

データ駆動型プロセッサによる ソフトウェア A/D・D/A 変換器の実装

小糸 啓介[†] 通堂 真[‡] 岩田 誠[†] 酒居 敬一[†]
高知工科大学[†] 株式会社コアシステムズ[‡]

デジタル信号処理を行なっている製品はたくさんあり、それらはおおむね センサー・A/D 変換器・デジタル処理・D/A 変換器・アクチュエータから成っている。そして、高速動作や高精度動作するような機器を除けば、たいていのシステムはノイマン型プロセッサとソフトウェアで処理している。演算能力の向上に伴って従来ハードウェアで行っていた多くの処理をソフトウェアで実装してきているが、実際には A/D 変換器や D/A 変換器といったいくつかのモジュールはソフトウェア実装から取り残されている。これらをソフトウェアで実装しようとした時ノイマン型プロセッサのアーキテクチャに適した実装がないため、一般的にそのようなモジュールは専用回路として実装されているのが現状である。

本稿では非同期のデータ駆動型プロセッサとソフトウェアを用いたよりプロセッサのアーキテクチャに適した実装について述べる。そして、実験によりソフトウェア化 A/D・D/A 変換器の有効性を示す。

Software Implementations of ADC and DAC on Data-Driven Processor

Keisuke KOITO[†] Makoto TSUDO[‡] Makoto IWATA[†] Keiichi SAKAI[†]
Kochi University of Technology[†] Coa Systems[‡]

There are various electrical appliances adopting a digital signal processing. Such an appliance typically consists of a sensor, A/D converter, a digital processing unit, D/A converter and an actuator. And almost system except for fast and high precision system may adopt von Neumann processor with the software as a processing unit. Additionally, usual modules in the system have been implemented by the software along with the development of the processing power in fact. However, several modules such as A/D-D/A converter in the system remain out of the software implementation. Those remained modules have been implemented by dedicated circuit in usual, because there are no reasonable implementation on von Neumann processor.

This report describes the reasonable implementation using an asynchronous data-driven processor with the software. Then the effectiveness of the software-defined A/D-D/A converters are demonstrated by the experiment.

1 はじめに

近年のプロセッサの処理速度向上により、オーディオ機器や通信機器といった信号処理を必要とする機器では、たいていプロセッサとソフトウェアによるデジタル信号処理を採用している。このようなデジタル信号処理は高精度化や低コスト化、調整の自動化や高い安定度による無調整化という利点があり、システム実装をアナログ信号処理からデジタル信号処理へ、そしてハードウェア処理からソフトウェア処理へ置き換える流れの

原動力となっている。このとき重要になるのは、アナログ系とデジタル系とのインターフェース、つまり、A/D 変換器 (以下 ADC) と D/A 変換器 (以下 DAC) であり、多くの処理方式が考案され実装されている。

現状では、MCU (Micro Control Unit) というプロセッサコアおよび周辺機能を 1 チップ化したものを使用するにしても、この ADC-DAC は専用回路として存在している。そのため、ADC-DAC は処理する対象に応じて標準化周波数 (以下 f_s) や分

解能といった要求仕様ごとに、MCUの再検討やシステムの再設計が必要である。そして製造メーカーではさまざまな要求に応えるように、仕様の異なる内部モジュールを持ったMCUを多種類準備している。デジタル信号処理はノイマン型プロセッサ(CPUやDSP、以下CPU)とそのソフトウェアで処理されることが多いのにもかかわらず、ADCとDACそのものに関しては未だにソフトウェア化が進んでいるとは言いがたい。

そこで本研究では、デジタル信号処理デバイスとして非同期式データ駆動型プロセッサ [1](以下DDP)に着目し、そしてADCとDACについては $\Delta\Sigma$ 変換方式に着目することで、ソフトウェア化可能な部分を検討し、実験によりその可能性を示す。さらに実験の考察より、ADC-DACの f_s や分解能などの特性をパラメータ化し、これらを柔軟に変更できる可能性があることを述べる。

2 目標と背景

2.1 目標

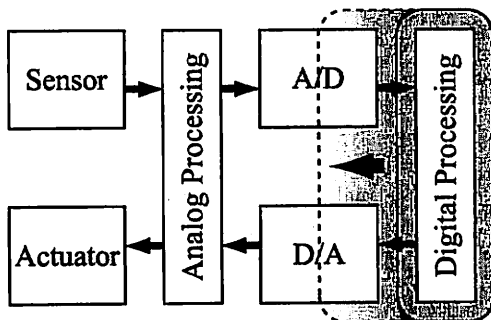


図 1: 信号処理システム

本稿では、図1のようなシステムを想定し、途中で枠で囲っているデジタルシステムの領域とその外側のアナログシステムの領域の境界を実線(従来)の位置からアナログ方向にシフトすることを目標とする。そこで本節では目標に関連することを簡潔に述べる。

2.2 さまざまな A/D 変換方式

フラッシュ方式 (全並列方式)

サンプル&ホールド回路(以下S&H)により標本化を行い、図2に示すように基準電圧を段階的に設定した比較器を並列に並べ、比較器の反応したレベルにより量子化を行うADCである。構造上高速動作に優れるが、分解能を上げるためには比較器数を増やす必要がある。高分解能が要求される用途には向かない。

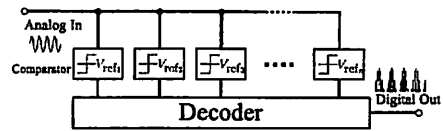


図 2: フラッシュ方式

逐次変換方式

S&Hにより標本化を行い、図3に示すように1つの比較器とレジスタから構成される回路により量子化を行うADCである。比較器の基準電圧を時間毎に段階的に変化させながらその出力をレジスタに記録する事で電圧レベルの量子化を行う。フラッシュ方式と比べて分解能を上げるのは容易である一方で量子化に時間を要するため f_s は下がる。

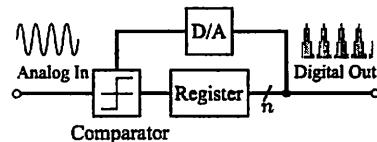


図 3: 逐次変換方式

積分方式

積分方式は図4に示すように積分器とクロックを使用したADCで、S&Hからの入力を積分し、積分値(電圧)が基準を超えるまでの時間を離散的に計測することで量子化する。一般的には直流オフセットによる変換誤差を打ち消すため、正方向の積分だけでなく負方向の積分も行う二重積分方式が用いられる。

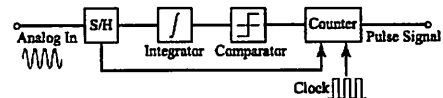


図 4: 積分方式

$\Delta\Sigma$ 方式

$\Delta\Sigma$ 変換方式は $\Delta\Sigma$ 変調器とデジタルフィルタから構成される。入力信号を $\Delta\Sigma$ 変調し、オーバーサンプリングにより1ビットのデジタル信号を得て、それをデシメーションにより所望の f_s と分解能を持つPCMデータを得るADCである。また、原理的にはDACにも用いることが可能である。本稿では、実装すべき回路が小さくて済むことから、ADC-DACとして本方式に着目している。

2.3 デジタル信号処理

信号処理には、アナログ信号処理やデジタル信号処理がある。前者では、専用回路による実装やFPAA[2]のような再構成可能な実装がある。後

者については、速度や精度に依存して次のような実装が考えられる。

1. デジタル回路によるハードウェア処理
いわゆる専用 LSI による実装であり、最高性能を得ることができるが、他と比較して設計や実装に多くの費用を必要とし、仕様変更や欠陥の修正も困難である。
2. FPGA のような再構成可能チップで処理
汎用 LSI より高価であるが、専用 LSI より安価でカスタマイズ可能という利点がある。さらに、プロセッサを使ったソフトウェア処理より速い処理をこなすことができる。
3. CPU もしくは DSP によるソフトウェア処理
利点が多いことから近年もっとも多く見られる処理形態である。しかし、高頻度で発生する細かい処理を並行して処理することは一般に困難である。2.4 節で詳しく述べる。
4. DDP によるソフトウェア処理
DDP でソフトウェア処理することについては、3 と同じ利点を持つ。それに加えて、とりわけ非同期 DDP では、互いに無関係に発生する細かな処理を並行してさばくことが可能である。2.5 節で詳しく述べる。

2.4 フォンノイマン型プロセッサ (CPU)

現在世の中で一般的に使用されている CPU はフォンノイマン型である。これは、格納されたプログラムに従って逐次命令を実行する方式である。CPU の動作はプログラムカウンタ (PC) によって制御され、命令の実行に伴って PC を更新することによって動作する。また、ある時間における CPU の状態はレジスタコンテキストに記録されており、PC の指す命令とその時のレジスタコンテキストの整合性がとれている時に始めて動作秩序が保たれる。

CPU で複数のタスクを並行に処理、すなわち複数のプログラム列を実行させる場合を考える。1 つの CPU がある時間に実行できるプログラム列は当然 1 つだけである。したがって、CPU で複数のプログラム列をさばくには図 5 のように時間毎に実行中のプログラムを切替えることで実現する。

その際プロセッサの動作秩序を乱さないようにするためにコンテキストの切替えを行わなくてはならない。さらに A/D 変換のようなリアルタイム性を要求されるタスクの場合、プログラムの切替えを割り込みに基づきタスクスケジューリングす

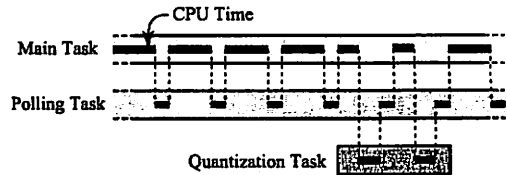


図 5: ノイマン型で処理を行う場合

る必要があり、プログラムのコーディングが難しくなる。

ここで、図 5 のように割り込みが高頻度で発生する場合の性能低下について考察する。タスクの処理時間を t_p 、割り込みが発生する割合を p 、割り込みで行う処理時間を t_{ip} とすると、本質的に必要な処理全体にかかる理想時間 T_{ideal} は

$$T_{ideal} = t_p + t_p \cdot p \cdot t_{ip} \quad (1)$$

である。しかし、実際の割り込み発生時には割り込み応答時間、コンテキスト退避時間、コンテキスト復帰時間、割り込み復帰時間といったオーバーヘッド時間 t_o を伴う。したがって、実際に処理全体にかかる時間 T_{real} は

$$T_{real} = t_p + t_p \cdot p \cdot (t_{ip} + t_o) \quad (2)$$

となる。(1) と (2) から本来必要な処理との割合を考えると

$$\frac{T_{ideal}}{T_{real}} = \frac{1 + p \cdot t_{ip}}{1 + p \cdot (t_{ip} + t_o)} \quad (3)$$

である。(3) を見ると割り込みで行う処理時間 t_{ip} が小さい場合、割り込みが発生する割合 p が大きい場合はプロセッサ時間の使用率が減ることが分かる。したがって、ノイマン型は少ない処理の割り込みを高頻度で発生させるような用途に対してはその処理性能を著しく低下する。

2.5 データ駆動型プロセッサ (DDP)

本稿で取りあげる DDP は自己タイミング型パイプライン (以下、STP) によって非同期に動作するアーキテクチャを持つプロセッサである。CPU では処理の制御情報をプロセッサが持っていたのに対し、DDP ではデータバケットが持っており STP 中で必要なデータがそらい次第実行される。このため CPU のような複雑なスケジューリングは必要無く、DDP の動作記述には並列性を表現しやすいデータフローグラフ (DFG) を用いるため、ソフトウェアの設計が容易である。同時にこれは CPU で存在したようなコンテキストスイッチなどのオーバーヘッドが DDP には本質的に存在しないことを意味する。つまり、並列処理において CPU に存在した開発時の設計困難性と実行時の処理オーバーヘッドによる性能低下という、アーキテクチャに

依存する問題が DDP には存在しない。したがって、DDP はリアルタイム性が要求される粒度の細かい処理を並列に動作させる用途にも親和性が高いと言える。

3 ソフトウェア A/D・D/A 変換器

本稿では、図1で境界線を左へシフトするとき、その右側には DDP を、左側には少量のアナログ回路を考えている。つまり、図1の破線の境界線で表すように、ADC や DAC の内部で直接 DDP へ接続する。本節では、そのようなソフトウェアによる ADC や DAC について述べる。

3.1 ADC

DDP によるソフトウェア $\Delta\Sigma$ ADC は図6のように実現できる。ここでは、 $\Delta\Sigma$ 変調器をアナログ回路で、デジタルフィルタを DFG により構成した。 $\Delta\Sigma$ ADC では入力されるアナログ信号をアナログ $\Delta\Sigma$ 変調器によりビットストリームに変換し、さらにビットストリームをデジタルフィルタにより PCM に変換する。得られる PCM の f_s や分解能はデジタルフィルタによって決定する。そこで、デジタルフィルタを書き換え可能なメモリ上のソフトウェアで実装する事で、全体のハードウェア構成を変更することなく f_s と分解能の調整を行うことができる構成となっている。

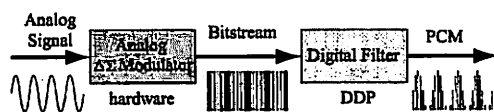


図 6: ソフトウェア ADC の構成

アナログ $\Delta\Sigma$ 変調器

アナログ $\Delta\Sigma$ 変調器は図7に示すように、積分器と1ビット量子化器、1ビット D/A 変換器によって構成される。入力信号の電圧を積分し、その値が1ビット量子化器のスレッシュホールドを越えるとデジタル的には1となる電圧 V_{dd} を出力し、同時に積分器をリセットする。その結果、アナログ $\Delta\Sigma$ 変調器の出力として入力信号のレベルに応じた1ビット信号の疎密波が得られる。

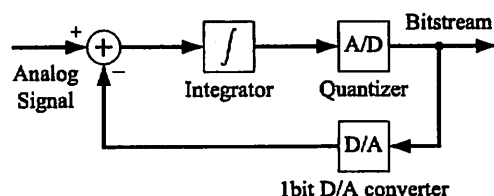


図 7: アナログ $\Delta\Sigma$ 変調器

デジタルフィルタ

アナログ $\Delta\Sigma$ 変調器から送出される1ビット PCM を目的の分解能の PCM に変換するため、デジタルフィルタ部分はいわゆるデシメーションフィルタになる [7]。デシメーションフィルタは図8に示すように LPF とダウンサンプラによって構成されている。



図 8: デシメーションフィルタ

実験では入力に 1kHz の正弦波信号を使うことにし、デシメーションフィルタとして $f_c = f_s/256$ のパワースペクトル特性 9 次 LPF を使用した。そして、LPF とダウンサンプラによって分解能が 2bit、3bit、4bit、5bit である PCM を出力するデシメーションフィルタを実装した。

3.2 DAC

ソフトウェアによる $\Delta\Sigma$ DAC は図9に示すように、DFG により実装したデジタル $\Delta\Sigma$ 変調器と電子回路で作成したアナログ LPF から構成されている。デジタル $\Delta\Sigma$ 変調器では入力される PCM 信号をビットストリームに変換し、出力されるビットストリームをアナログ LPF に通すことで出力のアナログ信号を得る。

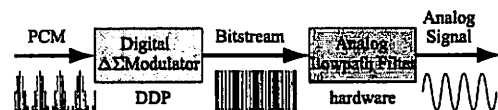


図 9: ソフトウェア DAC の構成

デジタル $\Delta\Sigma$ 変調器

PCM から1ビット信号列に変換するデジタル $\Delta\Sigma$ 変調器は図10に示すように基本的な構成はアナログ $\Delta\Sigma$ 変調器と同じである。

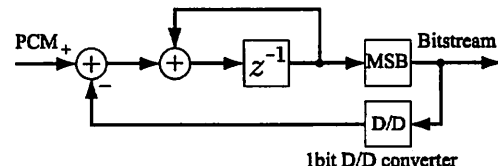


図 10: デジタル $\Delta\Sigma$ 変調器

アナログ LPF

デジタル $\Delta\Sigma$ 変調器からのビットストリーム出力からアナログ信号を得るためにはアナログ LPF が必要となる。実験では 1kHz の正弦波を入力としたため、 $f_c = 2\text{kHz}$ のパワースペクトル型 2 次 LPF を OP アンプを使用して実装した。

3.3 クロック発振器

本実験で用いたクロック発振器はソフトウェアにより実装した。DDP 内部でパケットを周回させコピーにより出力する図 11 のような構成で、トリガパケットを入力すると一定時間毎にパケットを出力する事ができる。この時、1パケットを1クロックパルスとみなすとパケットの周回時間を周期とした発振器と考える事ができる。

Trigger Packet

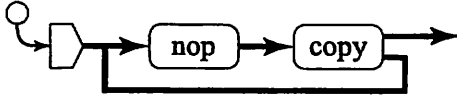


図 11: クロック発振器

3.4 実験による評価

本稿で提案したソフトウェアによる ADC-DAC が動作し、 f_s や分解能などの特性を変更できる事を確認するために実験を行う。そのため、今回の実験では信号を ADC で変換して得られた PCM に対して何もせずにそのまま DAC に入力し、デジタルフィルタを差し替える事による DAC 出力の変化により評価を行う。実験環境は図 12 に示すようになっている。まず、波形発生器 (Textronix 製 AFG3021) によって発生させた周波数 1kHz、電圧レベルを High が 3.95V で Low が 1.05V の正弦波の信号をアナログ $\Delta\Sigma$ 変調器に入力する。アナログ $\Delta\Sigma$ の出力を評価実験ボードのデジタル信号入力線に接続し、出力されるビットストリームを 3.3 のクロック発振器によって 1MHz で生成されるパケットに載せる。入力された信号はデシメーションフィルタによって PCM に変換され、そのままデジタル $\Delta\Sigma$ 変調器によってビットストリームに変換されて評価実験ボードから出力される。その出力結果をアナログ LPF に入力する事でアナログ信号の出力結果を得る。そして、この実験環境のデジタルフィルタ部分を変更して、その動作を DSO (Tektronix 製 TD3034) で観測することで評価した。

3.5 実験結果

ソフトウェア $\Delta\Sigma$ 変換器を、入力信号を 1kHz の正弦波として、ソフトウェアで実装したデシメーションフィルタからの出力 PCM の分解能が、2bit/3bit/4bit/5bit とそれぞれ差し替えながら動作させて観測した。その際の DSO での観測結果のうち、2bit の分解能を持つ PCM を出力するデシメーションフィルタを適用したものを図 13、同 5bit のものを図 14 に示す。

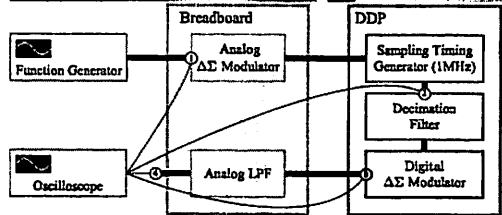
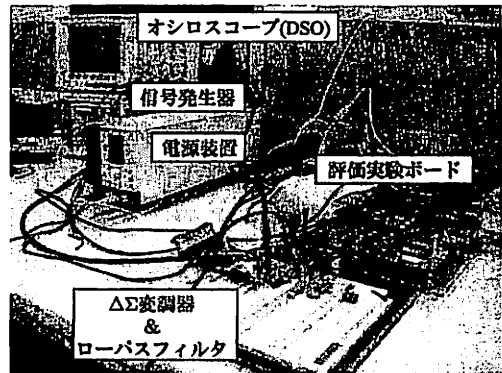


図 12: 実験環境

Ch1 は、参照用として、ソフトウェア $\Delta\Sigma$ 変換器に入力する信号を観測したもので、波形発生器からの 1kHz の正弦波である。Ch2 は、アナログ $\Delta\Sigma$ 変調器の出力したビットストリームを DDP パケットに載せたときの各々のパケットの持つデータ (0 or 1) を DDP の汎用出力端子を通じて出力し、観測したものである。Ch3 は、デジタル $\Delta\Sigma$ 変調器の出力を観測したもので、デシメーションフィルタによって出力された PCM がデジタル $\Delta\Sigma$ 変調器によってビットストリームに変換された結果である。Ch4 は、アナログ LPF の出力を観測したもので、理想的には Ch1 の入力信号と同じ結果が得られるはずである。

図 13 の Ch4 見てみると信号が 4 段の階段状のようになっている。これはデシメーションフィルタによって一度 PCM に変換されたが、その際の量子化レベルが 0、1、2、3 のいずれかしか取り得ることができず、入力信号を表現するのに不十分であったために出力のアナログ信号にその形が現れたと考えられる。それに対して、5bit の場合の図 14 の Ch4 を見てみると PCM の量子化レベルが 32 段階と多くなっている事から正弦波に近い出力結果が得られている。以上の観測結果からハードウェアに一切手を加える事無くソフトウェアのデシメーションフィルタの変更のみで分解能が変更できる事が確認できた。

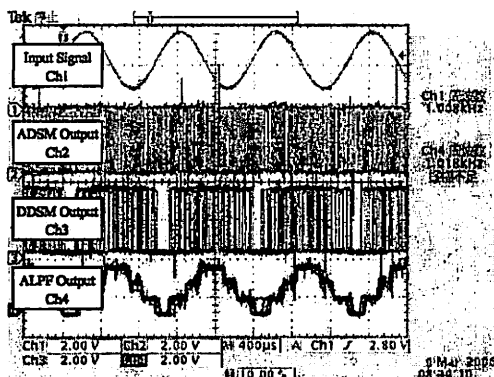


図 13: 分解能 2bit のデシメーション後の波形

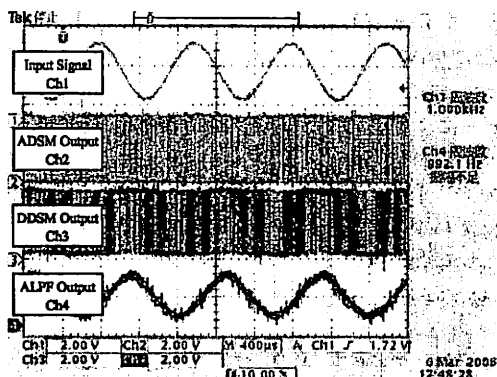


図 14: 分解能 5bit のデシメーション後の波形

4 まとめ

本稿では、まず、従来ハードウェア処理していた処理をソフトウェア処理にするという流れがあるにもかかわらず、ADC-DAC に関してはソフトウェア化されていないことから、ADC-DAC のソフトウェア化の検討を行った。つぎに、 $\Delta\Sigma$ 方式という比較的ソフトウェア化に適している変換方式が存在しながら、これまで ADC-DAC がソフトウェア化されなかった原因を指摘し、ノイマン型プロセッサの代わりに、DDP アーキテクチャ上のソフトウェアで ADC-DAC を構築した。そして、ソフトウェアで実装された $\Delta\Sigma$ 変換器に対して実際に信号を入力してその動作を観測する事で ADC-DAC として動作する事を確認した。さらに、ソフトウェアで実装した部分である、デジタルフィルタの変更だけで得られる PCM の分解能が変更できる事を確認し、ADC-DAC の特性を柔軟に変更できる可能性を示した。

将来の課題

まず、本稿では内部で生成したパケットによりビットストリームを入力する端子を監視しサンプリングしていたが、 f_s を安定させるためデジタル入力回路の検討を行いたい。つぎに、ある f_s を得る時、オーバーサンプリング係数とデジタルフィルタ次数は必要とする演算能力に関してトレードオフの関係にあるため、DDP の演算能力の分配を検討したい。最後に、オーディオや動画など、用途ごとに適応できるようなパラメータ設定幅の広いソフトウェア ADC-DAC にしたい。

謝辞

データ駆動型プロセッサに関する技術的なご支援・ご議論を頂いたシャープ株式会社の関係諸氏に深く感謝の意を表します。

参考文献

- [1] H. Terada, S. Miyata, M. Iwata, "DDMP's: Self-Timed Super-Pipelined Data-Driven Multimedia Processors," Proc. of the IEEE, Vol. 87, No. 2, Feb. 1999.
- [2] E. K. F. Lee and P. G. Gulak "A CMOS field programmable analog array," IEEE J. of SSC, Vol. 26, Issue 12, pp. 1860-1867, Dec. 1991.
- [3] U. Beis, "An Introduction to Delta Sigma Converters," <http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html>, 29th Dec. 2005.
- [4] R. H. Walden, T. Cataltepe, G. C. Temes, "Architectures for High-Order Multibit $\Delta\Sigma$ Modulators," IEEE Proc. of ISCAS'90, pp. 895-898, May 1990.
- [5] B. P. Agrawal, K. Shenoi, "Design Methodology for $\Sigma\Delta$," IEEE Trans. Commun., Vol. COM-31, pp. 360-370, Mar. 1983.
- [6] J. Thompson, "Care and Feeding of the One Bit Digital to Analog Converter," <http://www.ee.washington.edu/conselec/CE/kuhn/onebit/primer.htm>, 8th Jun. 1995.
- [7] R. E. Crochiere, L. R. Rabiner, "Interpolation and Decimation of Digital Signals - A Tutorial Review" Proc. IEEE, Vol. 69, pp. 300-331, Mar. 1981.