

# Advanced Computer Architecture

高级计算机系统结构（复习题）

-考试时间：1月20号下午

根据曹强老师的写的，不知道会不会有错误，祝大家好成绩——by刘梦博

# Amdahl's Law

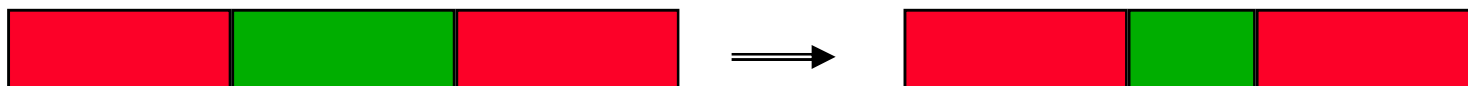
---

$$\text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} \times \left[ (1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right]$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Best you could ever hope to do:

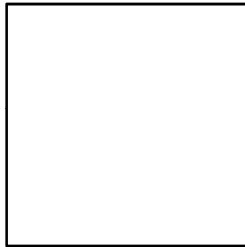
$$\text{Speedup}_{\text{maximum}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}})}$$



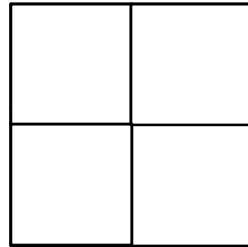
## 例题选讲 (1)

- 考虑下面3个处理器 (X,Y和Z)，都在一定硅面积 (16A) 上制造。假设单个线程的性能随着其使用面积的平方根增长。在上述3种处理器运行某一程序，这个程序串行的比例是S，而 (1-S) 为完全可并行的，完全使用Z中一个小核完成该程序的时间为T。

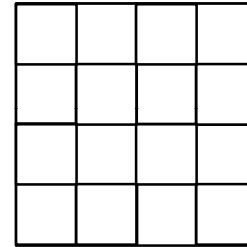
**Processor X**  
1 large core of area 16A



**Processor Y**  
4 medium cores of area 4A



**Processor Z**  
16 small cores of area A



- 分别计算3个处理器完成程序的时间。
- (1) X: 面积是Z小核的16倍，则程序执行时间为1/4;
  - (2) Y: 单个核面积是Z小核的4倍，则串行部分需要 $S \cdot T/2$ ，4个核并行部分执行时间为  $((1-S) \cdot T/2)/4$ ，总时间为 $3S \cdot T/8 + T/8$
  - (3) Z:  $S \cdot T + (1-S) \cdot T/16 = 15S \cdot T/16 + T/16$

## 例题选讲 (2)

- 某处理器能够使用**DVFS**技术来降低处理器的能耗，如果电压降低**15%**，则相应的频率也下降**15%**。请问使用**DVFS**之后，对于动态能耗和动态功率的影响？

解 **Energy<sub>dynamic</sub> = Capacitive load × Voltage<sup>2</sup>**

**Power<sub>dynamic</sub> = 1/2×capacitive load×Voltage<sup>2</sup>×Frequency switched**

- 处理器的晶体管数量不变，则动态能耗之比为：

$$\frac{Energy_{new}}{Energy_{old}} = \frac{(Voltage \times 0.85)^2}{Voltage^2} = 0.85^2 = 0.72$$

- 动态功率为：

$$\frac{Power_{new}}{Power_{old}} = 0.72 \times \frac{Frequency\ switched \times 0.85}{Frequency\ switched} = 0.61$$

- 动态能量为原来的**72%**，动态功率为原来的**61%**

## 例题选讲（3）

| Component type | Product                | Performance | Power                       |
|----------------|------------------------|-------------|-----------------------------|
| Processor      | Sun Niagara 8-core     | 1.2GHz      | 72-79 W peak                |
|                | Intel Pentium 4        | 2GHz        | 48.9-66 W                   |
| DRAM           | Kingston X64C3AD2 1 GB | 184-pin     | 3.7 W                       |
|                | Kingston D2N3 1 GB     | 240-pin     | 2.3 W                       |
| Hard driver    | DiamondMax 16          | 5400rpm     | 7.0 W read/seek. 2.9 W idle |
|                | DiamondMax 9           | 7200rpm     | 7.9 W read/seek. 4.0 W idle |

- 假设每个部件处于最大负载，电源功率效率为**80%**，**2GB 240针**内存，**7200RPM**硬盘。计算**Intel P4**服务器实际功率是多少？
- 使用该服务器使用**7200RPM（DM9）**硬盘，硬盘**60%**的时间空闲，磁盘的实际功率是多少？
- 如果**DM9**存取数据的时间为**DM16**的**75%**，如果两个盘的能耗相同，则**DM9**的空闲时间比例是多少？

解:

---

➤ (1)  $0.8x=66+2*2.3+7.9$  则  $x=0.99W$

➤ (2)  $0.6*4+0.4*7.9=5.56W$

➤ (3)  $W_{DM9}=0.75*W_{DM16}$

$$W_{DM9}=1 - I_{DM9}$$

$$W_{DM16}=1 - I_{DM16}$$

➤  $W_{DM9}*7.9+I_{DM9}*4=$

$$W_{DM16}*7+I_{DM16}*2.9$$

➤ So:  $I_{DM9}=29.8\%$

## CPU的性能公式

- **CPI**是衡量**CPU**执行指令效率的重要指标。让我们先考虑一个标准测速程序的全部执行时间 **$T_e$** 和其中所有第*i*种指令的累计时间 **$T_i$** ，易知

$$T_e = IC \times CPI \times CYCLE, \quad T_i = IC_i \times CPI_i \times CYCLE$$

- 其中： $CYCLE = \frac{1}{f}$ ， $IC = \sum_{i=1}^n IC_i$

- 另一方面，我们又可以写

$$T_e = \sum_{i=1}^n T_i = \sum_{i=1}^n (IC_i \times CPI_i \times CYCLE) = \left[ \sum_{i=1}^n IC_i \times CPI_i \right] \times CYCLE$$

- 比较上面第一式与最后一式，可以得到**CPI**与 **$CPI_i$** 的关系

$$IC \times CPI = \sum_{i=1}^n (IC_i \times CPI_i)$$

- 或者写为  $CPI = \sum_{i=1}^n \left( \frac{IC_i}{IC} \times CPI_i \right)$ ，它表明**CPI**为所有 **$CPI_i$** 的加权平均值

$$CPU \text{ time} = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

## 例题选讲（4）

- **A**计算机指令系统中含有一条特殊的多媒体处理指令，如果不使用这条指令，**A**计算机的 **MIPS** 为**400**，但执行该指令时，其执行时间是其他指令的 **4** 倍；**B**计算机 **MIPS** 为**600**，没有这条指令，但可以用其他**10**条指令构成子程序来代替这条指令。有**10000**行某段程序在**A**计算机上顺序执行时，这条多媒体指令出现的比例是**20%**。假设两台计算机除那条多媒体指令外，所有指令和指令的执行周期数都相同。问：哪台计算机先完成同样的计算任务？（需要通过计算结果比较）。
- 答案：**A**计算机的指令条数为**10000**条，**8000**条以**MIPS**为**400**的速度执行，**2000**条以**MIPS**为**100**的速度执行。**B**计算机的指令条数为**8000+2000\*10=28000**条，以**MIPS**为**600**的速度执行。

$$\text{执行时间} = \frac{\text{指令数}}{\text{MIPS} * 10^6}$$

- **A**计算机执行时间= $8000 / (400 * 10^6) + 2000 / (100 * 10^6)$   
= $20\mu\text{s} + 20\mu\text{s} = 40\mu\text{s}$
- **B**计算机执行时间= $28000 / (600 * 10^6) = 280 / 6 = 46.6\mu\text{s}$
- 所以，**A**计算机先完成



## 例题选讲 (5)

➤ **CPU性能问题:**

➤ **CPU的动态功率为:**

$$P_{dyn} \approx \sum_{i \in units} k_i C_i V^2 A_{if}$$

➤ 某**8核MCPUx**能够动态启动和关闭工作的核，假设某一程序**80%**部分可以并行执行，不考虑**L1, L2 Cache**对于计算过程和功耗的影响，**CPI**在两种情况下都不变；在单核运行时，程序执行的时间为**T0**，**P0**为运行功率。下面有两个调度方案，请分别计算该程序运行在两种情况下的实际功耗。

(1) 只使用一个核，其它核关闭，**CPU**电压不变，电压增加**10%**，主频增加**25%**；

(2) 使用全部**8**个核，**CPU**电压增加**10%**，主频不变。

## 解答

---

- **T0**为程序原来执行的时间，**P0**为单核执行不改变时的功率，其执行程序总能耗为 **$W_0 = T_0 * P_0$**
- (1)  **$T_1 = T_0 / (1+0.25) = 0.8 * T_0$**   
 **$P_1 = (1+10\%)^2 * (1+25\%) * P_0$**   
 **$= 1.5125 * P_0$**   
 **$W_1 = T_1 * P_1 = 1.21 W_0$**
- (2)  **$T_2 = (20\% + 80\% / 8) * T_0 = 30\% T_0$**   
 **$P_2 = (1+10\%)^2 * 8 * P_0$**   
 **$= 9.68 * P_0$**   
 **$W_2 = T_2 * P_2 = 2.904 W_0$**

## 副本可靠性模型 (6)

- **S**: 站点可用性 e.g., 99%
- **L**: 连接可用性 e.g., 95%

$$A = SL + (1 - SL)SL$$

Primary site  
can be reached

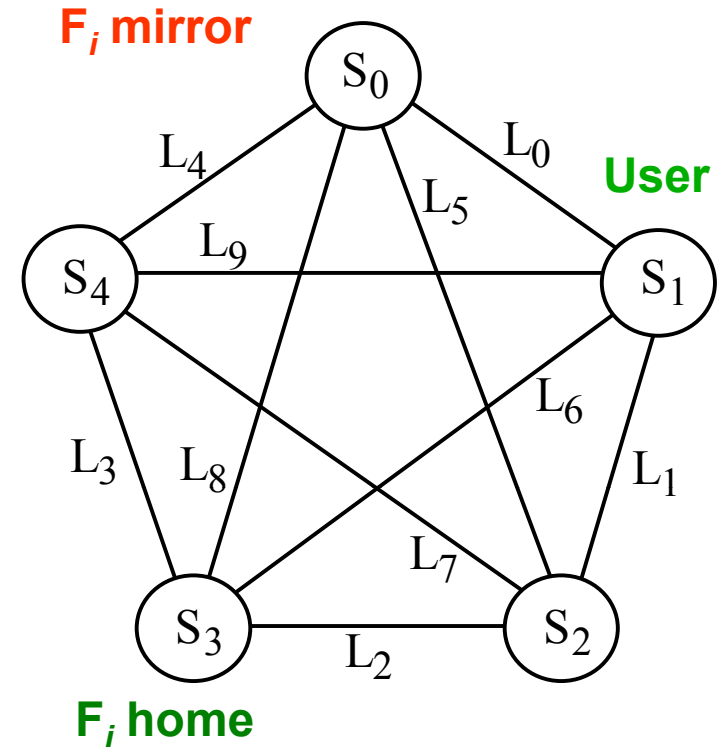
Mirror site  
can be reached

Primary site  
inaccessible

$$\begin{aligned} \text{Duplicated availability} &= 2SL - (SL)^2 \\ \text{Unavailability} &= 1 - 2SL + (SL)^2 \\ &= (1 - SL)^2 = 0.35\% \end{aligned}$$

Data unavailability reduced from 5.95% to 0.35%

Availability improved from  $\approx 94\%$  to 99.65%



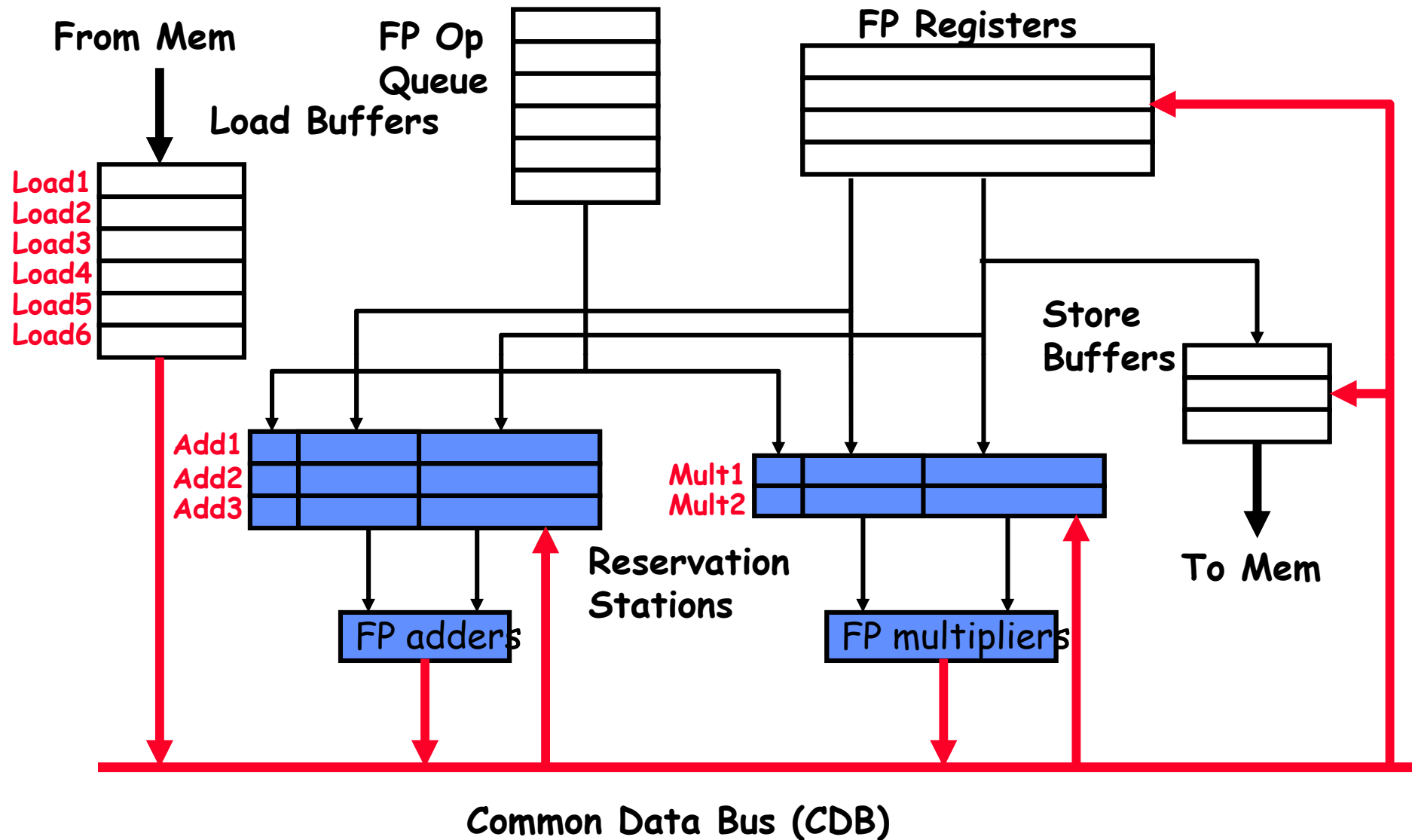
## 例题选讲 (7)

- 在下面**Tomasulo**结构上执行指令，分别计算每条指令写结果阶段的时间（时钟周期数），并画出相应的时空图。

（注：写不下了图在下页）

|       |     |     |    | Latency |
|-------|-----|-----|----|---------|
| LD    | F6  | 34+ | R2 | 1       |
| LD    | F2  | 45+ | R3 | 1       |
| MULTD | F0  | F2  | F4 | 10      |
| SUBD  | F8  | F6  | F2 | 2       |
| DIVD  | F10 | F0  | F6 | 40      |
| ADDD  | F6  | F8  | F2 | 2       |

# Tomasulo Organization



# Tomasulo Example

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy  | Address |
|-------------|----------|----------|-------------------|-------------|---------------|-------|---------|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |       |         |
| LD          | F6       | 34+      | R2                |             |               | Load1 | No      |
| LD          | F2       | 45+      | R3                |             |               | Load2 | No      |
| MULTD       | F0       | F2       | F4                |             |               | Load3 | No      |
| SUBD        | F8       | F6       | F2                |             |               |       |         |
| DIVD        | F10      | F0       | F6                |             |               |       |         |
| ADDD        | F6       | F8       | F2                |             |               |       |         |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| Add1        | No          |             |           |           |           |           |           |
| Add2        | No          |             |           |           |           |           |           |
| Add3        | No          |             |           |           |           |           |           |
| Mult1       | No          |             |           |           |           |           |           |
| Mult2       | No          |             |           |           |           |           |           |

*Register result status:*

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 0     | <i>FU</i> |           |           |           |           |            |            |     |            |

# Tomasulo Example Cycle 1

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | Issue | Exec | Write | Comp | Result | Busy | Address |
|-------------|----------|----------|-------|------|-------|------|--------|------|---------|
| LD          | F6       | 34+      | R2    | 1    |       |      |        | Yes  | 34+R2   |
| LD          | F2       | 45+      | R3    |      |       |      |        | No   |         |
| MULTD       | F0       | F2       | F4    |      |       |      |        | No   |         |
| SUBD        | F8       | F6       | F2    |      |       |      |        | No   |         |
| DIVD        | F10      | F0       | F6    |      |       |      |        | No   |         |
| ADDD        | F6       | F8       | F2    |      |       |      |        | No   |         |

## Reservation Stations:

| Time  | Name | Busy | Op | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------|------|------|----|-----------|-----------|-----------|-----------|
|       |      |      |    | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| Add1  |      | No   |    |           |           |           |           |
| Add2  |      | No   |    |           |           |           |           |
| Add3  |      | No   |    |           |           |           |           |
| Mult1 |      | No   |    |           |           |           |           |
| Mult2 |      | No   |    |           |           |           |           |

## Register result status:

| Clock | F0 | F2 | F4 | F6    | F8 | F10 | F12 | ... | F30 |
|-------|----|----|----|-------|----|-----|-----|-----|-----|
| 1     |    |    |    | Load1 |    |     |     |     |     |

# Tomasulo Example Cycle 2

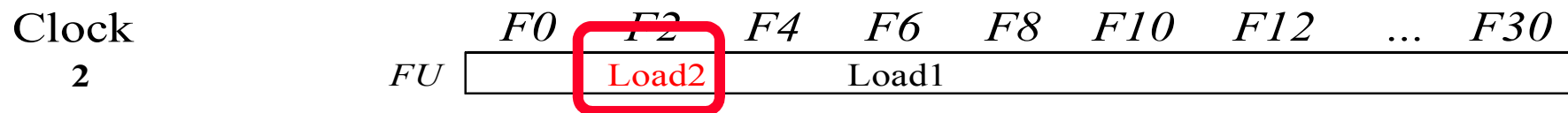
*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> |              |                    | Busy | Address |
|-------------|----------|----------|--------------|--------------------|------|---------|
|             |          |          | <i>Issue</i> | <i>Comp Result</i> |      |         |
| LD          | F6       | 34+      | R2           | 1                  | Yes  | 34+R2   |
| LD          | F2       | 45+      | R3           | 2                  | Yes  | 45+R3   |
| MULTD       | F0       | F2       | F4           |                    | No   |         |
| SUBD        | F8       | F6       | F2           |                    |      |         |
| DIVD        | F10      | F0       | F6           |                    |      |         |
| ADDD        | F6       | F8       | F2           |                    |      |         |

*Reservation Stations:*

| Time  | Name | Busy | Op | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------|------|------|----|-----------|-----------|-----------|-----------|
|       |      |      |    | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| Add1  |      | No   |    |           |           |           |           |
| Add2  |      | No   |    |           |           |           |           |
| Add3  |      | No   |    |           |           |           |           |
| Mult1 |      | No   |    |           |           |           |           |
| Mult2 |      | No   |    |           |           |           |           |

*Register result status:*



Note: Unlike 6600, can have multiple loads outstanding



# Tomasulo Example Cycle 3

*Instruction status:*

| Instruction | j   | k   | Exec  |      | Result | Busy  | Address   |
|-------------|-----|-----|-------|------|--------|-------|-----------|
|             |     |     | Issue | Comp |        |       |           |
| LD          | F6  | 34+ | R2    | 1    | 3      | Load1 | Yes 34+R2 |
| LD          | F2  | 45+ | R3    | 2    |        | Load2 | Yes 45+R3 |
| MULTD       | F0  | F2  | F4    | 3    |        | Load3 | No        |
| SUBD        | F8  | F6  | F2    |      |        |       |           |
| DIVD        | F10 | F0  | F6    |      |        |       |           |
| ADDD        | F6  | F8  | F2    |      |        |       |           |

*Reservation Stations:*

| Time  | Name | Busy | Op    | S1 | S2    | RS    | RS |
|-------|------|------|-------|----|-------|-------|----|
|       |      |      |       | Vj | Vk    | Qj    | Qk |
| Add1  |      | No   |       |    |       |       |    |
| Add2  |      | No   |       |    |       |       |    |
| Add3  |      | No   |       |    |       |       |    |
| Mult1 |      | Yes  | MULTD |    | R(F4) | Load2 |    |
| Mult2 |      | No   |       |    |       |       |    |

*Register result status:*

| Clock | F0    | F2    | F4 | F6    | F8 | F10 | F12 | ... | F30 |
|-------|-------|-------|----|-------|----|-----|-----|-----|-----|
| 3     | Mult1 | Load2 |    | Load1 |    |     |     |     |     |

- Note: registers names are removed ("renamed") in Reservation Stations; MULT issued vs. scoreboard
- Load1 completing; what is waiting for Load1?

# Tomasulo Example Cycle 4

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |           |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|-----------|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |           |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No        |
| LD          | F2       | 45+      | R3                | 2           | 4             |      | Load2   | Yes 45+R3 |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No        |
| SUBD        | F8       | F6       | F2                | 4           |               |      |         |           |
| DIVD        | F10      | F0       | F6                |             |               |      |         |           |
| ADDD        | F6       | F8       | F2                |             |               |      |         |           |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| Add1        | Yes         | SUBD        | M(A1)     |           |           |           | Load2     |
| Add2        | No          |             |           |           |           |           |           |
| Add3        | No          |             |           |           |           |           |           |
| Mult1       | Yes         | MULTD       |           | R(F4)     |           | Load2     |           |
| Mult2       | No          |             |           |           |           |           |           |

*Register result status:*

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 4     | Mult1     | Load2     |           | M(A1)     | Add1      |            |            |     |            |

- Load2 completing; what is waiting for Load2?

# Tomasulo Example Cycle 5

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           |               |      |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                |             |               |      |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| 2           | Add1        | Yes         | SUBD      | M(A1)     | M(A2)     |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 10          | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 5     | Mult1     | M(A2)     |           | M(A1)     | Add1      | Mult2      |            |     |            |

# Tomasulo Example Cycle 6

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Issue</i> | <i>Exec Comp</i> | <i>Write Result</i> | Busy | Address |    |
|-------------|----------|----------|--------------|------------------|---------------------|------|---------|----|
| LD          | F6       | 34+      | R2           | 1                | 3                   | 4    | Load1   | No |
| LD          | F2       | 45+      | R3           | 2                | 4                   | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4           | 3                |                     |      | Load3   | No |
| SUBD        | F8       | F6       | F2           | 4                |                     |      |         |    |
| DIVD        | F10      | F0       | F6           | 5                |                     |      |         |    |
| ADDD        | F6       | F8       | F2           | 6                |                     |      |         |    |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1 Vj</i> | <i>S2 Vk</i> | <i>RS Qj</i> | <i>RS Qk</i> |
|-------------|-------------|-------------|-----------|--------------|--------------|--------------|--------------|
| 1           | Add1        | Yes         | SUBD      | M(A1)        | M(A2)        |              |              |
|             | Add2        | Yes         | ADDD      |              | M(A2)        | Add1         |              |
|             | Add3        | No          |           |              |              |              |              |
| 9           | Mult1       | Yes         | MULTD     | M(A2)        | R(F4)        |              |              |
|             | Mult2       | Yes         | DIVD      |              | M(A1)        | Mult1        |              |

*Register result status:*

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 6     | FU        |           |           |           |           |            |            |     |            |
|       | Mult1     | M(A2)     |           | Add2      | Add1      | Mult2      |            |     |            |

- Issue ADDD here vs. scoreboard?

# Tomasulo Example Cycle 7

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Issue</i> | <i>Exec Comp</i> | <i>Write Result</i> | Busy | Address |    |
|-------------|----------|----------|--------------|------------------|---------------------|------|---------|----|
| LD          | F6       | 34+      | R2           | 1                | 3                   | 4    | Load1   | No |
| LD          | F2       | 45+      | R3           | 2                | 4                   | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4           | 3                |                     |      | Load3   | No |
| SUBD        | F8       | F6       | F2           | 4                | 7                   |      |         |    |
| DIVD        | F10      | F0       | F6           | 5                |                     |      |         |    |
| ADDD        | F6       | F8       | F2           | 6                |                     |      |         |    |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1 Vj</i> | <i>S2 Vk</i> | <i>RS Qj</i> | <i>RS Qk</i> |
|-------------|-------------|-------------|-----------|--------------|--------------|--------------|--------------|
| 0           | Add1        | Yes         | SUBD      | M(A1)        | M(A2)        |              |              |
|             | Add2        | Yes         | ADDD      |              | M(A2)        | Add1         |              |
|             | Add3        | No          |           |              |              |              |              |
| 8           | Mult1       | Yes         | MULTD     | M(A2)        | R(F4)        |              |              |
|             | Mult2       | Yes         | DIVD      |              | M(A1)        | Mult1        |              |

*Register result status:*

| Clock | F0    | F2    | F4 | F6   | F8   | F10   | F12 | ... | F30 |
|-------|-------|-------|----|------|------|-------|-----|-----|-----|
| 7     | Mult1 | M(A2) |    | Add2 | Add1 | Mult2 |     |     |     |

- Add1 completing; what is waiting for it?

# Tomasulo Example Cycle 8

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           |               |      |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
| 2           | Add2        | Yes         | ADDD      | M(A2)     | M(A2)     |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 7           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 8     | FU        |           |           |           |           |            |            |     |            |
|       | Mult1     | M(A2)     |           | Add2      | (M-M)     | Mult2      |            |     |            |

# Tomasulo Example Cycle 9

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           |               |      |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
| 1           | Add2        | Yes         | ADDD      | (M-M)     | M(A2)     |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 6           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 9     | FU        | Mult1     | M(A2)     |           | Add2      | (M-M)      | Mult2      |     |            |

# Tomasulo Example Cycle 10

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            |      |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
| 0           | Add2        | Yes         | ADDD      | M(A2)     | M(A2)     |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 5           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 10    |           |           |           |           |           |            |            |     |            |
|       | <i>FU</i> | Mult1     | M(A2)     |           | Add2      | (M-M)      | Mult2      |     |            |

- Add2 completing; what is waiting for it?



# Tomasulo Example Cycle 11

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 4           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

*Register result status:*

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 11    | FU        |           |           |           |           |            |            |     |            |
|       | Mult1     | M(A2)     |           | (M-M+M    | (M-M)     | Mult2      |            |     |            |

- Write result of ADDD here vs. scoreboard?
- All quick instructions complete in this cycle!

# Tomasulo Example Cycle 12

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 3           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 12    | Mult1     | M(A2)     |           | (M-M+M)   | (M-M)     | Mult2      |            |     |            |

# Tomasulo Example Cycle 13

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 2           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 13    | <i>FU</i> |           |           |           |           |            |            |     |            |
|       | Mult1     | M(A2)     |           | (M-M+M)   | (M-M)     | Mult2      |            |     |            |

# Tomasulo Example Cycle 14

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           |               |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 1           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

## Register result status:

| Clock | F0    | F2    | F4 | F6      | F8    | F10   | F12 | ... | F30 |
|-------|-------|-------|----|---------|-------|-------|-----|-----|-----|
| 14    | Mult1 | M(A2) |    | (M-M+M) | (M-M) | Mult2 |     |     |     |

# Tomasulo Example Cycle 15

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           | 15            |      | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
| 0           | Mult1       | Yes         | MULTD     | M(A2)     | R(F4)     |           |           |
|             | Mult2       | Yes         | DIVD      |           | M(A1)     | Mult1     |           |

*Register result status:*

| Clock | F0    | F2    | F4 | F6      | F8    | F10   | F12 | ... | F30 |
|-------|-------|-------|----|---------|-------|-------|-----|-----|-----|
| 15    | Mult1 | M(A2) |    | (M-M+M) | (M-M) | Mult2 |     |     |     |

# Tomasulo Example Cycle 16

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           | 15            | 16   | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
|             | Mult1       | No          |           |           |           |           |           |
| 40          | Mult2       | Yes         | DIVD      | M*F4      | M(A1)     |           |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 16    | <i>FU</i> | M*F4      | M(A2)     |           | (M-M+M)   | (M-M)      | Mult2      |     |            |

# Tomasulo Example Cycle 55

## Instruction status:

| Instruction | <i>j</i> | <i>k</i> | <i>Exec Write</i> |             |               | Busy | Address |    |
|-------------|----------|----------|-------------------|-------------|---------------|------|---------|----|
|             |          |          | <i>Issue</i>      | <i>Comp</i> | <i>Result</i> |      |         |    |
| LD          | F6       | 34+      | R2                | 1           | 3             | 4    | Load1   | No |
| LD          | F2       | 45+      | R3                | 2           | 4             | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4                | 3           | 15            | 16   | Load3   | No |
| SUBD        | F8       | F6       | F2                | 4           | 7             | 8    |         |    |
| DIVD        | F10      | F0       | F6                | 5           |               |      |         |    |
| ADDD        | F6       | F8       | F2                | 6           | 10            | 11   |         |    |

## Reservation Stations:

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
|             | Add1        | No          |           |           |           |           |           |
|             | Add2        | No          |           |           |           |           |           |
|             | Add3        | No          |           |           |           |           |           |
|             | Mult1       | No          |           |           |           |           |           |
| 1           | Mult2       | Yes         | DIVD      | M*F4      | M(A1)     |           |           |

## Register result status:

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 55    | FU        | M*F4      | M(A2)     |           | (M-M+M    | (M-M)      | Mult2      |     |            |

# Tomasulo Example Cycle 56

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Issue</i> | <i>Exec Comp</i> | <i>Write Result</i> | Busy | Address |    |
|-------------|----------|----------|--------------|------------------|---------------------|------|---------|----|
| LD          | F6       | 34+      | R2           | 1                | 3                   | 4    | Load1   | No |
| LD          | F2       | 45+      | R3           | 2                | 4                   | 5    | Load2   | No |
| MULTD       | F0       | F2       | F4           | 3                | 15                  | 16   | Load3   | No |
| SUBD        | F8       | F6       | F2           | 4                | 7                   | 8    |         |    |
| DIVD        | F10      | F0       | F6           | 5                | 56                  |      |         |    |
| ADDD        | F6       | F8       | F2           | 6                | 10                  | 11   |         |    |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1 Vj</i> | <i>S2 Vk</i> | <i>RS Qj</i> | <i>RS Qk</i> |
|-------------|-------------|-------------|-----------|--------------|--------------|--------------|--------------|
|             | Add1        | No          |           |              |              |              |              |
|             | Add2        | No          |           |              |              |              |              |
|             | Add3        | No          |           |              |              |              |              |
|             | Mult1       | No          |           |              |              |              |              |
| 0           | Mult2       | Yes         | DIVD      | M*F4         | M(A1)        |              |              |

*Register result status:*

| Clock | F0 | F2   | F4    | F6 | F8      | F10   | F12   | ... | F30 |
|-------|----|------|-------|----|---------|-------|-------|-----|-----|
| 56    | FU | M*F4 | M(A2) |    | (M-M+M) | (M-M) | Mult2 |     |     |

- Mult2 is completing; what is waiting for it?



# Tomasulo Example Cycle 57

*Instruction status:*

| Instruction | <i>j</i> | <i>k</i> | <i>Issue</i> | <i>Exec</i> | <i>Write</i>  | <i>Busy</i> | <i>Address</i> |
|-------------|----------|----------|--------------|-------------|---------------|-------------|----------------|
|             |          |          |              | <i>Comp</i> | <i>Result</i> |             |                |
| LD          | F6       | 34+      | R2           | 1           | 3             | 4           | Load1          |
| LD          | F2       | 45+      | R3           | 2           | 4             | 5           | Load2          |
| MULTD       | F0       | F2       | F4           | 3           | 15            | 16          | Load3          |
| SUBD        | F8       | F6       | F2           | 4           | 7             | 8           |                |
| DIVD        | F10      | F0       | F6           | 5           | 56            | 57          |                |
| ADDD        | F6       | F8       | F2           | 6           | 10            | 11          |                |

*Reservation Stations:*

| <i>Time</i> | <i>Name</i> | <i>Busy</i> | <i>Op</i> | <i>S1</i> | <i>S2</i> | <i>RS</i> | <i>RS</i> |
|-------------|-------------|-------------|-----------|-----------|-----------|-----------|-----------|
|             |             |             |           | <i>Vj</i> | <i>Vk</i> | <i>Qj</i> | <i>Qk</i> |
| Add1        |             | No          |           |           |           |           |           |
| Add2        |             | No          |           |           |           |           |           |
| Add3        |             | No          |           |           |           |           |           |
| Mult1       |             | No          |           |           |           |           |           |
| Mult2       |             | Yes         | DIVD      | M*F4      | M(A1)     |           |           |

*Register result status:*

| Clock | <i>F0</i> | <i>F2</i> | <i>F4</i> | <i>F6</i> | <i>F8</i> | <i>F10</i> | <i>F12</i> | ... | <i>F30</i> |
|-------|-----------|-----------|-----------|-----------|-----------|------------|------------|-----|------------|
| 56    | <i>FU</i> | M*F4      | M(A2)     |           | (M-M+M    | (M-M)      | Result     |     |            |

- Once again: In-order issue, out-of-order execution and completion.



## 例题选讲 (8)

```

Loop: LD    F0,0(R1)    ;F0=vector element
      ADDD  F4,F0,F2    ;add scalar from F2
      SD    0(R1),F4    ;store result
      SUBI  R1,R1,8     ;decrement pointer 8B (DW)
      BNEZ  R1,Loop     ;branch R1!=zero
      NOP                ;delayed branch slot
  
```

| <i>Instruction<br/>producing result</i> | <i>Instruction<br/>using result</i> | <i>Execution Latency in<br/>clock cycles</i> | <i>Use Latency in<br/>clock cycles</i> |
|---|-------------------------------------|--|--|
| FP ALU op                               | Another FP ALU op                   | 4  | 3                                      |
| FP ALU op                               | Store double                        | 4  | 2                                      |
| Load double                             | FP ALU op                           | 2  | 1                                      |
| Load double                             | Store double                        | 2  | 0                                      |
| Integer op                              | Integer op                          | 1  | 0                                      |

- Where are the stalls?

# FP Loop Showing Stalls

---

```

1 Loop: LD      F0, 0(R1)    ;F0=vector element
2          stall
3          ADDD   F4, F0, F2  ;add scalar in F2
4          stall
5          stall
6          SD     0(R1), F4   ;store result
7          SUBI   R1, R1, 8   ;decrement pointer 8B (DW)
8          BNEZ   R1, Loop   ;branch R1!=zero
9          stall             ;delayed branch slot

```

| <i>Instruction<br/>producing result</i> | <i>Instruction<br/>using result</i> | <i>Use Latency in<br/>clock cycles</i> |
|---|-------------------------------------|--|
| FP ALU op                               | Another FP ALU op                   | 3                                      |
| FP ALU op                               | Store double                        | 2                                      |
| Load double                             | FP ALU op                           | 1                                      |

- **9 clocks: Rewrite code to minimize stalls?**

# Revised FP Loop Minimizing Stalls

---

```
1 Loop: LD    F0, 0(R1)
2         stall
3         ADDD F4, F0, F2
4         SUBI R1, R1, 8
5         BNEZ R1, Loop    ;delayed branch
6         SD    8(R1), F4  ;altered when move past SUBI
```

Swap BNEZ and SD by changing address of SD

| <i>Instruction<br/>producing result</i> | <i>Instruction<br/>using result</i> | <i>Use Latency in<br/>clock cycles</i> |
|---|-------------------------------------|--|
| FP ALU op                               | Another FP ALU op                   | 3                                      |
| FP ALU op                               | Store double                        | 2                                      |
| Load double                             | FP ALU op                           | 1                                      |

**6 clocks: Unroll loop 4 times code to make faster?**

## 例题选讲 (8)

➤ 考察如下循环程序

```
for (i=0; i<100; i=i+1) {  
    A[i] = A[i] + B[i];      /* S1 */  
    B[i+1] = C[i] + D[i+1]; /* S2 */  
}
```

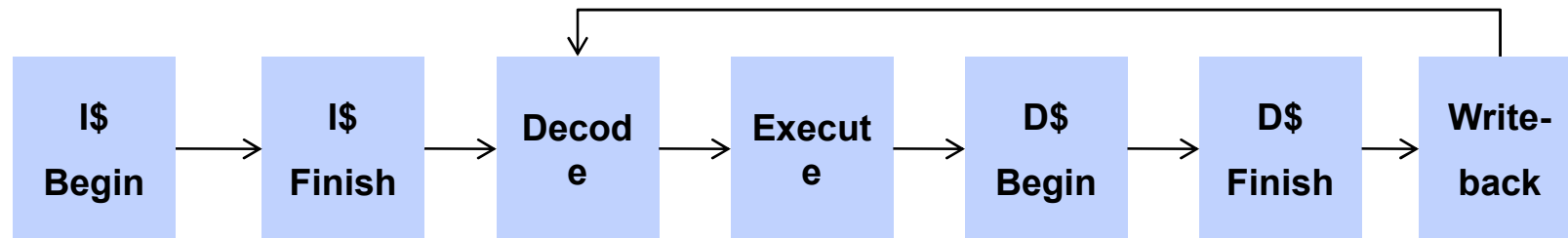
➤ **S1**与**S2**之间存在何种相关关系？这个循环程序是否可以并行，若可以，如何使其并行？

➤ 解：**S1**中使用的值是上次循环**S2**计算的结果，即**S2**和**S1**之间存在循环体间相关；但**S2**并不相关于**S1**。只要相关关系不形成环状，循环程序就可以并行。

1. 不存在**S1**到**S2**的相关路径，因此可以把**S1**和**S2**对调不影响**S2**的执行；
2. 在第一次循环中，**S1**中使用初始值**B[1]**，是在循环初始化之前就已经计算出来

- 
- **$A[1]=A[1]+B[1]$**
  - **for (i=1; i<=100; i=i+1) {**
    - $B[i+1] = C[i] + D[i+1];$**
    - $A[i+1] = A[i+1] + B[i+1];$**
  - **}**
  - **$B[101] = C[100] + D[100];$**
  - 现在这两条语句之间不再存在循环体间的相关关系，从而可以让不同循环体的语句重叠执行。

## 例题选讲 (9)



➤ 按照标准五段流水线设计7段顺序发射流水线，增加指令和数据 **cache** 存取阶段到2个时钟周期，如上图所示。并且流水线不具有任何分支预测机制，并且分支计算比较简单，在**Decode**段就能完成。请问：

- (1) 在此是流水线中分支目标最早在那个段获得？
- (2) 分支产生多少时钟的延迟？
- (3) 假设1 / 6的指令是分支指令，并且3 / 5分支成功，除分支之外的指令CPI为1，使用分支失败预测策略，请问此流水线实际CPI是多少？

答：(1) **Decode**

(2) **2**

(3)  **$1 + 1 / 6 * 3 / 5 * 2 = 1.2$**



## 例题选讲（10）

---

- 某处理器采用按照分支地址索引的二位历史预测器**PR**，处理器执行下述程序段，**R1**的初始值为**5**，代码段**Project1**中不会修改**R1**的值，预测器**PR**的初始值为**00**，表示预测分支失败。填写下表：
- **Project1: {**
- **.....**
- **}**
- **SUBI  R1, 1+, R1**
- **X1:       BNEQ Project1**

---

| 执行状态                 | 预测器PR值 | 预测   | 实际   | 预测是否成功 |
|----------------------|--------|------|------|--------|
| <b>Project1:R1=5</b> | 00     | 分支失败 | NULL | NULL   |
| <b>X1: R1=4</b>      | 00     | 分支失败 | 分支成功 | 否      |
| <b>X1:R1=3</b>       | 01     | 分支失败 | 分支成功 | 否      |
| <b>X1:R1=2</b>       | 10     | 分支成功 | 分支成功 | 是      |
| <b>X1:R1=1</b>       | 11     | 分支成功 | 分支成功 | 是      |
| <b>X1:R1=0</b>       | 11     | 分支成功 | 分支失败 | 否      |

## 分支预测 (11)

---

- 代码有四个分支 (B1, B2, B3和B4), 如果分支成功, 则对应大括号中的代码执行。
- ```
for ( int i=0; i< N; i++) { /*B1*/  
    val = i+2;           /*TAKEN PATH for B1*/  
    if (val % 2 == 0){   /*B2*/  
        sum += val;     /*TAKEN PATH for B2*/  
    }  
    if (val % 5 == 0){   /*B3*/  
        sum += val;     /*TAKEN PATH for B3*/  
    }  
    if (val % 10 == 0){ /*B4*/  
        sum += val;     /*TAKEN PATH for B4*/  
    }  
}
```
- }
- 上述代码执行在一个具有全局分支预测器的处理器。全局预测器具有以下特性:
- 1) 全局预测器 (GHR) 2位
- 2) 分支历史表 (PHT) 4项
- 3) 分支历史表项 (PHTE) 11位有符号饱和计数器 (-1024~1023)
- 4) 在代码运行之前, 所有PHTE初始值为0
- 5) 当代码运行, 如果相应分支成功, PHTE为增加1, 如果分支失败, 则PHTE减1

- 
- (1) 哪些分支是相关的
  - **B1**是本地相关, **B2**, **B3**, **B4**不是本地相关
  - **B4**全局关联与**B2**, **B3**关联。如果**B2**或者**B3**不发生, **B4**也不发生
  - (2) 当**N=2**时, 给出**GHR**的变化过程

|        |           |
|--------|-----------|
| 初始值    | <b>00</b> |
| B1 i=0 | 00        |
| B2 i=0 | 01        |
| B3 i=0 | 10        |
| B4 i=0 | 00        |
| B1 i=1 | 00        |
| B2 i=1 | 00        |
| B3 i=1 | 00        |
| B4 i=1 | 00        |

## BTB 习题12

---

- 有一个深度流水线化的处理器，仅为条件分支配备一个分支目标缓冲器，加速预测错误的损失为4个时钟周期，缓冲器缺失损失3个时钟周期，并假设命中率为90%，精度为90%，分支频率为15%。比较该处理器与一个固定2个时钟周期分支损失的加速比。假设无分支停顿的CPI为1。
- 答：
- 加速比= $\text{CPI (noBTB)} / \text{CPI (BTB)}$   
 $= [\text{CPI (base)} + \text{Stall (noBTB)}] / [\text{CPI (base)} + \text{Stall (BTB)}]$

## BTB 习题12

➤ **Stall (noBTB) = 15%\*2=0.3**

➤ **对于Stall (BTB) :**

| BTB结果 | BTB预测 | 出现频率             | 开销（时钟频率） |
|-------|-------|------------------|----------|
| 未命中   | 无     | 15%*10%=1.5%     | 3        |
| 命中    | 正确    | 15%*90%=12.1%    | 0        |
| 命中    | 错误    | 15%*90%*10%=1.3% | 4        |

➤ **Stall (BTB) = (1.5%\*3) + (12.1%\*0) + (1.3%\*4) = 0.097**

➤ **1+0.3 / (1 + 0.097) = 1.2**

## 例题选讲 (13)

---

- **IBM研究：超标量流水线限制？**
- **内存带宽**
  - ◆ **Fetch 1 instr / cycle from I-cache**
  - ◆ **40% of instructions are load / store (D-cache)**
- **Code characteristics (dynamic)**
  - **Loads – 25%**      从内存读取具有**2**个时钟周期的延迟
  - **Stores 15%**      可以通过**CPU**内存缓存，隐藏延迟
  - **ALU / RR – 40%**
  - **Branches – 20%**    分支具有**2**个时钟周期的延迟
    - **1 / 3 unconditional (always taken).**
    - **1 / 3 conditional taken.**
    - **1 / 3 conditional not taken.**

# 假设

---

## ➤ Cache Performance

- Assume 100% hit ratio (upper bound)
- Cache latency:  $I = D = 1$  cycle default

## ➤ Load and branch scheduling

### – Loads

- » 25% cannot be scheduled (delay slot empty)
- » 65% can be moved back 1 or 2 instructions
- » 10% can be moved back 1 instruction

### – Branches

- » Unconditional – 100% schedulable (fill one delay slot)
- » Conditional – 50% schedulable (fill one delay slot)



## CPI优化

---

### ➤ Goal and impediments

– CPI = 1, prevented by pipeline stalls

### ➤ No cache bypass of RF, no load/branch scheduling

– Load penalty: 2 cycles:  $0.25 \times 2 = 0.5$  CPI

– Branch penalty: 2 cycles:  $0.2 \times 2/3 \times 2 = 0.27$  CPI

– Total CPI:  $1 + 0.5 + 0.27 = 1.77$  CPI

### ➤ Bypass, no load/branch scheduling

– Load penalty: 1 cycle:  $0.25 \times 1 = 0.25$  CPI

– Total CPI:  $1 + 0.25 + 0.27 = 1.52$  CPI

## 更多CPI优化

---

### ➤ Bypass, scheduling of loads/branches

#### – Load penalty:

- »  $65\% + 10\% = 75\%$  moved back, no penalty
- »  $25\% \Rightarrow 1$  cycle penalty
- »  $0.25 \times 0.25 \times 1 = 0.0625$  CPI

#### – Branch Penalty

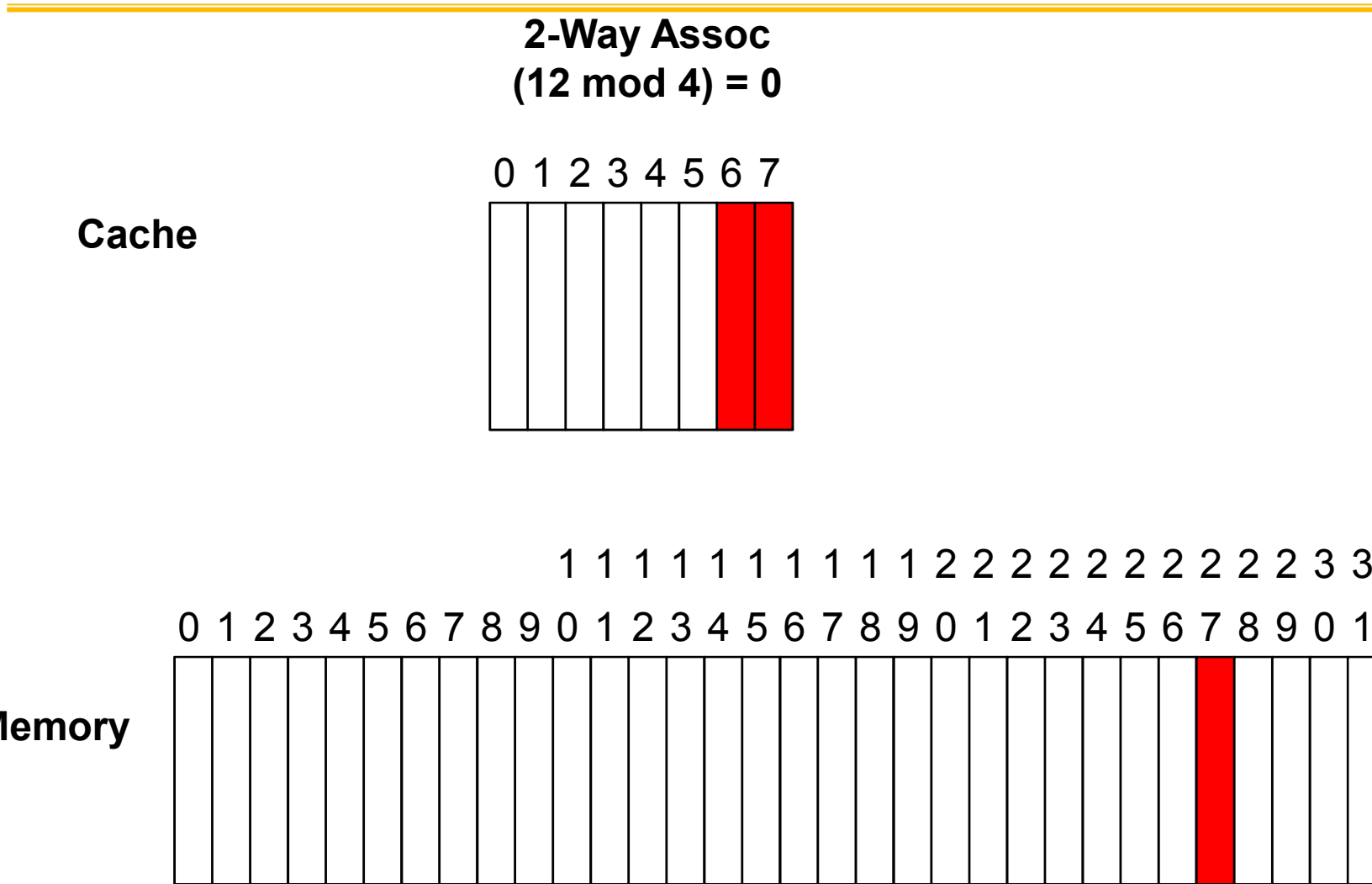
- »  $1/3$  unconditional 100% schedulable  $\Rightarrow 1$  cycle
- »  $1/3$  cond. not-taken,  $\Rightarrow$  no penalty (predict not-taken)
- »  $1/3$  cond. Taken, 50% schedulable  $\Rightarrow 1$  cycle
- »  $1/3$  cond. Taken, 50% unschedulable  $\Rightarrow 2$  cycles
- »  $0.20 \times [1/3 \times 1 + 1/3 \times 0.5 \times 1 + 1/3 \times 0.5 \times 2] = 0.167$

### ➤ Total CPI: $1 + 0.063 + 0.167 = 1.23$ CPI

## 例题选讲（14）

---

- 一个四路相联**cache**，CPU字长为4字节，内存和**cache**都是以字节编址，**cache**和内存交换单位为块，每个块大小为512字节，**cache**能够容纳1024个块。如果物理内存为32位地址：
  - （1）请画出**cache**和内存的地址格式，标明块内偏移地址、组号和标签位置；
  - （2）请计算内存地址**FAB12389**（16进制）在**cache**中可能的位置块号。



## 解答

➤ 解： (1) 

|             |          |             |
|-------------|----------|-------------|
| Tag标签 (15位) | 组地址 (8位) | 块内偏移地址 (9位) |
|-------------|----------|-------------|

➤ (2) 二进制地址

**1111 1010 1011 0001 0010 0011 1000 1001**

➤ Cache中包含数组为 $1024/4 = 256$ 组

➤ Cache内组地址：**1 ■ 0010 ■ 001**

➤ 也就是第**145**组，

➤ 可能在cache内中的块号为**580,581,582和583**

# Cache performance

---

## ➤ Miss-oriented Approach to Memory Access:

$$CPUtime = IC \times \left( CPI_{Execution} + \frac{MemAccess}{Inst} \times MissRate \times MissPenalty \right) \times CycleTime$$

## ➤ Separating out Memory component entirely

◆ AMAT=Average Memory Access Time

$$CPUtime = IC \times \left( CPI_{AluOps} + \frac{MemAccess}{Inst} \times AMAT \right) \times CycleTime$$

$$AMAT = HitTime + MissRate \times MissPenalty$$

$$= (HitTime_{Inst} + MissRate_{Inst} \times MissPenalty_{Inst}) + (HitTime_{Data} + MissRate_{Data} \times MissPenalty_{Data})$$

## 例题选讲（15）

---

- 给定以下假设，试计算直接映象**Cache**和两路组相联**Cache**的平均访问时间以及**CPU**的性能。
- 假设：
  - 1. 理想**Cache**情况下的**CPI**为**2.0**，时钟周期为**2ns**，平均每条指令访存**1.2**次；
  - 2. 两种**Cache**容量均为**64KB**，块大小为**32**个字节；
  - 3. 组相联**Cache**中的多路选择器使得**CPU**的时钟周期增加了**10%**；
  - 4. 这两种**Cache**的失效开销都是**80ns**；
  - 5. 命中时间为**1**个时钟周期；
  - 6. **64KB**直接映象**Cache**失效率为**1.4%**，**64KB**两路组相联**Cache**的失效率为**1.0%**。

## 解答

- 1. 因为 平均访存时间 = 命中时间 + 失效率 × 失效开销  
 平均访存时间直接 = 命中时间直接 + 失效率直接 × 失效开销直接  

$$= 1 \times 2 + 1.4\% \times 80 = 3.12\text{ns}$$
 平均访存时间两路 = 命中时间两路 + 失效率两路 × 失效开销两路  

$$= 1 \times 2(1 + 10\%) + 1.0\% \times 80 = 3.00\text{ns}$$
- 2. CPU的性能

$$\begin{aligned} \text{CPU时间} &= \text{IC} \times \left( \text{CPI}_{\text{execution}} + \frac{\text{失效次数}}{\text{指令数}} \times \text{失效开销} \right) \times \text{时钟周期} \\ &= \text{IC} \times \left[ (\text{CPI}_{\text{execution}} \times \text{时钟周期时间}) + \left( \frac{\text{访存次数}}{\text{指令数}} \times \text{失效率} \times \text{失效开销} \times \text{时钟周期} \right) \right] \end{aligned}$$

- CPU时间<sub>直接</sub> =  $\text{IC} \times ((2 \times 2) + 1.2 \times 1.4\% \times 80) = \text{IC} \times 5.344$
- 总CPU时间<sub>两路</sub> =  $\text{IC} \times ((2 \times 2.2) + 1.2 \times 1.0\% \times 80) = \text{IC} \times 2.36$

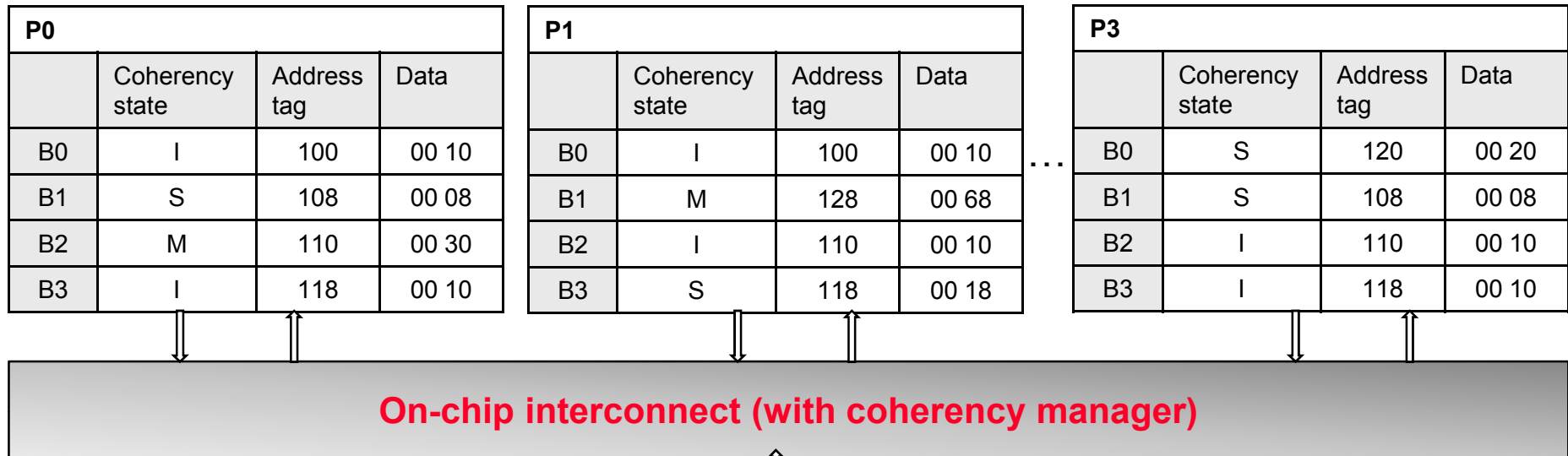


## 例题选讲 (16)

---

- **Suppose a processor executes at**
  - **Clock Rate=200MHz(5ns per cycle), Ideal (no misses) CPI=1.1**
  - **50% arith/logic,30% ld/st, 20% control**
- **Miss Behavior:**
  - **10% of memory operations get 50 cycle miss penalty**
  - **1% of instructions get same miss penalty**
- **CPI = ideal CPI + average stalls per instruction**  
 $1.1(\text{cycles/ins}) + [0.30(\text{DataMops/ins}) \times 0.10(\text{miss/DataMop}) \times 50(\text{cycle/miss})] + [1(\text{instMop/ins}) \times 0.01(\text{miss/instMop}) \times 50(\text{cycle/miss})]$   
 $= (1.1 + 1.5 + 0.5) \text{ cycle/ins} = 3.1$
- **AMAT=(1/1.3) × [1+0.01 ×50]+(0.3/1.3) ×[1+0.1 ×50] = 2.54**

# 例题选讲 (17)



- 本图描述一个典型的SMP多处理器结构，使用snooping协议。每个cache使用直接映射，包含4个块，每个块包含两个字，为了简化，cache的地址标签包含全地址，每个字显示两个16进制。

| Memory  |        |
|---------|--------|
| Address | Data   |
| ...     | ... .. |
| 100     | 00 10  |
| 108     | 00 08  |
| 110     | 00 10  |
| 118     | 00 18  |
| 120     | 00 20  |
| 128     | 00 28  |
| 130     | 00 30  |
| ...     | ... .. |

- 一致性协议状态为M，S和I状态。
- 每条指令有下述形式：
- **P#:<op><addr>[<value>]**  
P#代表CPU号；<op>代表读写；<addr>代表内存地址；<value>代表写操作的新值

## 解答

---

- 在初始状态（上图所示）分布进行下面操作，请列出内存和 **cache** 的结果状态（例如一致性状态，标签和数据）。仅仅描述状态改变的块。例如 **P0.B0: (I, 120, 00 01)** 表示处理器 **P0** 的块 **B0** 有最后状态 **I**，标记为 **120**，数据位 **00** 和 **01**。
- **a. P0: read 120**
- **b. P0: write 120 ← 80**
- **c. P3: write 120 ← 80**
- **d. P1: read 110**
- **e. P0: write 108 ← 48**
- **f. P0: write 130 ← 78**
- **g. P3: write 130 ← 78**

## 解答

---

- a. P0: read 120 -> P0.B0: (S, 120, 0020)  
returns 0020
- b. P0: write 120<-80 -> P0.B0: (M, 120, 0080)  
P3.B0: (I, 120, 0020)
- c. P3: write 120<-80 -> P3.B0: (M, 120, 0080)
- d. P1 read 110->P1.B2: (S, 110, 0010)  
returns 0010
- e. P0: write 108<-48 ->P0.B1: (M, 108, 0048)  
P3.B1: (I, 108, 0008)
- f. P0: write 130<-78 ->P0.B2: (M, 130, 0078)  
M: 110<-0030  
(write back to memory)
- g. P3: write 130<-78 ->P3.B2:(M, 130, 0078)

## 例题选讲（18）

➤ **例3：**对于目前一般的磁盘而言，读或写一个**512**字节的扇区的平均时间是多少？假设此时磁盘空闲，这样没有排队延迟，公布的平均寻道时间是**9ms**，传输速度是**4M/s**，转速是**7200rpm**，控制器的开销是**1ms**。

➤ **解：**平均磁盘访问时间=平均寻道时间+平均旋转延迟+传输时间+控制器开销

$$9ms + \frac{0.5}{7200RPM} + \frac{0.5KB}{4.0MB/s} + 1ms = 9 + 4.2 + 0.125 + 1 = 14.3ms$$

➤ 假设实际测得的寻道时间是公布值的**33%**，请求块大小为**4KB**，读取和传输速度增加到**40MB/s**，则答案是：

$$3ms + 4.2ms + 0.1ms + 1ms = 8.3ms$$

## 例题选讲 (19)

---

- 假设处理器每秒发送**40 I/O**请求，请求间隔满足负指数分布，磁盘处理每个**I/O**的时间为**20ms**，请问：
  - 1. 磁盘的使用率
  - 2. 请求的排队时间
  - 3. 请求的响应时间
  
- $u$  (server utilization) =  $\lambda \times T_{ser} = 40/s \times 0.02s = 0.8$
- $T_q$  (avg time/customer in queue) =  $T_{ser} \times u / (1-u)$   
 $= 20 \times 0.8 / (1-0.8) = 20 \times 4 = 80ms (0.08s)$
- $T_{sys}$  (avg time/customer in system) =  $T_q + T_{ser} = 100ms$

## 例题选讲（20）

---

- 设有如下所示的磁盘子系统：
- **10个磁盘**，都标示为**MTTF=1,000,000**小时；
- **1个SCSI控制器**，**MTTF=500,000**小时；
- **1个电源****MTTF=200,000**小时；
- **一个风扇****MTTF=200,000**小时；
- **一个SCSI缆线****MTTF=1,000,000**小时；
- 各个部件的生命周期是各自独立的，故障时独立的，计算整个系统的**MTTF**。

## 解答

---

- 故障率之和为:
- 系统故障率= $10 \times 1 / 1,000,000 + 1 / 500,000 + 1 / 200,000 + 1 / 200,000 + 1 / 1,000,000 = 23 / 1,000,000$ 小时
- 系统MTTF为故障率的倒数:
- $MTTF_{\text{系统}} = 1,000,000 / 23 = 43500$ 小时=5年