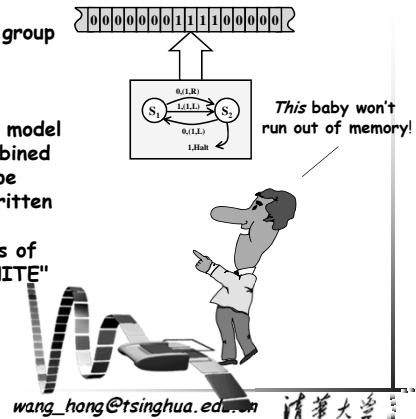


## Turing Machines

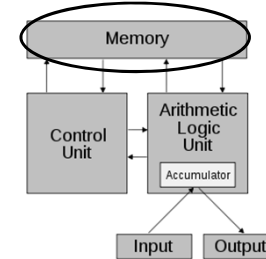
Alan Turing was one of a group of researchers studying alternative models of computation. He proposed a conceptual model consisting of an FSM combined with an infinite digital tape that could be read and written at each step. Turing's model (like others of the time) solves the "FINITE" problem of FSMs.



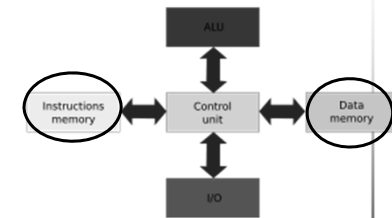
wang\_hong@tsinghua.edu.cn

清华大学

## • Von Neumann architecture



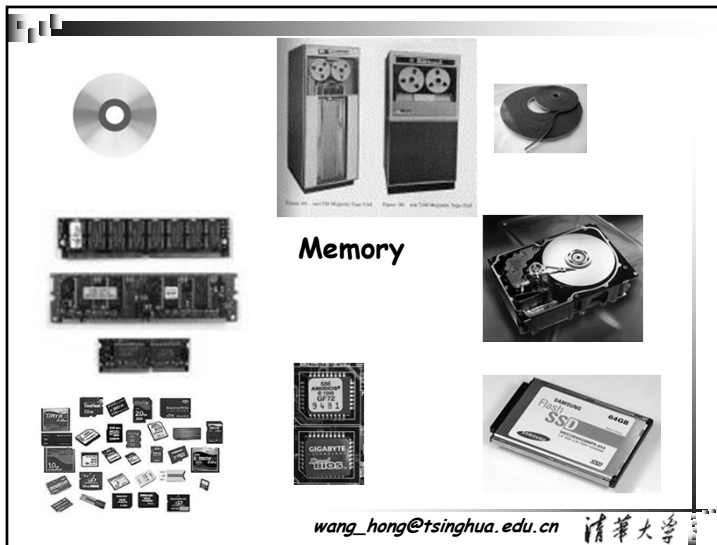
## • Harvard architecture



wang\_hong@tsinghua.edu.cn

清华大学

## Memory



wang\_hong@tsinghua.edu.cn

清华大学

## 第七章 半导体存储器

wang\_hong@tsinghua.edu.cn

清华大学

一些术语:

- 存储器单元 **cell**  
用于存储一个bit的电路单元
- 字节 **Byte = 8bits**
- 字 **Word = 1-8 Bytes**  
1个字中有8-64 bits

1B=8bit; 1KB=1024B; 1MB=1024KB; 1GB=1024MB

wang\_hong@tsinghua.edu.cn

清华大学

一些术语:

- 容量  
表示特定存储器单元或整个存储器系统能够存储多少bits
- 密度  
表示容量的另一术语
- 地址  
表示word在存储系统中位置

wang\_hong@tsinghua.edu.cn

清华大学

## 第七章 半导体存储器

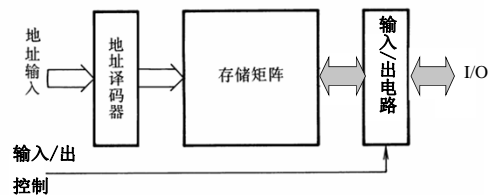
### 7.1 概述

能存储大量二值信息的器件

! 单元数庞大

一、一般结构形式

! 输入/输出引脚数目有限



wang\_hong@tsinghua.edu.cn

清华大学

### 二、分类

#### 1、从存/取功能分:

##### ①只读存储器

(Read-Only-Memory)

掩模 ROM

可编程 ROM

可擦除的可编程 EPROM

##### ②随机读/写

(Random-Access-Memory)

静态RAM

动态RAM

#### 2、从工艺分:

##### ①双极型

##### ②MOS型

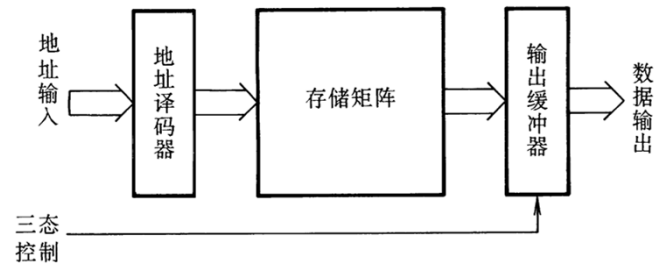
wang\_hong@tsinghua.edu.cn

清华大学

## 7.2 ROM

### 7.2.1 掩模ROM

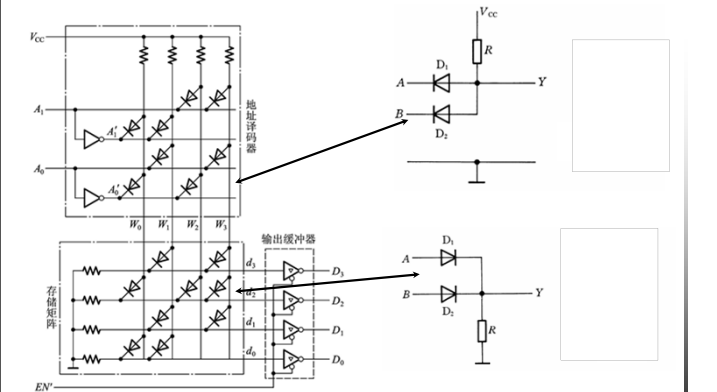
#### 一、结构



wang\_hong@tsinghua.edu.cn

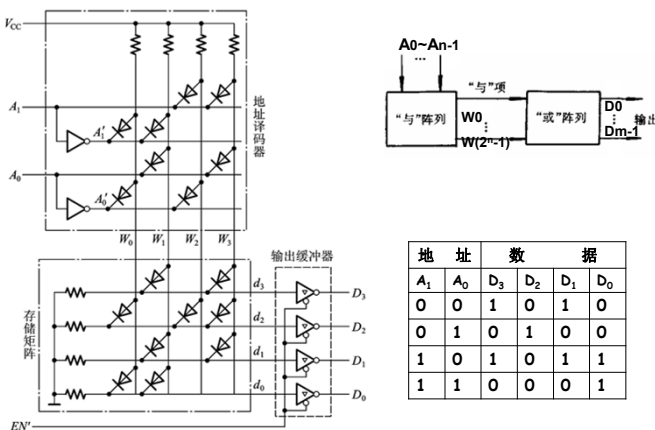
清华大学

#### 二、举例



wang\_hong@tsinghua.edu.cn

清华大学

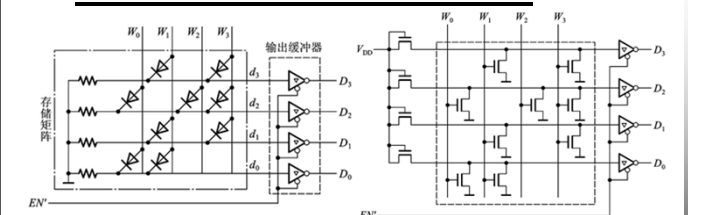


wang\_hong@tsinghua.edu.cn

清华大学

#### 两个概念：

- 存储矩阵的每个交叉点是一个“存储单元”，存储单元中有器件存入“1”，无器件存入“0”



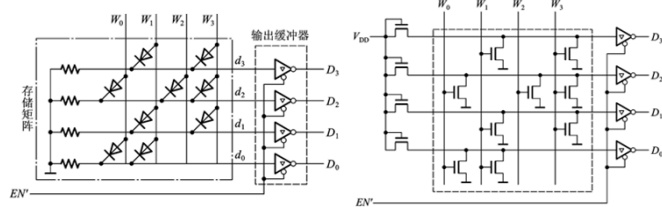
- 存储器的容量：“字数 × 位数”

wang\_hong@tsinghua.edu.cn

清华大学

掩模ROM的特点:

出厂时已经固定, 不能更改, 适合大量生产  
简单, 便宜, 非易失性

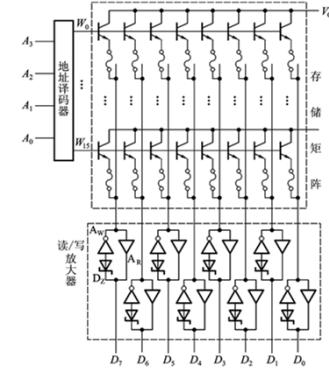


wang\_hong@tsinghua.edu.cn

清华大学

## 7.2.2 可编程ROM (PROM)

总体结构与掩模ROM一样, 但存储单元不同



写入时, 要使用编程器

wang\_hong@tsinghua.edu.cn

清华大学

## 7.2.3 可擦除的可编程ROM (EPROM)

总体结构与掩模ROM一样, 但存储单元不同  
一、用紫外线擦除的PROM (UVEPROM)

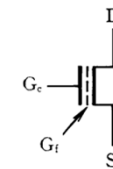
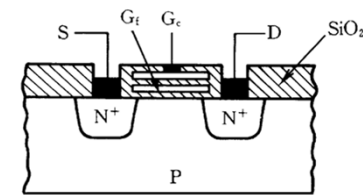


wang\_hong@tsinghua.edu.cn

清华大学

## SIMOS (Stacked-gate Injection MOS)

叠栅注入MOS管



$G_c$ : 控制栅  
 $G_f$ : 浮置栅

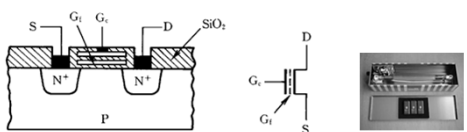
工作原理:

若  $G_f$  上充以负电荷, 则  $G_c$  处正常逻辑高电平下不导通

若  $G_f$  上未充负电荷, 则  $G_c$  处正常逻辑高电平下导通

wang\_hong@tsinghua.edu.cn

清华大学



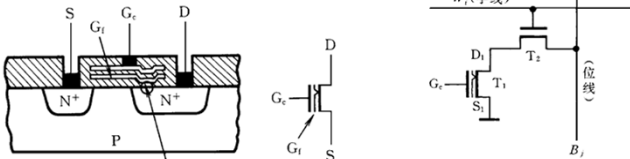
“写入”：雪崩注入， $D-S$ 间加高压（ $20 \sim 25V$ ），发生雪崩击穿  
同时在  $G_c$  上加  $25V$ ,  $50ms$  宽的正脉冲，  
吸引高速电子穿过  $SiO_2$  到达  $G_f$ ，形成注入电荷

“擦除”：通过照射产生电子-空穴对，提供泄放通道  
紫外线照射  $20 \sim 30$  分钟（阳光下一周，荧光灯下3年）

wang\_hong@tsinghua.edu.cn 清华大学

### 二、电可擦除的可编程ROM（E<sup>2</sup>PROM）

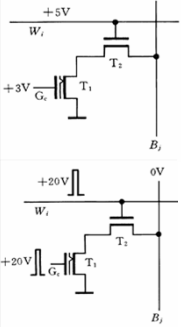
总体结构与掩模ROM一样，但存储单元不同  
为克服UVEPROM擦除慢，操作不便的缺点  
采用FLOTOX（浮栅隧道氧化层MOS管）



隧道区

$G_f$ 与 $D$ 之间有小的隧道区， $SiO_2$ 厚度  $< 2 \times 10^{-8}m$   
当场强达到一定大小（ $10^7V/cm$ ），电子会穿越隧道  
——“隧道效应”

wang\_hong@tsinghua.edu.cn 清华大学



工作原理：  
 $G_f$ 充电后，正常读出  $G_c$  电压（ $3V$ ）下， $T$ 截止  
未充电时，正常读出  $G_c$  电压（ $3V$ ）下， $T$ 导通

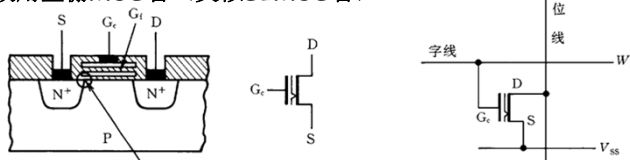
充电： $W_i, G_c$ 加 $20V$ ,  $10ms$ 的正脉冲， $B_j$ 接0  
电子隧道区  $\rightarrow G_f$

放电： $G_c$ 接0， $W_i, B_j$ 加正脉冲  
 $G_f$ 上电荷经隧道区放电

wang\_hong@tsinghua.edu.cn 清华大学

### 三、快闪存储器（Flash Memory）

为提高集成度，省去 $T_2$ （选通管）  
改用叠栅MOS管（类似SIMOS管）

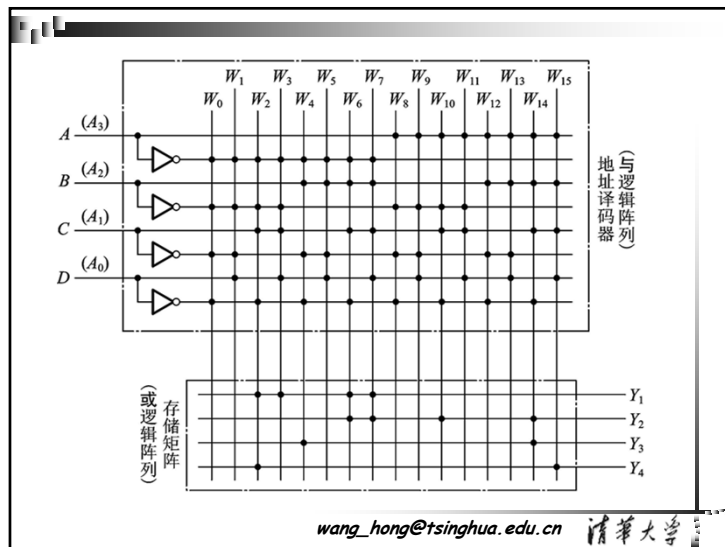


隧道区

$G_f$ 与衬底间 $SiO_2$ 更薄（ $10 \sim 15nm$ ）  
 $G_f$ 与 $S$ 区有极小的重叠区（隧道区）

\*工作原理：  
 $G_f$ 充电，利用雪崩注入方式  
 $G_f$ 放电，利用隧道效应  
 $D-S$ 加正压（ $6V$ ）， $V_{ss}$ 接0  
 $G_c = 0, V_{ss}$ 加 $12V$ ,  $100ns$ 的正脉冲  
 $G_c$ 加 $12V$ ,  $10us$ 的正脉冲  
 $G_f$ 上电荷经隧道区放电

wang\_hong@tsinghua.edu.cn 清华大学

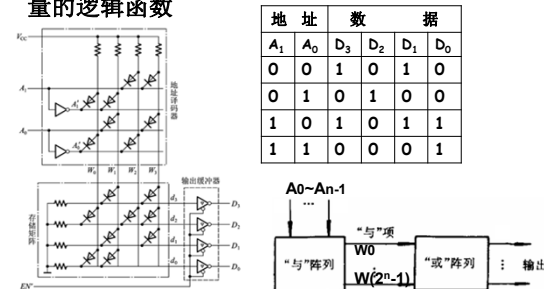


## 7.5 用存储器实现组合逻辑函数

### 一、基本原理

从ROM的数据表可见:

若以地址线为输入变量, 则数据线即为一组关于地址变量的逻辑函数

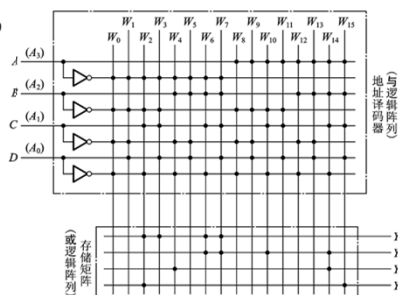


### 二、举例

用ROM产生:

$$\begin{cases} Y_1 = A'BC + A'B'C \\ Y_2 = AB'CD' + BCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$

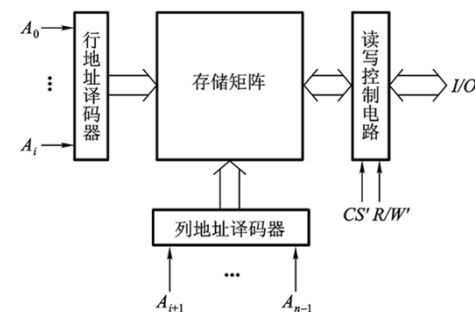
$$\begin{cases} Y_1 = \sum m(2,3,6,7) \\ Y_2 = \sum m(6,7,10,14) \\ Y_3 = \sum m(4,14) \\ Y_4 = \sum m(2,15) \end{cases}$$

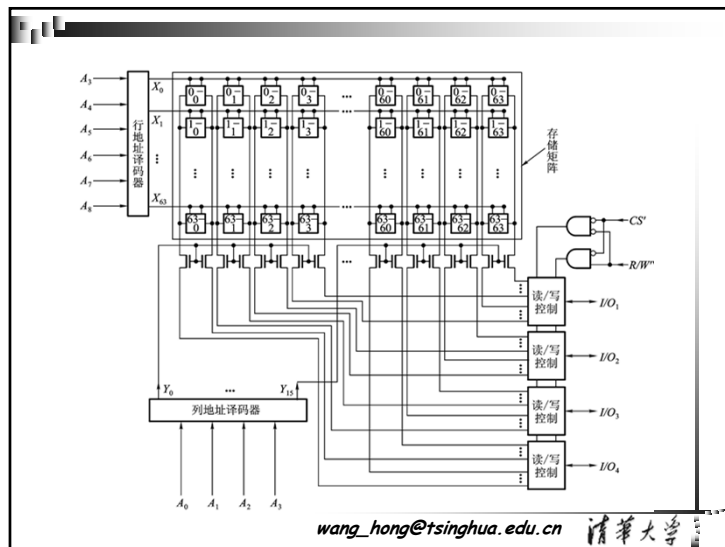


## 7.3 随机存储器RAM

### 7.3.1 静态随机存储器 (SRAM)

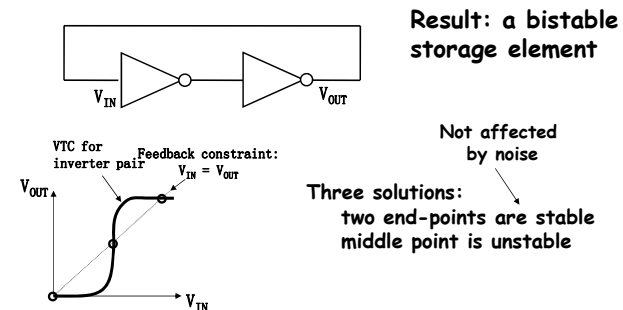
#### 一、结构与工作原理



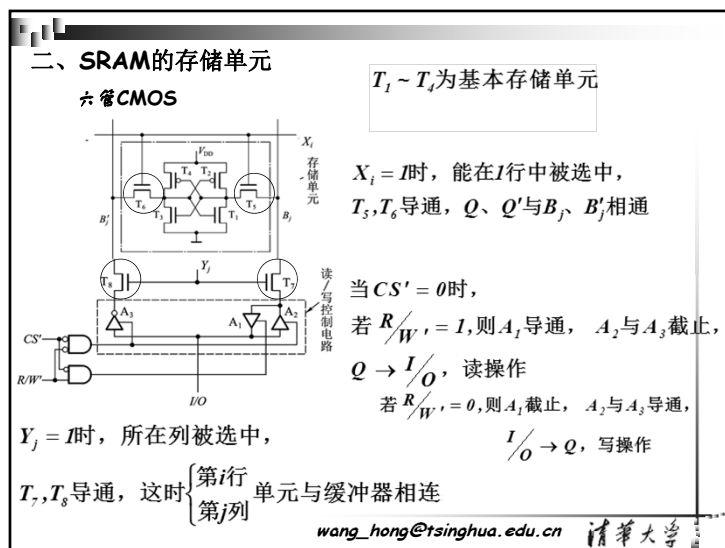


## Storage: Using Feedback

IDEA: use positive feedback to maintain storage indefinitely.  
Our logic gates are built to restore marginal signal levels, so noise shouldn't be a problem!



wang\_hong@tsinghua.edu.cn 清华大学

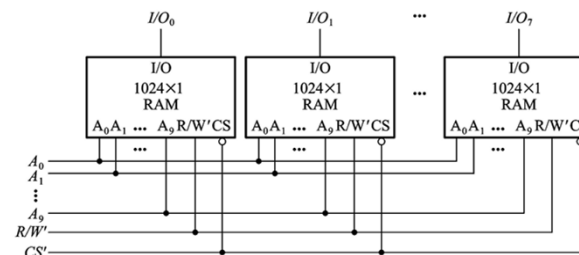


## 7.4 存储器容量的扩展

### 7.4.1 位扩展方式

适用于每片RAM, ROM字数够用而位数不够时  
接法: 将各片的地址线、读写线、片选线并联即可

例: 用八片  $1024 \times 1$  位  $\rightarrow 1024 \times 8$  位的RAM

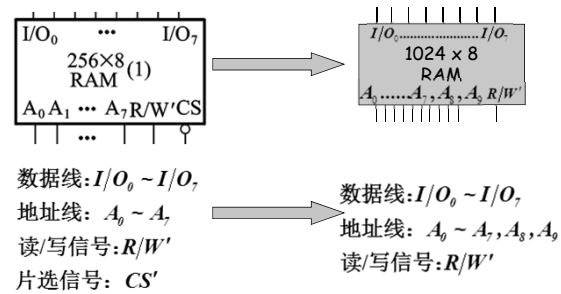


wang\_hong@tsinghua.edu.cn 清华大学

## 7.4.2 字扩展方式

适用于每片RAM,ROM位数够用而字数不够时

例：用四片  $256 \times 8$  位  $\rightarrow 1024 \times 8$  位 RAM



wang\_hong@tsinghua.edu.cn

清华大学

每一片提供 256 个字，需要 256 个地址 ( $A_{0-7}: 0 \sim 255$ )

用  $A_8, A_9$  两位代码区分四片

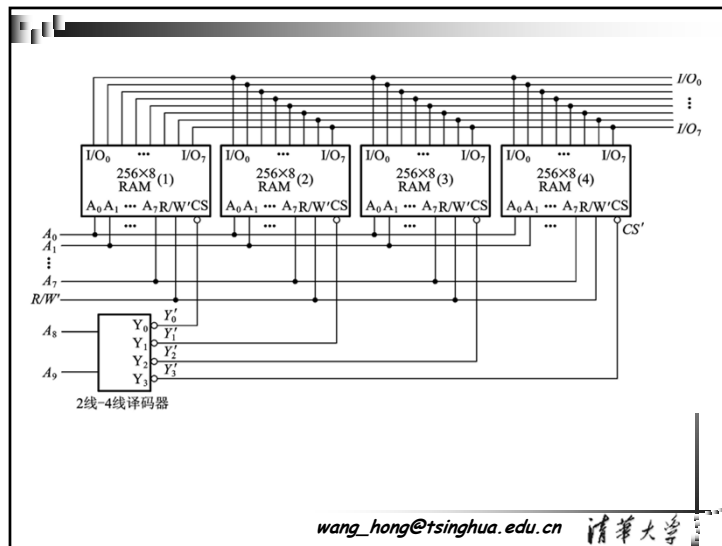
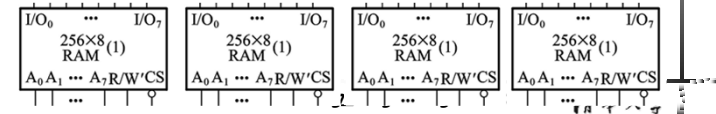
即将  $A_8, A_9$  译成  $Y'_0 \sim Y'_3$ ，分别接四片的  $CS'$

$A_9$	$A_8$	$CS'_1$	$CS'_2$	$CS'_3$	$CS'_4$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

四片的地址分配就是：

$00 A_7 \sim A_0$ ,  $01 A_7 \sim A_0$ ,  $10 A_7 \sim A_0$ ,  $11 A_7 \sim A_0$

$0 \sim 255$ ,  $256 \sim 511$ ,  $512 \sim 767$ ,  $768 \sim 1023$



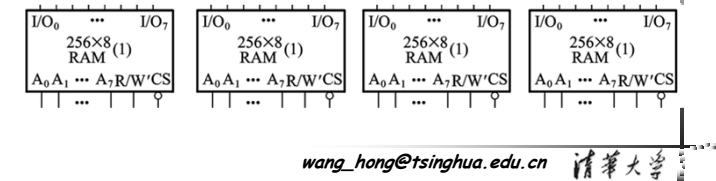
wang\_hong@tsinghua.edu.cn

清华大学

$A_1$	$A_0$	$CS'_1$	$CS'_2$	$CS'_3$	$CS'_4$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

四片的地址分配就是：

$A_9 \sim A_2 00$ ,  $A_9 \sim A_2 01$ ,  $A_9 \sim A_2 10$ ,  $A_9 \sim A_2 11$



wang\_hong@tsinghua.edu.cn

清华大学