

楽しい
MMU

MMUの機能

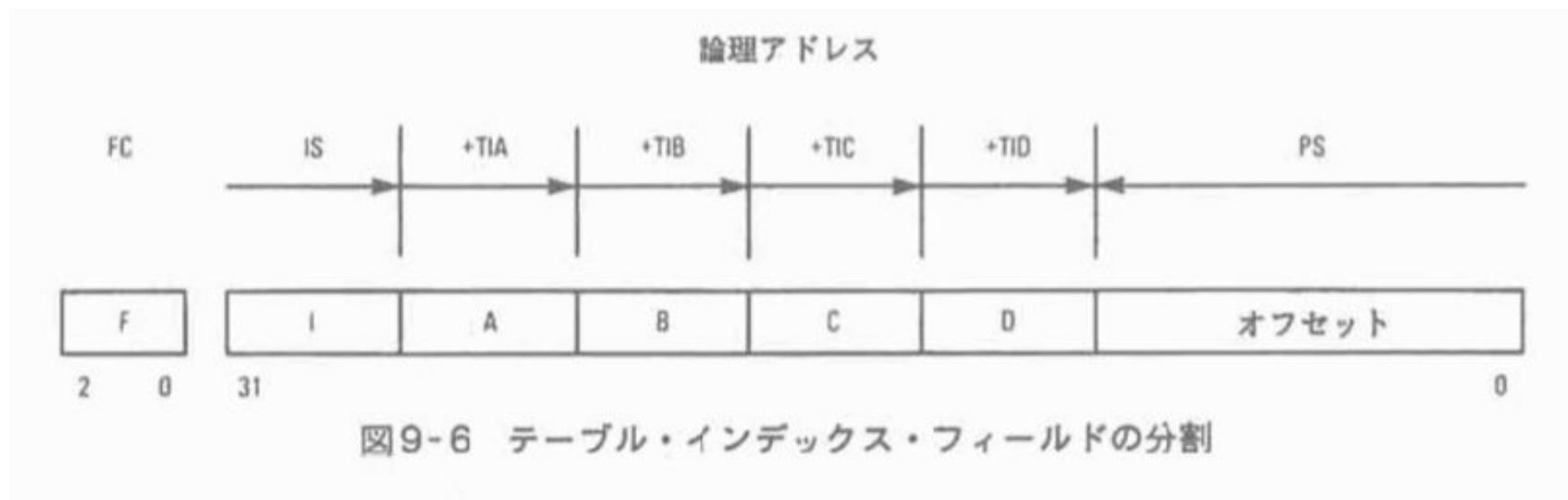
- 論理アドレスを物理アドレスに変換する
 - アドレスの範囲検査、アクセス権限の検査
 - メモリ領域の属性(キャッシュ禁止など)を出力
-
- TLB (アドレス変換キャッシュ)の管理
 - ページテーブル探索、管理 (ものによる)
 - MMU関連の特権命令の実行

68030のMMU

- 論理アドレスは35bit
 - 3-bit Function Code (SV Inst, SV data, Usr Inst, ...)
 - 32-bit 線形アドレス空間
- pagetableの引き方は複雑にプログラム可能
 - 最大6回ポインタチェース
 - ページテーブルは最大5階層
 - ページサイズは256 bytes ~ 32 kbytes

68030のページ探索構造

- 4段階のページオフセットのサイズを自由に制御可能
 - FC Lookupは有効・無効を切り替え可能
- 設定時に $IS+TIA+TIB+TIC+TID+PS=32$ をチェックし、違えば割り込みを発生させる



ページテーブルの構造

- ページディスクリプタは5種類!
 - 無効 - バスエラー発生させれば良い
 - アーリーターミネーション - ある意味largepage
 - タイプ2 - 4-byte、リミットチェックなし
 - タイプ3 - 8-byte、リミットチェックする
 - インダイレクション - ページテーブルの最下層でディスクリプタを共有するための、別ディスクリプタへのポインタ

68030のMMU命令

- CPU SPACEにマップされたコプロセッサとして見える- FPUと同様の扱い
 - PMOVE - 制御レジスタ<->論理アドレスでデータを移動
 - PFLUSH - FC, アドレスマスクを指定して特定のTLB エントリを無効化
 - PTEST - pagetable searchを例外を記録しつつ、外に例外を出さず実行 事実上2つ目のページ探索機能
 - PLOAD - TLBにエントリを読む。意味不明だが、普通にメモリアクセスが来たと思って処理している

それでもシンプルになった

- 68852 (68020のMMU, 外付け) には...
 - TLBのエントリロック機能
 - PBcc (MMU状態レジスタのビットをみて分岐)
 - PTRAPcc (MMU状態レジスタのビットをみてtrap)
 - IOMMU的な機能 (FCが4-bitで、最上位はDMA)
- などの退廃的な機能が満載だった
- なお、68040では更に単純化されている模様

その他のMMU

- 大体は68030のMMUよりわかりやすい

x86

- real, protected, longそれぞれのモードで異なる管理メカニズムが動く
 - VM86などの面白いサブモードもあったりする
- realはsegmented, protectedはsegmented+paging
- ここからはprotected modeの話をします

x86: protected mode address translation

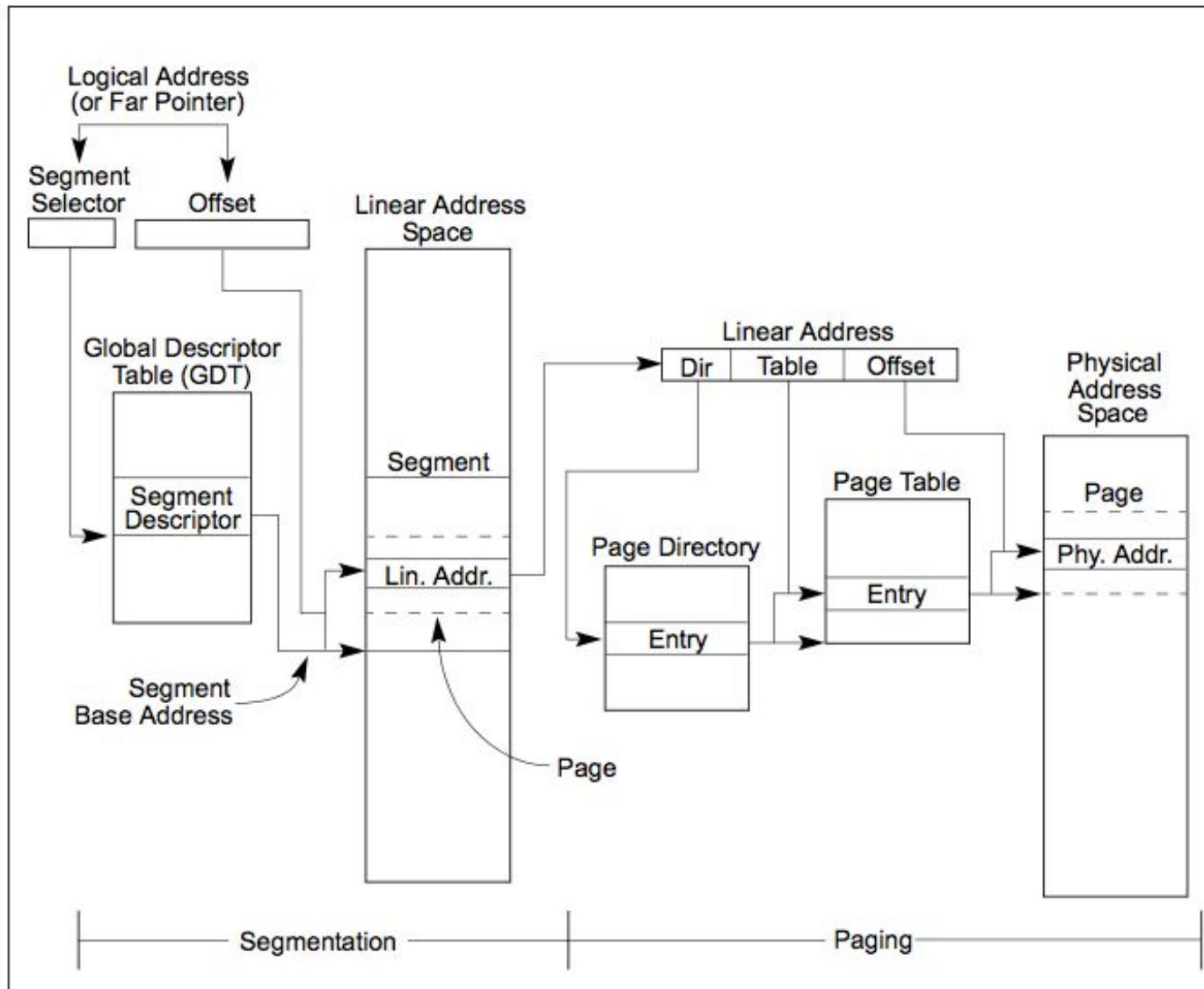


Figure 3-1. Segmentation and Paging

要点

- セグメントセレクタがGDT entryのidx
- segmented modelは検査と加算
- pagingは10-bit, 10-bitで固定
- ディスクリプタは4-byte固定

SH-4のページング

- 物理29bit
- TLBはタスクIDをセットしておける
- TLBはメモリにマップされていて、OSで管理
- 68030の場合もpagetable searchはバス律速なので、OSで管理が最適な気がする

結論

- ページ単位でtransactionを管理したい
- ページのverをTLBの外の物理メモリ上の構造に持つ
 - 後でハードウェアで嘗める
- 詳細不定