

## 组合逻辑作业

1. 化简逻辑表达式:  $F = BC + \bar{A}\bar{B}\bar{C} + B\bar{C}$ 。

答案:  $F = B + \bar{A}\bar{C}$

过程:

$$F = B(C + \bar{C}) + \bar{A}\bar{B}\bar{C}$$

$$F = B \cdot 1 + \bar{A}\bar{B}\bar{C}$$

$$F = B + \bar{A}\bar{B}\bar{C}$$

$$F = B + \bar{B}\bar{A}\bar{C}$$

$$\mathbf{F = B + \bar{A}\bar{C}}$$

2. 化简逻辑表达式:  $F = \overline{A + \bar{A}B + \bar{A}\bar{B} + A + \bar{B}}$

答案:  $F = \bar{A}B$

过程:

分为两部分: 1)  $\overline{A + \bar{A}B + \bar{A}\bar{B}}$  和 2)  $\overline{A + \bar{B}}$

$$1) \overline{A + \bar{A}B + \bar{A}\bar{B}}$$

$$\overline{A + B + \bar{A}\bar{B}}$$

$$\overline{A + B + \bar{A}}$$

$$\overline{B + 1}$$

$$\bar{1} = 0$$

$$2) \overline{A + \bar{B}}$$

$$\bar{A} \cdot B$$

合并:

$$F = 0 + \bar{A}B$$

$$\mathbf{F = \bar{A}B}$$

3. 化简逻辑表达式:  $F = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + A\bar{B}C\bar{D} + ABD + \bar{B}C\bar{D} + \bar{A}$

答案:  $F = \bar{B}\bar{C} + \bar{B}\bar{D} + \bar{A} + BD$

过程:

$$F = \bar{B}\bar{C}(\bar{A}\bar{D} + A) + A\bar{B}C\bar{D} + ABD + \bar{B}C\bar{D} + \bar{A}$$

$$F = \bar{B}\bar{C}(\bar{D} + A) + A\bar{B}C\bar{D} + ABD + \bar{B}C\bar{D} + \bar{A}$$

$$F = \bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + A\bar{B}C\bar{D} + ABD + \bar{B}C\bar{D} + \bar{A}$$

$$F = A\bar{B}\bar{C} + \bar{B}C\bar{D} + \bar{B}\bar{D}(AC + \bar{C}) + ABD + \bar{A}$$

$$F = A\bar{B}\bar{C} + \bar{B}C\bar{D} + \bar{B}C\bar{D} + ABD + \bar{A}$$

$$\begin{aligned}
F &= A\bar{B}\bar{C} + \bar{B}\bar{D}(\bar{C} + C) + ABD + \bar{A} \\
F &= A\bar{B}\bar{C} + \bar{B}\bar{D} + ABD + \bar{A} \\
F &= A(\bar{B}\bar{C} + BD) + \bar{B}\bar{D} + \bar{A} \\
\mathbf{F} &= \mathbf{\bar{B}\bar{C} + BD + \bar{B}\bar{D} + \bar{A}}
\end{aligned}$$

4. 化简逻辑表达式:  $F = AC + \bar{A}\bar{B}C$

答案:  $F = AC + \bar{B}C$

过程:

$$\begin{aligned}
F &= C(A + \bar{A}\bar{B}) \\
F &= C(A + \bar{B}) \\
\mathbf{F} &= \mathbf{AC + \bar{B}C}
\end{aligned}$$

5. 化简逻辑表达式:  $F = ABC + ABD + ABE + ACD + ACE + \overline{A + D + E} + \bar{B}\bar{C}D + \bar{B}\bar{C}E + \bar{B}\bar{D}\bar{E} + \bar{C}\bar{D}\bar{E}$

答案: 先完成  $\overline{A + D + E} = \bar{A}\bar{D}\bar{E}$ ; 卡诺图 (注意要把 ABC 展开成 4 个由 5 变量组成的乘积项); 最简表达式为  $A + \bar{B}\bar{C} + \bar{D}\bar{E}$

ABC DE	000	001	011	010	110	111	101	100
00	1	1	1	1	1	1	1	1
01	1				1	1	1	1
11	1				1	1	1	1
10	1				1	1	1	1

【注意】000 列是不能与 111、101、100 构成 4 列的!!!

过程:

Column code: COL-ROW

ABC → 6-1, 6-2, 6-3, 6-4

ABD → 5-3, 5-4, 6-3, 6-4

ABE → 5-2, 5-3, 6-2, 6-3

ACD → 6-3, 6-4, 7-3, 7-4

ACE → 6-2, 6-3, 7-2, 7-3

$\sim(ADE)$  → 1-1, 2-1, 3-1, 4-1

$\sim B\sim CD$  → 1-1, 1-2, 1-3, 8-1, 8-2, 8-3

$\sim B\sim CE$  → 1-3, 1-4, 8-3, 8-4

$\sim B\sim D\sim E$  → 1-1, 2-1, 7-1

$\sim C\sim D\sim E$  → 1-1, 4-1, 5-1, 8-1

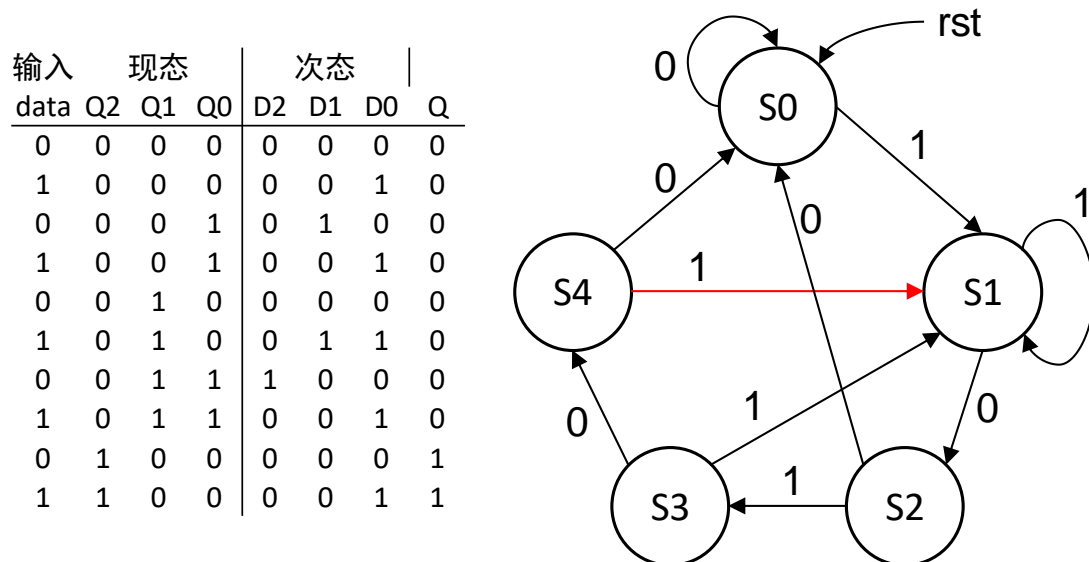
ABC DE	000	001	011	010	110	111	101	100
00	1	1	1	1	1	1	1	1
01	1				1	1	1	1
11	1				1	1	1	1
10	1				1	1	1	1

## 时序逻辑作业

6. 某序列检测器，输入为 data，输出为 Q。当检测到 1010 时，Q 输出 1 个周期的 1，否则 Q 输出 0。给出状态机次态逻辑表达式和 Q 的逻辑表达式。注意：{1010} 为独立检测，不与后续序列拼接，即 {101010} 视为 1 次匹配成功。

答案：假设寄存器 D2D1D0 的编码 000、001、010、011 及 100 分别对应 S0 至 S4。由于 Q 输出宽度以周期为单位，因此 Q 只能是状态的函数，即该状态机是摩尔型状态机。

答案：



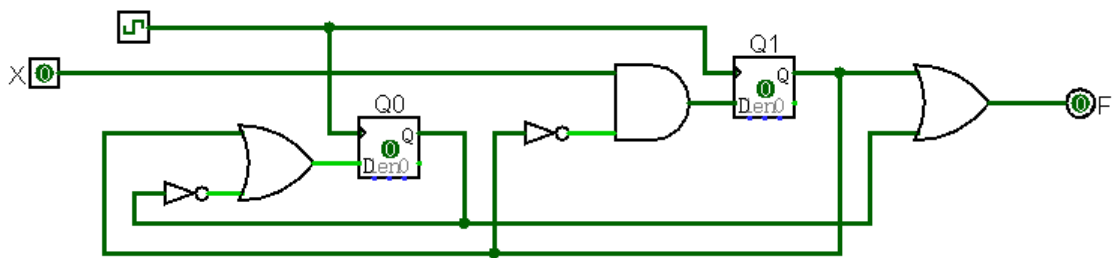
$$D2 = \overline{Q2} \cdot Q1 \cdot Q0 \cdot \overline{data}$$

$$D1 = \overline{Q2} \cdot \overline{Q1} \cdot Q0 \cdot \overline{data} + \overline{Q2} \cdot Q1 \cdot \overline{Q0} \cdot data$$

$$D0 = \overline{Q2} \cdot data + \overline{Q1} \cdot \overline{Q0} \cdot data$$

$$Q = Q2 \cdot \overline{Q1} \cdot \overline{Q0}$$

7. 状态机如下图所示。构造次态逻辑表达式和输出表达式，写出状态转换及输出表，画出状态图，分析状态机类型。



答案:

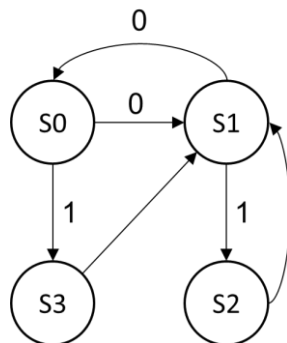
Moore 型状态机，因为 F 只与状态寄存器相关。

$$Q1 = \overline{Q1} \cdot X$$

$$Q0 = Q1 + \overline{Q0}$$

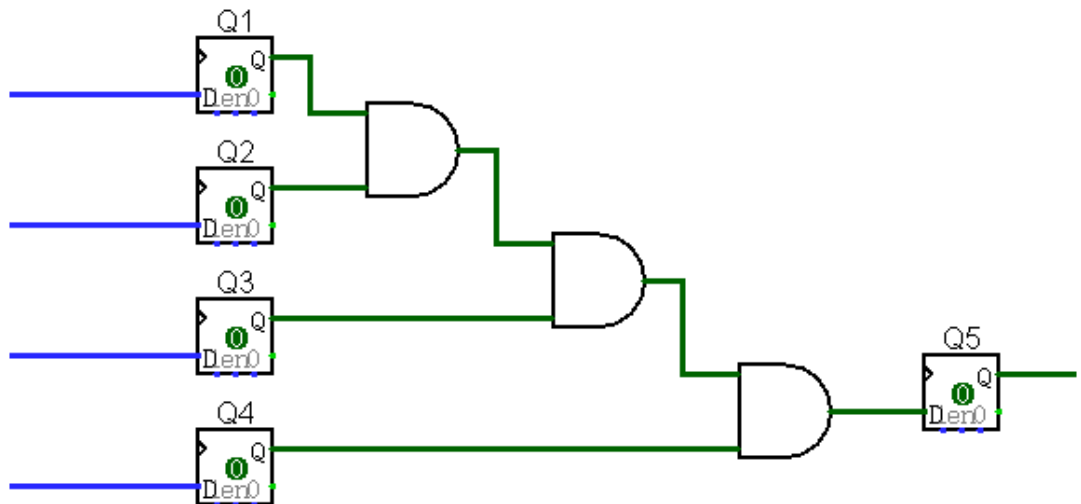
$$F = Q1 + Q0$$

Q1	Q0	X	Q1 <sup>n</sup>	Q0 <sup>n</sup>	F	现态	X	次态
0	0	0	0	1	0	S0	0	S1
0	0	1	1	1	0	S0	1	S3
0	1	0	0	0	1	S1	0	S0
0	1	1	1	0	1	S1	1	S2
1	0	0	0	1	1	S2	0	S1
1	0	1	0	1	1	S2	1	S1
1	1	0	0	1	1	S3	0	S1
1	1	1	0	1	1	S3	1	S1



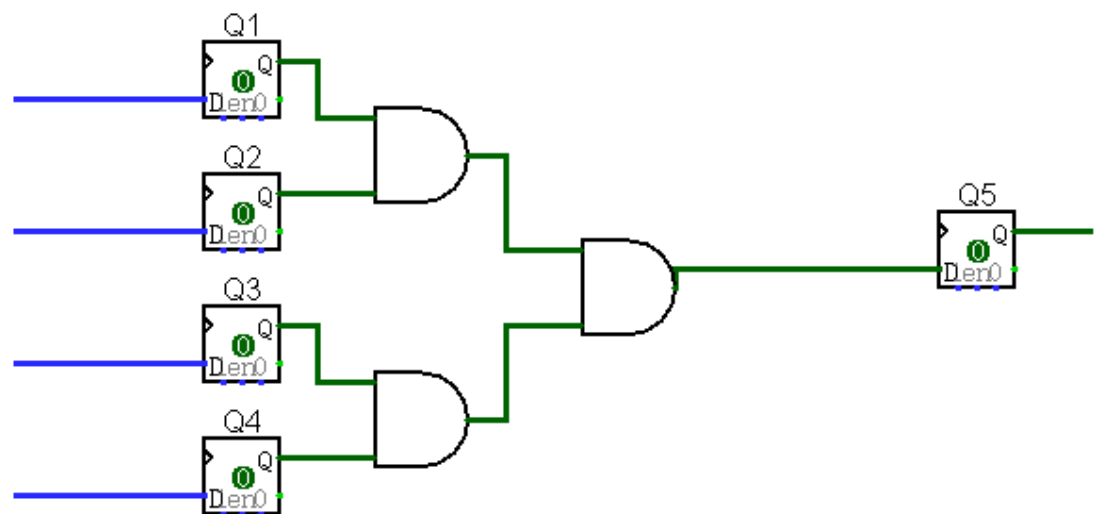
8. 对于如下电路，假设每个与门的延迟均为 T，连线延迟、寄存器固有延迟等均为 0。

- 计算关键路径的延迟以及寄存器的最大时钟频率。
- 重构电路以提高时钟频率，并计算提高比例。



答案:

- 关键路径为 3 个 AND 门的串接，延迟为  $3T$ 。最大时钟频率为  $1/3T$ 。
- 优化后为 2 层 AND，延迟为  $2T$ 。最大时钟频率为  $1/2T$ ，比原设计频率提高 50%。



9. 《数字设计和计算机体系结构》：第 3.20 题。

答案:

- 由于输出与输入相关，因此这是一个 Mealy 型状态机。
- 2 个寄存器 Q1 和 Q0，编码值分别为：S0=0b00；S1=0b01；S2=0b10
- 根据状态图得到如下真值表。关键点在于对 S2 转 S0 的分析（表中绿色部分。无论是否对于化简有用，都一定先补全!）

现态 S1S0	A	B	次态 S1S0	F
00	0	X	00	0

00	1	X	01	0
01	X	0	00	0
01	X	1	10	0
10	1	1	10	1
10	0	0	00	0
10	0	1	00	0
10	1	0	00	0

4) 表达式如下:

$$S1 = \overline{S1} \cdot S0 \cdot B + S1 \cdot \overline{S0} \cdot A \cdot B$$

$$S0 = \overline{S1} \cdot \overline{S0} \cdot A$$

$$F = S1 \cdot \overline{S0} \cdot A \cdot B$$

5) 功能: 检测 A 和 B 先后输入为 1, 此后 AB 同时为 1。

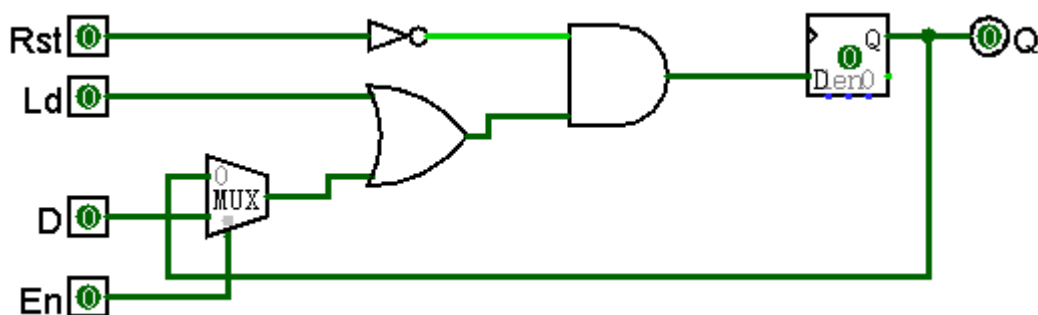
10. 用 D 触发器为核心, 设计一个支持使能 (En 若为 0, 则寄存器值不变)、预置 (Ld 为 1, 则寄存器值为 1)、清除 (Rst 为 1, 则寄存器被清除) 功能的寄存器。寄存器输入信号为 D, 输出为 Q。所有控制信号都是同步控制信号。3 个信号的优先级从高到低为: Rst、Ld、En。

**答案:**

1) 因为 3 个控制信号都是同步控制信号, 所以 D 触发器的输入端是 3 个控制信号及 D 的函数。

Rst	Ld	En	D	Q	D 触发器输入
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	Q	Q
0	0	1	0	X	0
0	0	1	1	X	1

2) 电路结构如下图



## 数据表示与运算方法作业

11. 计算  $222_3$ ,  $222_4$ ,  $222_5$  对应的十进制值。

**答案:**

$$222_3 = 26;$$

$$222_4 = 42;$$

$$222_5 = 62$$

12. 分别计算  $0b1011101$  与  $0xB23$  的十进制值（均按无符号数处理）

**答案:**

$$0b1011101 = 1+4+8+16+64 = 93;$$

$$0xB23 = 0b1011\_0010\_0011 = 1+2+32+256+512+2048 = 2851$$

13. 用 16 进制方式表示无符号数  $1001111110_2$ 。

**答案:**

$$0x27E$$

14. 除了十进制转换为二进制的方法，下面给出了从十进制转换为五进制和九进制的 2 个具体案例。根据该案例总结出从 10 进制转换为 N 进制的一般性方法。

表 0-1 十进制 2007 转换为 5 进制的计算过程

步骤	被除数	商	余数	位序	备注
1	2007	401	2	0	2007 除以 5
2	401	80	1	1	401 除以 5
3	80	16	0	2	80 除以 5
4	16	3	1	3	16 除以 5
5	3	0	3	4	被除数小于除数 5，计算结束

43210 位序

$$2007_{10} = 31012_5$$

表 0-2 十进制 2018 转换为 9 进制的计算过程

步骤	被除数	商	余数	位序	备注
1	2018	224	2	0	2018 除以 9
2	224	24	8	1	224 除以 9

3	24	2	6	2	24 除以 9
4	2	0	2	3	被除数小于除数 9，计算结束

$$\begin{array}{c} 3210 \text{位序} \\ 2018_{10} = 2682_9 \end{array}$$

**答案:**

组织循环：被除数对除数进行除法直至商为 0；被除数调整为上一次的商。  
将所有的余数倒序排列就是结果。

15. 给出快速估算  $2^{28}$  的方法。

**答案:**

$$2^{28} = 2^{20} \times 2^8 \approx 256 \times 10^6$$

16. 给出 6 位二进制补码的表示范围。

**答案:**

{-32, +31}

17. 将下列十进制数转换为 6 位二进制补码并完成计算，同时指出结果是否存在溢出。

①16+15    ② 16+18    ③16-8    ④-16-16    ⑤-24-13

**答案:**

1) 011111;

2) 溢出;

3) 001000;

4) 100000;

5) 溢出

18. 下列代码执行结束后，请用 32 位二进制补码方式分别表示 c、s 与 us 的值。

```

1 | char      c ;
2 | short     s ;
3 | unsigned short us ;
4 |
5 | c = -1 ;
6 | s = c ;
7 | us = (unsigned short)c ;

```

**答案:**

c: 0xFFFF\_FFFF;

s: 0xFFFF\_FFFF;

us=0x0000\_FFFF



## 指令和汇编作业

19. 请从指令编码格式的角度分析 MIPS 寄存器个数为 32 个的合理性。提示：主要分析如果寄存器个数多于 32 时，对于指令编码会带来哪些负面影响。

**答案：**

多于 32 个，则编码位数就至少为 6 位。

- a) R 型移位指令：移位位数为 4 位，无法单条指令实现在 32 位内任意移位。
- b) I 型计算：立即数部分则会减至 14 位。扩展至 32 位时至少需要 2 条指令。
- c) I 型 beq：转移地址范围大幅度减少。

20. 请以指令为单位分析 jal 的被调用函数的入口地址的范围。

**答案：**

如果将 4G 划分为 16 个 256MB，则可以在 jal 所在的那个 256MB 区段内任意跳转。

21. 请结合 C 语言的特点，分析 beq 指令格式中 imm 域的合理性。

**答案：**

beq 的 imm 有 16 位，相当于 64K 条指令的范围。意味着以 beq 为基准，其上下可跳转的范围为  $\{-32K, +32K\}$ 。一条 C 语句大于在 10 条指令左右。这意味着 if-else 语句块的大下为 6.4K 条 C 语句。从程序设计合理性角度，不应该有这么大的语句块。因此 imm 的范围足够了。

22. 请把下列 C 代码翻译为汇编代码。假设 \$s0 存储着变量 i，可用的寄存器只有 \$s0~\$s3。

```
do {  
    循环体;  
while ( 0<i && i<100 );
```

**答案：**

```
Loop :  
    循环体  
    blez $s0, Loop_End      // i ≤ 0 时退出  
    slt $s0, $s1, 100       // i ≥ 0 时，$s1 为 0  
    beq $s1, $0, Loop_End    //  
    j Loop  
Loop_End :
```

23. 请用 beq 和 bne 以及与分支无关的指令来完成下面这条语句的功能。提示：  
要防止溢出。

```
slt $s0, $s1, $s2
```

**答案:**

```
s1 小于 s2: s0 为 1
s1 等于 s2: s0 为 0
s1 大于 s2: s0 为 0
```

```
sub $t0, $s1, $s2
beq $t0, $0, SET0    // s1 等于 s2
srl $t0, $t0, 31     // 只保留符号位
beq $t0, $0, SET0    // 符号位为 0: S1>S2
addi $s0, $0, 1      // 符号位为 1: S1<S2
j END
```

SET0:

```
addi $s0, $0, 0
```

END:

24. 程序员编写了如下汇编程序来完成 1000 个字节的复制任务。假设循环开始前，\$s0 和 \$s1 分别指向源字符串首地址和目的字符串首地址。

```
LOOP:  lb $t0, 0($s0)
        beq $t0, $0, TAIL
        sb $t0, 0($s1)
        addi $s0, $s0, 1
        addi $s1, $s1, 1
        j LOOP
```

TAIL:

- 1) 请计算上述代码执行的指令总数。

**答案:**

1000 字节，还得增加 1 个字节的结束符（即 0），因此总计循环 1001 次。

所以执行的总指令数=6\*1001=6006

- 2) 请最大化优化上述代码。优化代码只能使用本章讲授的指令。优化后代码仍然采用循环结构，且每次循环只能有 1 次复制操作。

**答案:**

循环 250 次用于复制，然后 S1 就指向了结束符所在的【字】。为此，需要写入字内的最高字节（因为最高字节是结束符）

```
addi $t1, $0, 250
```

Loop:

```

lw $t0, 0($s0)
sw $t0, 0($s1)
addi $s0, $s0, 4
addi $s0, $s1, 4
addi $t1, $t1, -1
bne $t1, 0, LOOP
sb $0, -3($s1)

```

3) 请计算优化后代码执行的指令总数。

**答案:**

指令总数=6\*250+1=1501

25. 请编写一个名为 fib(n) 的递归函数来计算第 n 个斐波那契数。斐波那契数列的计算公式为： $f(n+2)=f(n+1)+f(n)$ ， $f(1)=f(2)=1$ 。

1) 用 MARS 模拟器测试程序是否正确。

**答案:**

```

.data
    vars: .space 0
.align 2
.align 1
    str0: .asciiz "%d "
    str1: .asciiz "Enter total number of terms: "
    str2: .asciiz "Fibonacci series is : "
    str3: .asciiz "%d\\t%d\\t"
    str4: .asciiz "\\n"
.text
main :
    addi $sp, $sp, -8
    addi $sp, $sp, -20
    addi $t3, $0, 0
    move $s0, $t3
    sw $s0, 8($sp)
    addi $t0, $0, 1
    move $s1, $t0

```

```
sw $s1, 12($sp)
la $a0, str1
li $v0, 4
syscall
li $a0, '\n'
li $v0, 11
syscall
li $v0, 5
syscall
sw $v0, 20($sp)
la $a0, str2
li $v0, 4
syscall
li $a0, '\n'
li $v0, 11
syscall
la $a0, str3
li $v0, 4
syscall
li $a0, '\n'
li $v0, 11
syscall
lw $a0, 8($sp)
li $v0, 1
syscall
li $a0, '\n'
li $v0, 11
syscall
lw $a0, 12($sp)
li $v0, 1
```

```

syscall
li $a0, '\n'
li $v0, 11
syscall
lw $t0, 8($sp)
move $s0, $t0
lw $t1, 12($sp)
move $s1, $t1
lw $t3, 20($sp)
addi $t4, $0, 2
sub $t2, $t3, $t4
move $s2, $t2
jal LABEL1
la $a0, str4
li $v0, 4
syscall
li $a0, '\n'
li $v0, 11
syscall
addi $sp, $sp, 8
j exit

```

LABEL1:

```

addi $sp, $sp, -8
addi $sp, $sp, -4
addi $sp, $sp, -12
sw $s2, 16($sp)
sw $s1, 12($sp)
sw $s0, 8($sp)
sw $ra, 4($sp)
lw $t0, 16($sp)

```

```

addi $t1, $0, 0
ble $t0, $t1, LABEL3
lw $t3, 8($sp)
lw $t4, 12($sp)
add $t2, $t3, $t4
move $s0, $t2
sw $s0, 20($sp)
la $a0, str0
li $v0, 4
syscall
lw $a0, 20($sp)
li $v0, 1
syscall
li $a0, '\n'
li $v0, 11
syscall
lw $t0, 12($sp)
move $s0, $t0
sw $s0, 8($sp)
lw $t0, 20($sp)
move $s1, $t0
sw $s1, 12($sp)
lw $t0, 8($sp)
move $s0, $t0
lw $t1, 12($sp)
move $s1, $t1
lw $t3, 16($sp)
addi $t4, $0, 1
sub $t2, $t3, $t4
move $s2, $t2

```

```

        jal LABEL1
        b LABEL3
LABEL3:
        lw $s2, 16($sp)
        lw $s1, 12($sp)
        lw $s0, 8($sp)
        lw $ra, 4($sp)
        addi $sp, $sp, 4
        addi $sp, $sp, 20
        jr $ra
exit:

```

- 2) 如果栈空间容量为 4KB，请估算递归调用次数的极限。

答案：

假设函数需要入栈的寄存器总数为  $n$ （含 PC 以及通用寄存器等），则栈容量需求为  $4n$  字节。调用次数的极限为  $4K/4n=1K/n$ 。

# CPU 作业

26. 为了将 DM 结果向 ALU 转发，本章采用如图 6-16(a)的思路：从最后一级流水线寄存器 MEM/WB 向 ALU 转发。但这个设计在执行如下指令序列时必须暂停一个时钟周期。设计师认为采用如图 6-16(b)的思路就可以解决这个问题：从直接从 DM 向 ALU 转发。虽然图 6-16(b)可以解决上述问题，但却使得流水线时钟频率下降了，请分析具体原因（假设 IM 读出、RF 读出、ALU、DM 读出的延迟均为 L）。

lw \$1, xxx

add yyy, \$1,

zzz

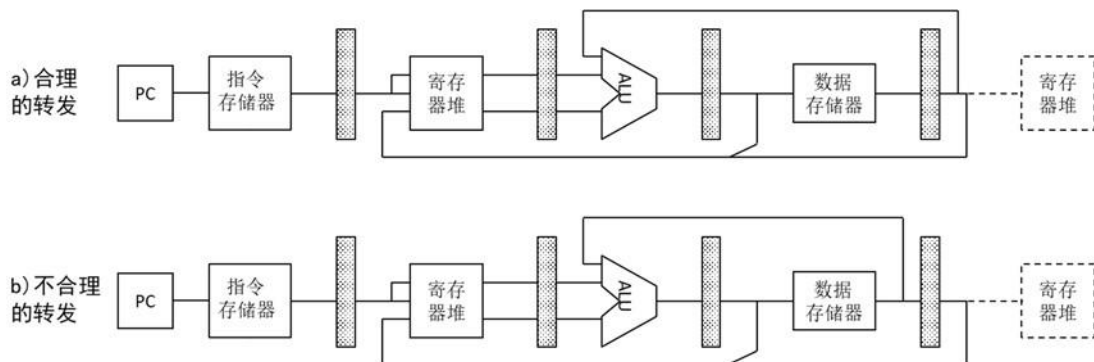


图 6-16 DM 转发的两种思路

**答案：**

由于从 DM 向 ALU 反馈，使得从 EX/MEM 寄存器到其自身之间存在了组合逻辑，即 DM+ALU，其延迟为 2L。流水线频率变为 1/2L，因此图 b 流水线性能是图 a 的 50%。

27. 在流水中插入寄存器可以提高时钟频率。但是，随着级数增多，流水线性能提升会遇到瓶颈。首先，寄存器自身的时序开销（如寄存器建立时间与保持时间等）对于性能改善的影响越来越大。其次，随着流水线级数的增长，分支冒险会导致流水线排空的问题愈发严重。第三，数据冒险造成的暂停也会随之增多。

假设：5 级流水线 CPI 为 1.2，且每增加 1 级，CPI 增加 0.1；单周期 CPU 关键路径延迟为 800ps，寄存器自身时序开销为 50ps。

- a) 建立 CPI 与流水线级数 N ( $N \geq 5$ ) 的计算公式。

**答案：**

$$CPI = 1.2 + 0.1(N - 5) = 1.2 + 0.1N - 0.5 = 0.1N + 0.7$$

- b) 建立时钟周期延迟  $T_c$  与流水线级数 N 的计算公式。



答案:

$$T_c = \frac{800}{N} + 50$$

c) 给出一条指令执行时间的计算公式。

答案:

$$\text{一条指令执行时间 } CPI \times T_c = (0.1N + 0.7) \times \left( \frac{800}{N} + 50 \right)$$

$$= 80 + \frac{560}{N} + 5N + 35$$

$$= 115 + \frac{560}{N} + 5N$$

d) 请指出 N 为多少时, 流水线性能最好。

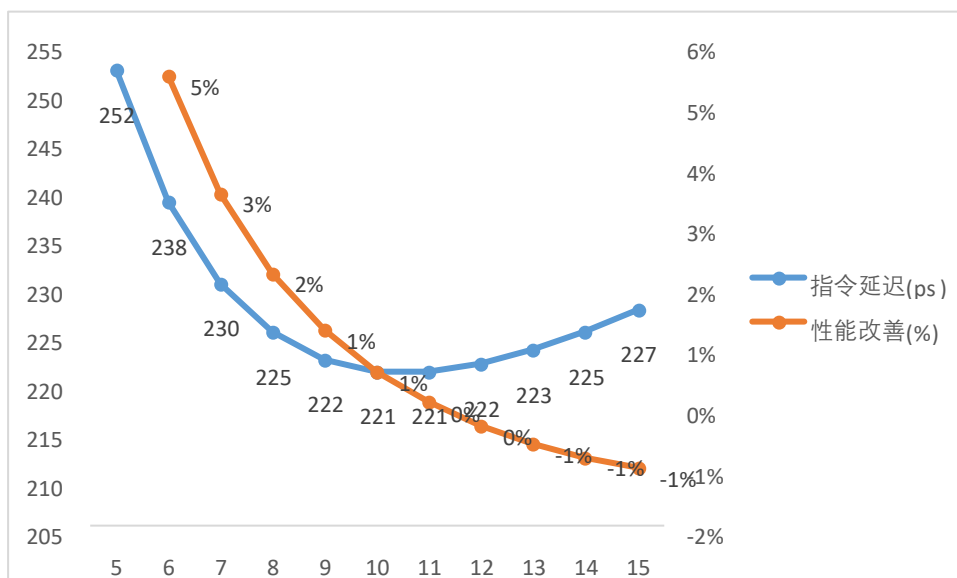
答案:

N = 10, 性能最好。

e) 请指出 N 为多少时, 流水线性能改善最为显著。

答案:

N = 6, 性能改善最为显著。



28. 设计师将单周期数据通路改造为如图 6-17 所示的 3 级流水线。假设 RF 不支持内部转发。

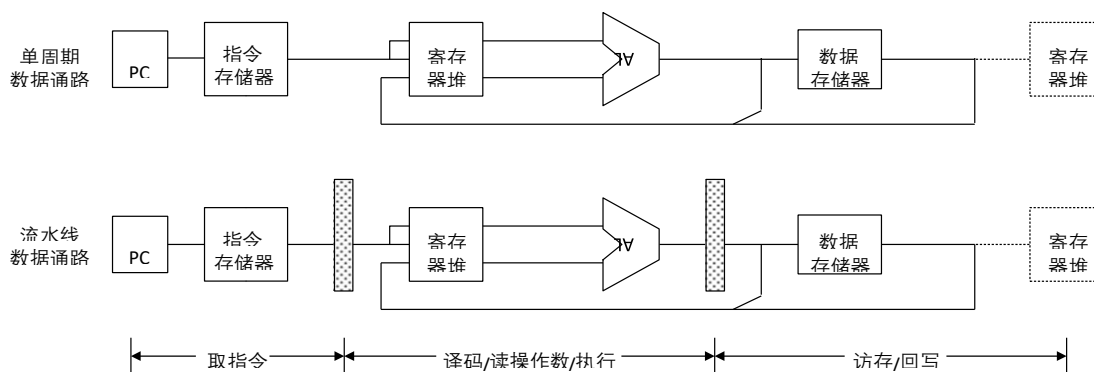


图 6-17 改造单周期数据通路为 3 级流水线

- 1) 流水线是否仍然可能会因为 beq 而需要清空流水线？如果会，最多有几条指令会被清除？

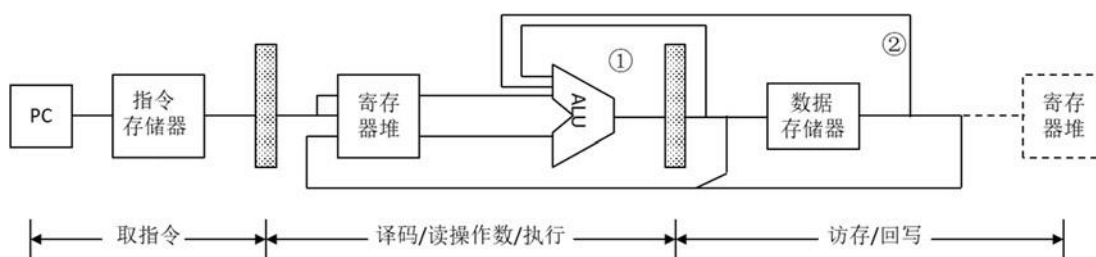
**答案：**

会。beq 的比较判断电路只能前移至译码/读操作数/执行阶段。这与本章介绍的处理是完全一致的。可能被清除的指令仍然是 beq 后面的那条指令，因此只有 1 条。

- 2) 假设指令集只有 {lw, add}，请以 rs 寄存器为例，增加旁路以应对所有的数据冒险可能。给出思路即可，不用讨论因此带来的 MUX 及其控制。

**答案：**

从 ALU 和 DM 分别向 ALU 的 A 端转发数据。注意，由于 RF 无内部转发，因此必须有路径 2。



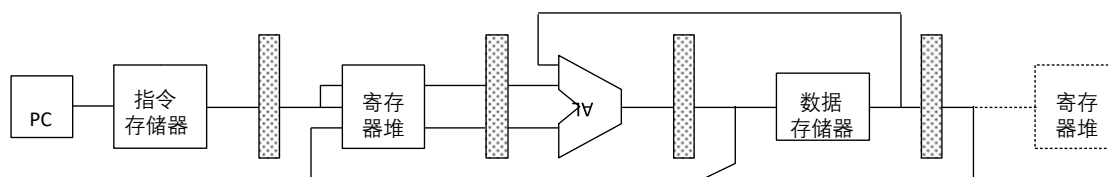
- 3) 对于第 2 问的指令集，能否消除 rs 寄存器相关的全部数据冒险？为什么？

**答案：**

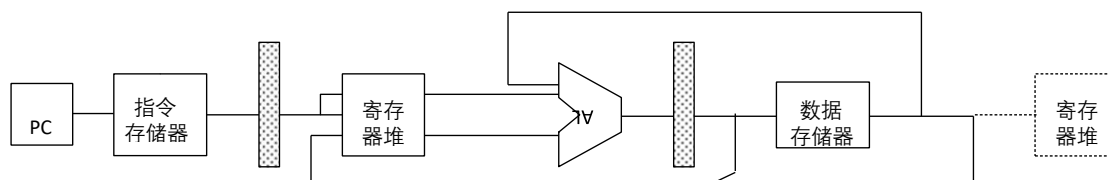
可以。因为对于 {lw, add} 来说，rs 只在 ALU 的 A 端使用。无论当前哪条指令需要使用 ALU，其前面只可能有 1 条指令位于最后一级。而无论是哪条指令位于最后一级，结果均已产生，因此必然可以通过旁路转发来消除数据冒险。

29. 同样是从 DM 向 ALU 的转发，图 6-18(a)的设计会使得 5 级流水线性能下降。请分析图 6-18(b)的设计会导致 3 级流水线性能下降吗？假设 IM

读出、RF 读出、ALU、DM 读出的延迟均为  $L$ ，忽略所有控制器延迟及 MUX 延迟。



5 级流水线：不合理的转发



3 级流水线：合理的转发

图 6-18 从 DM 向 ALU 的转发

**答案：**

- 1) 在 3 级流水线中，中间那段的延迟已经是最坏的  $2L$  了。
- 2) 转发电路的组合逻辑为 DM 读出+ALU，同样也是  $2L$ 。
- 3) 这表明转发并没有增加最坏延迟，因此不会导致性能下降。

30. 下图所示的流水线 CPU 执行如下指令序列。

```
I1 lw $1, 0($2)
I2 addi $1, $1,
$1
I3 sw $1, 0($2)
I4 lw $1, 4($2)
I5 sw $1, 8($2)
```

- a) 分析上述指令执行过程中流水线共计需要暂停多少个时钟周期。

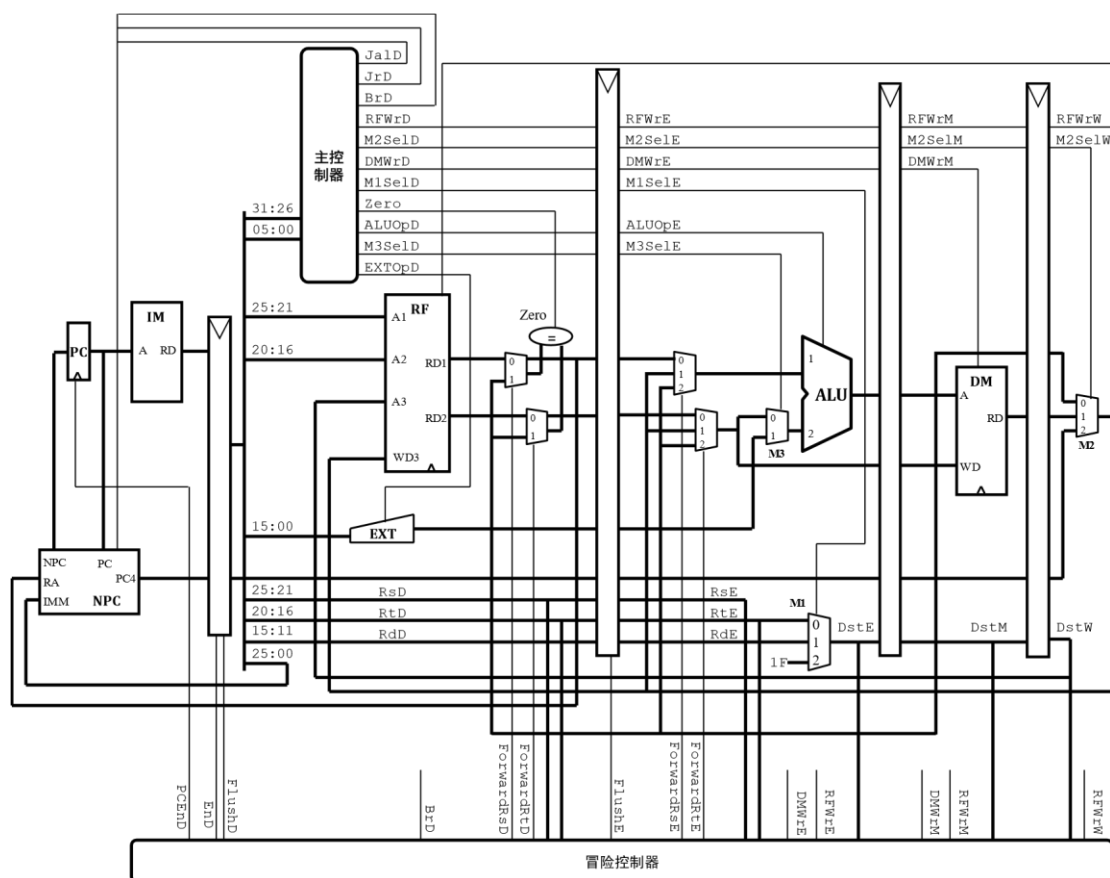
**答案：**

指令 1 和指令 2 之间在  $\$1$  有数据冒险，在转发电路存在的前提下，仍然必须暂停 1 个时钟周期。指令序列调整为：

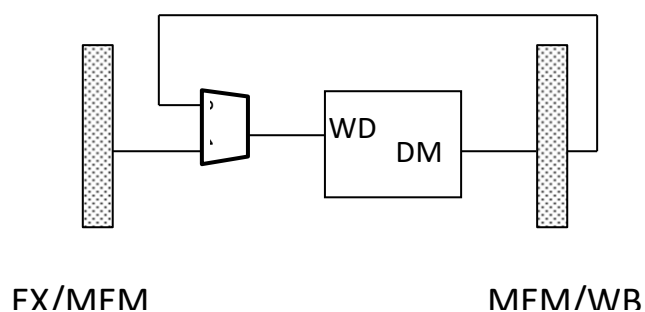
1	lw \$1, 0(\$2)
2	nop
3	addi \$1, \$1,
4	\$1 sw \$1,
5	0(\$2) lw \$1,
6	4(\$2) sw \$1,
	8(\$2)

原指令 4（现指令 5）与原指令 5（现指令 6）之间存在数据冒险。但原设计没有转发电路，因此必须将 sw 冻结在 IF/ID，直至 lw 进

b) 是否可以增加转发来提升流水线性能？如果可以，请简述设计思路。



前文的 2 个暂停周期是因为解决 1w-sw 之间的数据冒险而产生的, 因此需要从 MEM/WB 增加一个转发至 DM 的 WD。



31. 在第 5 题图中所示的流水线 CPU 执行某程序, 其指令分布如下: load 占 15%, store 为 10%, 分支指令为 10%, R 型计算类指令为 65%。假设: load-R 导致暂停概率为 30%; load-store 导致的暂停概率为 5%; 分支指令预测成功率为 75%。计算流水线执行该程序的 CPI。

- 1) load-R: 没有数据数据相关时, load 的 CPI 为 1。如果有数据相关, 需暂停 1 个时钟周期, 其 CPI 为 2。

$$CPI_{load-R} = 1 \times (1 - 30\%) + 2 \times 30\% = 1.3$$

- 2) load-store: 没有数据数据相关时, load 的 CPI 为 1。如果有数据相关, 需暂停 2 个时钟周期, 其 CPI 为 3。

$$CPI_{load-store} = 1 \times (1 - 5\%) + 3 \times 5\% = 1.1$$

- 3) store: 题目中未出现数据相关, 因此  $CPI_{store}$  为 1。

- 4) 分支：预测成功，分支的 CPI 为 1。如果预测失败，需暂停 1 个时钟周期，分支的 CPI 为 2。

$$CPI_{branch} = 1 \times 75\% + 2 \times (1 - 75\%) = 1.25$$

- 5) R 型: CPI 为 1。

$$\begin{aligned} CPI &= CPI_{load-R} \times 30\% + CPI_{load-store} \times 5\% + CPI_{store} \times 10\% \\ &\quad + CPI_{branch} \times 10\% + CPI_{R-Type} \times 65\% \\ &= 1.3 \times 30\% + 1.1 \times 5\% + 1 \times 10\% + 1.25 \times 10\% + 1 \times 65 \\ &= 1.32 \end{aligned}$$

32. 如图 6-19 所示, 某 MIPS 标准 5 级流水线仅支持 M 级向 D 级的转发 (注意: 寄存器堆无内部转发)。某程序员编写了如下 MIPS 代码, 请回答下列问题。

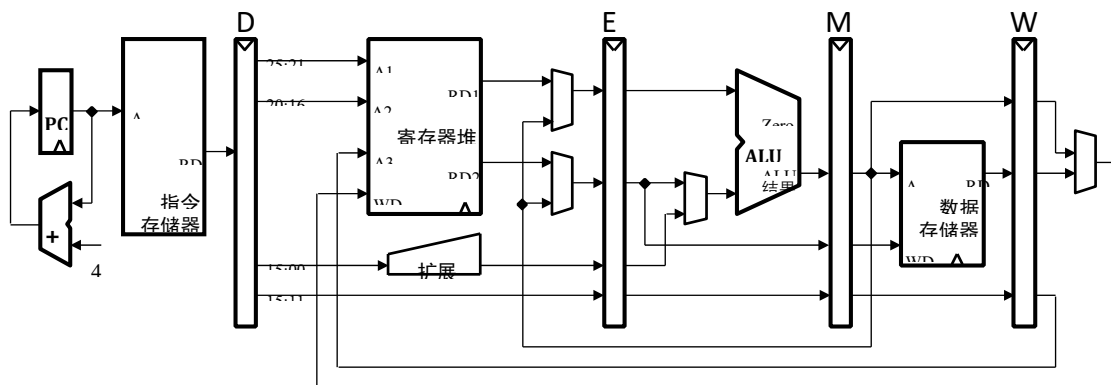


图 6-19 仅支持 M 级向 D 级转发的 5 级流水线

- I1: lw \$1, 0(\$2)  
I2: sw \$1, 0(\$1)  
I3: add \$3, \$2, \$2  
I4: sub \$4, \$1, \$3  
I5: or \$5, \$5, \$6

- a) 请指出上述指令片段在上述流水线中执行时存在的所有数据相关。

答案:

lw-sw 在\$1 有 2 次相关；lw-sub 在\$1 相关；add-sub 在\$3 相关。

b) 请通过调整指令顺序来优化上述指令片段以最大化减少暂停。

**答案:**

```
I1: lw  $1, 0($2)
I3: add $3, $2, $2
I5: or  $5, $5, $6
I2: sw  $1, 0($1)
I4: sub $4, $1, $3
```

c) 对于优化前和优化后指令片段，分别给出流水线的执行时间，并说明理由。示例：对于 2 条无冒险的指令片段，则流水线执行时间为 6 个时钟周期。

**答案:**

优化前：13 个 cycle

由于仅在 W 和 D 之间存在转发，因此 lw-sw 以及 add~sub 之间必须分别插入 2 个 NOP。指令总数从 5 条变为 9 条，因此流水线共计需要  $9 + (5 - 1) = 13$  个时钟周期。 优化后：9 个 cycle

优化后，lw~sw 和 add~sub 的数据相关均通过转发解决了，因此无需插入 NOP，故执行时间  $= 5 + (5 - 1) = 9$  个时钟周期

## 存储作业

33. 一标准流水线 CPU 的理想 CPI 为 1，其执行的程序片段中 load 和 store 类指令占 30%，其余指令均为 R 型指令。主存延迟为 50 个时钟周期。L1 级 cache 性能为：命中时间为 1 个时钟周期，缺失率为 2%。L2 级 cache 性能为：命中时间为 10 个时钟周期，缺失率为 5%。针对如下情况分别计算流水线的实际 CPI：没有 cache；只有 L1 级 cache；L2 级 cache。

**答案：**

- 1) 由于没有 cache，因此指令和数据的缺失率均为 100%；

暂停周期数 = 指令缺失率 × 指令缺失代价 + 数据缺失率 × 数据缺失代价

$$= 50 + 30\% \times 50 = 65$$

$$CPI_{\text{无}} = CPI_{\text{基准}} + \text{暂停周期数} = 1 + 65 = 66$$

- 2) 只有 L1 级 cache

暂停周期数 = 指令缺失率 × 指令缺失代价 + 数据缺失率 × 数据缺失代价

$$= 2\% \times 50 + 30\% \times 2\% \times 50$$

$$= 1 + 0.3$$

$$= 1.3$$

$$CPI_{L1} = CPI_{\text{基准}} + \text{暂停周期数} = 1 + 1.3 = 2.3$$

- 3) 有 L2 级 cache：

由于不分指令 cache 和数据 cache，因此 L1 指令缺失代价与 L1 数据缺失代价是相同的。

暂停周期数 = 指令缺失率 × 指令缺失代价 + 数据缺失率 × 数据缺失代价

$$= 2\% \times \text{L1 缺失代价} + 30\% \times 2\% \times \text{L1 缺失代价}$$

$$= (2\% + 0.6\%) \times (10 + 5\% \times 50)$$

$$= 0.026 \times 12.5$$

$$= 0.325$$

$$CPI_{L2} = CPI_{\text{基准}} + \text{暂停周期数} = 1 + 0.325 = 1.325$$

34. 直接映射 cache 参数如下：cache 的数据容量为 16KB，cache 块为 16B。

现该 cache 接入一个 32 位 cpu。请给出 cache 的 TIO 结构以及包含标记后的 cache 块容量。

**答案:**

- 1) Offset:  $\log_2(16)=4$
- 2) Index:  $16KB/16=1024$  块,  $\log_2(1024)=10$
- 3) Tag:  $32-4-10=18$
- 4) cache 块总容量: 数据+Tag+Valid= $16B+18b+1b=16B+19b$

35. 组相联映射 cache 参数如下: cache 的数据容量为 512KB, 16 路相联, cache 块为 32B。现该 cache 接入一个 32 位 cpu。请给出 cache 的 TIO 结构。

**答案:**

- 1) offset:  $\log_2(32)=5$
- 2) Index:  $512KB/32B=16K$  块,  $16K/16=1K$  组,  $\log_2(1K)=10$
- 3) Tag:  $32-5-10=17$

36. 对于第 3 问的系统, cache 采用写回策略, cache 命中时间为 1 个时钟周期, 从主存调入一块或回写一块的代价均为 100 个时钟周期。某程序片段将起始地址为 0000\_0000h 的 1MB 内存单元全部初始化为 0, 代码如下所示。

```
int *p=0x0 ;
for ( int i=0; i<1024*1024/4; i++ )
    *p++ = 0 ;
```

- 1) 计算该程序片段中的数据访问缺失率 (忽略指令缺失等因素)。

**答案:**

- i. 1MB 主存对应的块数:  $1MB/32B=32K$  (块)
- ii. 由于是线性初始化, 因此这 32K 块均会访问 1 次。可以看成是如下循环:
- iii. 第 1 次循环:
  1. 当初始化主存块 0 字 0 时, 缺失。于是从将主存块 0 写入 cache 的组 0 块 0。之后的 7 个字 (8 字-1 字) 均命中。
  2. 当初始化主存块 1 字 0 时, 缺失。于是从将主存块 1 写入 cache 的组 1 块 0。之后的 7 个字 (8 字-1 字) 均命中。
  3. 类似的, 主存块 15, 映射到 cache 组 15 块 0。



- iv. 第 2 次循环：
    - 1. 主存块 16 字 0 时，缺失，会被映射到 cache 组 0 块 1；之后 7 个字均命中。
    - 2. 依次类推，主存块 31 字 0 时，缺失，会被映射到 cache 组 15 块 1。之后 7 个字均命中。
  - v. 重复上述过程，直至第 1K 次循环。在第 1K 次循环中，主存块 (16K-16) 至主存块 (16K-1) 被分别装入 cache 组 0 块 15 和 cache 组 15 块 15。
  - vi. 至此，主存的前 16K 块已经访问结束了，cache 所有块均被写满了，再无空闲块了。
  - vii. 而主存的后 16K 块的行为与前 16K 块完全一样，区别在于需要把前 16K 块写回主存。
  - viii. 主存访问时，其对应的每个主存块的访问情况完全相同：字 0 缺失，字 1~字 7 均命中。因此，数据访问缺失率=1/8=12.5%
- 2) 计算该程序片段的数据访问实际需要多少个时钟周期。
- i. 对于每个块来说，有 1 次缺失，其代价是 100 个时钟周期；7 次命中，7 个周期。故每个块是 100+7=107 个时钟周期。
  - ii. 总共有 32K 块，因此访问时间=107\*32K（时钟周期）。
  - iii. 但是后 16K 块还会导致 16K 次替换，每次替换都会有 100 时钟周期，因此替换时间=100\*16K（时钟周期）
  - iv. 总时间=107\*32K+100\*16K=5024K（时钟周期）
- 3) 计算程序片段的存储访问次数与实际时钟周期数之比。
- i. 程序总存储访问次数=1MB/4=256K
  - ii. 程序总存储访问次数：实际时钟周期数=256K：5024K $\approx$ 1:20
  - iii. 意味着每 20 个时钟周期才能完成一次存储访问。
37. 假设某系统的虚页和物理页尺寸均为 8KB，40 位虚地址，物理主存容量为 32GB。请问虚页号与物理页号的位数分别是多少。

**答案：**

- 1) 8KB 页面，则页内地址为 13 位。
- 2) 虚页号 (VPN) 位数 = 40-13 = 27 位

3) 物理页号 (PPN) 位数 =  $\log_2(32G) - 13 = 35 - 13 = 22$  位

38. 设计师给第 5 问的系统设计了 TLB。TLB 采用 2 路组相联结构，共有 256 个页表项。TLB 中的每个页表项结构如下图所示。

Valid	Dirty	访问权限	TLB Tag	PPN
1 位	1 位	2 位		

1) 请问每个页表项的位数是多少。

**答案:**

TLB 的组数 =  $256 / 2 = 128$  组

TLB 的 index 位数 =  $\log_2(128) = 7$

TLB 的 tag 位 = 虚地址位数 - 页内偏移位数 - index 位数

$$= 40 - 13 - 7$$

$$= 20 \text{ 位}$$

页表项位数 =  $1 + 1 + 2 + 20 + 22 = 46$  位

Valid	Dirty	访问权限	TLB Tag	PPN
1 位	1 位	2 位	20	22

2) 页表的总存储容量是多少位？

**答案:**

TLB 总容量 = TLB 每个页表项 x 表项数

$$= 46 \times 256$$

$$= 11776 \text{ 位}$$

3) 操作系统设计团队希望将页面尺寸从 8KB 降低至 4KB，但硬件设计团队认为会增加硬件开销，因此不同意改变页面尺寸。你作为硬件设计师团队负责人，请陈述理由。

**答案:**

若页面尺寸改成 4KB，页内偏移便会变成 12 位，虚页号变成 28 位，实页号变成 23 位。这样一来，页表项位数也会增大，TLB 存储容量也会变大，所以硬件开销也自然会增大。

39. 现有  $4K \times 8$  位容量的 DRAM 存储芯片。

- 1) DRAM 芯片内置译码器输出的行/列选择线各是多少条。

**答案:**

DRAM 芯片共计 4K 个单元，因此内部共需

地址线的位数=  $\log_2(4K)=12$ ,

即内部地址为 A[11:0]。由于 DRAM 采用行地址和列地址共享，因此行/列地址选择线均为  $12/2=6$ 。

- 2) DRAM 芯片内置的刷新地址计数器位数为多少位。

**答案:**

由于 DRAM 每次刷新一行，因此刷新地址计数器位数与行地址的位数相同，即刷新地址计数器位数为 6 位。

- 3) DRAM 每刷新周期刷新一行存储单元是多少位。

**答案:**

一行的单元数由列地址数决定，即一行单元数= $2^6=64$  个。故总的存储单元位数为  $64 \times 8=512$  位。

- 4) 主存容量为  $8K \times 8$  位，需使用 DRAM 芯片的数量。

**答案:**

需要的 DRAM 芯片数量 =  $(8K \times 8) / (4K \times 8) = 2$  片。

- 5) 给出每个 DRAM 芯片的片选控制信号的逻辑表达式。

**答案:**

主存地址为 A[12:0]。每片 DRAM 芯片的位数为 A[11:0]，因此 A12 用来选择 2 个芯片。假设芯片 1 对应 0~4K，芯片 2 对应 4K-8K，则：

芯片 0 的片选=  $\neg A_{12}$

芯片 1 的片选=  $A_{12}$