

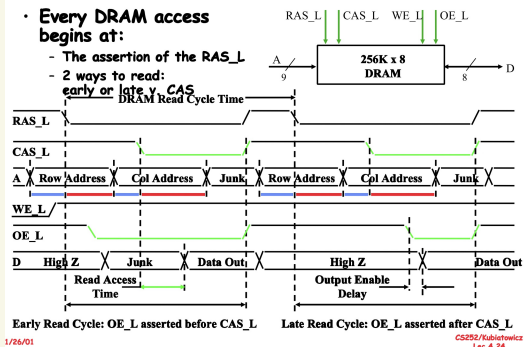
Week 9 Day 2 / 11A 90

①

DRAM Read Timing

Every DRAM access begins at:

- The assertion of the RAS_L
- 2 ways to read: early or late CAS



High Z 高阻态

RAS: row access strobe.

CAS: column access strobe

WE: write enable

OE: out enable

Main Memory performance

$$AMAT_{mem} = T_{addr} + T_{access} + T_{transfer \text{ and send data}}$$

②

MutiBank

Fast Bank Number

Chinese Remainder Theorem

As long as two sets of integers a_i and b_i follow these rules

$$b_i = x \bmod a_i, 0 \leq b_i < a_i, 0 \leq x < a_0 \times a_1 \times a_2 \times \dots$$

and that a_i and a_j are co-prime if $i \neq j$, 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

Bank Number:	Seq. Interleaved			Modulo Interleaved		
Address within Bank:	0	1	2	0	1	2
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

$$\text{module } p = 2^l - 1 \text{ or } p = 2^l + 1$$

for address d to integer pair (u, v)

$$0 \leq d \leq pm - 1$$

$$0 \leq u \leq m - 1 \quad 0 \leq v \leq p - 1$$

$$\text{PMS1} \quad v = d \bmod p \quad u = \lfloor \frac{d}{p} \rfloor$$

缺点是除法运算复杂度高

$$\text{PMS2} \quad v = d \bmod p \quad u = \lfloor \frac{d}{p} \rfloor$$

其中 $p = 2^l + 1$ 可以用 shift 计算除法但很费时

$$\text{PMS} \text{ 中国剩余定理} \quad v = d \bmod p.$$

$$\text{并且 } a = d \bmod m.$$

12月5日

Week 13 Day 1

Running Example

- This code, a scalar to a vector:
for (i=1000; i>0; i=i-1)
 x[i] = x[i] + s;
- Assume following latency all examples

A Instruction producing result	B Instruction using result	Execution in cycles	Latency in cycles	执行所需周期	下一条执行的指令要间隔多少时间 延迟需要 stall (暂停)
FP ALU op	Another FP ALU op	4	3		
FP ALU op	Store double	3	2		
Load double	FP ALU op	1	1		
Load double	Store double	1	0		
Integer op	Integer op	1	0		

```
Loop: L.D    F0,0(R1) ;F0=vector element
      ADD.D  F4,F0,F2 ;add scalar from F2
      S.D    0(R1),F4 ;store result
      DSUBUI R1,R1,8  ;decrement pointer 8B (DW)
      BNEZ   R1,Loop ;branch R1!=zero
      NOP                    ;delayed branch slot
```

调整指令顺序减少 stall

改变编译模式

Hardware scheduling

unrolling { 优点 并行, 大大减少 overhead 时间
 缺点 会要求大量 reg 代码膨胀

unrolling 条件 每段代码没有相关性

Software pipeline

基本思路: 把循环用 unrolling 展开, 然后分析 pipeline

super scalar (多路化) 一个部件一个时刻可以处理多个数据

VLM (Very long Instruction Word)

History o TI-DSP

o unrolling

o S/W pipeline

Kill Branch Instruction

- ① 用 LWC 条件执行指令
- ② super block
- ③ trace cache