

划线

Jalr rd,rs

1)rd=PC+4

2)PC=rs

1.op:6 rs:5,0:5,rd: 5

2.Datapath

3.

```
1) IF, PC+=4
IorD=0(PC),
MemRead
IRWrite
ALUSrcA=0 #PC
ALUSrcB=01 #4
ALUop=00 #+
PCsource=00 #ALU
PCwrite

2)ID PC+Eft->ALUout
ALUSrcA=0 #PC
ALUSrcB=11 #Eft
ALUop=00

3)rd,PC
RegDst=0 #R3
MemtoReg=10 #PC(New)
PegWrite
ALUSrcA=1 #A
ALUSrcB=00 #B
ALUop=00/01 #+
PCsource=0
```

微程序

微程序的意思，和组合逻辑（快，可靠，复杂，板子的点点）的区别

控制器，微程序类似查表，在使用时直接调用

单时钟直接给出操作对应的所有信号，多时钟是由于是一段一段的，分支，

存储器的字位扩展

译码器和片选（大门钥匙）

大小头

0x12345678写到存储器

小头: , 78, 56, 34, 12,

Cache

32-bit byte-addressing, 32-bit=4G , 若说是2G则为31

cache:128KB, block:64B Blocks=128KB/64B=2K

TAG,Total (记得除以8)

一个数在哪个块

0x1234567

15TAG 11index 6offset

(0123)>>1 (1456>>2) (27)

cache行(index/64)%8

44. (12 分)某计算机的主存地址空间大小为 256MB，字节编址。指令 Cache 和数据 Cache 分离，均有 8 个 Cache 行，每行 64B。数据 Cache 采用直接映射。现有两个功能相同的程序 A 和 B，其伪代码如下：

程序 A: <pre>int a[256][256]; int sum(){ int i, j, sum=0; for(i=0; i<256; i++) for(j=0; j<256; j++) sum += a[i][j]; return sum; }</pre>	程序 B: <pre>int a[256][256]; int sum(){ int i, j, sum=0; for(j=0; j<256; j++) for(i=0; i<256; i++) sum += a[i][j]; return sum; }</pre>
---	---

假定 int 为 32 位补码，变量 i, j, sum 均分配在寄存器中。数组 a 按行优先方式存放，其首地址为 320(十进制)。请回答下列问题，说明理由或给出计算过程。

- 1) 若不考虑用于 Cache 一致性维护和替换算法的控制位，则数据 Cache 的总容量是多少？
- 2) 数组元素 a[0][31]和 a[1][1]各自所在的主存块对应的 Cache 行号分别是多少？
- 3) 程序 A 和 B 的数据访问命中率各是多少？哪个程序的执行时间更短？

答：

1) 数据 Cache 有 8 行(block)，每行 64B。Cache 中每块的 Tag 位数为：28-9=19 位。
则数据 Cache 的总容量为：8*64+(19+1)/8*8=532B

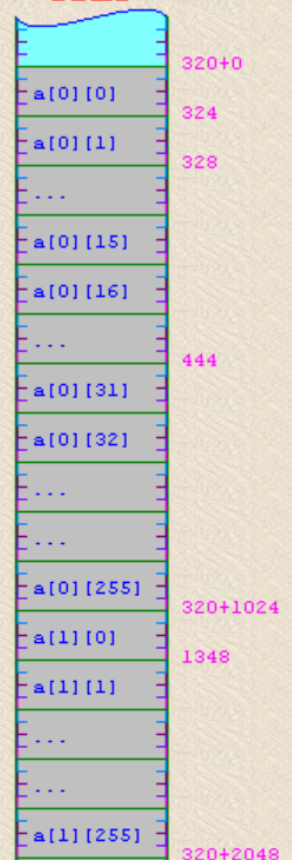
2) 数组 a 首地址为 320，所以：a[0][31]，a[1][1]在内存的地址分别为：
a[0][31]：320+0*256*4+31*4 = 444，对应的 Cache 行为：(444/64)%8 = 6
a[1][1]：320+1*256*4+1*4 = 1348，对应的 Cache 行为：(1348/64)%8 = 5

3) 程序 A 按行优先累加，命中率：15/16=93.75%

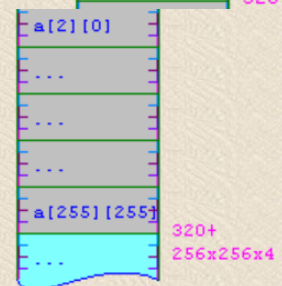
程序 B 按列优先计算：命中率：0。

注：Cache 容量不足以存放数组 a 的 16 行数据。

RAM



0,0	0,1	0,2	...	0,15	0,16	0,17	...	0,31	0,32	...	0,255
1,0	1,1	1,2	...	1,15	1,16	1,17	...	1,31	1,32	...	1,255
...
15,0	15,1	15,2	...	15,15	15,16	15,17	...	15,31	15,32	...	15,255
16,0	16,1	16,2	...	16,15	16,16	16,17	...	16,31	16,32	...	16,255
...
255,0	255,1	255,2	...	255,15	255,16	255,17	...	255,31	255,32	...	255,255



$T = T_c + (1-H)T_m$ (cache那边有没有命中算时间)

虚拟存储与Cache技术的比较

$PageTable = VM / PageSize * (1 + \log(PM / PageSize) + n) / 8 \rightarrow VM / PageSize * 4B$

Overhead of Polling in an I/O System

Overhead of Interrupt-Driven I/O

注意比大小：

计算题：一个tag,一个根据地址求TAG+INDE+OFFSET，pageTable，磁盘时间，一个overhead

总线同步异步总线的分类，总线的特点，总线的分类，总线的性能（存储器总线，IO总线，系统总线）

三种方式

流水线

冒险和解决方案（结构资源分支）

(structural control data)

如何加强。

控制Hazard

阻塞，猜，换。