Main Memory

Ref: CSAPP Page 399(435 in PDF)

display buffer `->` frame buffer：帧缓冲区，内存中的一块区域，用来处理图像

DMA(direct memory access): 是一个controller，把display buffer 和 output 相连，可能有多个

- DRAM(dynamic，时序逻辑，电容)

  主存、帧缓冲区

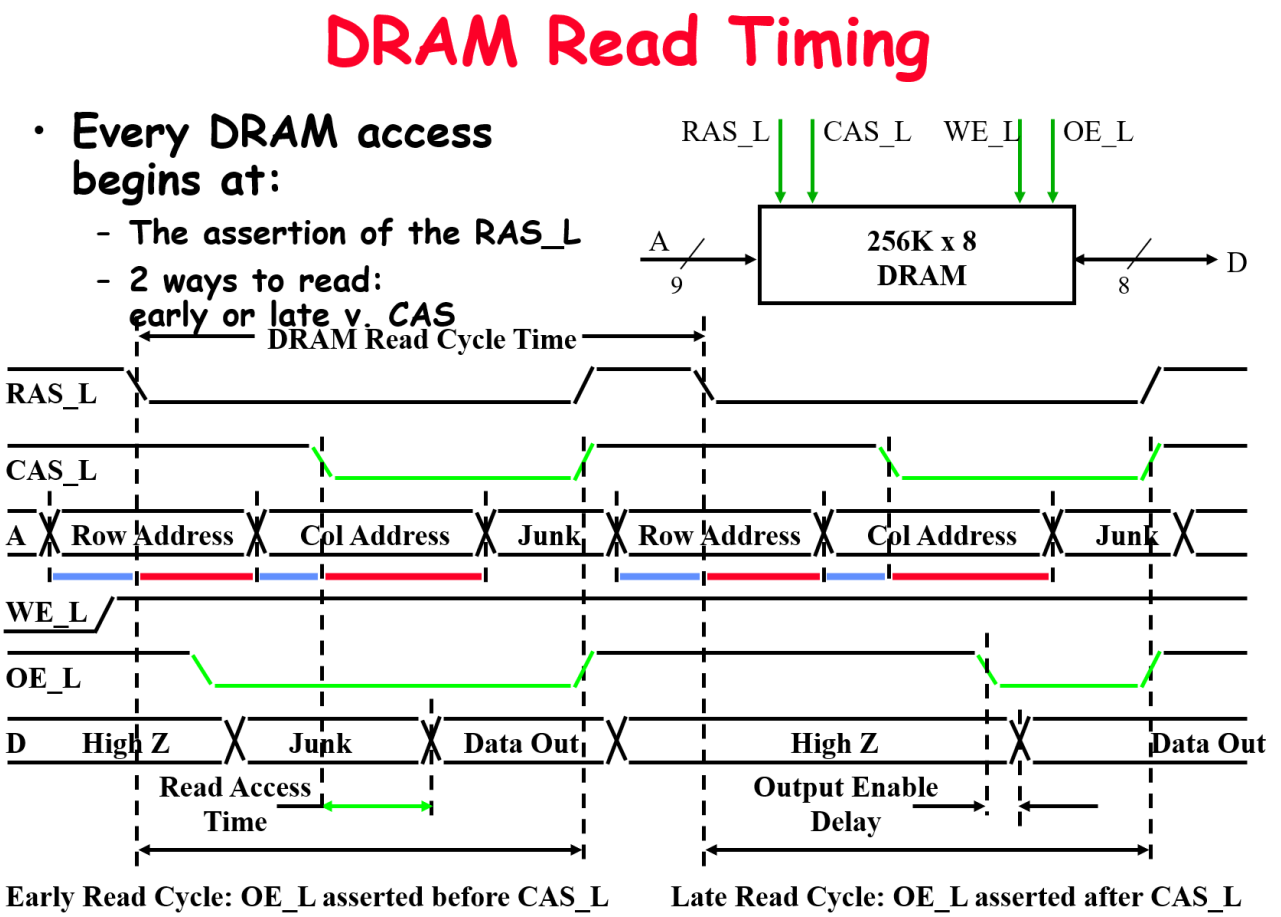
  需要刷新，对环境敏感

- SRAM(static，组合逻辑`->`状态逻辑，晶体管)

  cache

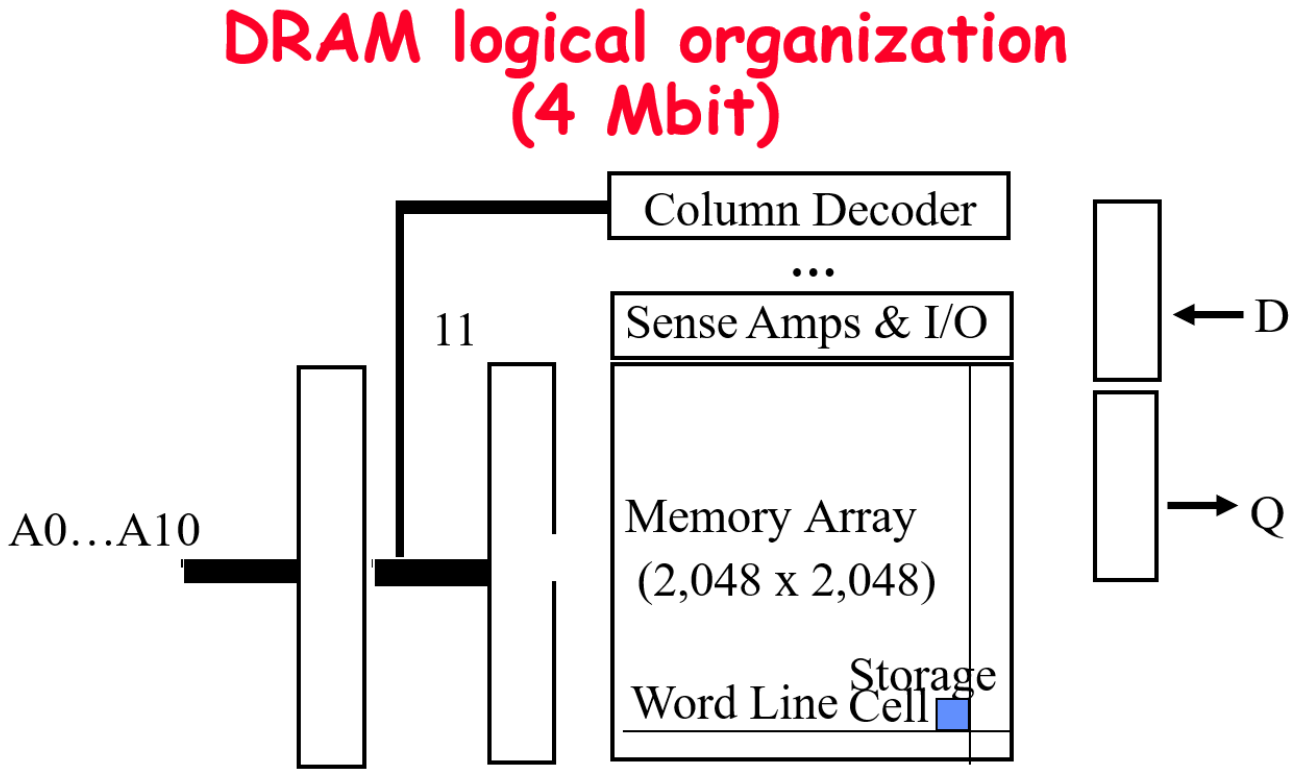
  双稳态，亚稳态

DRAM



- RAS: Row Access Strobe(脉冲)，得到行地址
- CAS: Column，得到列地址

二者的高低电平有4种组合，可以表示四种状态。



- **Square root of bits per RAS/CAS**

设计成二维数组的目的，是防止引脚过多。

Main Memory Organizations

- Simple

  CPU, Cache, Bus, Memory same width (32 or 64 bits)

- Wide（非常宽的总线，假设字长为N）

  CPU/Mux 1 word

  Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits; UtraSPARC) 512)

- Interleaved（总线字长仍为1）

  CPU, Cache, Bus 1 word:

  Memory N Modules (4 Modules for instance);

$$AMAT_{Mem} = T_{addr} + T_{access} + T_{trans}$$

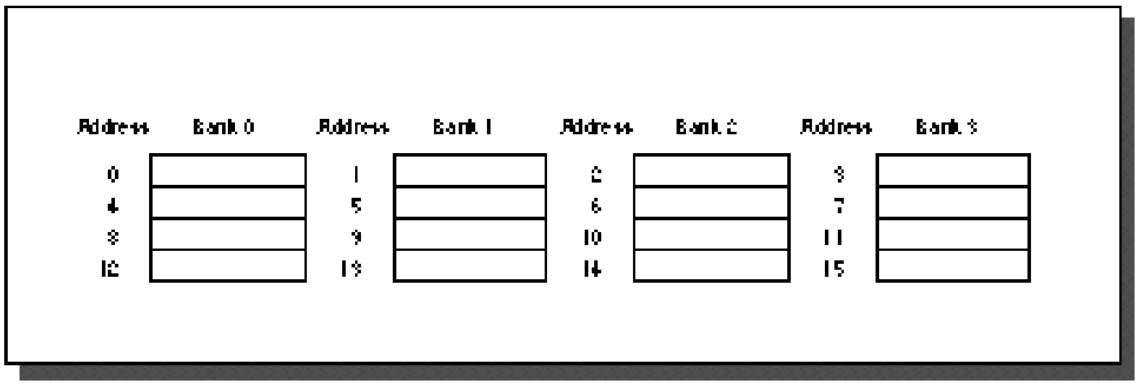
transfer：send the data

multi-bank

把连续的内存地址，划分为 N个 bank（内部模N同余），下图划分成了4个bank

Main Memory Performance

- Timing model (word size is 32 bits)
  - 1 to send address,
  - 6 access time, 1 to send data
  - Cache Block is 4 words
- Simple M.P. = 4 x (1+6+1) = 32
- Wide M.P. = 1 + 6 + 1 = 8
- Interleaved M.P. = 1 + 6 + 4x1 = 11



对于Interleaved：

- address/transfer 是在总线中进行的，因此一次只能处理一个字长。
- access 是在 bank 内部进行，一层一层往下查找（0123 -> 4567...）

所以access的查找只需要花费一倍的时间，而transfer的返回需要每个bank依次返回，也就是 4*1

> 可以把多个bank合成为一个superbank，构成递归的结构

bank数目=N，不是越多越好！

实际中采用并行，也就是在上一个transfer阶段执行的同时，开始下一个的address，那么要求该transfer执行结束前，下一个数据最多只能执行到access结束。

简单来说，$T_{addr} + T_{access} \leq N * T_{trnas}$

这样就可以利用类似 pipeline 的操作，大大提高效率。

同时需要一个buffer，来支持上一个数据 transfer 与 下一个access 的并行。

Bank Conflict

为了保证access的效率，要求内存访问时应该连续访问（0123 -> 4567...）

而如果出现了 0 -> 4 -> 8...的访问模式，access就会退化到simple的效率。比方说一个二维数组 a[n][4]，而访问 a[0][0] -> a[1][0] -> a[2][0]...

因此，需要在编译器层面上对代码进行优化

Fast Bank Number

Fast Bank Number

- **Chinese Remainder Theorem**

  As long as two sets of integers ai and bi follow these rules

  $$b_i = x \bmod a_i, 0 \le b_i < a_i, 0 \le x < a_0 \times a_1 \times a_2 \times \ldots$$

  and that ai and aj are co-prime if i ≠ j, then the integer x has only one solution (unambiguous mapping):

  - bank number = b_0, number of banks = a_0 (= 3 in example)
  - address within bank = b_1, number of words in bank = a_1 (= 8 in example)
  - N word address 0 to N-1, prime no. banks, words power of 2

		Seq. Interleaved			Modulo Interleaved		
Bank Number:		0	1	2	0	1	2
Address within Bank:	0	0	1	2	0	16	8
	1	3	4	5	9	1	17
	2	6	7	8	18	10	2
	3	9	10	11	3	19	11
	4	12	13	14	12	4	20
	5	15	16	17	21	13	5
	6	18	19	20	6	22	14
	7	21	22	23	15	7	23

CS252/Kubiatow

下文的 a_i 与上述无关。

把传统的bank内地址排列方式（左边，$x_i = a_i \mod m, y_i = a_i/m$），修改为右边的排列方法（$x_i = a_i \mod m, y_i = a_i \mod l$，Gao Q.S），可以加快排列时的计算速度（不需要大量除法，取模显然比）

1. 映射的唯一性

   取bank数（列数）$m = 2^k - 1$，行数 $l = 2^s$，显然二者互质，由Chinese Remainder Theorem(中国剩余定理) 可以保证有唯一解

2. 具体的映射操作：

   计算时，$x_i = a_i \mod m, y_i = a_i \mod l$

   因为 l 是 2 的幂，所以 y_i 计算很快，难点在于 x_i

   $\because a_i = q(2^k - 1) + x_i = q * 2^k + (x_i - q)$

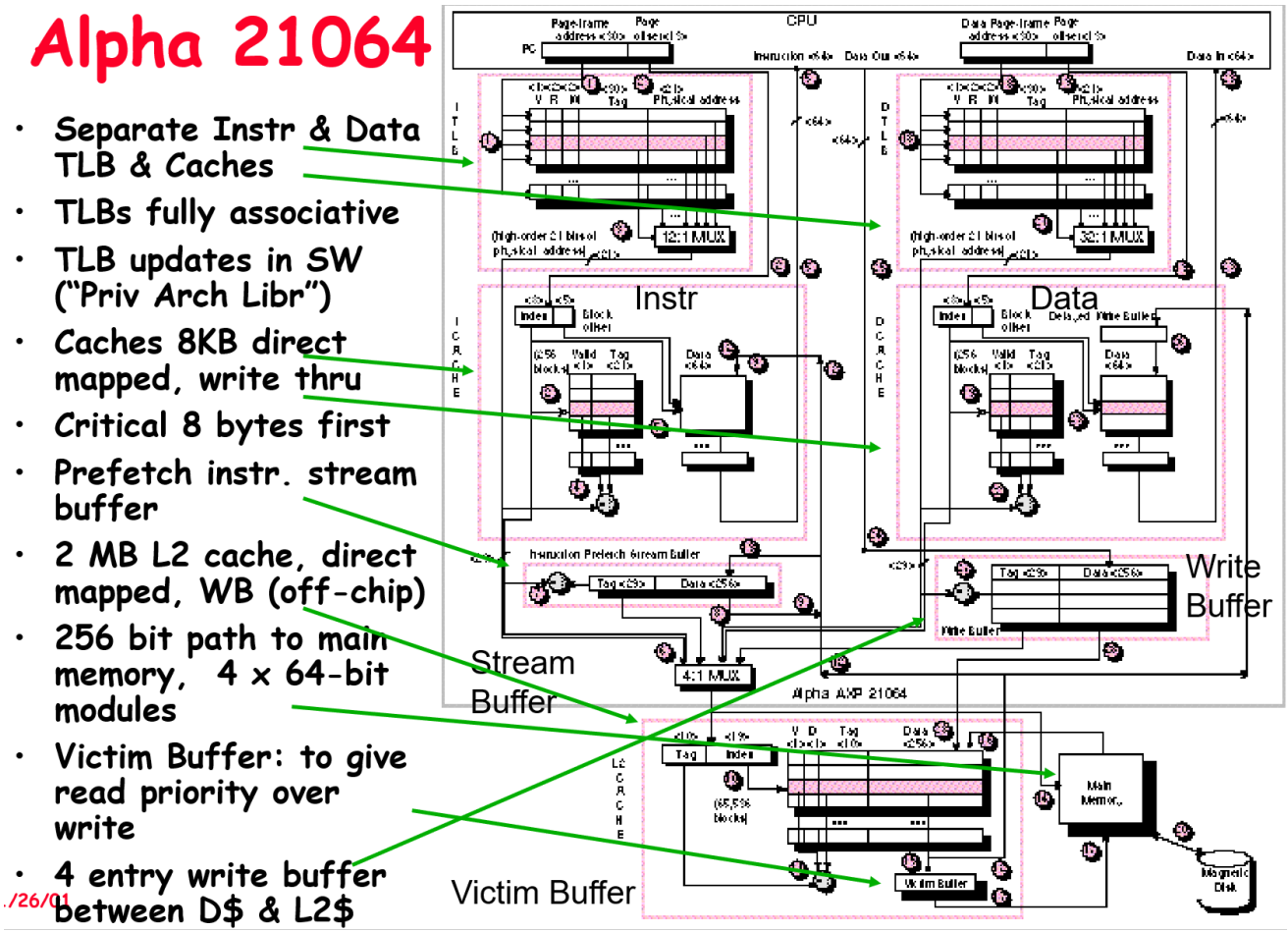
   所以就分段，a_i (默认是二进制) 的前 k 位取出来，即为q

   把 a_i 以长度 k 划分，每一段数值上=c_i，那么$a_i = \sum c_i * 2^{ik} = \sum c_i (2^k - 1 + 1)^i$，$a_i \mod m = a_i \mod (2^k - 1) = \sum c_i$（**cycle add**），因此加快了计算的速度。

> DOS：拒绝服务攻击
>
> 占用资源，不进行操作
>
> D-DOS：分布式...
>
> DNS劫持：提前回应请求

MEMS-based storage

Micro-Electro-Mechanical-System，微机电系统



RAIDs 磁盘阵列