

コンピュータの仕組み (11)

柴山 潔

コンピュータの仕組み

- 1 コンピュータシステム
- 2 ハードウェア
- 3 内部装置
- 4 プロセッサ
- 5 メモリ(2)
- 6 外部装置
- 7 論理回路
- 8 オペレーティングシステム

5 メモリ(2)

5.1 メインメモリ —主要なメモリ—

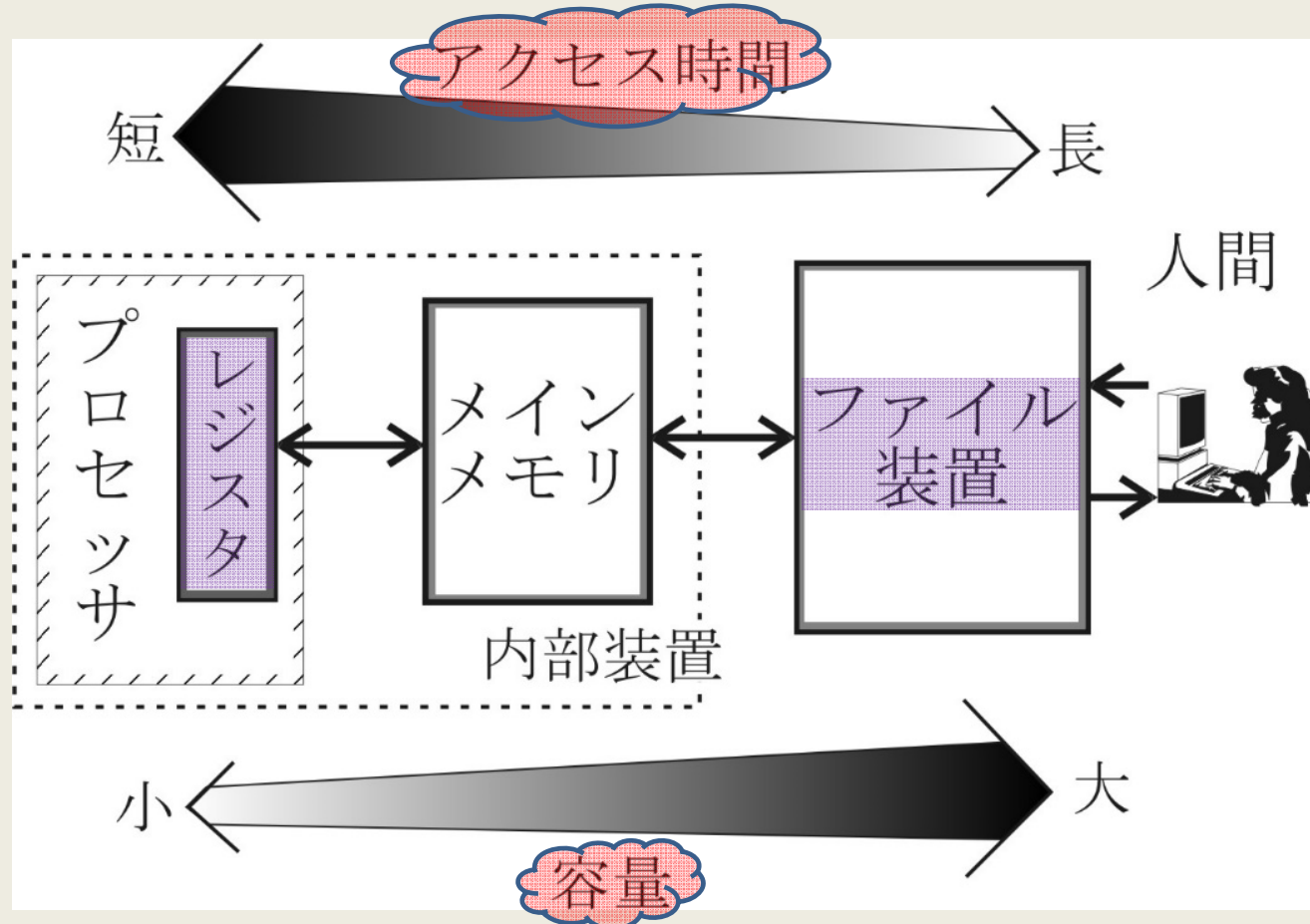
5.2 メモリの機能

5.3 メインメモリの高機能化

メモリの機能とその指標

- **メモリの機能**: 情報の**格納**と(プロセッサなどによる)情報への**アクセス**との2つ
 - この2機能の高低を測る指標は異なる
- **容量**: **格納**能力に対する「どれくらいの**量**が格納できるか？」という指標
- **アクセス時間**(アクセス速度): **アクセス**能力に対する「どれくらいの**時間**(速さ)でアクセスできるか？」という指標

主要なメモリ階層間の関係(図)



メインメモリの高機能化(1)

- **メインメモリ**: 容量とアクセス時間という両立しない性能指標を適度に**バランス**

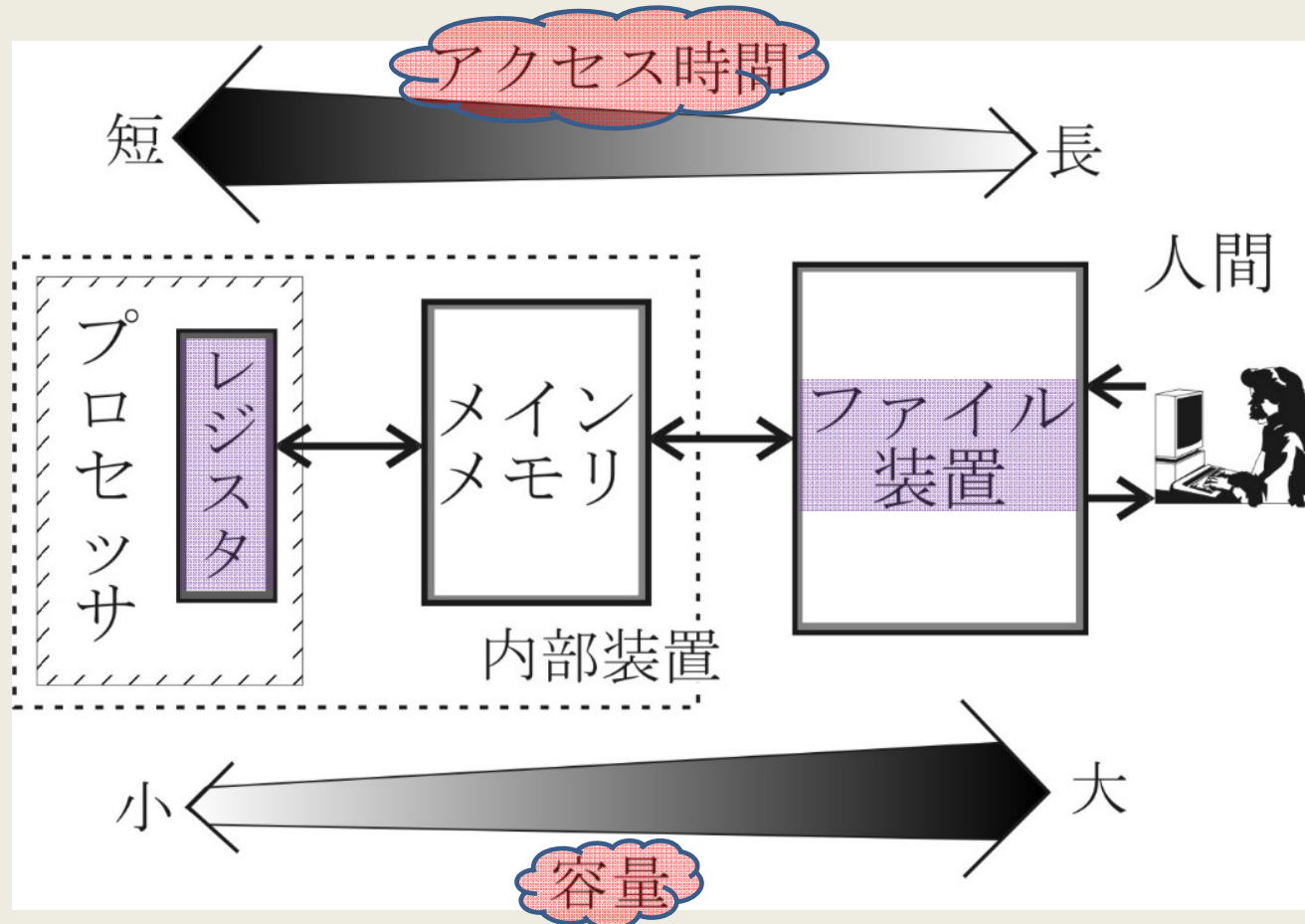


- 人間の要求は限りないから, **メインメモリ**の**高機能化**も強い要求

メインメモリの高機能化(2)

- メモリ機能は2つの指標で測定可
- **メインメモリの高機能化**には2つの観点
 - (1) メインメモリ**容量**の増大
 - (2) メインメモリへの**アクセス時間**の短縮
- 2種類の高機能化は**両立し得ない**要求
- どちらかに絞れば、メインメモリ階層に**隣接するメモリ階層**を利用して**メインメモリの高機能化**(**大容量化**か**アクセスの高速化**かのいずれか)を図ることは可能

主要なメモリ階層間の関係(図)



メインメモリの高機能化(3)

■ 仮想メモリ(virtual memory)

- メインメモリをより大きく見せかける！
- **大容量化** → メインメモリよりも容量性能で優(大)の隣接メモリ階層(ファイル装置が代表的)を利用

■ キャッシュメモリ(cache memory)

- メインメモリ動作をより速く見せかける！
- **アクセス時間の高速化** → メインメモリよりアクセス時間性能で優のメモリ階層を新たに設定・利用
- レジスタ階層とメインメモリ階層間に**新規に**設定するメモリ階層

メインメモリとプログラム(1)

- **メインメモリ**にはプロセッサが実行しようとする**プログラム**が格納
 - **プログラム**が巨大で一時に**メインメモリ**に入りきらない場合はどうする？
- とりあえずは、**メインメモリ以外のメモリ**に格納しておいて、**必要時**に必要なプログラムを**メインメモリ**に持ってくる
 - **メインメモリ**は「**プログラム内蔵**」というコンピュータの原理を支える**ハードウェア**装置

メインメモリとプログラム(2)

- コンピュータは種々の仕事を同時に行うから、メインメモリには複数のプログラムが同時に存在
- 多数のプログラムをあらかじめメインメモリに格納しておくのは、「アクセス時間との適度なバランスを保つ性能」として設定したメインメモリ容量では不足

仮想メモリ方式(1)

- 実際には、プログラムは人間が作成→「ファイル」としてファイル装置に格納
 - ファイル装置には人間が作成したファイルが(必要か不必要かにかかわらず)格納
- 実行するか使用する可能性があるときにファイル装置からメインメモリに転送
 - それをプロセッサが必要時(まさに実行時)にアクセス

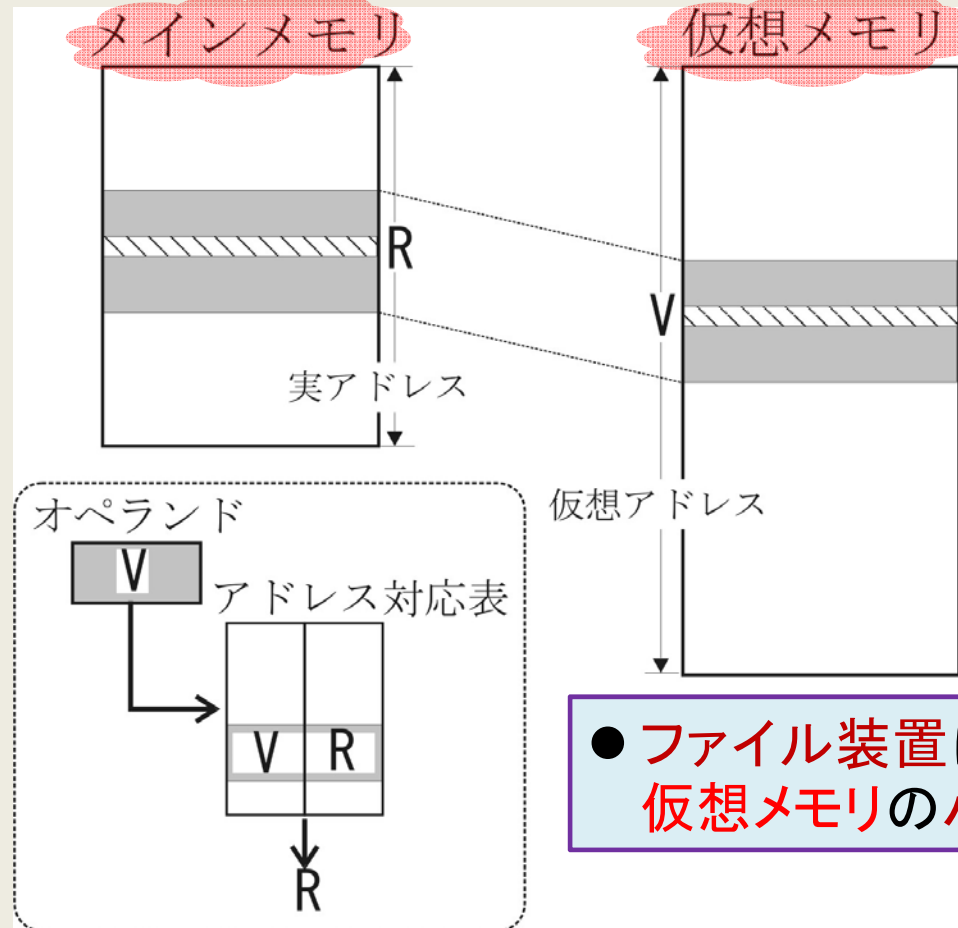
仮想メモリ方式(2)

- プロセッサが実行時にアクセスするプログラムは必ず**ファイル装置**にも存在
- **ファイル装置** = メインメモリの「**バックアップ(backup; 予備, 控え) 装置**」
 - **メインメモリ**には, 「**実行する可能性が高いファイル**」だけを置く

仮想メモリ方式(3)

- 「メインメモリとファイル装置の2メモリ階層を利用して、プロセッサには、仮想(の)メモリを見せる」機能
- 仮想メモリは、実(実際の)メモリであるメインメモリ容量を、見かけ上、大きく及び一定にした仮想のメモリ

仮想メモリ方式(図)



- **ファイル装置はメイン(実)メモリと仮想メモリのバックアップメモリ**

仮想メモリ機能の実現(1)

- メインメモリとファイル装置の2メモリ階層間で対応関係をとる
- 実際には、**仮想アドレス**という「プロセッサから見る場合の(仮想の)アドレス」を付けた**仮想のメモリ**を、**ファイル装置**(ハードディスク装置が代表的)を**実(実際の)メモリ**である**メインメモリのバックアップメモリ**にして、実現

仮想メモリ機能の実現(2)

- プロセッサがプログラムカウンタやマシン命令のオペランドとして指定する命令やデータのアドレスは仮想アドレス

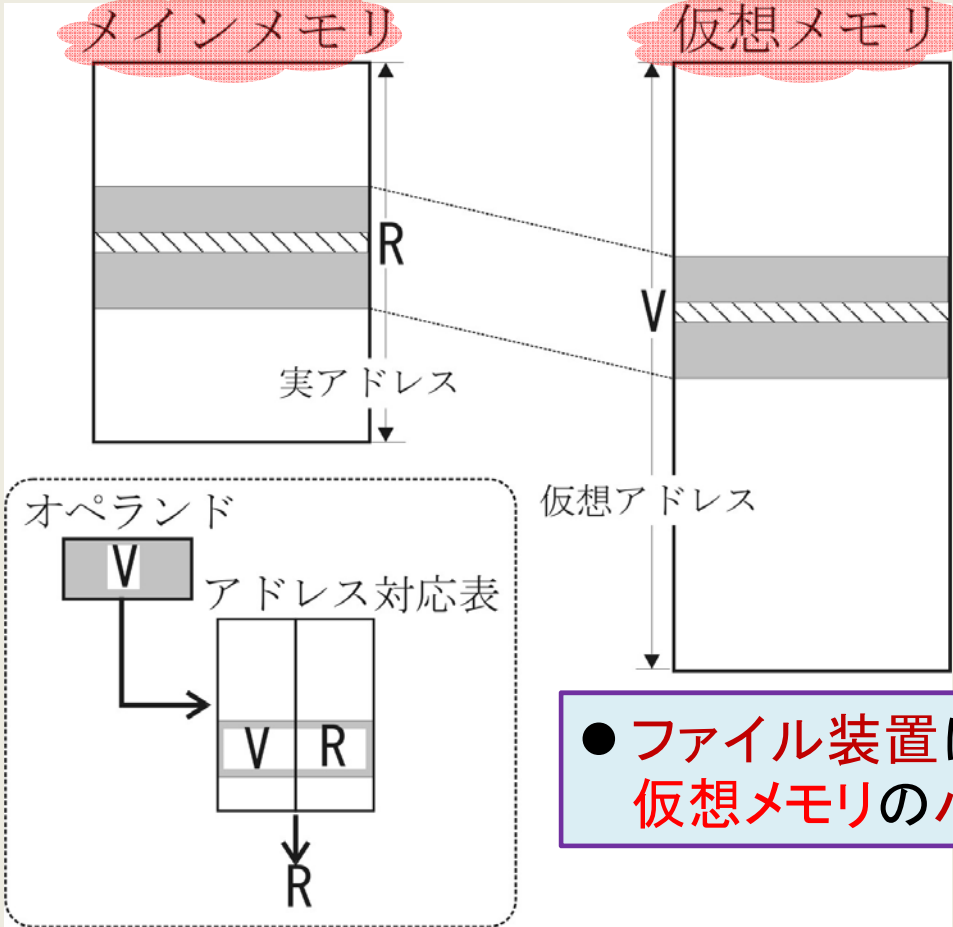
仮想メモリ機能の実現(3)

- **実アドレス**という「プロセッサから見る場合の**メインメモリのアドレス**」と**仮想アドレス**との対応関係を取り(実際には, 仮想アドレスと実アドレスとの対応表をメインメモリ上に置く), その**対応関係**をOSが管理
- **実アドレス**と**仮想アドレス**との対応関係は, **ブロック(block)**というある**一定**の長さの**連続**アドレス領域を単位

仮想メモリ機能の実現(4)

- **ブロック(block)**: 大きさを固定値にするかどうかで, 次の2種類に分類
 - (1) **ページ(page)**: 大きさをあらかじめ規定・**固定**
 - (2) **セグメント(segment)**: 大きさをプログラムとかデータに合わせて**変える**ことが可能
- **メインメモリ**には, **必要**とされる(←その判断もOSが行う)ブロックを置く

仮想メモリ方式(図)



● **ファイル装置はメイン(実)メモリと仮想メモリのバックアップメモリ**

仮想メモリ機能の動作(1)

- プロセッサが命令実行サイクルの命令取り出し、オペランド取り出し、結果格納の各ステージでメインメモリにアクセスする場合
 1. プログラムカウンタに設定するか、あるいはマシン命令からオペランドとして取り出された、仮想アドレスVによって対応関係をチェック(アドレス対応表を引く)

仮想メモリ機能の動作(2)

2. 対応する命令あるいはデータが

(a) 既にメインメモリにある → アドレス対応表によって実アドレス R が求まる(=アドレス変換)

➤ この実アドレス R でメインメモリにアクセス

仮想メモリ機能の動作(3)

2. 対応する命令あるいはデータが

(b) まだメインメモリにない(←アドレス対応表を引けば判明)

→ 仮想アドレスVを含むブロックをファイル装置からメインメモリへコピー

➤ メインメモリにある不要(=OSが「不要」と判定)なブロックをファイル装置に戻す必要 → メインメモリーファイル装置間でのブロック入れ替え

➤ この後, (a)を実行

仮想メモリ機能の実現(まとめ)

- メインメモリとファイル装置間でのアドレス対応表などのハードウェア機構の管理をOSというソフトウェア機能の支援によって実現
 - ハードウェアとソフトウェアの協調動作によるコンピュータシステムとしての高度な機能の実現例

メインメモリの高機能化(3)

■ 仮想メモリ(virtual memory)

- メインメモリをより大きく見せかける！
- **大容量化** → メインメモリよりも容量性能で優(大)の隣接メモリ階層(ファイル装置が代表的)を利用

■ キャッシュメモリ(cache memory)

- メインメモリ動作をより速く見せかける！
- **アクセス時間の高速化** → メインメモリよりアクセス時間性能で優のメモリ階層を新たに設定・利用
- レジスタ階層とメインメモリ階層間に**新規に**設定するメモリ階層

キャッシュメモリによるメインメモリの高速化(1)

- 「プログラムの格納」の外に、**メインメモリ**のもう1つの大きな仕事として、「**プロセッサ**によるプログラム(命令とデータ)への**アクセス**」
 - **プロセッサ**の要求に応じて**命令**の読み出しや**データ**のアクセス(読み出しと書き込み)を実現する必要
- **メインメモリ**は**容量**と**アクセス時間**との適度な**バランス**を保つ必要
 - プロセッサとメインメモリそれぞれの動作**速度**で**差** → 必然

キャッシュメモリによるメインメモリの高速化(2)



- プロセッサが要求する速度で、メインメモリ内の命令やデータにアクセスできない
 - 「プロセッサがメインメモリへのアクセスのために待たされる」という事態が多発

キャッシュメモリによるメインメモリの高速化(3)



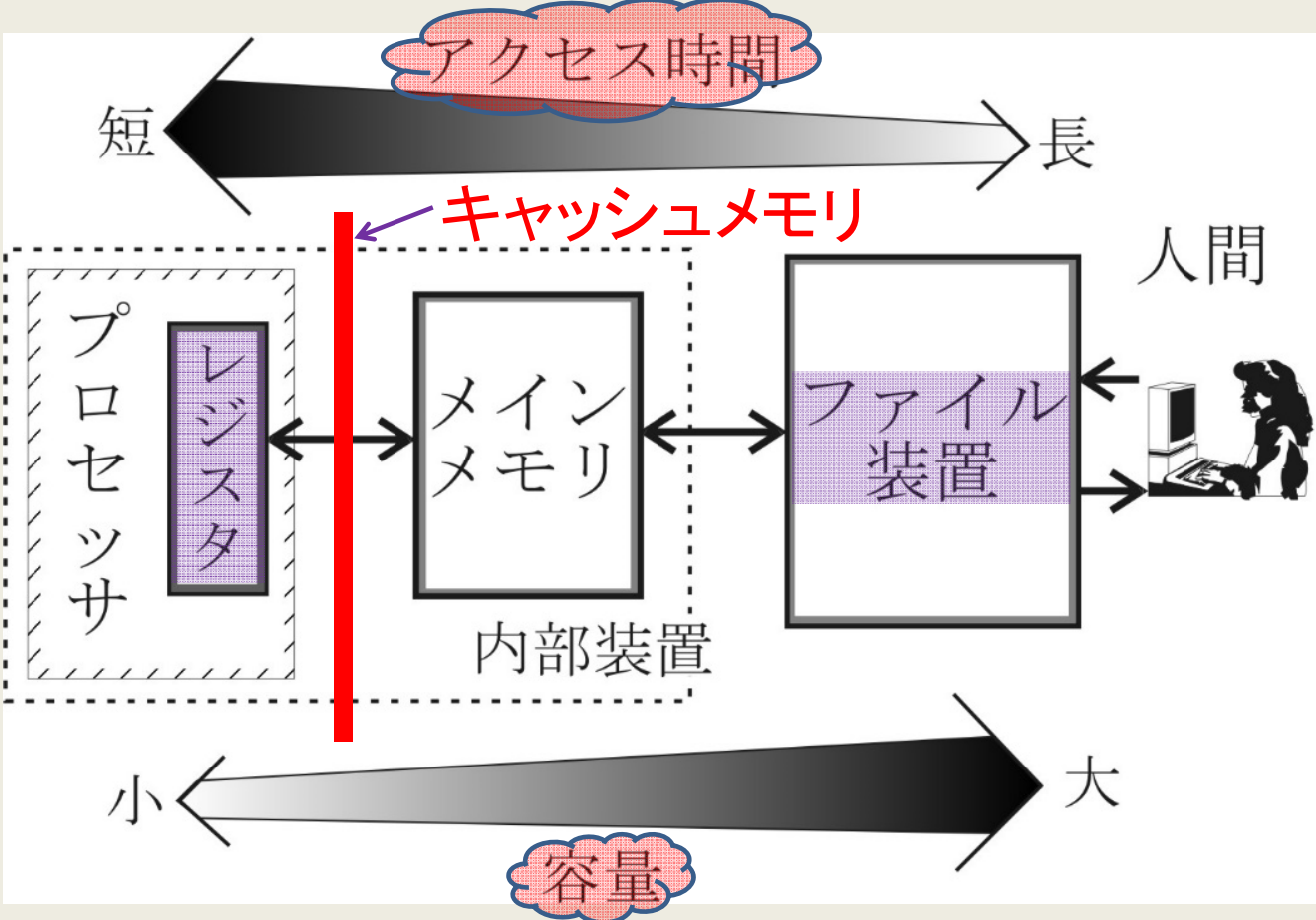
- **キャッシュメモリ方式, キャッシュ方式**:メインメモリへのアクセス時間の短縮
 - **メインメモリのもう1つの高機能化**
 - 「**キャッシュ(cache)**」=「隠し場所」, 「貯蔵所」という意味

キャッシュメモリ階層

- レジスタ階層とメインメモリ階層との中間に設ける新たなメモリ階層
- キャッシュメモリ: レジスタと比べるとアクセス時間は長だが容量は大, メインメモリと比べると容量では小だがアクセス時間は短
- この「メインメモリよりもアクセス時間が短」というメモリ性能の特長を活用

(再掲)

主要なメモリ階層間の関係(図)



キャッシュメモリの実際

- **キャッシュメモリ**は、レジスタとメインメモリとの**中間位置**(例：マイクロプロセッサのように、**プロセッサと同じIC上**)に実装
 - ◆ **キャッシュメモリ性能の最近の実例**：**容量**：数十キロ～十数メガバイト、**アクセス時間**：数ナノ～十数ナノ秒程度

キャッシュメモリ方式(1)

- キャッシュメモリとメインメモリとの2メモリ階層で対応関係
 - キャッシュメモリとメインメモリとを対応付け
- メインメモリにあるプログラムのうち「実際に今使っている」プログラムあるいはその一部をキャッシュメモリに置く
 - キャッシュメモリには、メインメモリの一部のコピー(copy; 写し)を置く

キャッシュメモリ方式(2)

- キャッシュメモリとメインメモリとの2メモリ階層を利用 → プロセッサからのメインメモリへのアクセス時間を実質的に向上

キャッシュメモリ機能の実現(1)

- キャッシュメモリとメインメモリとの間で、アドレスの対応関係をとる
- 対応関係は、ある一定の長さの連続アドレス(ブロック)を単位(=ライン(line))
 - ラインの大きさは、あらかじめ規定・固定
 - ラインの大きさは、仮想メモリのページよりもずっと小

キャッシュメモリ機能の実現(2)

- 「プロセッサがアクセスしようとする命令やデータが**キャッシュメモリ**にあるかどうか？」などの対応関係の管理や制御は「**ハードウェア機構**」が ← **キャッシュメモリ**は**メインメモリ**を**速く**見せかける機構

キャッシュメモリ機能の実現(3)

- **キャッシュメモリ**には, (1) 最も**頻繁**に使用; (2) **最近**アクセス; (3) 今実行(使用)しているプログラムに**引き続く**あるいは**近い**アドレスに置かれている; などの理由で**最もアクセスされ**そうな(その判定も**ハードウェア**によって行う)命令やデータを置く

キャッシュメモリの動作(1)

- 命令実行サイクルの命令取り出し、オペランド取り出しあるいは結果格納の各ステージで、プロセッサがメインメモリにアクセスする場合
1. プログラムカウンタや命令オペランドとしてあるメインメモリアドレスによってアクセスしようとした命令やデータのコピーがキャッシュメモリにあるかどうかをチェック

キャッシュメモリの動作(2)

2. 命令やデータのコピーが

(a) **キャッシュメモリにあり**(=ヒット(hit))

→ **それにアクセス**

➤ この場合の(メインメモリへの)**アクセス時間**は, 実質的に**キャッシュメモリへのアクセス時間**に

キャッシュメモリの動作(3)

2. 命令やデータのコピーが

(b) **キャッシュメモリ**にない(=**ヒットミス(hitmiss)**, **ミスヒット(mishit)**)

→ 原則として, 対象とするブロック(**ライン**)をメインメモリからキャッシュメモリに**コピー**, その後その命令あるいはデータに**アクセス**

キャッシュメモリの動作(4)

- ヒットミスの場合: キャッシュメモリー-メインメモリ間でのブロック(ライン)のコピー操作が必要に
 - キャッシュメモリによるメインメモリアクセスの高速化は実現不可

ヒット率

- アクセスしようとした命令またはデータがキャッシュメモリにある(ヒット)割合
 - キャッシュメモリ方式の効果を左右

■ 実質的メインメモリへのアクセス時間 T_E

$$T_E = R \times T_c + (1 - R) \times T_m \quad (5.1)$$

- 実際のメインメモリへのアクセス時間: T_m , 実際のキャッシュメモリへのアクセス時間: T_c , ヒット率: R の場合

キャッシュメモリ機能の実現(まとめ)

- キャッシュメモリ機構とメインメモリ装置間におけるアドレス対応表などの**ハードウェア機構**の管理を**ハードウェア**自身によって
 - **ハードウェア機構**によってキャッシュの制御や管理を行う意味 → 「キャッシュメモリ方式がメインメモリの**アクセス時間の高速化(短縮)**」という**時間的な性能改善**を狙う

例題5.1: メインメモリへのアクセス時間 T_m が100ナノ秒, キャッシュメモリへのアクセス時間 T_c が5ナノ秒であるコンピュータで, ヒット率 R ($0 \leq R \leq 1$) が 99%, 95%, 90%, 80% のとき, メインメモリへの実質的なアクセス時間 T_E はいくらになるか?

[解答]

メインメモリへの実質的なアクセス時間 T_E は, 次式5.1

$$T_E = R \times T_c + (1 - R) \times T_m \quad (5.1)$$

[解答](続き)

$T_m = 100$, $T_c = 5$ で, R に 0.99, 0.95, 0.9, 0.8 を順に代入して, T_E を求めると,

$$T_{E99} = 0.99 \times 5 + 0.01 \times 100 = \underline{5.95}$$

$$T_{E95} = 0.95 \times 5 + 0.05 \times 100 = 9.75$$

$$T_{E90} = 0.9 \times 5 + 0.1 \times 100 = 14.5$$

$$T_{E80} = 0.8 \times 5 + 0.2 \times 100 = \underline{24}$$

T_{E99} は T_{E80} の約4分の1