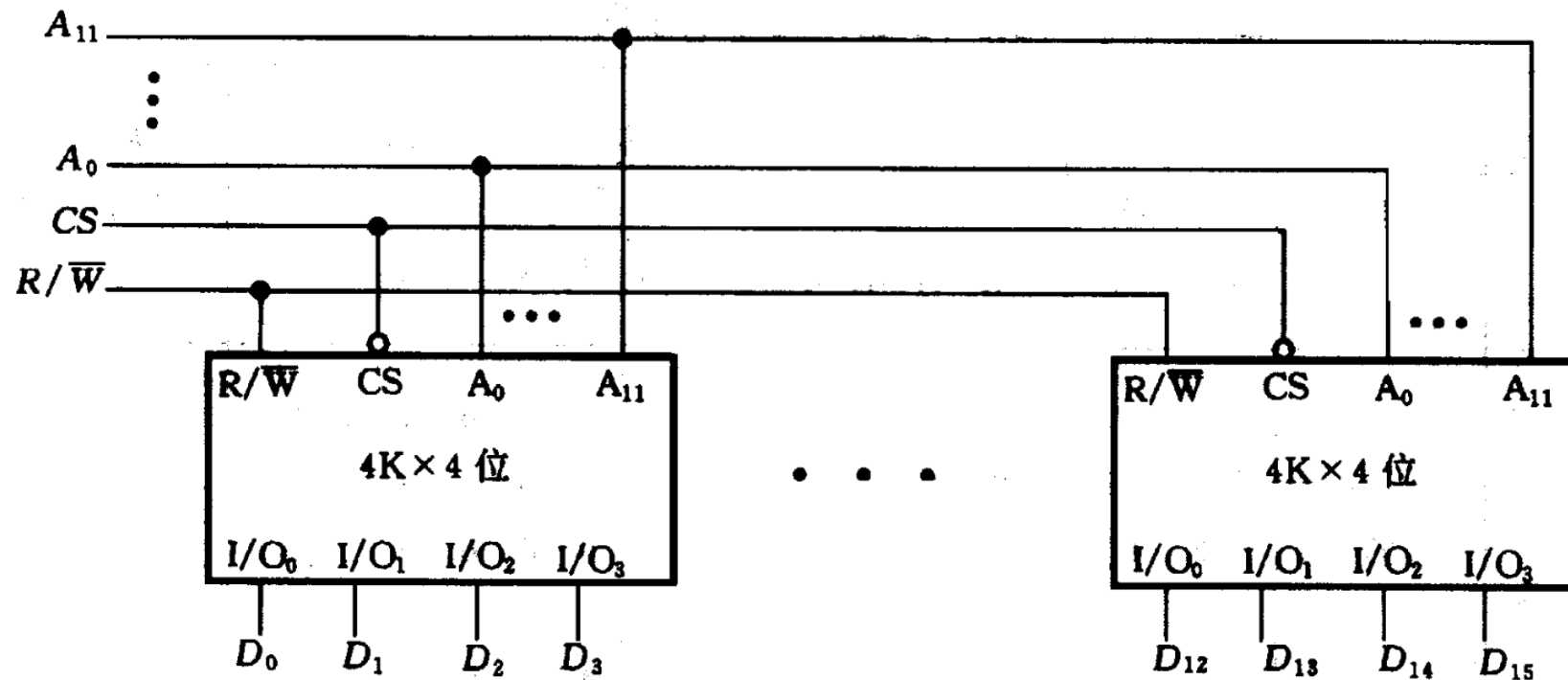
1、SRAM：数据由触发器记忆，不断电，数据就能永久保存。速度快，功耗大。

2、DRAM：每次读出后，必须及时对读出单元刷新（重写）。



3、RAM存储容量扩展：一片RAM不能满足存储容量时，可以通过增加字长或字数的方法，将多个芯片适当连接，实现容量扩展。

字长扩展：芯片并联，即将RAM的地址线、读/写控制线和片选信号对应地并联在一起，而各个芯片的数据输入/输出端作为字的各个位线。

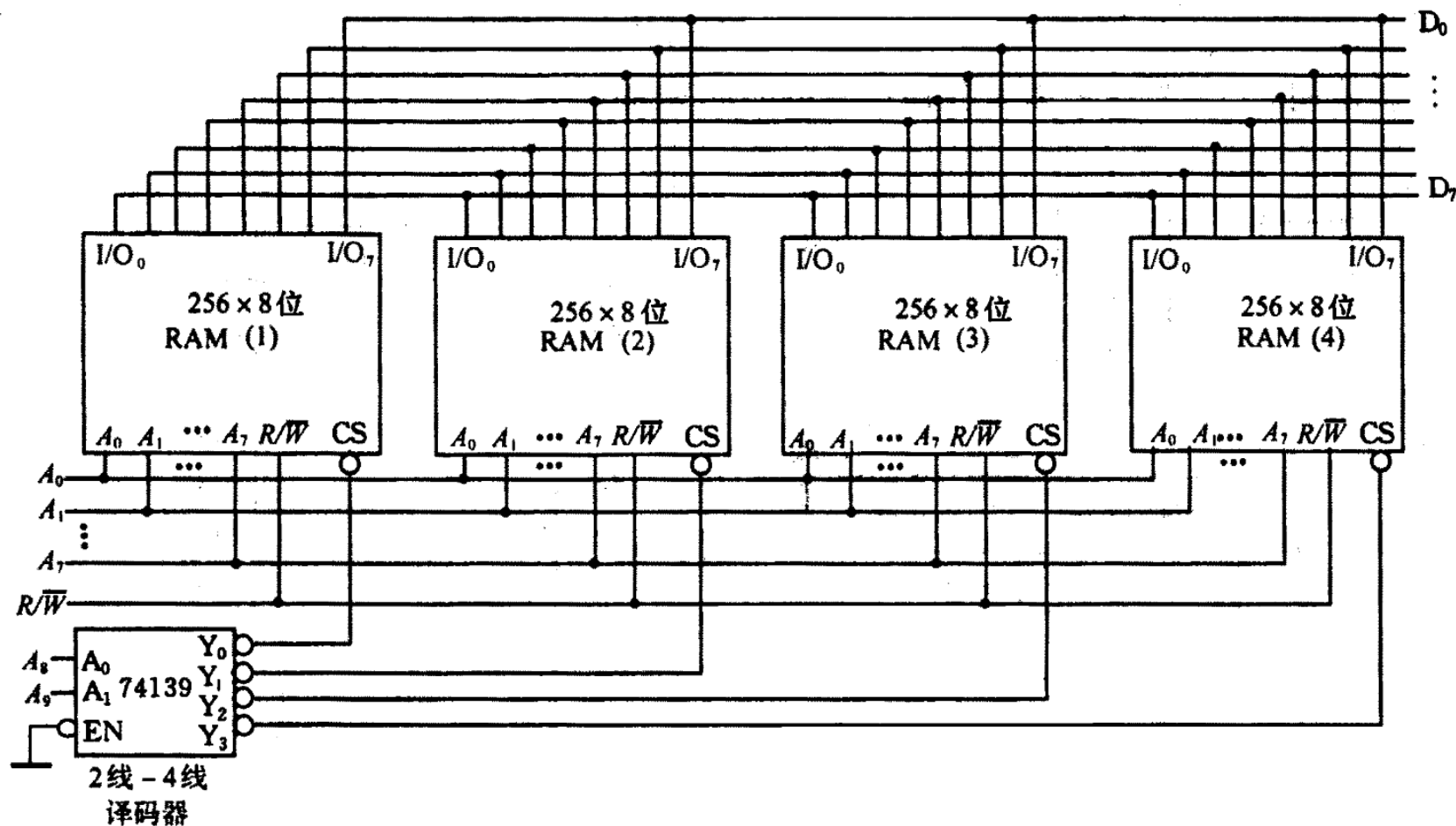


4片4K × 4位的RAM扩展成4K × 16位原理图



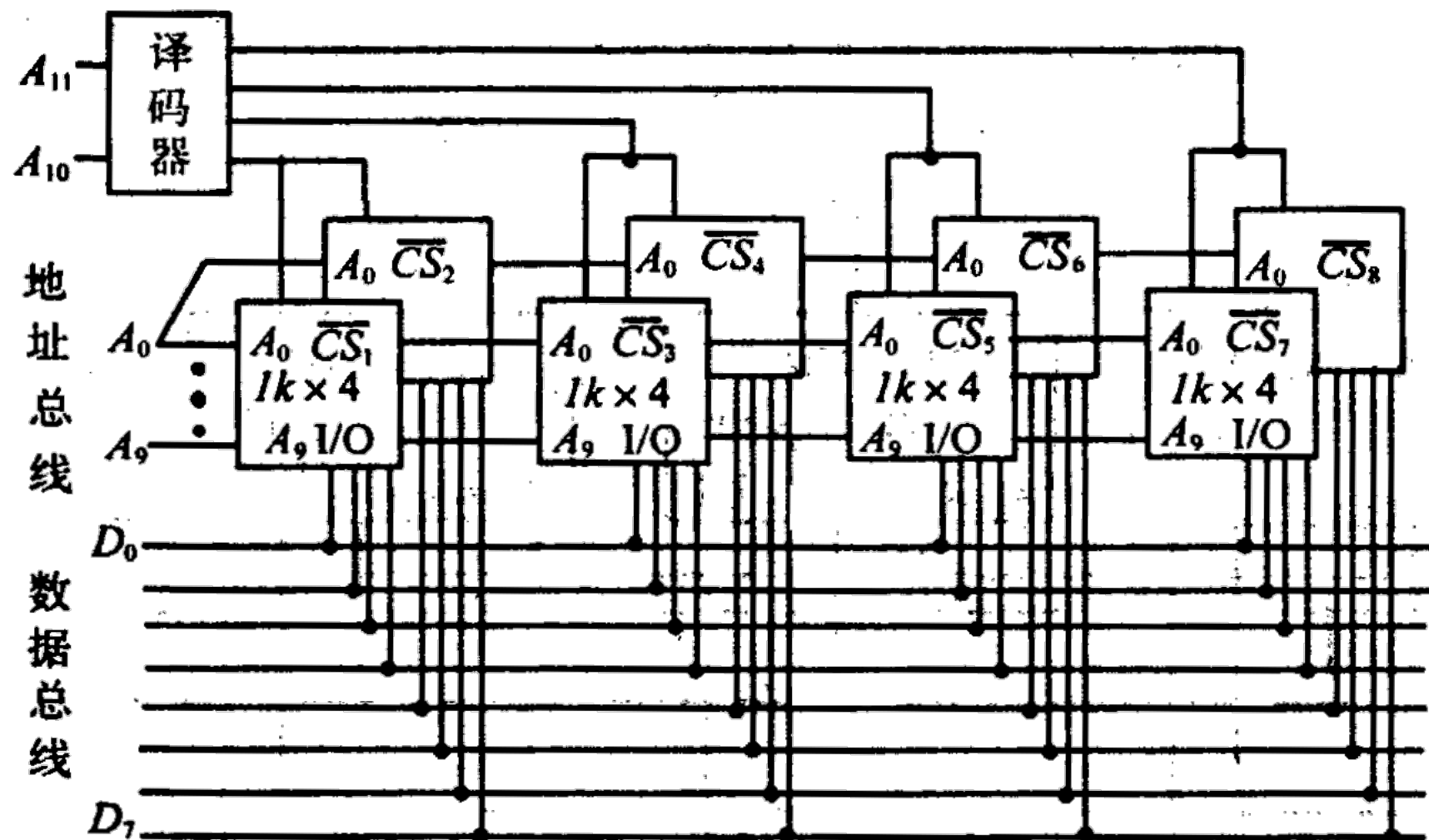
## 4、字数的扩展

例如，将 $256 \times 8$ 位的RAM扩展成 $1K \times 8$ 位的RAM  
需要4片 $256 \times 8$ 的RAM芯片和一片2-4译码器




例、用 $1K \times 4$  RAM实现 $4K \times 8$ 位存储器

字数由1K扩展到4K，需4片 $1K \times 4$ 的RAM和一片2-4译码器，每一片4位要扩展到8位，需2片 $1K \times 4$ 的RAM并联，因而共需要8片。



## 5、RAM生成实例



# RAM: 1-PORT

[About](#)[Documentation](#)

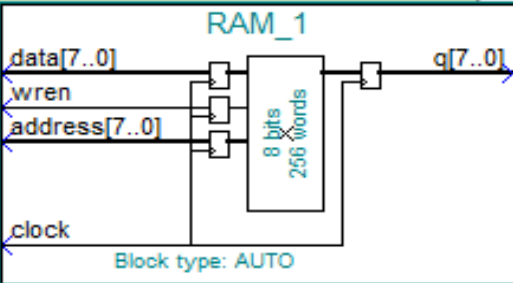
1 Parameter Settings

2 EDA

3 Summary

Widths/Blk Type/Cls>Regs/Clen/Byte Enable/Aclrs>Read During Write Option>Mem Init>

RAM\_1



Block type: AUTO

Resource Usage

1 M10K

Currently selected device family: 

Cyclone V

☒ Match project/default

How wide should the 'q' output bus be? 

8

 bits

How many 8-bit words of memory? 

256

 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?

☒ Auto

☐ MLAB

☐ M10K

☐ M-RAM

☐ LCs

Options...

Set the maximum block depth to 

Auto

 words

What clocking method would you like to use?

☒ Single clock

☐ Dual clock: use separate 'input' and 'output' clocks

Cancel

< Back

Next >

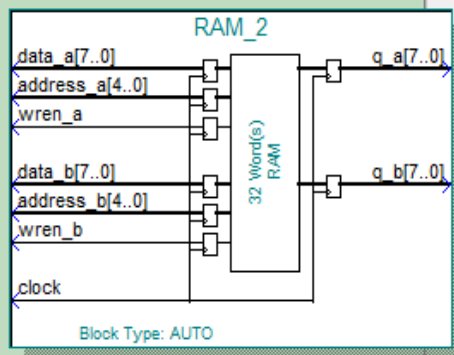
Finish

# RAM: 2-PORT

[About](#)[Documentation](#)

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clocks/Adrs Output1 Output2 Mem Init



Currently selected device family: Cyclone V

☒ Match project/default

How will you be using the dual port RAM?

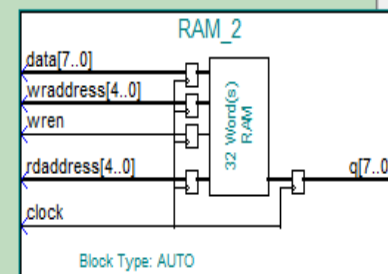
- ☐ With one read port and one write port
- ☒ With two read/write ports

How do you want to specify the memory size?

- ☒ As a number of words
- ☐ As a number of bits

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clocks/Adrs Output1 Mem Init



What docking method do you want to use?

- ☒ Single clock
- ☐ Dual clock: use separate 'read' and 'write' clocks
- ☐ Dual clock: use separate 'input' and 'output' clocks
- ☐ No clock (fully asynchronous)
- ☐ Customize clocks for A and B ports

☐ Create a 'rden' read enable signal

Byte Enable Ports

- ☐ Create byte enable for port A
- ☐ Create byte enable for port B

What is the width of a byte for byte enables? 8 bits

☐ Enable error checking and correcting (ECC) to check and correct single bit errors and detect double errors

☐ Enable ECC pipeline registers before the output decoder to achieve the same performance as non-ECC mode at the expense of one cycle of latency

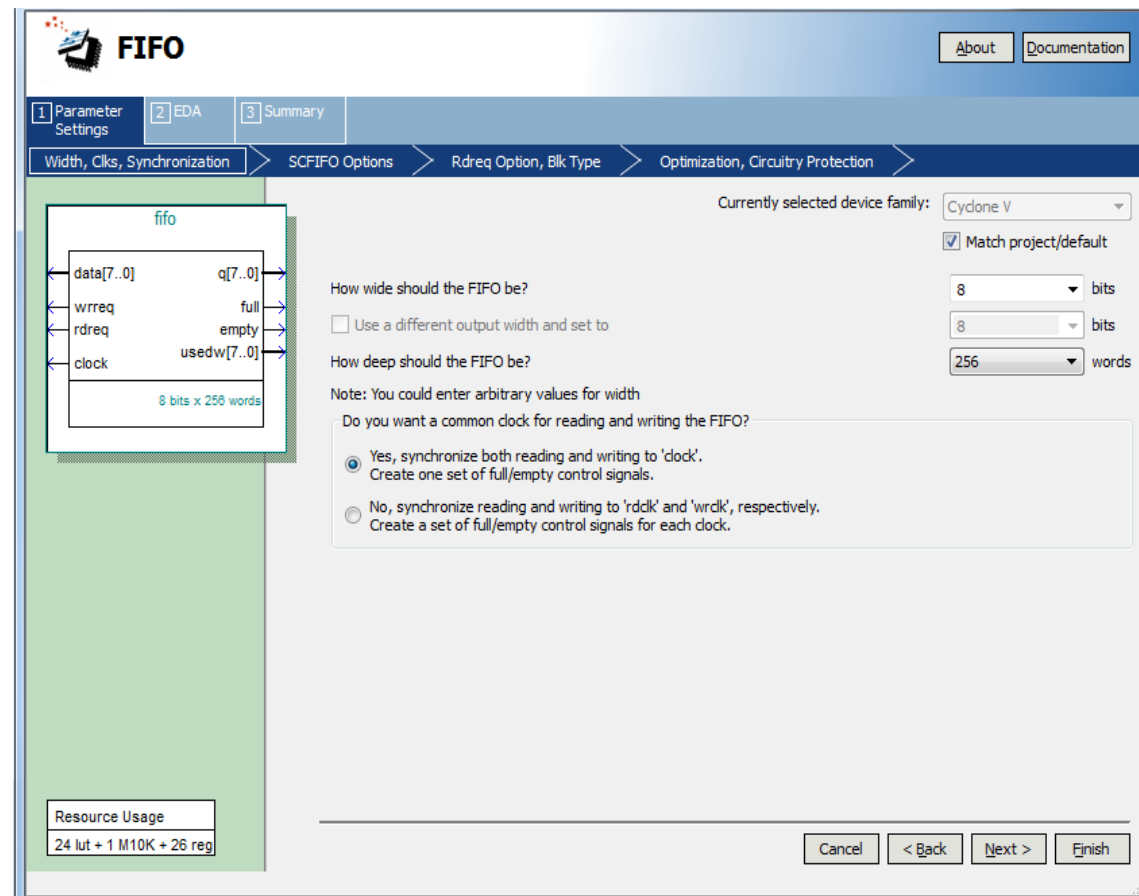
## 6、FIFO存储器

FIFO：先入先出。是一种存储器结构。

- 输入数据速率和输出速率不匹配时，作为临时存储单元。
- 用于不同时钟域之间的同步。
- 输入数据和输出数据间数据宽度不匹配时，用于数据宽度调整。

分类：

- 同步FIFO：单一时钟写入和读取
- 异步FIFO：两个时钟域之间传送数据。



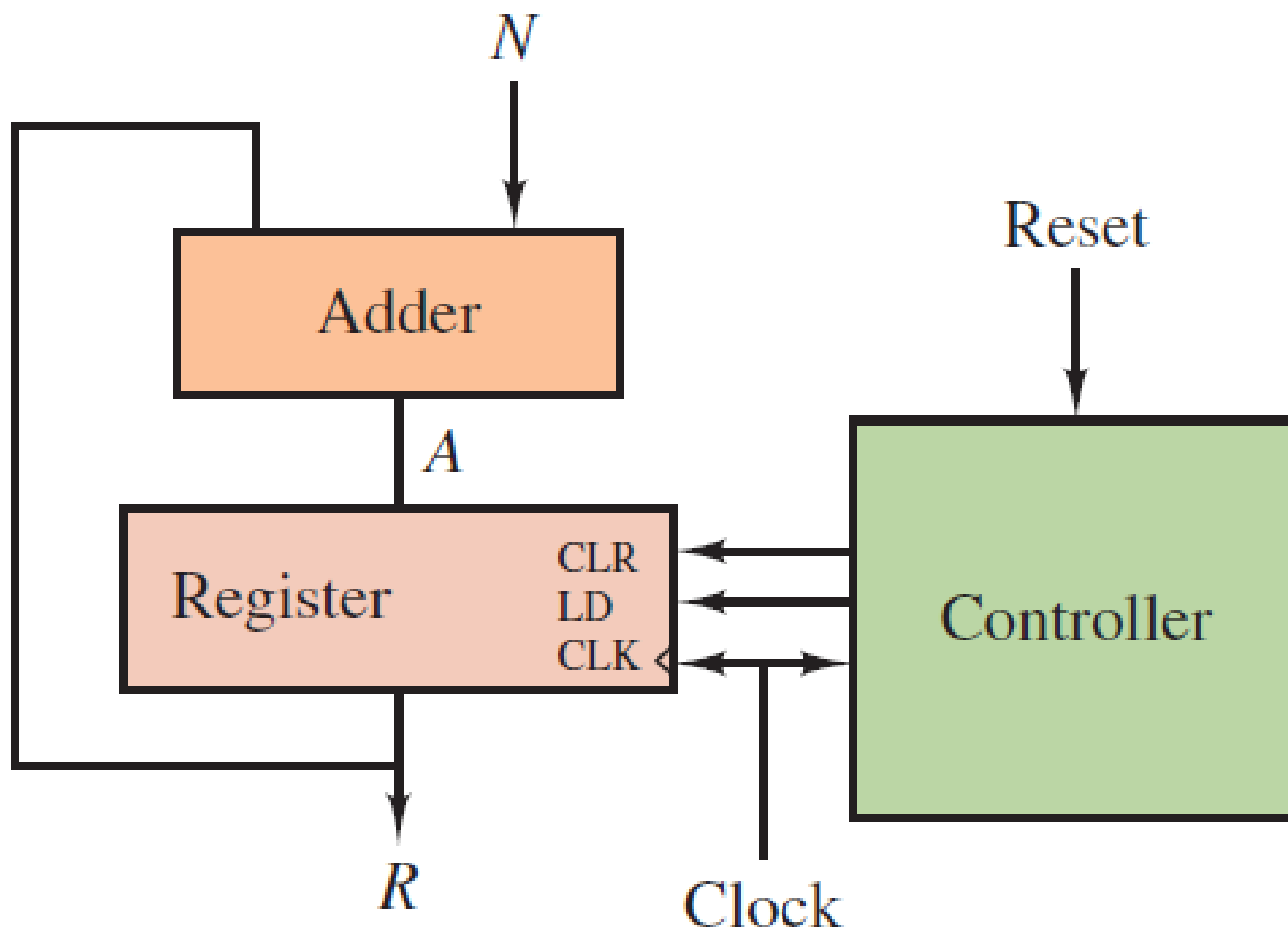
- 开始工作（电路复位），FIFO为空，写指针和读指针指向同一个位置（地址），通常为零。
- 将数据写入FIFO时，write\_en信号有效，同时数据需要被提供给写入端口，write\_data端口输入的数据被依次写入FIFO的不同存储位置。持续向FIFO写数据时，write\_ptr将不断增加，当达到存储地址的最大值时，write\_ptr重回0地址。
- 读FIFO数据时，read\_en信号有效，下一个时钟周期开始时，总线read\_data的数据可用，read\_ptr指向下一个地址，当达到存储地址的最大值时，重回0地址。
- 可以同时FIFO进行写入和读取操作。

- 特殊情况：确保避免向满FIFO写数据和从空FIFO读数据；
- FIFO满或空时，写入（ write\_ptr ）和读取（ read\_ptr ）指针相同
- 需要FIFO\_full、FIFO\_empty信号用于表示FIFO满还是空，FIFO满时，禁止写入数据；FIFO空时，禁止读取数据。

## 输入输出端口：

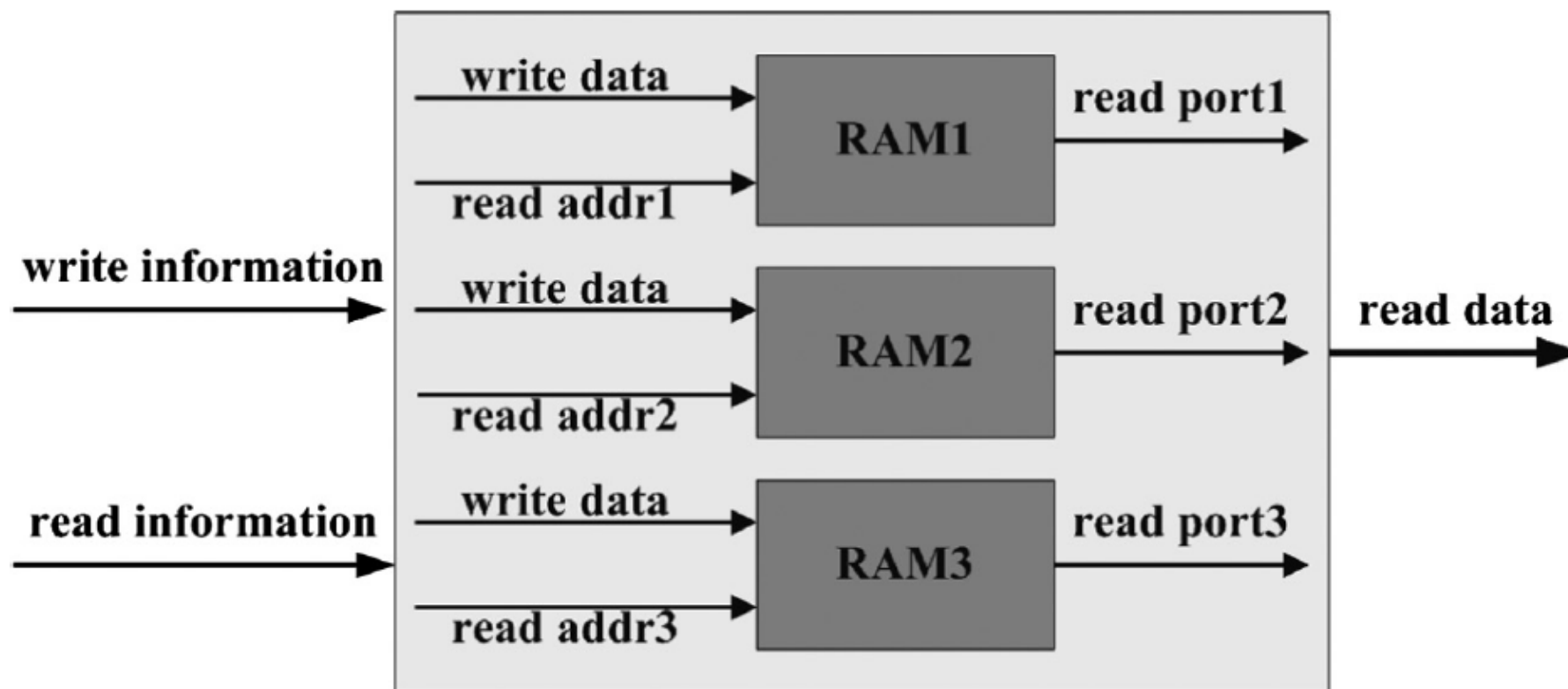
- clk、rst
- wren、wdata、rden、addata
- full、empty
- room\_avail、data\_avail、almost\_full、almost\_empty

## 7、寄存器文件 (register file)





寄存器文件又称寄存器堆，是CPU中多个寄存器组成的阵列，通常由快速的静态随机读写存储器（SRAM）实现。



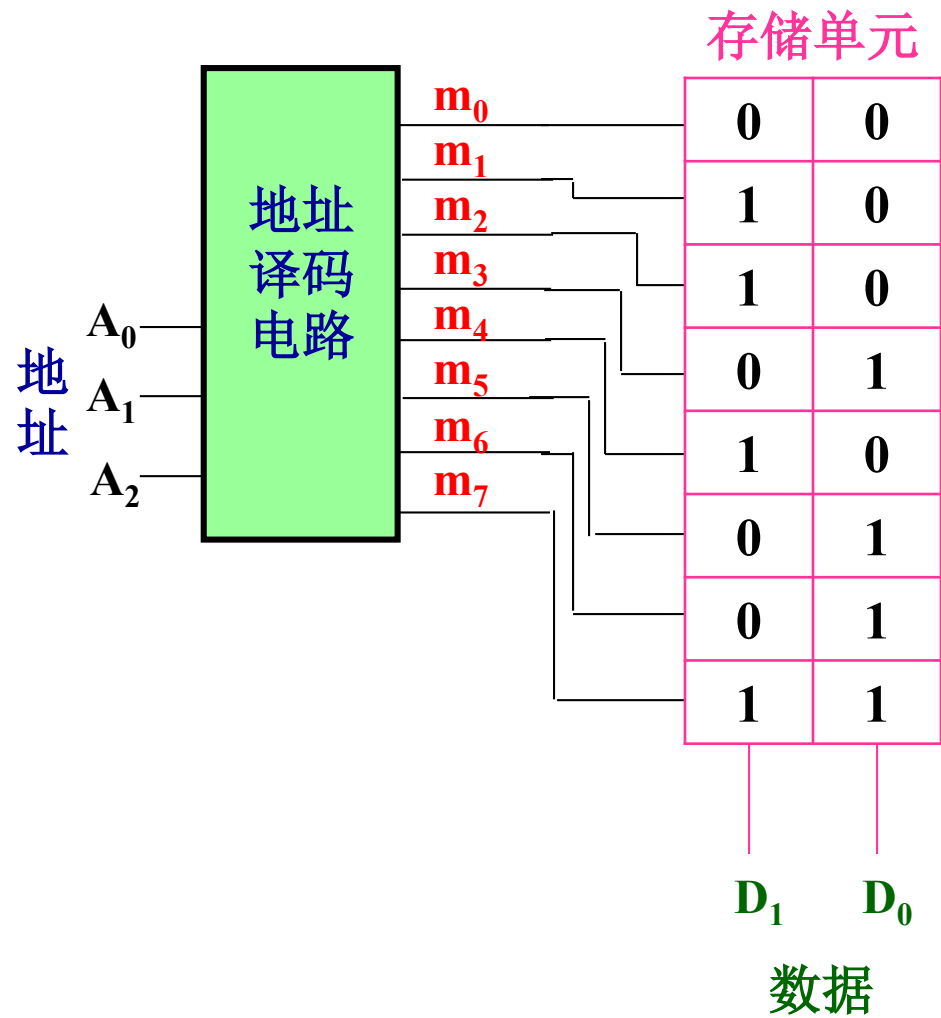
## 7.3 只读存储器ROM

ROM是存储固定信息的存储器。预先把信息写入到存储器中，在操作过程中，只能读出信息，不能写入，去掉电源，所存信息不会丢失。优点是结构简单、电路形式和规格也比较统一，经常用它存放固定的数据和程序。

ROM器件的种类很多，按存储内容存入方式的不同又可以分成固定ROM和可编程ROM。

固定只读存储器	ROM
可编程只读存储器	PROM
可擦除可编程只读存储器	EPROM
电可擦除可编程只读存储器	EEPROM

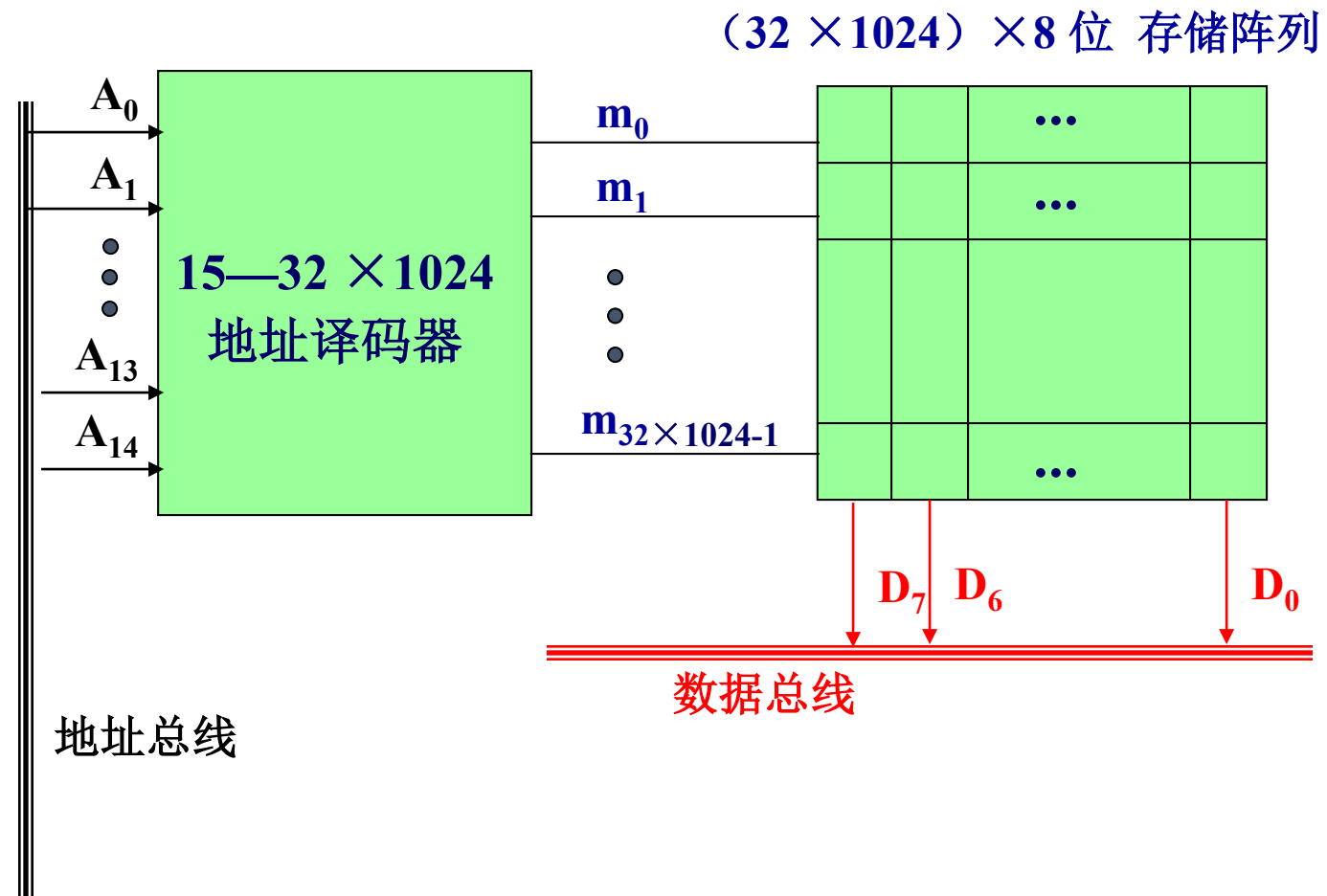
## 8 × 2 bit ROM的逻辑图



$$D_0 = m_3 + m_5 + m_6 + m_7$$

$$D_1 = m_1 + m_2 + m_4 + m_7$$

# 32K × 8 EPROM组成框图



ROM: 1-PORT

AboutDocumentation

1Parameter Settings2EDA3Summary

General>Regs/Clock/Adrs>Mem Init>

Currently selected device family: Cyclone V

☒ Match project/default

How wide should the 'q' output bus be? 8 bits

How many 8-bit words of memory? 256 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?  
☒ Auto ☐ MLAB ☐ M10K  
☐ M-RAM ☐ LCs Options...

Set the maximum block depth to Auto words

What clocking method would you like to use?  
☒ Single clock  
☐ Dual clock: use separate 'input' and 'output' clocks

ROM: 2-PORT

AboutDocumentation

1Parameter Settings2EDA3Summary

Widths/Blk Type>Clks/Rd, Byte En>Regs/Clocks/Adrs>Mem Init>

Currently selected device family: Cyclone V

☒ Match project/default

How do you want to specify the memory size?  
☒ As a number of words  
☐ As a number of bits

How many 8-bit words of memory? 32 words

☐ Use different data widths on different ports

Read Ports  
How wide should the 'q\_a' output bus be? 8  
How wide should the 'q\_b' output bus be? 8  
Note: You could enter arbitrary values for width and depth

What should the memory block type be?  
☒ Auto ☐ M10K ☐ M144K  
Set the maximum block depth to Auto words

ROM: 1-PORT

AboutDocumentation

1Parameter Settings2EDA3Summary

General>Regs/Clock/Adrs>Mem Init>

Which ports should be registered?  
☐ 'data' input port  
☒ 'address' input port  
☒ 'q' output port

☐ Create one clock enable signal for each clock signal.  
☐ Note: All registered ports are controlled by the enable signal(s) More Options...

☐ Create byte enable for port A

What is the width of a byte for byte enables? 8 bits

☐ Create an 'adr' asynchronous clear for the registered ports More Options...

☐ Create a 'rden' read enable signal

ROM: 2-PORT

AboutDocumentation

1Parameter Settings2EDA3Summary

Widths/Blk Type>Clks/Rd, Byte En>Regs/Clocks/Adrs>Mem Init>

Which ports should be registered?  
☒ Write input ports More Options...  
☒ Read output port(s) 'q\_a' and 'q\_b'

☐ Create one clock enable signal for each clock signal More Options...

☐ Use different clock enables for registers

☐ Create an 'adr' asynchronous clear for the registered ports More Options...