2007年度 いずみ研 新人課題

三角関数のハードウェアモジュールをつくれ

「コロンブスの卵」な単純さ

と

実用設計の難儀さ

正弦関数のモジュール化仕様

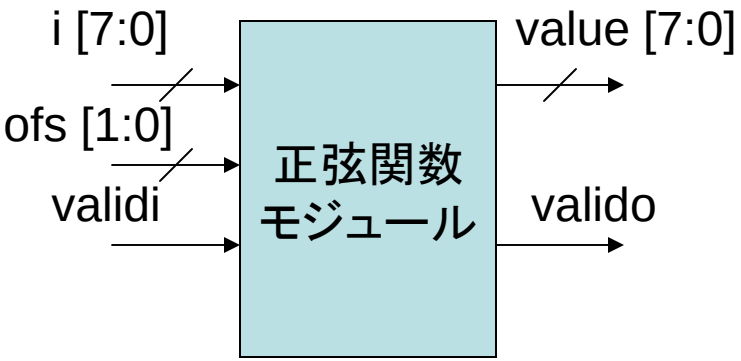
$$\mathrm{value} = \sin\left(\frac{2\pi}{2^n} \times \mathrm{i} + \frac{\pi}{2} \times \mathrm{ofs}\right)$$

- n ... ビット数
- i ... 角度（$0 \leqq \theta < 2\pi$を$0 \leqq i < 2^n$に正規化）
- ofs ... 角度のオフセット（π／2単位）

  (00: sin, 01: cos, 10: -sin, 11: -cos になる)
- value ... 正弦値

正弦関数ハードウェアモジュール インターフェース

- 入力側
  - （入力）角度 i (8bit)
  - （入力）オフセット ofs (2bit)
  - （入力）入力指示 validi
- 出力側
  - （出力）値 value
  - （出力）出力指示 valido



- モジュールの設計方法、インターフェースの考え方などについては泉の“演算機能回路”の講義資料、2007年度、第12回、第13回を参照されたし

表参照型演算器

- 複雑な関数を実現する「コロンブスの卵」的な単純な方法
- 入力値に対する出力値を、表として持っておく
- 与えられた x に対応する y を表から引いて出力する

表 → メモリで実現

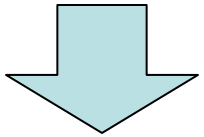
参照表

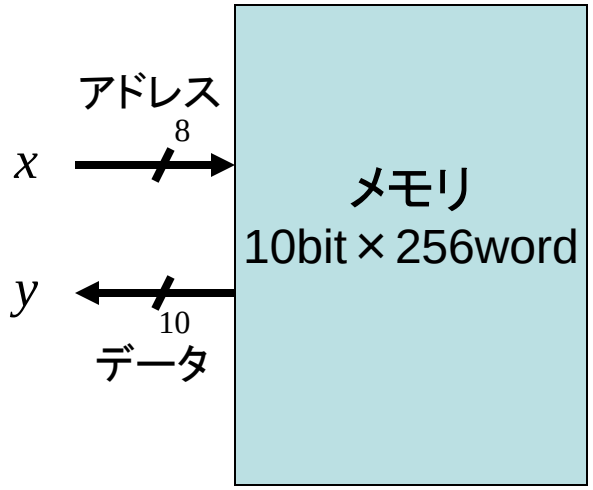
例: $y = 512 \sin(2\pi x / 128)$

x ... 8bit, y ... 10bit

- 入力値 x (n bit)
  → メモリのアドレス
- 出力値 y (m bit)
  → メモリのデータ

x	y
127	6
:	:
2	12
1	6
0	0
-1	-6
-2	-12
:	:
-127	6





y の値は別途計算して
事前に書き込んでおく

- $m \times 2^n$ bit のメモリ

表参照型演算器の構成

- 表を保持するメモリ（ROM）
- メモリを読み出すための制御回路
- 通常、メモリの読み書きに数サイクル必要
  - 組合せ回路による実現は不可
  - マルチサイクル演算
  - パイプライン演算（メモリがパイプライン的な読出可能であれば）

正弦関数の表

- 正弦関数の周期性、対称性を利用
  - 表を読むアドレスの正順と逆順
  - 結果の符号反転

  0.0π～0.5πの範囲のみ表に保持

  0.5π～1.0πは表を逆順に読む

  1.0π～1.5πは値をマイナスにする

  1.5π～2.0πは表を逆順に読んで値をマイナスにする
- オフセットを反映した角度 $i + 2^{n-2}ofs$ の上位2ビットで判定できる

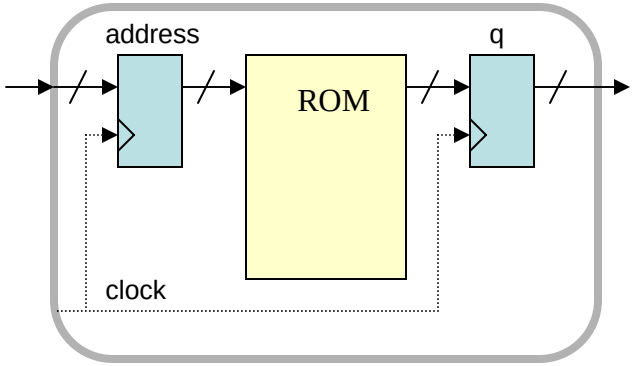
正弦関数の表

- 表中の正弦値は -1.0≦sin≦+1.0
- 固定小数点表現
  - 1ビット符号＋1ビット整数部＋残り小数部

    ↓効率アップ↓
- 整数部が1になるのはふたつだけ
  - ＋1.0 になるのは 0.5π (b01000...00) のとき
  - －1.0 になるのは 1.5π (b11000...00) のとき
- 符号は角度から容易に判定できる
- 小数部のみ格納し、絶対値が1.0になる場合と符号は別途処理

表メモリとパイプライン

- 表はメモリで実現
- メモリはそれほど高速ではない
- 複数サイクルで動作
- 読出結果が出る前に次の読出要求を出してよい
- パイプライン化が可能



Altera MegaCore ROM の例 (2cycles delay)

表引き／パイプライン型正弦関数