

1.7 CPU ユニットとその内部

- コンピュータの重要な部品である
- 一般的には高価な部品
- (現在では) コンピュータの種類に関係する

CPU の内部動作について話題・・・

- コンピュータ概論 A で学習した 5 大装置の「 」と「 」

CPU の性能はどうやって決まる？

- 1 秒間に実行できる命令の数は？
- 処理を効率よく実行するには？



- 1) 一連の処理に必要な命令数
- 2) 1 命令当たりの実行に必要なクロック数 (CPI : Clock cycles Per Instruction)
- 3) 動作周波数(クロック周波数)



$$1 \text{ 秒間に実行できる命令数} = \text{動作周波数} / \text{CPI}$$

これら 3 つの性能の積で決まる

- さらに、複数の処理を一度に実行する多機能命令があれば所要命令数は減る
最近の拡張命令 (MMX, SSE 他) はこれに該当

1) 一連の処理に必要な命令数

制御ユニット

ワイヤード・ロジック

命令の実行制御に必要な論理回路を直接順序回路を用いて実行する。 速い。

マイクロコード

マイクロプロセッサの命令の動作を記述するプログラム

命令セットを簡単に拡張できるが実行速度は遅くなる

→ **ワイヤードロジック化**

2) CPI の改善

→ 1 命令当たりの実行に必要なクロック数

→ **パイプライン**

ベルト・コンベアのような流れ作業（命令をオーバーラップして実行）

CPU の実行過程をみると次の 5 つのステージで処理が完了する

命令を読み込む (Instruction Fetch)

命令コードの解釈 (Instruction Decode)

命令の実行 (Execution)

メモリーやレジスタへのアクセス (Memory Access)

結果をメモリーやレジスタへの書き込み (Write Back)

仮に各ステージが 1 クロックなら全部で 5 クロック必要 (= CPI は 5)

パイプライン化された CPU では各ステージの処理は別々の機能ユニットが担当する

注意)

パイプラインを採用しても実際に命令を発行してから結果が得られるまでの実行時間 (**レイテンシ**) が短くなるわけではない

しかし、単位時間内に結果が得られる命令数 (スループット) は増える

3) 動作周波数の高速化

CPU の最高動作周波数は回路の最小線幅 (製造プロセス) によって決まる

最小線幅が 1 / 2 になれば動作周波数もほぼ 2 倍

製造プロセス技術 (プロセッサの実装技術)

→ 1 チップに集積できる機能や性能に影響する

ムーア (Moore) の法則 (1975 年), Gordon Moore

マイクロプロセッサに集積するトランジスタの数は年率 40%、**2 年で約 2 倍**になっていく

・・・今までのところ守られている (達成されている)

→ でも無限にというのは無理! 今後の方向は?

微細になることのメリット

- プロセッサが小さくなる
- プロセッサの機能を豊富にする
- コストを下げる
- 動作周波数上がる・・・遅延が短くなる
- 消費電力が下がる・・・チャンネルが細くなる ← 別の問題もある



CPUの実装命令

命令セットの拡充は、CPU性能向上要因と並列が難しい

→ 複雑になる → 開発コストがかさむ

豊富だが複雑な命令セットを備えている

→ **CISC** Complex Instruction Set Computer

頻繁に使用される命令だけに絞りそれを高速に実行する

→ **RISC** Reduced Instruction Set Computer

アーキテクチャが簡単のため、高い動作周波数での動作が可能

RISC では・・・ CISCで1命令で実行していた複雑な処理は、
コンパイラが複数の単純な処理に置き換える

→ プログラム設計

メモ：RISCとCISCのすみわけと混在

80年代の後半にはRISCとCISCの優劣の議論が盛んに行われた

現在、両者の違いは意外なほどに小さい

パレートの法則!?

頻繁に使用されるのはごく一部の命令であって、

「プログラムの全実行時間の90%は、その命令セットの10%の実行に費やす」

(CISCへのアンチテーゼ)

→ ほとんどの(90%)命令はあまり使われない!