Published at « Intelligent Communication technologies and Applications, with Emphasis on Mobile Communications »,
International COST 254 Workshop, May 5-7, 99, Neuchâtel, Switzerland

Efficient Sub-band Coder Implementation for Portable Low-power Applications

Institute of Microtechnology, Rue Breguet 2, CH-2000 Neuchâtel
Andreas Drollinger, Dequn Sun, Christophe Waelchli, Alexandre Heubi, Peter Balsiger, Fausto Pellandini
Contact: www-imt.unine.ch

Dspfactory, Waterloo, Ontario, Canada N2V 1K8
Robert Brennan, Todd Schneider
Contact: www.dspfactory.com

Abstract

This paper presents a low-power co-processor designed for power-efficient, real-time mono and stereo Weighted Overlap-Add (WOLA) filterbank processing. The filterbank easily interfaces to general-purpose DSP processors as a co-processor and provides high fidelity filterbank processing with low group delay for a wide range of applications. The co-processor architecture is very flexible and generates from 4 up to 128 complex channels. It is implemented in a 0.35 um CMOS five-layer metal technology. The chip size is less than 8 mm^2, which is very small compared to standard DSP processor chips. Additionally, the power consumption is only 250 μA with a power supply of 1 volt. The co-processor has also been targeted to a 0.18 um CMOS four-layer metal technology for use in digital hearing aids where it provides the flexibility required to support advanced algorithms while meeting the ultra low-power requirements of this application.

1. Introduction

Deep sub-micron technology opens the door to the advanced implementation of complex algorithms specifically targeted for extremely low-power and portable applications. These applications are severely constrained by small physical size and extremely low power consumption requirements.

This paper presents the design and implementation of an oversampled Weighted Overlap-Add (WOLA) filterbank co-processor. This design provides a highly flexible time-frequency representation amenable to sub-band adaptive, sub-coding and other similar applications [Bre98a, Bre98b, Sch97].

The co-processor architecture offers maximum throughput at low clock speed in order to minimize power consumption (256-point FFT-WOLA stereo signal processing at 50 kHz sampling rate at only 1.28 MHz chip clock rate). This high throughput is achieved by intelligent DMA management combined with concurrent operating computation units specifically targeted to radix2 and radix4 FFT butterfly structures. Block floating point has been used to increase the numerical accuracy and realize a noise floor of –115 dB.

The co-processor easily interfaces with any standard DSP processor, AD-converter and DA-converters. It has two main sub-blocks presented in Figure 9. The first one is the Input-Output Processor (IOP). It receives audio signals from an AD-converter and performs decimation. To minimize aliasing distortion, the IOP uses two highly selective, digital IIR filters for anti-alias decimation and anti-image interpolation. The filters have been design to provide 78 dB SNR and use an architecture that is free from limit cycles.

The second sub-block is the WOLA filterbank processor. Input samples to the WOLA are stored in a circular input FIFO. Every R (input block size) samples a WOLA analysis transformation is performed on L samples (L >> R). A general purpose DSP (the “control DSP”) is used to analyze the spectrum and to apply, via the shared RAM, gains for each frequency band. Then, the WOLA coprocessor performs a WOLA synthesis transformation and stores the results in the output FIFO. Finally, the IOP sends the out-going samples to an interpolator which incorporates the digital IIR filter described above. The up-sampled signal is then sent to a DA-converter.

The WOLA can also operate in stereo mode. In this mode, the WOLA processes two simultaneous data streams. Following analysis, the control DSP performs a final butterfly separation step, applies gains to the separated stereo channels and then mixes both channels. The mixed frequency-domain signals are then returned to the time domain via a WOLA synthesis transformation. Stereo supports the implementation of phase-dependent algorithms such as sub-band beam-forming.

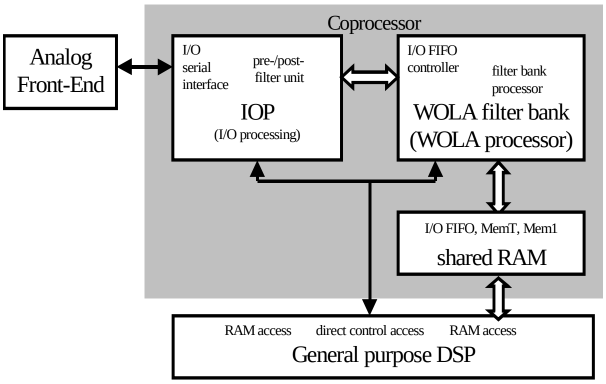

Figure 1. Overview of the co-processor’s environment

The remainder of this paper describes the design and implementation of the WOLA co-processor. Section 2 presents some theoretical aspects of the WOLA. Section 3 describes the WOLA co-processor’s architecture. Section 4 provides information about the Input-Output Processor (IOP) while Section 5 details the performance that has been achieved. Section 6 describes some typical applications of the WOLA coprocessor. Finally, conclusions are presented in Section 7.

2. Theoretical aspect of the WOLA filterbank

Over the last two decades, multi-rate digital signal processing techniques have been considerably developed and widely practiced in various engineering disciplines. The conditions to obtain a perfect reconstruction (PR) maximally decimated (or critically sampled) filter bank have been extensively investigated and well-documented [Vai93]. PR systems impose severe constraints that are not suitable in some applications. For example, in applications requiring significant adjustment in the frequency bands, other structures are preferable [Bre98b].

Two commonly used structures that yield an efficient implementation of a DFT filter bank are the polyphase structure and Weighted Overlap Add (WOLA) structure. The basic mathematical framework for these two structures is identical – they differ only in the manner of data manipulation. Thus, they both have similar implementation efficiency [Cro83].

The WOLA structure, which is based on a block transform interpretation, provides a more familiar view of the processing. Furthermore, as will be shown below, when this block processing presentation is implemented using Block-Floating-Point (BFP) arithmetic, excellent performance can be obtained in a 16-bit fixed-point implementation due to a reduction in data quantisation artifacts.

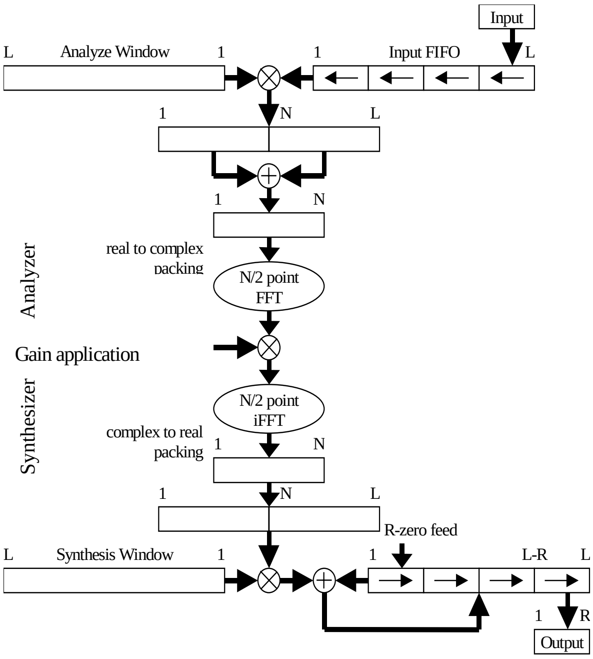

Figure 2. Simplified block diagram of a WOLA filter bank

WOLA structure

Figure 2 shows a simplified block diagram of an oversampled WOLA filter bank [Bre98, Cro83]. For an oversampling of two times, the input step size (R) is half of the FFT size (N). When R and N are equal the filter bank is critically sampled since the analysis window is designed with cutoff frequency equal to 2π/N. The use of oversampling provides two benefits:

1. the gain of the filterbank bands can be adjusted over a wide range without the introduction of audible aliasing and
2. a group delay versus power consumption trade-off can be made (i.e., more oversampling implies reduced group delay and increased power consumption).

In operation, the input FIFO is shifted and R new samples are stored. The input FIFO is then windowed with a prototype low pass filter of length L. The resulting vector is added modulo N (i.e., “folded”) and the FFT of the resulting windowed time segment is computed. Because an FFT is used, the outputs from the analysis filterbank provide both magnitude and phase information (i.e., they are complex).

To generate a modified time-domain signal, the channel gains are applied to the N/2 FFT outputs (channel signals) and an inverse FFT is computed. The resulting time-domain “slice” is then windowed with a synthesis window and accumulated into the output FIFO. This generates R samples that are shifted out of the output FIFO. Finally, R zeros are shifted into the output FIFO and the entire process repeats for the next block of R input samples.

The data flow shown in Figure 2 is simplified. Additional operations are required:

- The time shift of the input and output FIFO must be compensated with a pre-/post-rotation of the complex values before the FFT is performed.
- After analysis two half-bands at DC and the Nyquist frequency are generated. Equal width bands can be generated by using an Odd FFT and a square-wave modulator on the time-domain input and output signals.
- For stereo processing, two mono signals are interleaved and processed as a complex signal. After analysis the complex result is separated to get the individual mono signal spectra.

Block floating point arithmetic

Fixed-point arithmetic applied to radix-2 and radix-4 butterfly structures and MAC operations require optimised corrections for overflow and quantisation effects. Block-floating-point (BFP) computation units are used to increase the dynamic range and lower the quantisation error in order to improve the SNR of the WOLA filterbank. The BFP strategy decreases the quantisation error without increasing the computation complexity. This is achieved by dividing data into non-overlapped groups (passes) and formatting the data at each node in data flow path with common exponent.

Both silicon area and power consumption are saved by using BFP to “scale down” the data and prevent overflow. The combination of radix-2 and radix-4 structures implies a worst case growing factor of 5.06. A 2-bit overflow detection/scaling mechanism combined with saturation ability was chosen to prevent overflow and to improve the dynamic range.

3. Implementation of the WOLA

The implementation of the WOLA coprocessor was guided by three primary constraints:

a) minimal size (both gate count and silicon area)
b) minimal power consumption
c) flexibility: we required a WOLA filterbank design that supported programmable configurations (input block step size R=2...128, DFT size N=8...256, prototype lowpass filter order L=32...256, synthesis window decimation factor 1...127; with all factors with a power of 2.)

General considerations

FFT-based filterbank algorithms may use data-path local registers to store the intermediate results of basic radix2/radix4 and Multiplication/Accumulation unit (MAC) operations. However, the main memory must be a RAM because the amount of data for an FFT of 256 points (mono and stereo) is quite high. An advanced data and coefficient-addressing concept is one of the keys to a successful low-power FFT filterbank implementation.

The design intelligently exploits the regularity of the WOLA algorithm. A simple example of an 8-point FFT, using radix4/radix2 fixed memory location structure, is shown in Figure 3 to highlight the essential components of the design.

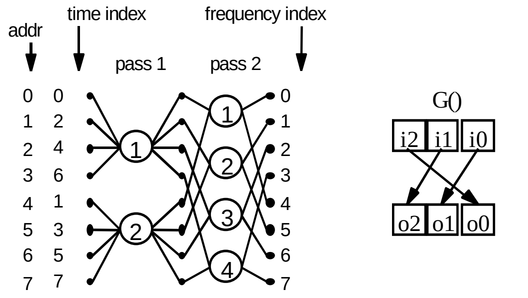

Figure 3. Example of an 8-point radix4/radix2 FFT

In the first pass, the WOLA processes the radix4 operation; the radix2 operation is processed in the second pass. The addresses to read and write data from/to the memory are generated using a simple binary counter and bit-reverse address generation unit.

Every pass of the forward and reverse WOLA algorithm (including FFT, windowing, pre-rotation and frequency separation) uses the bit reverse address generation unit to access data and coefficients.

Architecture

The implementation of the WOLA focuses on an architecture which simultaneously minimizes the overhead of the control/addressing blocks and maximizes the parallelism of the data processing in order to achieve continuous one-cycle memory accesses. Satisfying these two constraints allows the use of a low-frequency system clock which results in a significant power savings. The control/addressing overhead was minimized by using a microcoded design and an optimized architecture. The high degree of parallelism was achieved by using a number of processing units in parallel, all of which work simultaneously.

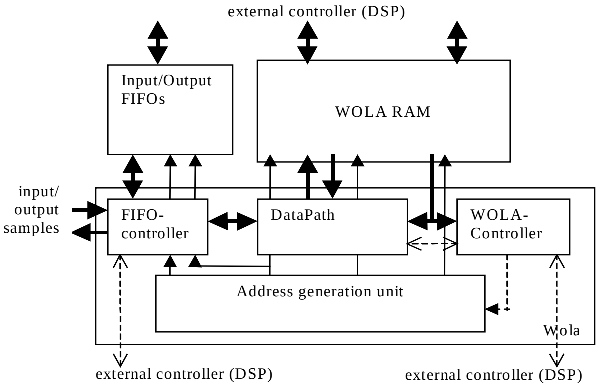

Figure 4. The WOLA coprocessor architecture

The signal flow is shown in Figure 4. Data moves from the input FIFO through a WOLA analysis transformation under the control of a general purpose DSP. Gains are computed by the general-purpose DSP and written to WOLA RAM where they are applied to the (complex) results of the WOLA analysis transformation. Then, the control DSP initiates a WOLA synthesis transformation and the modified time-domain signal is available in the output FIFO.

The input/output FIFO controller

The input and output FIFOs are realized as circular buffers. The input FIFO has two data domains: the WOLA-processing domain that is used by the WOLA processor and the write-in domain that stores the new samples.

The output FIFO has also two data domains: the WOLA-processing domain and the read-out domain (which contains the data that are ready to send out).

Every R samples, if the write-in domain of the input FIFO is full, all domains of both FIFOs are shifted forward by a step of R samples.

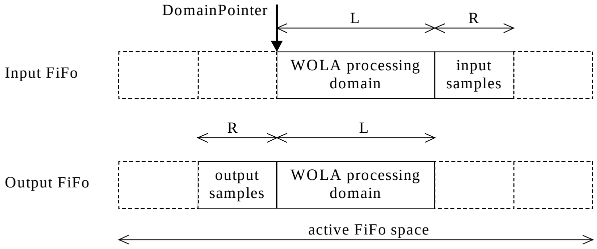

Figure 5. Input and output FIFO memory domains

The FIFO controller controls the different memory domains of the input and output FIFOs. It indicates the position of these domains via a signal sent to the general purpose (control) DSP.

In order to facilitate access to the input samples by the WOLA-processor, the FIFO controller stores and reads the samples in different ways, depending on the mode of the FIFO controller (mono, stereo, digital-mix mode[1]).

The FIFO controller may optionally work only with a half, a quarter or an eighth of the whole FIFO space. In these cases, the unused memory space is available for use by the general purpose (control) DSP.

The WOLA control system

The WOLA processor is commanded by a simple, yet very flexible controller. As explained previously, the WOLA transformation (analysis and synthesis) and gain application are subdivided into different passes. The characteristics of a pass are:

- all operations are the same (radix2, radix4, etc)
- every read and write address for the data and coefficients can be generated by a bit-reverse addressing unit.

The consequence of these two characteristics is that the data-path and the address generators need a fixed configuration during one pass. This configuration defines the basic data-path operation and the bit-reverse addressing that must be applied.

To understand the idea behind the control system, one has to define the term "function". The passes are organized in functions, such as WOLA analysis, WOLA synthesis and gain application. All parameters of the sub-blocks are fixed during one pass, but some of the parameters are fixed during one whole function.

The pass parameters depend on the function, on the pass number and on the WOLA configuration. Because this dependence is complicated, a microcoded design was used. This approach makes efficient use of hardware resources and provides a flexible implementation.

The external DSP starts the WOLA coprocessor giving the function number to be performed. The function configuration contains the parameters that are fixed during the whole function and the number of passes of the function.

Data-path and address generators receive two counter values. The first one has to execute the correct arithmetic sequence in accordance with the operation counter. The second one controls the bit-reverse addressing unit defined by the pass configuration to get read and write addresses of data and coefficients.

Data-path

The data-path is specifically designed for efficient computation of WOLA filterbank processing; it consists of a multiplier array and an ALU bank (Figure 6).

The datapath controller steers not only the multiplexers and registers but also the multiplier array and ALU bank. It defines the correct sequence of operations required for a particular function, which is defined by the WOLA controller.

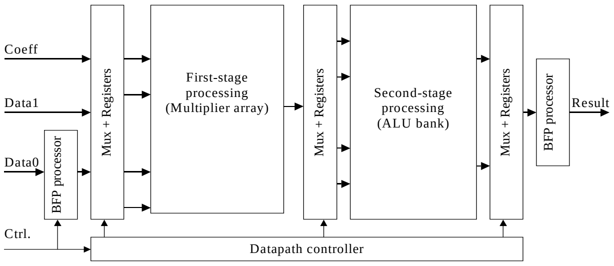

Figure 6. Block diagram of the data-path.

This structure was developed to achieve an efficient data-flow for the basic WOLA operations. N-point FFT processing is achieved by computing a specific number of radix4 or radix2 transforms. A radix4 butterfly is a 4-point FFT, which is computed with three complex multiplications plus sixteen additions/subtractions.

$$X(n) = \frac{1}{4}\sum_{k=0}^{3} x(k)\cdot W_4^{nk}$$

Twelve real multiplications are needed for one radix4 pass. The sixteen last additions/subtractions are made by the ALU bank. Then, the four complex outputs are stored in the WOLA RAM. This structure allows for parallelisation of the computations. This results in faster FFT computations; also, the use of one single radix4 as a basic FFT operation minimizes area and consumption.

Because the WOLA data-path is the main power consumer, this block was carefully designed using gated clocks to minimize power consumption.

4. I/O processing and interfacing

The IOP unit (input-output processor) contains an interface module (SSI = synchronous serial interface) and a pre- and post processing unit (FILT) as shown in Figure 7. The IOP does both input and output operations.

The SSI converts incoming stereo audio samples into an internal parallel representation. The FILT unit contains a combined DC-removal / decimation filter and an interpolation filter.

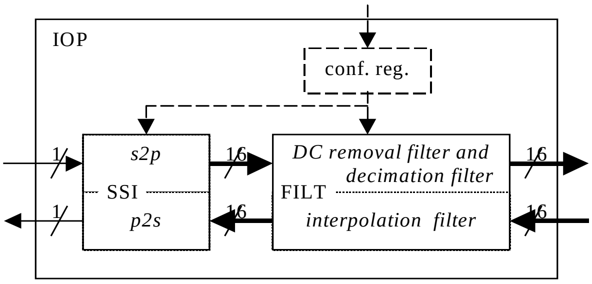

Figure 7. The input/output unit

DC removal, decimation and interpolation filters

A simple filter with a pole (x)-zero (o) pair is used as a DC removal filter. The closer the pole is to the zero, the lower cut-off frequency that will be obtained. One of three cut-off frequencies can be selected or the DC

[1] The input FIFO contains two independent time-domain signals that can be preprocessed before WOLA analysis.

removal feature can be disabled (feed-through). Both decimation and interpolation filters have been realized using IIR filters.

Implementation

The filter unit may work in mono or in stereo mode. In order to save area and power the number of arithmetic units is limited to one MAC unit, one adder, one shifter and one rounding unit. These resources are shared between the DC-removal, the decimation and the interpolation filters. The whole filter block has been generated with dp_gen, a custom program. It reads a data-path description file and automatically generates an optimized VHDL code of the data-path together with its control unit and a postscript file to visualize the data-flow-timing diagram and the resource allocations.

Similar to the data-path of the WOLA processor, the IOP data-path was designed carefully in order to get the best performance in terms of area, power and numerical performance. The IIR decimation and interpolation filters are an especially good choice and need minimal resources. To save power, temporarily unused registers are directly controlled by gated clock strategy. In addition, the number of data transfers was minimized.

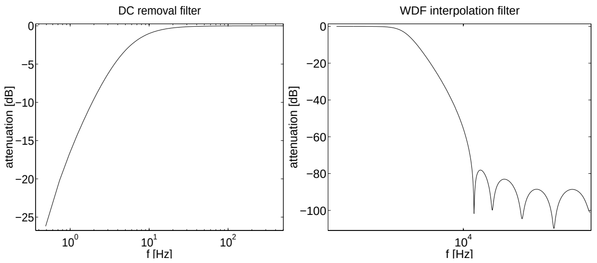

Figure 8. The characteristics of the DC removal and interpolation filters

The filter coefficients are chosen to provide 78-dB attenuation at 10.35 kHz. The ripple in the pass band is extremely low (< 6e-8 dB) and the maximum group delay is less than 0.4 ms. Figure 8 shows the frequency responses for the DC removal and interpolation filters.

5. Results

The noise floor and THD+N measurements of the coprocessor's data-path units show that the noise floor is approximately –115 dB. A highly-selective filterbank configuration (16-channels with 14 ms group delay) provides about –65 dB THD+N. Note that the realized THD+N is dependent on the selected WOLA parameters (FFT size, oversample factor and window length) as well as the window coefficients. A large number of combinations that trade-off fidelity (THD+N) versus power consumption versus group delay are possible.

The time used to perform WOLA analysis and synthesis is strongly configuration dependent. Table 1 shows the time required for analysis and synthesis and also gives the ratio between the used time and the time that is available if a system clock of 1 MHz and a sampling frequency of 16 kHz are used.

N	L	OF	DF	M	#cycles (analysis + synthesis)	$t_{used}/t_{available}$ (analysis + synthesis)
16	128	2	2	no	430	86%
32	256	2	2	no	770	77%
32	128	2	2	yes	578	58%
32	128	2	2	no	734	72%
128	128	4	4	no	1778	89%

N : FFT size
L : prototype filter length (window size)
OF : oversampling factor (R=N/OF)
M : input/output sign modulation (odd stack)

Table 1. Computation cycle and time for different configurations

6. Applications

The WOLA filterbank is ideal for use in a wide range of frequency domain processing applications. These include dynamic filtering and equalisation (dynamic range compression, noise reduction), sub-band coding/decoding, sub-band directional processing, voice activity detection and echo cancellation. Other applications include personal listening devices where head-related transforms and other similar algorithms can be implemented in the frequency domain. The WOLA is especially suited to applications that require low delay (< 10 ms).

In wireless applications the WOLA can be used to perform simultaneous high-fidelity equalization, noise reduction and AGC (dynamic range compression). Using the stereo processing mode, the WOLA can be used to implement two-microphone, adaptive frequency domain noise cancellation for use in handsets. The stereo processing mode is also well-suited for frequency domain echo cancellation.

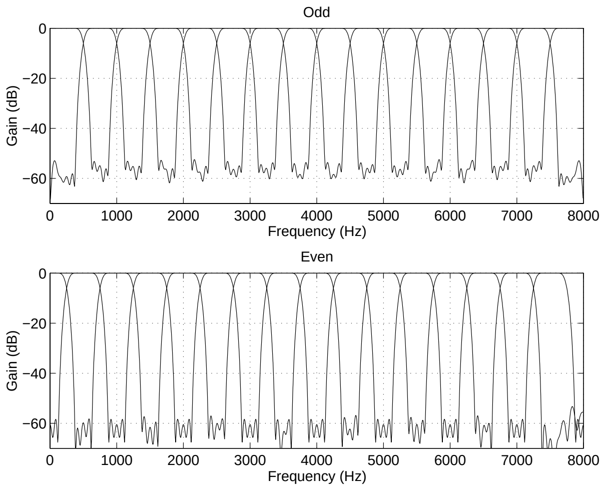

Figure 9. Channel frequency responses for 16-channel filterbank in even and odd stacking with 14-ms group delay

The large amount of signal separation between the WOLA output channels offers a high-degree of orthogonality between all channels. This greatly speeds the convergence of adaptive algorithms and makes for very efficient subband coders and decoders. The low delay is also attractive for many subband coding applications.

The combination of a general-purpose (control) DSP and the WOLA provides an ideal flexibility versus power consumption tradeoff.

7. Conclusions

This paper presents an efficient design and the results for a novel real-time WOLA filterbank. The co-processor has been implemented in two deep sub-micron technologies. The entire co-processor requires approximately 30 kgates.

A first implementation was done on a 0.35 µm CMOS five-metal layer technology (Figure 10). The total die size (including pads) is 7.86 mm^2. A second implementation has been done on a 0.18 µm CMOS four-layer metal technology. The measured power consumption is 0.4 mW @ 1.7 volt supply voltage and less than 250 µW @ 1 volt for a typical algorithm.

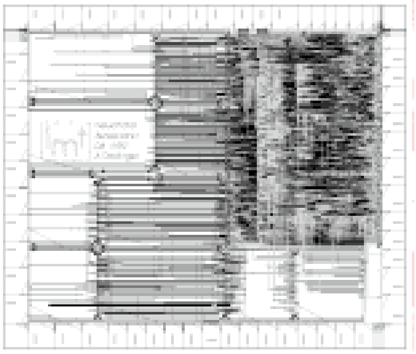

Figure 10. The WOLA coprocessor's implementation on a 0.35 µm five-layer metal technology

Figure 11 compares the co-processor performance (in terms of execution time in microseconds at an equivalent chip clock frequency of 10 MHz) to standard DSP processors. Note that the BDSP9124 and the DSP-24 are dedicated FFT DSPs, which use significantly more resources than the WOLA coprocessor described here does.

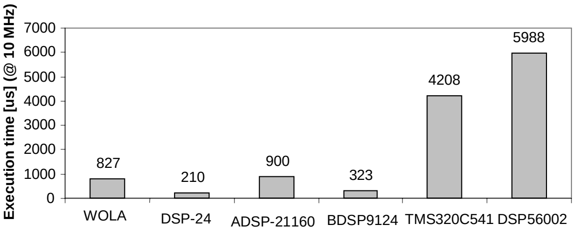

Figure 11. Execution time in µsec for1024 complex point FFT sizes @ 10 MHz. ADSP-21160 (Analog Devices), BDSP9124 (Butterfly DSP), WOLA (IMT, Dspfactory), DSP-24 (DSP Architectures) and C6x and C50 (Texas Instruments)

Because the maximal FFT size of the WOLA co-processor is limited on 128 complex points (256 reel points) its 1024 point complex-FFT execution time is an interpolated value.

The most important value for lowest power applications is the effective power consumption used to perform an FFT. Figure 12 shows the effective power consumption for different processors.

The processors from Analog Devices, Butterfly DSP and DSP Architectures are focussed on high throughput rate and not on low-power implementations. Figure 12 clearly shows that the WOLA offers roughly 4.5 times the performance when power consumption is considered.

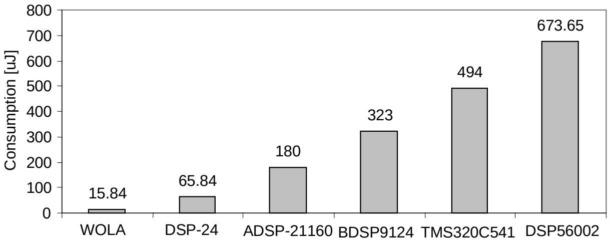

Figure 12. Effective power consumption for one 1024 complex point FFT. Power supply: WOLA 1.7V, DSP-24 3.3V, ADSP-21160 5V, BDSP9-124 5V, TMS320C541 5V, DSP56002 5V

References

[Bre98a] R. Brennan & T. Schneider, "Filterbank Structure and Method for Filtering and Separating an Information Signal into Different Bands, Particularly for Audio Signals in Hearing Aids", PCT Patent Publication WO09847313A210, October 22, 1998.

[Bre98b] R. Brennan & T. Schneider "A Flexible Filterbank Structure for Extensive Signal Manipulations in Digital Hearing Aids," *Proc. ISCAS-98, Monterey, CA.*

[Cro83] R. E. Crochiere & L. R. Rabiner, "Multirate Digital Signal Processing", Prentice-Hall, 1983

[Fli94] N. J. Fliege, "Multirate Digital Signal Processing", JOHN WILEY & SONS, 1994

[Sch97] T. Schneider & R. Brennan "A Multichannel Compression Scheme for a Digital Hearing Aid," *Proc. ICASSP-97,* Munich, Germany.

[Smi95] W. W. Smith & J. M. Smith, "Handbook of Real-Time Fast Fourier Transforms", IEEE Press, 1995.

[Vai93] P. P. Vaidyanathan, "Multirate Systems And Filter Banks", Prentice-Hall, 1993.