目次

★1　キャッシュメモリ

★1.1　キャッシュメモリとは

不景気で「ほげらんどり」をくびになったほげお君は，大学院進学を決意し，図書館の机に座って勉強をはじめました．ところが，机のある場所から本棚まで距離があるので，本を一冊とってきたり返しにいったりするのに結構時間がかかります．最初のうちは，机には一冊だけ置いといて，別の本を読みたくなったらまず机の本を返しに行ってそれから新しい本をとってくるようにしていたのですが，一度読んで返してしまった本をあとでまた読みたくなることが多くて，その度に本棚にとりにいくのがばからしくなってきました．そこで，一度とってきた本は机の上に置いておき，あとですぐ読めるようにすることを思いつきました．ただし，机の上はそんなに広くないので，一杯になったら一冊返しにいってから新しい本をとってくることにしました．

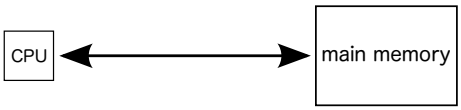
ほげお君：「ぐっどあいであ〜♪」

レジスタ間の加減算などのように CPU 内で完結する演算に比べて，主記憶装置（メインメモリ）へのアクセスをともなう命令（ロード／ストアなど）の実行には長い時間がかかる (☆ 1)．そのため，CPU が主記憶装置を直接アクセスするようにしていると，CPU はメモリアクセスの度に待たされることになって十分な性能を発揮できない．そこで，こんにちのコンピュータでは，CPU と主記憶装置の間に，記憶容量は小さいけれどアクセス速度の速い記憶装置を接続して記憶装置を階層化することでこの問題に対処している．中間におかれるこのような記憶装置のことを**キャッシュメモリ** と呼ぶ．

☆ 1) 後者に要するクロックサイクル数は前者の数十倍になることもある

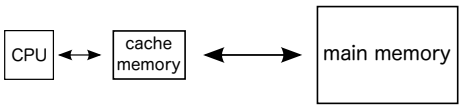
キャッシュ: 英語では cache. 現金を意味する cash とは別.

● キャッシュメモリなし



CPU は，1 つ命令を実行しようとする度に，その命令をフェッチするために主記憶にアクセスする．ロード／ストア等の命令の場合は，さらにデータを読み出す／書き込むためにも主記憶にアクセスする．そのため，CPU 内部の演算を高速化しても，主記憶へのアクセスがボトルネックとなって性能を向上させられない．

● キャッシュメモリあり



命令のフェッチやデータの読み書きの多くを，メインメモリよりもアクセス時間の短いキャッシュメモリ相手に行えれば，CPU が待たされる時間が減り，効率よく命令を実行できるようになる．

CPU が命令やデータにアクセスしようとした際，キャッシュメモリ中にそれらが見つかる場合を**キャッシュヒット**といい，見つからない（主記憶中にしかない）場合を**キャッシュミス** (☆ 2) という．キャッシュはなるべくヒットした方がよいので，キャッシュメモリの容量は大きくしたいが，そうすると高価になる (☆ 3)．そのため，コンピュータ設計者は，なるべく小容量のキャッシュメモリでも有効なように，メモリアクセス全体に対してキャッシュがヒットする割合（**ヒット率**）が高くなるように設計する．

☆ 2) キャッシュ「ミスヒット」と呼ばれることもある．

☆ 3) より高速にアクセスできるようにするため，一般に，キャッシュメモリに用いられる回路（例えば SRAM）は主記憶に用いられるもの（例えば DRAM）よりも容量当たりの価格が高いことが多い．

★ 1.2 キャッシュの方式いろいろ

キャッシュメモリや仮想記憶 (☆ 4) のように階層化された記憶システムでは，**メモリアクセスの空間的局所性** (☆ 5) を活用するために，ある程度連続した記憶領域をひとまとめに扱う．キャッシュメモリではその 1 区画を**ブロック**（または**ライン**）と呼び，仮装記憶では**ページ**と呼ぶことが多い．

キャッシュメモリの構成方式は，ブロックをどのようにキャッシュに格納するか，書き込み動作をどのように実現するか，等によって分類することができるが，この授業では省略する (☆ 6)．

また，1 次キャッシュ，2 次キャッシュ，というように階層化してメモリアクセス時間を短縮しようとする（マルチレベルキャッシュという）ことも多い．

☆ 4) 仮想記憶は，この授業の後半に登場する．

☆ 5) 「局所性」については，★ 1.4 参照．

☆ 6) データ格納方式には，フルアソシアティブ，ダイレクトマップ，セットアソシアティブなどがあり，書き込み方式には，ライトスルー，ライトバックなどがある．2007 年度の高橋の「計算機アーキテクチャ」の第 10 回講義資料に少し解説があります．

Q1. いんちき計算機 II にキャッシュを接続したとする．この計算機の CPU は，キャッシュが必ずヒットする理想的な条件では，どんな命令でも 2 クロックサイクルで実行できる（つまりこの場合の CPI は 2 である）一方，命令をフェッチする場合もデータを読み書きする場合も，キャッシュミスした場合，メモリアクセスのために余分に 40 クロック必要となるとする．この計算機で次の命令を実行した場合，必要なクロック数はそれぞれいくつになるか．(1) 全てキャッシュヒットする場合，(2) 全てキャッシュミスする場合，のそれぞれについて答えなさい．ただし，この CPU はパイプライン処理を行わないものとする．

```
LD      GR1,A
ADDA    GR1,GR2
```

★ 1.3 キャッシュを備えたシステムの性能を評価してみよう

Q2. キャッシュを備えたあるコンピュータを考える．このコンピュータの CPU は，キャッシュが必ずヒットする理想的な条件では，どんな命令でも 2 クロックサイクルで実行できる．一方，命令フェッチの際もデータアクセスの際も，キャッシュミスの場合には余分に 40 クロックサイクルが必要となる．

このコンピュータであるプログラム（P とする）を実行してみたところ，命令フェッチ時のヒット率が 98%，データアクセス時のヒット率が 96%であり, データアクセスを伴うロード／ストア命令は命令全体の 36%を占めることがわかった．これらの値を用いて，P を実行する場合のこのコンピュータの性能を評価してみよう．P の実効命令数を n として，以下の問に答えよ．

(1) 理想的な条件では，P の実行に要するクロックサイクル数はいくつか
(2) 命令フェッチ時のキャッシュミスによって余分にかかるクロック数は，プログラム全体でいくつか
(3) データアクセス時のキャッシュミスによって余分にかかるクロック数は，プログラム全体でいくつか
(4) このプログラムの実行には何クロックかかるか．また，そこから計算される CPI はいくつになるか

★ 1.4 メモリアクセスの局所性

CPU のメモリアクセスには，時間的にも空間的にも**局所性**がある．キャッシュメモリは，それを活かしてアクセスに要する時間を短縮する技術である．

● 時間的局所性の利用

メモリ領域へのアクセスは，時間的に集中することが多い（これを「時間的局所性がある」という）．すなわち，一度アクセスされたメモリ領域は，近いうちに再度アクセスされる可能性が高い．

⇓

キャッシュは，過去にアクセスのあったデータを保持しておくので，次に同じデータへのアクセスが要求された際のアクセス時間を短縮できる．

● 空間的局所性の利用

メモリ領域へのアクセスは，空間的に集中することが多い（これを「空間的局所性がある」という）．すなわち，あるメモリ領域がアクセスされたら，その近くのメモリ領域もアクセスされる可能性が高い．

⇓

上述のように，キャッシュはメモリの内容を適当なサイズのブロック単位でキャッシュにもってくるので，同じブロック内の別のデータへのアクセス時間を短縮できる．

C 言語などの高級言語を用いるプログラマも，このことを知っているとより効率のよいプログラムが書ける．例えば，右の (1) と (2) はどちらも同じ計算結果となるが，(?) の方がメモリアクセスの空間的局所性が高いので，実行時間も短いと考えられる（ただし，実際にはキャッシュのブロックサイズや方式，メモリのデータ転送方式の話もからむので事態はもう少し複雑である）．

```
#define N 5000
double A[N][N], x[N], y[N];

/***** y = Ax (1) *****/
for(i = 0; i < N; i++){
   y[i] = 0.0;
}
for(i = 0; i < N; i++){
  for(j = 0; j < N; j++){
    y[i] += A[i][j] * x[j];
  }
}
```

```
#define N 5000
double A[N][N], x[N], y[N];

/***** y = Ax (2) *****/
for(i = 0; i < N; i++){
   y[i] = 0.0;
}
for(j = 0; j < N; j++){
  for(i = 0; i < N; i++){
    y[i] += A[i][j] * x[j];
  }
}
```

Q3. 上記の (?) に入る正しいものは (1),(2) のいずれか．理由をつけて答えなさい．