



MAX II 応用製作

周波数可変の正弦波発生器

上田 智章
Tomoaki Ueda

2006年4月号 付録基板

2006年4月号の付録CPLD基板に実装されているアルテラ社のMAX IIは、8192ビットのユーザ・フラッシュ・メモリと240個のロジック・エレメントが内蔵された高機能なCPLDです。ここでは、内蔵フラッシュ・メモリを活用できるアプリケーションとして、写真1に示すようなDDS(Direct Digital Synthesizer)を製作しました。DDSとは、任意周波数のサイン波を発生することができるデジタル型ファンクション・ジェネレータのことです。

本稿ではDDSの動作原理について説明を行ってから、今回製作したDDSの仕様と実験基板などについて解説します。表1に、製作したDDSの仕様を示します。

表1 製作したDDSの仕様

項目	仕様
チャンネル数	1チャンネル
出力波形	正弦波
サイン波形の定義	アドレス幅：9ビット(512アドレス) データ幅：16ビット
出力電圧	±10V
基本クロック	ユーザ・フラッシュ・メモリのオシレータ出力を使用(約5MHz)
最小時間分解能	基本クロックを32分周(156.25kHz)
累積加算器	演算データ幅：24ビット
DIPスイッチの設定	100000hのとき10kHzが出力される 最小周波数分解能：0.0095367Hz/ステップ

DDSの動作原理と回路方式

DDSの動作原理

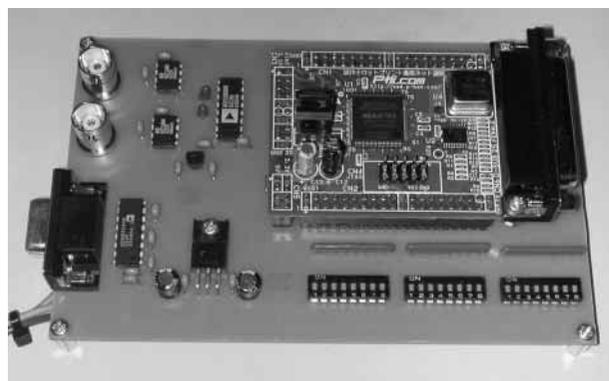
DDSの最も基本的な構成を図1に示します。

加算器の演算結果は位相レジスタで保持され、周波数指定レジスタの値を順次、累積加算する構成になっています。加算結果の上位ビットが正弦波変換ROMのアドレスとして供給されています。

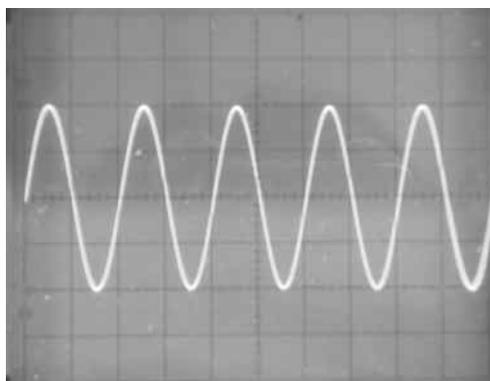
正弦波変換ROMのメモリ空間には、正弦波(サイン波)の波形1周期ぶんのデータが符号付き整数値で順次記録されています。今、この回路にクロックとして例えば16.777216MHzが供給されており、周波数指定レジスタに1が格納されているとすれば、24ビット($2^{24} = 16777216$)の累積加算結果は16777216クロックごと、すなわち1秒ごとにフル・カウントに達します。

また、ROMのアドレスも順次000hからFFFhまで増加します。つまり、1秒周期(1Hz)で正弦波が出力されるのです。もし、周波数指定レジスタに2が入っていたら、加算器は1秒間に2回フル・カウントに達し、2Hzの正弦波を出力するわけです。

このように、DDSの基本クロックを適切に選択(例えば2MHzなど)しておくことにより、周波数指定レジスタに設定された数値xに対応する周波数xHzで正弦波を出力することができます。



(a) 4月号付録CPLD基板を載せた試作基板



(b) 出力波形(5V/div., 5ms/div.)

写真1 製作したDDS基板と出力波形

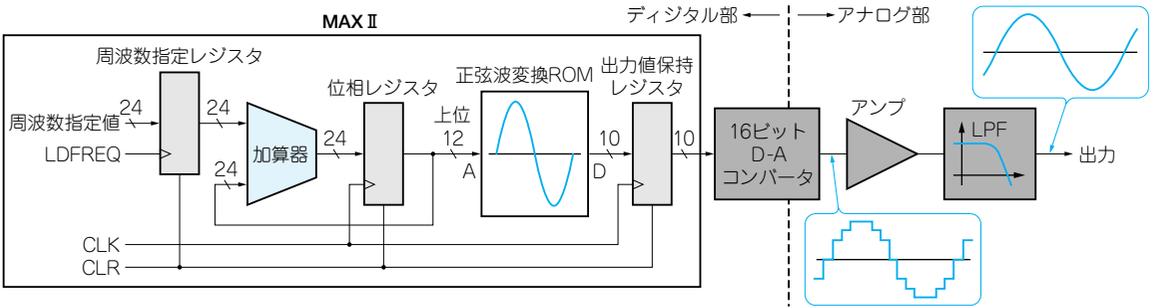


図1 DDSの基本構成

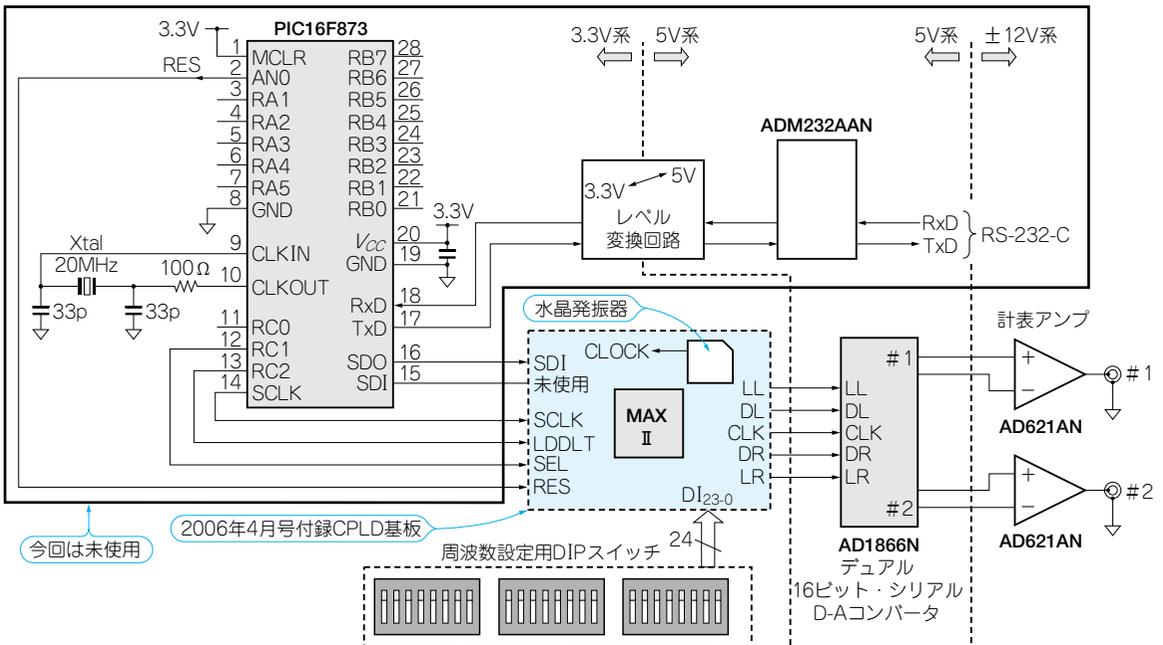


図2 製作したDDSジェネレータの全体構成

製作するDDSの構成と回路

● 全体の構成

回路全体の構成を図2に示します。

D-Aコンバータは、2チャンネル、シリアル16ビットD-AコンバータAD1866(アナログ・デバイス)を使用しており、 $\pm 10V$ の正弦波が出力できる設計にしています。実験用のDIPスイッチによる周波数設定が行えるパラレル・モードと、PICマイコンからSPI経由での周波数設定と位相シフトが行えるシリアル・モードが実現できる構成にしています。

そのほか、外部パソコンとのシリアル通信インターフェース(RS-232-C)も用意しています。ただし、今回は基本的な1チャンネルDDSとDIPスイッチによる周波数設定のみを実験することにします。

将来的には2チャンネル同期型DDSまで拡張していく予定で、外付け回路のプリント基板を製作していま

す。

● CPLDの内部ブロック

シンプルな1チャンネルDDSの基本回路をCPLDに実装します。図3にCPLDの内部ブロック図を示します。

クロックは付録基板上の水晶発振器は使わずに、MAX IIに内蔵されているフラッシュ・メモリのブロックから出力される信号を使っています。8192ビットのフラッシュ・メモリを、通常のROMに似た512アドレス入力、16ビット・データ出力のパラレル・インターフェースのROMとして使っています。

このフラッシュ・メモリは実際にはシリアルのフラッシュ・メモリを使っているらしく、リードに $6\mu s$ 弱を要しています。正弦波の対称性を活用すれば1/4波ぶんしか記録の必要はありませんが、設計をシンプルにするために、正弦波1波ぶんのデータを記録して