1 总结指令流水线 CPI 的优化技术，如基于硬件还是软件，主要思想，对应了流水线 CPI 公式的哪个分量等。

2 简单描述什么是真数据依赖（True data dependence），输出依赖（Output dependence），反依赖（Antidependence），寄存器重命名（Register renaming）

3 请回答下列关于 cache 的问题

（1）一个计算机系统用 32 位内存地址。它有 128KB 8 路组相联缓存，每个块大小为 64B。计算标签(tag)、缓存索引(cache index) 和块偏移(block offset)的位数。

（2）处理器有一个小型直接映射缓存，能够容纳四个缓存块。 内存是按字节寻址的，每个缓存块由 32 字节组成。 处理器使用 12 位内存地址。 假设每个缓存块的初始标签值如下：

块	标签
00	00110
01	00001
10	00000
11	Invalid

处理器从以下十进制地址顺序读取数据：32， 48， 64， 128

对于上述每个地址，指出缓存访问将导致命中还是未命中。并给出详细过程。

4.

（1）简述平均内存访问时间公式，总结附录 B 的 6 种缓冲优化技术。

（2）计算题：假设存在一个计算机， CPI 是 2（没有储存器停顿（Memory stalls）），仅有载入/储存指令进行数据访问，并且载入/储存的指令的占比是 36%。假如缺失代价（Miss penalty）是 40 个时钟周期，指令缓存缺失率（Instruction miss rate）是 2%， 数据缓存缺失率（Data miss rate）是 4%。请计算平均内存访问时间、不使用 cache 技术的 CPU 的平均 CPI，和使用了 cache 技术获得的加速比。**题目 8：**（10 分）

a)考虑具有 32 字节块和2^{19}字节的数据的写分配、写回和直接映射缓存。假设一个字节寻址机器有 32 位地址

5. 1）以下地址位分配的每个字段的宽度是多少

TAG	Set index	Block offset

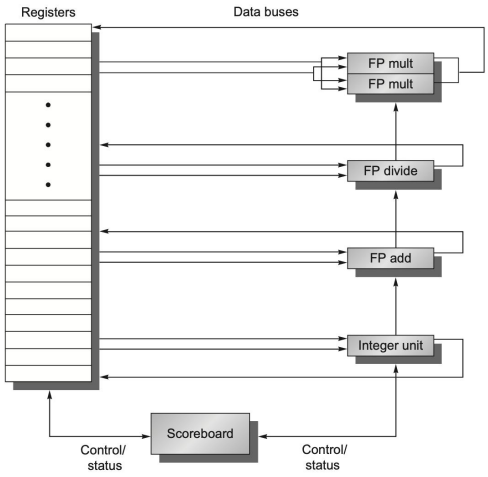
2）构成缓存的总位数是多少（**数据和缓存管理**）？请给出计算过程。

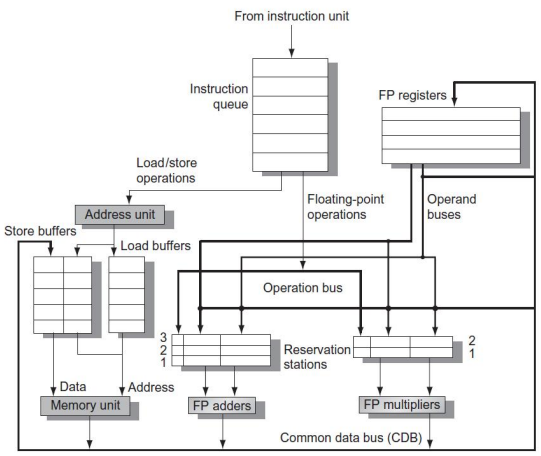
b)假设有一个单级、1KB 直接映射 L1 缓存和 16 字节块。有 4GB 的内存。一个整数是 4 字节。数组是块对齐的.

1）计算 L1 缓存中的标签(tag)、索引(index)和偏移位(offset bits)的个数

TAG	Set index	Block offset

5.简述下图各个主要部件的功能。填写 Tomasulo 算法与记分牌的运行状态表，讲述两者异同。





对于以下代码序列，写出各个周期运行时状态（信息表已给出）：

$$fld \quad f6, 32(\times 2)$$

$$fld \quad f2, 44(\times 3)$$

$$fmul.d\ f0, f2, f4$$

$$fsub.d\ f8, f2, f4$$

$$fdiv.d\ f8, f0, f6$$

$$fadd.d\ f6, f8, f2$$

计分牌功能部件：

指令	指令状态			
	发射	读操作数	执行	写回
$fld \quad f6, 32(\times 2)$				
$fld\ f2, 44(\times 3)$				
$fmul.d\ f0, f2, f4$				
$fsub.d\ f8, f2, f4$				
$fdiv.d\ f8, f0, f6$				
$fadd.d\ f6, f8, f2$				

名称	保留站								
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
$Integer$									
$Add1$									
$Add2$									
$Add3$									
$Mult1$									
$Mult2$									

字段	寄存器状态								
	$f0$	$f2$	$f4$	$f6$	$f8$	$f10$	$f12$	…	$f30$
Qi									
Val									

Tomasulo 算法功能部件：

指令	指令状态		
	发射	执行	写结果
$fld \quad f6, 32(\times 2)$			
$fld\ f2, 44(\times 3)$			
$fmul.d\ f0, f2, f4$			
$fsub.d\ f8, f2, f4$			
$fdiv.d\ f8, f0, f6$			
$fadd.d\ f6, f8, f2$			

名称	保留站						
	Busy	Op	Vj	Vk	Qj	Qk	A
$Load1$							
$Load2$							
$Add1$							
$Add2$							
$Add3$							
$Mult1$							
$Mult2$							

字段	寄存器状态

	$f0$	$f2$	$f4$	$f6$	$f8$	$f10$	$f12$	···	$f30$
Qi									
Val									

6 解释下图每条指令的功能，怎么理解 ISA 设计的正交性原则。

Example instruction	Instruction name	Meaning
ld x1,80(x2)	Load doubleword	Regs[x1]←Mem[80+Regs[x2]]
lw x1,60(x2)	Load word	Regs[x1]$\leftarrow_{64}$ Mem[60+Regs[x2]]$_0$)^{32} ## Mem[60+Regs[x2]]
lwu x1,60(x2)	Load word unsigned	Regs[x1]$\leftarrow_{64}$ 0^{32} ## Mem[60+Regs[x2]]
lb x1,40(x3)	Load byte	Regs[x1]$\leftarrow_{64}$ (Mem[40+Regs[x3]]$_0$)^{56} ## Mem[40+Regs[x3]]
lbu x1,40(x3)	Load byte unsigned	Regs[x1]$\leftarrow_{64}$ 0^{56} ## Mem[40+Regs[x3]]
lh x1,40(x3)	Load half word	Regs[x1]$\leftarrow_{64}$ (Mem[40+Regs[x3]]$_0$)^{48} ## Mem[40+Regs[x3]]
flw f0,50(x3)	Load FP single	Regs[f0]$\leftarrow_{64}$ Mem[50+Regs[x3]] ## 0^{32}
fld f0,50(x2)	Load FP double	Regs[f0]$\leftarrow_{64}$ Mem[50+Regs[x2]]
sd x2,400(x3)	Store double	Mem[400+Regs[x3]]$\leftarrow_{64}$ Regs[x2]
sw x3,500(x4)	Store word	Mem[500+Regs[x4]]$\leftarrow_{32}$ Regs[x3]$_{32..63}$
fsw f0,40(x3)	Store FP single	Mem[40+Regs[x3]]$\leftarrow_{32}$ Regs[f0]$_{0..31}$
fsd f0,40(x3)	Store FP double	Mem[40+Regs[x3]]$\leftarrow_{64}$ Regs[f0]
sh x3,502(x2)	Store half	Mem[502+Regs[x2]]$\leftarrow_{16}$ Regs[x3]$_{48..63}$
sb x2,41(x3)	Store byte	Mem[41+Regs[x3]]$\leftarrow_{8}$ Regs[x2]$_{56..63}$

Figure A.25 The load and store instructions in RISC-V. Loads shorter than 64 bits are available in both sign-extended and zero-extended forms. All memory references use a single addressing mode. Of course, both loads and stores are available for all the data types shown. Because RV64G supports double precision floating point, all single precision floating point loads must be aligned in the FP register, which are 64-bits wide.