ES：０２_CPU / メモリアーキテクチャ

割込み制御

- 概要と種類
  - 概要
    - 割込みによって実行されたプログラム
    - 実行中のプログラムを一時的に止めて別の処理を優先的に行わせるために利用される。
      - 割込みサービスルーチン（Interrupt Service Routine：ISR）／割込みハンドラ（Interrupt Handler）
      - 割込みベクタテーブル
      - 割込み発生より、レジストルの退避や復旧など、割込みを実行するために必要とな全処理や後処理などの処理が必要となる。これらの処理及びこれらにかかる時間を割込みオーバヘッド あるという
      - 割込みオーバヘッド
  - 種類
    - 外部割込み
      - CPUの周辺装置から要因が発生する割込み、代表例は入出力デバイス準備および、状態変更など
    - 内部割込み
      - CPU自体が発生させる割込み
        - 0除算
        - オーバフロー
        - 書き込み禁止領域への書込み
        - 割込み命令による割込み
    - NMI（Non-Mashable Interrupt）
      - ソフトウェアによって割込みの禁止をすることができない割込み。システムに致命的な問題が起きた場合に発生する割込みである
        - 例
          - ウォッチドッグタイマがタイムアウトし、システムをリセットする処理を行う場合
          - 電源電圧の低下によりシステム誤動作が懸念されるため、シャットダウン処理を行う場合
  - 割込みの優先順位と多重割込み
    - 多重割込みを許可しないシステム
    - 多重割込みを許可するシステム
      - リアルシステムなど
  - 割込みコントローラの接続方式
    - 割込みコントローラ
      - 割込みコントローラの割込み入力端子に別の割込みコントローラの出力を接続する方式である、一般に、CPUからの割込み応答信号ACKは共通に接続される
      - カスケード方式
      - 割込み信号線を共通に接続する方式、CPUからの割込み応答信号ACKは上位の割込みコントローラを介して下位の割込みコントローラに伝えられる
      - ディジーチェーン方式
  - 扱う外部事象のうち、割り込みで対応するものとポーリングで対応するものをそれぞれの要求特性から区分し、処理ｏ負荷のバランスを考慮しながら正しく使い分けることが重要
    - ポーリング方式では、事象の発生がポーリング間隔内複数発生の場合、取りこぼしが発生する可能。割込み方式では個々対応が必要ので遅くなる可能
    - 外部環境の変化を検知する方法の割込みとポーリング

メインメモリの制御

- エラー検出と訂正
  - ECCメモリ（Error Check and Correct Memory）
    - 暗夫最編码
      - ハミング符号
    - 奇偶校验
      - パリティ方式
- 大容量化技術
  - 概念図
    - CPUが直接アクセスできるアドレス空間に、使用するメモリの総容量が収まらない場合に用いられる
    - 同一アドレス空間に複数の実メモリを配置し、これらを切り替えて使用する方法である
    - バンクメモリ方式
  - 概念図
    - プログラム中では主にオフセットアドレスを使用することにより、アドレス指定を短くできるほか、プログラムをリロケータブル（relocatable：再配置可能）にすることができる
    - 実もメモリ上の絶対位置を示すベースアドレスと、ベースアドレスからの相対位置を示すオフセットアドレスとを組み合わせてアドレスを指定する
    - セグメント方式
  - 概念図
    - ページング方式における論理アドレスと物理アドレスの変換は、MMU（Memory Management Unit：メモリ管理ユニット）と呼ばれるハードウェアによって行われる
    - セグメント方式と類似が、主記憶をページと呼ばれる固定長の区画に区切り、プログラムからアクセスされる論理ページと実メモリ上の物理ページのたいおうを管理する
    - ページング方式
  - 主記憶体の容量を超える大きさのプログラムを実行する際に、プログラムをセグメントと呼ばれる単位に分割し、実行に必要なセグメントのみ外部記憶から主記憶にロードして実行する方式である
    - オーバレイ方式
  - 仮想記憶
    - 主記憶と外部記憶（ディスク装置など）を一元的にアドレス付けし、主記憶の物理容量を超えるメモリ空間を利用できるようにする技術である
    - ページイン
    - ページアウト
    - スラッシング
      - ページフォールトが頻繁に発生してシステムの処理能力を急激に低下する現象
    - ページ置換えアルゴリズム
      - LFU（Least Frequently Used）
      - LRU（Least Recently Used）
      - FIFO（First In First Out）
      - LIFO（Last In First Out）
- 高速化技術
  - 概念図
    - 主記憶を複数のバンクに分け、並列的にアクセスすることでアクセスを高速化する技術である
    - メモリインタリーブ（interleave）
  - 例：リング方式ーメモリのブロック単位で階層的に保護レベルを設定し、実行中のプログラムが属する階層より高い階層へのアクセスを禁止する方式
    - メモリ上のデータを誤った操作から保護する仕組み
      - 記憶保護
    - MMUのアドレス変換より記憶保護機能にも持っている

キャッシュメモリ

- 概念図
  - 上位のキャッシュに格納されているデータは全て下位のキャッシュにも格納されているように制御する動作方式
    - インクルージョンキャッシュ（Inclusion Cache）
    - 上位のキャッシュにデータを新たに格納した時、そこから追い出されたデータを下位キャッシュに格納する動作方式
      - ビクティムキャッシュ（Victim Cache）
    - 概念図
      - CPU → キャッシュ（2） → メインメモリ（16） → ハードディスク（1.6）
      - 速度の差よりのボトルネックを減らすため
- 効果評価
  - 主記憶のメモリ空間全域にわたって均等にアクセスが発生するような場面に不向き
    - ヒット率
      - キャッシュヒット
    - キャッシュミス
      - ミスヒット率
      - ミスペナルティ（penalty）
      - キャッシュミスに起因するCPUの停止時間
  - ヒット率×キャッシュアクセス時間＋ミスヒット率×主記憶アクセス時間
    - 平均アクセス時間
- 制御方式
  - データをキャッシュメモリに更新する際に、キャッシュと主記憶両方とも更新する方式
    - ライトスルー方式
  - キャッシュ上にデータがある場合には、キャッシュだけに書込み、主記憶へはキャッシュのページの入り替えなど、必要となった時に書き込む方式である
    - ライトバック方式
  - 主記憶へのアクセス回数が少なくてすむが、キャッシュメモリと主記憶の不一致が発生することがある、DMAで主記憶をアクセスする際に、DMAがCPUと違う、古いのデータが読み出されてしまう
    - 対策
      - コヒーレンシ（一貫性：coherency）問題
      - キャッシュとして保有しているメモリブロックを主記憶上で管理する方式である。メモリやバスを共有しないマルチプロセッサシステムで使用される
        - ディレクトリ方式
      - バス上でやりとりされる全てのデータを監視する、問題がある可能な領域へのアクセスを検出して一時停止させ、その後キャッシュ上のデータを主記憶にかきだしてから読出しを再開する
        - バススヌープ（bus snoop）
- データ構造
  - 通常、32バイトや64バイトと言ったまとまった単位（キャッシュライン）で主記憶と対応させる
    - ダイレクトマッピング方式
      - 主記憶体の複数のアドレス範囲で一個のキャッシュラインを共有する方式である。個々のキャッシュラインが対応するアドレス範囲をエントリという
      - エントリに対応する別のアドレス範囲にあるメモリが交互にCPUからアクセスされた場合、そのキャッシュラインに対するデータの入り込えが頻繁に発生、効率が逆に低下されたしまう可能
    - セットアソシアティブ（associative）方式
      - 同一エントリに対して2個以上のキャッシュラインを割り当てる方式である。一個のエントリに対して割り当てるキャッシュラインの数をウェイ数という
      - ウェイ数より交互使用の数が少ない場合、ダイレクトマッピングの類似問題が起こりにくい
    - フルアソシアティブ方式
      - キャッシュラインと主記憶上のアドレスの対応についての制約を設けず、任意の領記憶領域ブロックに対して任意のキャッシュラインを割り当てるられるようにした方式である
      - 自由度が高い、キャッシュ・スラッシングは最も起こりにくいが、構造は複雑になる

メモリアーキテクチャ

- 内蔵メモリ
  - CPUが搭載するシステムLSIチップ上又は同一パッケージうちに設けられ、CPUからはメモリ空間の一部として認識されるメモリ
  - バス経由ではなく、CPU並みの高速、レジストリなど
- DMA
  - CPUがDMACに対し、転送先のアドレスを指定した後、転送開始の指示を行うと、で一たの転送が開始される。DMA転送中はDMAがバスマスタとして機能するため、CPUからメリへのアクセスは制限を受ける
    - DMACコントローラ（DMAC）
  - 一回のDMA転送要求に対して、1バイト（またはワード）の転送を行う方式
    - シングル転送モード
  - 指定されたバイト数のDMA転送を連続して行う。転送中はDMACがバスを占有し、指定したバイト数の転送が終わるまでバス占有権をCPUに返却しない
    - バースト転送モード
  - バースト転送モードとの区別点は、途中でDMA要求が取り下げより転送中断可能、CPUがバス利用後、DMAに指示して再開可能です
    - デマンド転送モード
- 特殊メモリ
  - FIFOメモリ
    - 入出力インタフェースを2系統もち、いずれの側からもデータの読み書きができるメリである。ハードウェアレベルで同時アクセスの矛盾を発生させないように制御している
    - デュアルポートメモリ（Dual Port RAM：DPRAM）
    - 概要
      - 複数アドレスに渡る大きなデータを共有する場合には、データの整合性に注意する必要がある
        - 一番安全だが、遅い
        - 方式1：書き込み中はすべてのアクセスを禁止
        - 方式2：書き込み中のアドレス以外からの読み出しは許可
      - 最も性能が良いが、データ整合性を注意要
        - 方式3：書き込み中のアドレスからの読出しも許可
  - 画面表示のためのCPUリソースがかかるが、画面表示用のメモリを別途用意する必要がないため、コスト、消費電力、小型化の面でメリットがある（グラフィックカード（显卡）など）
    - 主記憶の一部をCPUだけではなく画面表示用のフレームバッファとしてなど、ほかのデバイスも共有して使うメモリの使い方
    - ユニファイドメモリ方式（Unified Memory Architecture：UMA）

ESの基本構成

- 入力装置n→1制御部(MPU(ROM,RAM))1→n(表示装置、外部機器…)
- ハーバードアーキテクチャ
  - プログラム格納用メモリのバスとデータ格納用メモリのバスを独立に持つことで、いのマンボトルネックの影響を低減することを狙ったアーキテクチャである
    - ハーバーアーキテクチャ構成図
      - プログラムメモリ
      - データメモリ
      - CPU
  - DSPの一部には、データメモリ上のデータ同士のえんざんを連続して実行するのを効率的に行うため、データメモリを2系統としたものもある
    - メモリを2系統にしたハーバーアーキテクチャ構成図
      - プログラムメモリ
      - データメモリ
      - CPU
- ノイマン型アーキテクチャ／プリンストンアーキテクチャ
  - データとプログラムを同一のメモリバス上に配置する構成
  - 構成図
    - プログラムメモリ
    - データメモリ
    - CPU
- 用語
  - ノイマン型
    - メモリから順次プログラムを読み出して実行する方式（John von Neumann）
  - ノイマンボトルネック
    - ノイマン型アーキテクチャでは、プログラムはメモリから読み出されるため、プロセスと記憶装置との間のデータ転送能力が処理性能上の制約となっている
  - 非ノイマン型
    - データフローマシンやニューロコンピュータ、バイオコンピュータなどのアーキテクチャ
  - DSP（Digital Signal Processor）
    - 積和演算をリアルタイムで高速に実行できるプロセッサ。音声や画像などの信号処理、モータの制御などに使用される
- IOアクセス方式
  - メモリマップドI／O方式
    - 入出力装置とメモリを特に区別せず、両者に対して共通のアクセス信号を用いるという概念である
      - CPUキャッシュの有効範囲?
        - コンパイラによる最適化を抑止するため、volatile型修飾子をつけて宣言した変数へのポインタとしてアドレスを指定してアクセスする
  - I／OマップドI／O方式
    - メモリ空間とは別にI／Oアドレス空間を設け、それぞれに対して別のアクセス信号を用いるという概念である
  - アドレスバス・データバスの共通化
    - 現実してもでは、アドレスバスとデータバスを同一の信号線として時分割で利用する場合がある
      - ALE（Address Latch Enable）
        - 共有バスがアドレスを示しているときはALE信号が出力左れ、外部ではこの信号を用いてアドレスをラッチすることができる

並列処理方式

- SISD（Single Instructionn Single Data Stream）
  - パイプライン手法による内部で処理並列かされていても、命令とデータストリームがそれぞれ一つであればSISDに分類され
- SIMD（Single Instructionn Multiple Data Stream）
  - 画像処理など、おなじしょりを複数のデータに対して行う場合多く採用されている
- MISD
  - 信頼性を上げるための冗長性を確保するのに適するときれるが、実施例は多くない
- MIMD
  - 近年のパソコンのようにマルチコアCPUによって複数のタスクをそれぞれのコアで実行する例がこれに相当する

CPUアーキテクチャ

- パイプライン制御
  - 図：F：フェッチ、D：デコード、E：実行、M：メモリ、W：ライトバック
  - プロセッサの処理を、フェッチ、データコード、実行、メモリなど複数の処理ステージに細分化し、複数の命令を並列に実行することを特徴とする高速化手法である
    - 命令全体の開始から終了までに要する時間は、一般的に、（D+I−1）xP秒で表すことができる
  - 阻害要因
    - パイプラインバブル
      - 分岐命令より、パイプライン上残っている実行中の後継命令を破棄しなければならない
      - 避ける方法：遅延分岐
        - 後続の命令を無条件に実行した後に、実際の分岐が発生する。分岐命令が直前の命令の実行結果に依存しない場合、分岐命令を先に実行し、直前の命令を後で実行すること
    - パイプラインハザード
      - データハザード
        - 直前の命令の実行結果を、次の命令が使用する場合、次の命令は直前の命令の結果が受け取れるようになるまで待たなければならない
      - 構造ハザード
        - 複数の命令スロットが、一つしかないCPU内部の資源を同時に利用しようとした場面
      - 制御ハザード
        - 分岐命令によるハザードである。分岐の次にフェッチすべき命令のアドレスが確定するタイミングが遅くなる
- スーパパイプライン
  - 前述したパイプラインの処理ステージをさらに細分化したものである
    - クロックとパイプラインの深さを上げることによってスループットを上げることができるが、分岐命令が発生する場合のダメージが大きい
- スーパスカラ
  - パイプラインを複数用意し、同時に複数の命令を実行できるようにしたものである
- 高速化アーキテクチャ
  - 用語
    - アウトオブオーダー
      - 処理実行の順番を入れ替えることで高速化が可能な場合を検出し、実行の順番をCPUの処理に最適なように並べ替えて実行すること
    - 分岐予測
      - 投機的実行、その結果をリネームレジスタに書き込みなどで、分岐予測が外れた場合に元の状態を復元できるようにする仕組みが必要である
    - ポラックの法則（経験則）
      - CPUの処理能力は面積の平方根に比例する。すなわち、面積を2倍にしても処理能力には約1.4倍にしかならない
  - RISC：Reduced Instruction Set Computer
    - 「命令セットを細小して」設計されたコンピュータ（プロセッサ）である
      - パイプラインの利用率を高めいる
      - 命令の機能が少ない分、コンパイル後のコードサイズは大きくなる傾向がある
  - CISC：Complex Instruction Set Computer
    - 高機能な命令を多く揃えるプロセッサ設計
      - 現代のCISCプロセッサでは、内部にRISC的な構成要素を取り入れている場合もあるので、一概にCISC、RISC型といった分類をする意味は薄れてきている
  - VLIW：Very Long Instruction Word
    - 一つの命令語の中に複数の命令を構成し、これらを同時に（出来るだけ、できない部分はNOP）実行する方式である
      - 命令同士の依存関係を解析して最適化し、再配置するためには、コンパイラにる最適化技術が必須である。ディジタルテレビ受像器に用いられるメディアプロセッサなどの用途がある
  - マルチコア／メニューコア
    - CPU単体を高機能化して処理能力を高める代わりに、CPUコア自体を複数個集積し、処理の並列化を行う高速化手法である
      - ホモジニアス（homogeneous）・マルチコア
        - 同一のCPUを複数個搭載するもの
      - ヘテロジニアス（Heterogeneous）・マルチコア
        - 異なる種類のCPUを組み合わせて搭載するもの
  - コンフィギャラブル・プロセッサ
    - CPU単体を高機能化して処理能力を高める代わりに、CPUコア自体を複数個集積し、処理の並列化を行う高速化手法である
    - 基本的な機能を揃えたCPUコアに、ユーザがソフトウェアによって命令を追加できるプロセッサである。追加する命令の内容を動的に変更することも可能である。複雑な繰り返し演算が要求される画像信号処理分野や高速通信機器分野などの用途がある

DSP

- 積和演算を高速に実行でき、簡単な条件分岐命令なども揃えたプロセッサ
  - 積和演算器：乗算と加算を同時に行うことができる
    - 積和演算はデータと同じ数のクロックで実行できるため、同じ処理をCPU上のソフトウェアで実行する場合と比較してはるかに高速である（同Hzの場合）
  - ディジタル信号処理
    - 例：移動平均を用いた平滑化処理の中心点を含む5点の加重平均を行う。
      平均値＝D(t)x0.1+D(t−1)x0.2+D(t−2)x0.4+D(t−3)x0.2+D(t−4)x0.1
      - 5回の積和演算にかかる時間がサンプリングの間隔より短ければ、実時間での処理が可能である
      - 加重平均の処理を中心を移動されながら次々と行うことを移動平均という
- 分類
  - 固定小数点型
    - ユーザが小数点の位置を決める、その位置によって表現できる数値が異なる
    - 32ビットの小数点が最下位ビットの右側にある場合の範囲
      正の数：1〜2.147483647x10の9次
      負の数：−2.147483648x10の9次〜−1
  - 浮動小数点型
    - 32ビットのIEEE 754では10の38次、精度10

バスアーキテクチャ

- 基本構成要件
  - バスマスタ
    - バスに接続されている装置のうち、自らバスを制御データの転送を行う機能を有するもの
    - 例：CPUやDMZコントローラ
  - バススレーブ
    - バスに接続されている装置のうち、自らバスを制御機能を有しないもの、データ転送時にはバスマスタに対して受動的に振る舞う
    - 例：I／O
  - バスアービタ
    - 複数のバスマスタが接続されているバスにおいて、バスマスタ間でのバス使用権の調停を行う装置
    - 独立型とバスマスタが調停機能を持つ方式型がある
- システムバスの構成
  - 単一バス構成
    - 低速IOの影響でCPU待ちが多い可能です
  - 階層バス構成
    - アクセス速度の違いによって複数のバスを設ける構成である、比較的に低速なI／OとCPUを分離する
  - 複数バス構成
    - 用途ごとに独立したバスを用意し、並列に動作させるの構成。大量の配線が必要です。
    - マルチレイヤバス構成
    - スイッチネットワークバス構成
  - 用語
    - 低速な入出力機器にCPUの動作を合わせの手法
      - ウェイト機能
        - アクセス時に挿入するダミーサイクルの数を設定することで、特定のデバイスに対するアクセス速度を遅くする機能である
      - レディ機能
        - 外部デバイスがCPUに対してその実行中のアクセスサイクルを延長するように指示する信号線（レディ信号）を用いてCPUのアクセスを遅くする機能
  - マルチプロセッサ
    - 密結合構成
      - 複数のプロセッサが同じメモリにアクセスしてデータを共有する方式である（このうち、複数のプロセッサが全てのメモリ空間を共有する方式は共有メモリ方式と呼ばれる）
      - 基本構成図
      - 各プロセッサがキャッシュメモリを持つ構成
      - デュアルポートメモリによる一部共有
    - 疎結合構成
      - 各プロセッサが専用のメモリを持ち、プロセッサ間でのデータ共有は通信によって行う方式である
      - 構成例
    - 用語
      - デュアルポートメモリ（Dual-ported RAM：DPRAM）
        - 一度に一つだけのアクセスを許可するシングルポートRAMとは異なり、複数の読み取りや書き込みを同時またはほぼ同時に行うことができる。
        - ビデオRAMまたはVRAMはデュアルポートDynamic Random Access Memoryの一般的な形態である。主にビデオメモリに使用され、CPUはビデオハードウェアが画面に読み出すと同時に、画像を描画することを可能にする。
      - 分散メモリ方式
        - 独立したコンピュータをネットワークで接続し、メッセージを送信することによって他のプロセッサのメモリにアクセスする方法式

XMind Trial Mode