**5.6.1**

Video streaming访问时间=512KB

直接映射缓存大小=64 KB

缓存行大小=32字节

两个流之间的差等于4。

未命中率与块大小成比例。

在64KB的直接映射高速缓存中，块大小=512KB/32bytes=16K，

未命中=64KB/16K/4=4\*8/4=8，所以，每8次访问都是一次失误，所以，未命中率=1/8=12.5%，

根据工作集或缓存大小，未命中率不会改变。

基于3C模型，这些失误可以分为冷失误。

**5.6.2**

（1）块大小=512KB/16bytes=32K，未命中率=32K/64KB/4=1/2B/4=25%

（2）块大小=512KB/64bytes=8K，未命中率=8K/64KB/4=6.25%

（3）块大小=512KB/128bytes=4K，未命中率=4K/64KB/4= 3.125%

因此，随着缓存行大小的增加，缺失率降低，这是因为更大的缓存行可以覆盖更多的连续地址访问。这种工作负载利用了空间局部性，因为连续的数据项通常被连续访问。这也解释了为什么更大的缓存行大小会导致更低的缺失率。

**5.6.3**

对于上述地址流，由于地址模式是顺序的，预取可以非常有效。在这种情况下，一旦第一个缓存行被访问，接下来的两个缓存行会被预取到流缓冲区中。因此，当计算机访问这些地址时，它们已经被预取并且存在于缓冲区中，这样就可以避免缺失（miss）。

考虑到流缓冲区有两个条目，这意味着它可以存储两个缓存行。因此，当访问第一个地址（0）时，接下来的两个缓存行（假设缓存行大小为64字节，那么这将涵盖地址4到131）将被预取。由于地址流是顺序的，这将导致所有后续的访问都是命中，直到超出预取缓冲区的范围。

因此，在这种情况下，只有在开始时和每次预取缓冲区耗尽并需要加载新的缓存行时才会发生缺失。由于我们假设缓存行可以在前一个缓存行的计算完成之前被加载，这将进一步减少缺失的可能性。

所以，对于给定的地址流，如果预取策略被正确实施，缺失率将非常低，接近于零。这是因为每次缓存行被访问时，下一个缓存行已经被预取并准备好被使用。这种预取策略有效地利用了地址流的顺序性，几乎消除了缺失。

**5.6.4**

为了找到给定缺失延迟为 ( 20 \times B ) 周期的情况下的最佳缓存块大小（B），我们需要计算每个块大小的总平均访问时间（Average Memory Access Time, AMAT）。AMAT 可以用以下公式计算：

AMAT=命中时间+(缺失率×缺失延迟)

在这个例子中，我们假设命中时间是一个周期（1-CPI），缺失延迟是 ( 20 \times B ) 周期。因此，对于每个缓存块大小，AMAT 为：

AMAT=1+(缺失率×20×B)

我们可以用这个公式来计算两个不同的缺失率（a 和 b）对应的每个缓存块大小的 AMAT，并找出哪个缓存块大小提供了最低的 AMAT。

对于缺失率 a：

对于 8 字节块：( AMAT = 1 + (0.08 \* 20 \* 8) = 1 + 12.8 = 13.8 )

对于 16 字节块：( AMAT = 1 + (0.03 \*20 \* 16) = 1 + 9.6 = 10.6 )

对于 32 字节块：( AMAT = 1 + (0.018 \* 20 \* 32) = 1 + 11.52 = 12.52 )

对于 64 字节块：( AMAT = 1 + (0.015 \* 20 \* 64) = 1 + 19.2 = 20.2 )

对于 128 字节块：( AMAT = 1 + (0.02 \* 20 \* 128) = 1 + 51.2 = 52.2 )

对于缺失率 b：

对于 8 字节块：( AMAT = 1 + (0.04 \* 20 \* 8) = 1 + 6.4 = 7.4 )

对于 16 字节块：( AMAT = 1 + (0.04 \* 20 \* 16) = 1 + 12.8 = 13.8 )

对于 32 字节块：( AMAT = 1 + (0.03 \* 20 \* 32) = 1 + 19.2 = 20.2 )

对于 64 字节块：( AMAT = 1 + (0.015 \* 20 \* 64) = 1 + 19.2 = 20.2 )

对于 128 字节块：( AMAT = 1 + (0.02 \* 20 \* 128) = 1 + 51.2 = 52.2 )

在这两种情况下，我们都在寻找最低的 AMAT。对于缺失率 a，最佳缓存块大小是 16 字节，因为它提供了最低的 AMAT（10.6）。对于缺失率 b，最佳缓存块大小是 8 字节，因为它提供了最低的 AMAT（7.4）。

因此，考虑到缺失延迟为 ( 20 \times B ) 周期的情况下，对于缺失率 a，最佳缓存块大小是 16 字节；对于缺失率 b，最佳缓存块大小是 8 字节。这些计算假设每条指令平均有 1.35 次引用（包括指令和数据）。

**5.6.5**

在这个例子中，我们假设命中时间是一个周期（1-CPI），缺失延迟是 ( 24 + B ) 周期。因此，对于每个缓存块大小，AMAT 为：

AMAT=1+(缺失率×(24+B))

我们可以用这个公式来计算两个不同的缺失率（a 和 b）对应的每个缓存块大小的 AMAT，并找出哪个缓存块大小提供了最低的 AMAT。

对于缺失率 a：

对于 8 字节块：( AMAT = 1 + (0.08 × (24 + 8)) = 1 + 2.56 = 3.56 )

对于 16 字节块：( AMAT = 1 + (0.03 × (24 + 16)) = 1 + 1.2 = 2.2 )

对于 32 字节块：( AMAT = 1 + (0.018 × (24 + 32)) = 1 + 1.008 = 2.008 )

对于 64 字节块：( AMAT = 1 + (0.015 × (24 + 64)) = 1 + 1.32 = 2.32 )

对于 128 字节块：( AMAT = 1 + (0.02× (24 + 128)) = 1 + 3.04 = 4.04 )

对于缺失率 b：

对于 8 字节块：( AMAT = 1 + (0.04 × (24 + 8)) = 1 + 1.28 = 2.28 )

对于 16 字节块：( AMAT = 1 + (0.04 × (24 + 16)) = 1 + 1.6 = 2.6 )

对于 32 字节块：( AMAT = 1 + (0.03 × (24 + 32)) = 1 + 1.68 = 2.68 )

对于 64 字节块：( AMAT = 1 + (0.015 × (24 + 64)) = 1 + 1.32 = 2.32 )

对于 128 字节块：( AMAT = 1 + (0.02 × (24 + 128)) = 1 + 3.04 = 4.04 )

在这两种情况下，我们都在寻找最低的 AMAT。对于缺失率 a，最佳缓存块大小是 32 字节，因为它提供了最低的 AMAT（2.008）。对于缺失率 b，最佳缓存块大小是 8 字节，因为它提供了最低的 AMAT（2.28）。

因此，考虑到缺失延迟为 ( 24 + B ) 周期的情况下，对于缺失率 a，最佳缓存块大小是 32 字节；对于缺失率 b，最佳缓存块大小是 8 字节。这些计算假设每条指令平均有 1.35 次引用（包括指令和数据）。

**5.6.6**

根据题上所给，选择最小的缺失率即可，a和b最小块大小都未64bytes

**5.7.1**

a)

给定P1的L1命中时间=0.62ns

P1的时钟周期=1/L1 hit time=1/0.62ns=1.613GHz

给定P2的L1命中时间=0.66ns

P2的时钟周期=1/L1 hit time=1/0.66ns=1.515GHz

b)

给定P1的L1命中时间=0.96ns

P1的时钟周期=1/L1 hit time=1/0.96ns=1.041GHz

给定P2的L1命中时间=1.08ns

P2的时钟周期=1/L1 hit time=1/1.08ns=0.925GHz

**5.7.2**

a.) 对于处理器1：给定数据：

P1的L1命中时间=0.62ns，L1未命中率=11.4%，未命中惩罚=AMATDRAM=70 ns

AMAT（平均内存访问时间）=（命中时间+（未命中率\*未命中惩罚））；

AMAT=（0.62+（11.4%\*70））；

= (0.62 + 7.98)

AMAT=8.6 ns

对于处理器2：给定数据：

P2的L1命中时间=0.66ns，L1未命中率=8.0%，未命中惩罚=AMATDRAM=70 ns

AMAT（平均内存访问时间）=（命中时间+（未命中率\*未命中惩罚））；

AMAT=（0.66+（8.0%\*70））；

= (0.66 + 5.6)

AMAT=6.26 ns

b) 对于处理器1：给定数据：

P1的L1命中时间=0.96ns，L1未命中率=4.3%，未命中惩罚=AMATDRAM=70 ns

AMAT（平均内存访问时间）=（命中时间+（未命中率\*未命中惩罚））；

AMAT=（0.96+（4.3%\*70））；

= (0.96 + 3.06)

AMAT=3.097 ns

对于处理器2：给定数据：

P2的L1命中时间=1.08ns，L1未命中率=3.4%，未命中惩罚=AMATDRAM=70 ns

AMAT（平均内存访问时间）=（命中时间+（未命中率\*未命中惩罚））；

AMAT=（1.08+（3.4%\*70））；

= (1.08 + 2.38)

AMAT=3.46 ns

**5.7.3**

a)主存访问时间=70ns

memory访问=36%=0.36

基础CPI =1.0

P1的CPI：

未命中惩罚=主存访问时间\*L1 hit time=70\*0.62=43.4

CPI=L1缺失率\*未命中惩罚+基础CPI=5.95

P1的CPI：

未命中惩罚=主存访问时间\*L1 hit time=70\*0.66=46.2

CPI=L1缺失率\*未命中惩罚+基础CPI=4.7

综上所述P2处理器要快一点

b) P1的CPI：

未命中惩罚=主存访问时间\*L1 hit time=70\*0.96=67.2

CPI=L1缺失率\*未命中惩罚+基础CPI=3.28

P1的CPI：

未命中惩罚=主存访问时间\*L1 hit time=70\*1.08=75.6

CPI=L1缺失率\*未命中惩罚+基础CPI=3.5

综上所述P2处理器要快一点

**5.7.4**

a) Miss PenaltyL1=AMATL2=Hit timeL2+Miss RateL2+Miss PenaltyL2=3.22+98%\*70

=3.22+68.6=71.82ns

AMAT= Hit timeL1+Miss RateL1+Miss PenaltyL1=0.62+11.4%\*71.82

=0.62+8.187=8.80ns

所以AMAT比L2 cache要差

b) Miss PenaltyL1=AMATL2=Hit timeL2+Miss RateL2+Miss PenaltyL2=11.48+73%\*70

=11.48+51.1=62.58ns

AMAT= Hit timeL1+Miss RateL1+Miss PenaltyL1=0.96+4.3%\*62.58

=0.96+2.64=3.60ns

所以AMAT比L2 cache要好

**5.7.5**

主存储器访问时间=70ns

所有指令的内存访问时间百分比=36%=36/100=0.36

基本CPI=1.0

L1命中时间=0.62 ns

L1未命中率=11.4%= 11.4/100= 0.114

a) P1未命中时间=70ns/0.62ns=112.9cycles

则对于113cycles则需要113\*0.62=70.06ns

CPI=基础CPI+0.36\*（L1命中时间\*（L2命中+L2缺失率\*memory））

=1+0.36\*(0.114\*(3.22+0.98\*70.06)/0.62)

=1+0.36\*13.216=5.76

所以总CPI=5.76

b)

CPI=基础CPI+0.36\*（L1命中时间\*（L2命中+L2缺失率\*memory））

=1+0.36\*(0.114\*(11.48+0.73\*70)

=1+0.36\*7.13412=3.56828

所以总 CPI 大约是 3.57

**5.7.6**

a)由5.7.5总CPI=5.76,故处理器P1的时间=总CPI\*L1 hit time= 5.76\*0.62=3.57

故P1处理器仍然更快

b) 由5.7.5总CPI=3.57,故处理器P1的时间=总CPI\*L1 hit time= 3.57\*0.96=3.4272

故P1处理器仍然更快