Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 52(66), Fascicola 1, 2007

An Extension of the Xilinx PicoBlaze Architecture for DDFS Applications

Ioan Jivet[1]

***Abstract* – The paper presents an extension of the Xilinx PicoBlaze soft processor with timers for digital signal synthesis applications. The novel direct digital synthesis (DDS) amplitude sequencing architecture is used targeting FPGA implementations. The Xilinx picoBlaze soft microprocessor assembly code for direct digital synthesis algorithm is presented and its verification using the graphical pBlazIDE tool. The implementation is used to explore the design space for specific applications. The results of simulations for two sample applications are presented.**

***Keywords* - picoBlaze architecture extension, amplitude based direct digital synthesis.**

I. INTRODUCTION

Direct digital synthesis (DDFS) has been accepted as the main frequency generation solution surpassing classical circuits like the analog phase locked loop (PLL)[4][5]. A novel method of direct digital frequency synthesis based on amplitude sequences was recently introduced [1].
Use of the amplitude based architecture in the design of frequency synthesis solutions require additional logic to augment the basic architecture characteristics. A firmware development environment was proposed [2]. Although the architecture of the PicoBlaze is a very optimized one it does incorporate any timing modules.
The paper presents an extension of the PicoBlaze microprocessor architecture with two eight bit timers for real time applications. The clock for the timer is derived using a Digital Clock Manager (DCM) module available in most recent Xilinx FPGA families [10].
The synthesis and simulation results are presented for two sample applications targeting FPGA implementation[6][8]. A twin low frequency high precision emulation case that demonstrates the usefulness of the environment for DDFS solutions design space exploration. The second example application presented is a stored binary step sequence for high frequency generation. It was found that using this approach with two phases (one for generation and storage and a second one for generation) the sample frequency rate can be raised to tens of MHz.

II. THE AMPLITUDE BASED DDFS ARCHITECTURE

According to the original Jordan's algorithm a rotating vector determines the coordinates sequentially. An incremental step on one or the other axis is determined according to the minimal distance to the circle. The distance is calculated based on the implicit equation evaluation [3]. A MathLab implementation of the algorithm used for simulations is presented ANNEXE 1.
The major drawback of Jordan's algorithm is that the approximating point tracking the circle does not move at a constant speed. Immediate use with uniform timing of the algorithmically determined values in the generation of sine and cosine functions results in phase distortions.
The principle of operation of the proposed DDFS architecture is presented in Fig.1.

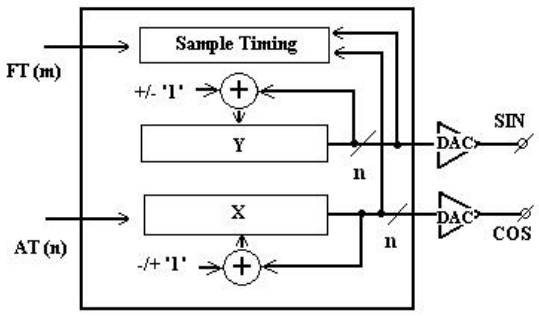

Fig 1. Block diagram of the amplitude based DDFS.

The solution of the problem for the compensation of the phase distortion in the generated signal is the preservation of the same non uniformity distribution of the samples in time as when the values where determined.

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Electronica Aplicata, Bd. V. Pârvan Nr. 2, 300223 Timişoara, e-mail ioan.jivet@etc.upt.ro

The sequence of coordinates generated are interleaved at periods proportional to calculated current step sine and cosine values [1].
The coordinate generator core implementing the algorithm in hardware receives as inputs AT(n) and FT(m), the amplitude and frequency tuning words of length n, m. Register X,Y implement sample value storage and also determine sample timing. The phase compensations sample timing can be set to any resolution smaller then n trading accuracy to maximum output signal frequency.
Simulations indicate that the compensation method generates low distortions signals. Due to the fact that the algorithm computes both the sine and cosine values of successive central angles the quadrature output is obtained.
The defining characteristic of the proposed architecture is that the desired resolution of the output signal is not generating a ROM like exponential resource demand. The main limiting factor only remaining is the rank of the DAC converter. Very pure signal generation is possible with minimal resources. The gate count cost involved with long word vector length is not exponential but linear. The spur frequencies in the output due to phase truncation have been eliminated entirely.
The amplitude tunning is simple to implement on a cycle basis by changing the central vector length. The amplitude of the generated signal and its output frequency can be adjusted to any value but must be correlated when a amplitude change is needed. Most PSK common modulation methods are supported .
A solution for frequency tunning is immediate as seen in the diagram by adding a period counter on the master frequency clock loaded at cycle start with the tuning word.
The very accurate and versatile FSK capacity of DDS architecture is preserved with frequency hopping speed dependent on frequency tuning counter loading speed only.
The quadrature output with a very high match inherent to DDS is preserved since the algorithm and there fore the hardware implementation determines both sine and cosine values simultaneously. In fact the two can not be determined independently.

III. AN EXTENDED PICOBLAZE ARCHITECTURE

The PicoBlaze microprocessor was designed and optimized for efficient and low cost implementation. A typical implementations in a FPGA together with a block RAM that stores the instructions can be simulated and its states tracked in the Xilinx ISE development environment.
The PicoBlaze soft processor is supported by development tools including an assembler, a graphical instruction set simulator in a integrated development environment (ISE). Furthermore, the PicoBlaze soft processor is provided as a free, source-level VHDL file with royalty-free re-use within Xilinx FPGAs.
For the purpose of this project the Mediatronix pBlaze IDE assembler was used. The DDFS module was implemented in assembly code using byte-wide arithmetic. The 16 byte-wide general-purpose data registers are just sufficient to accommodate the variables for two frequency synthesizer module simultaneously.
The basic amplitude generation module was implemented in PicoBase assembly code in both original eight bits wide world length and a extended double byte version.
The PicoBlaze assembly code in ANNEXE 2 outlines the calculus of the partial derivatives and implicit function values necessary for the decision of the step direction in the next iteration [1].
Considering the symmetry of the sine and cosine functions on the two axis results that values need to be determined for the first octant only. The same code is valid for other quadrants with increment or decrement coordinate steps per axis implemented by add or sub instructions respectively.
As it can be seen from the code used in the description only several compare, additions and increments are to be executed at each sample coordinates determination.
The timing necessary to generate the output signals was implemented as a extension of the basic PicoBlaze architecture as it is presented in Fig. 3. The top two upper output ports have been used to load the two timer/counters. The VHDL code of the original PicoBlaze was modified by adding two counter modules.
Sample timing was implemented by using the terminal counts of the timers to control the next sample load into the signal generator registers.

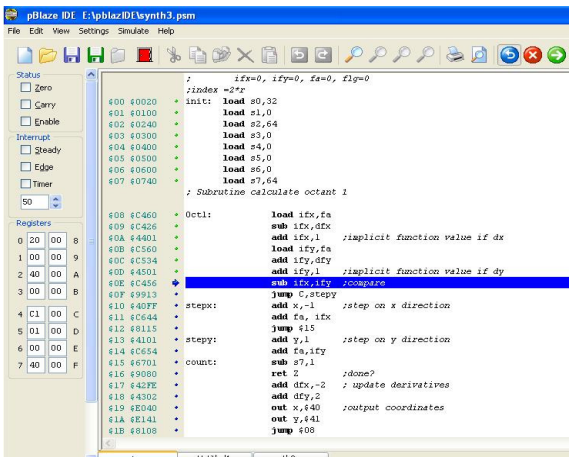

Fig. 2. PicoBlaze code graphical development and test environment for the direct digital synthesis module.

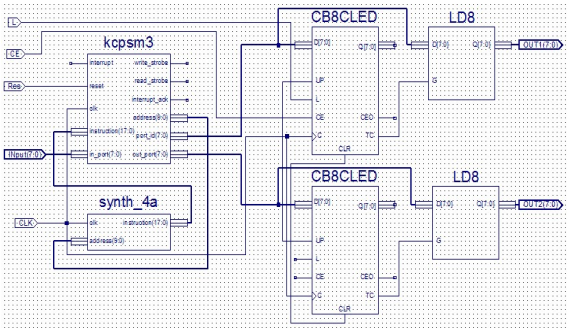

Fig 3. Timing architecture of a module for multiple frequency synthesis.

The lower first two output ports are used to hold output values. The frequency of the calculation and the timing compensation is resolved at this stage.

III. PICOBLAZE EXTENSION SYNTHESIS AND SIMULATION FOR SAMPLE APPLICATIONS

Every synthesized frequency application has its specific needs that are difficult to define in the early stages of the design. The project in this example is a ongoing project in the field of Electrical Bio-Impedance hardware development. The objective of the project is to design a flexible and 'tunable' frequency pair generator for use in the MIT tomography [9].

The amplitude based DDFS architecture as proposed was found as well suited for compact generator implementation. The design target device for the final implementation is the Xilinx Spartan 3 series.

The results of the ICE synthesis of the PicoBlaze and the timing circuit are presented in TABLE 1 clearly indicating the very low gate count. The complexity of the DDFS architecture is characterized by a similar economy of resources. A manually instantiated VHDL description of the amplitude based DDFS returned a resource count very close but lower.

The timing can be implemented by dividing of master clock. Direct implementation in VHDL of a flexible clock scheme using the capabilities the Xilinx DCM (Digital Clock Manager) would be difficult to accomplish [10].

TABLE I

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	76	3,840	1%
Number of 4 input LUTs	116	3,840	3%
Logic Distribution			
Number of occupied Slices	100	1,920	5%
Number of Slices containing only related logic	100	100	100%
Number of Slices containing unrelated logic	0	100	0%
Total Number of 4 input LUTs	186	3,840	4%

The PicoBlaze Digital Synthesizer as implemented in software was used to find solutions to generate simultaneously two frequencies using only slightly different amplitude tuning words on two instances of frequency synthesizer module and a same DCM scaled clock.

One simple solution identified was to generate the two frequencies on the same clock scaled and stabilized for jitter by one DCM. The amplitude minor differences can be resolved at the output buffer stage by truncating the values until identical amplitudes result.

A second sample example simulated was a stored chain of one bit step directions binary encoded. The code is very simple in this case and implies just one addition. Knowing that the picoBlaze can execute at 40 MIPS the generated sample frequency will approach 10 Mhz.

Further work is necessary to determine other methods of selectively calculate points on the circle as a mean to accommodate applications with MHz signal frequencies.

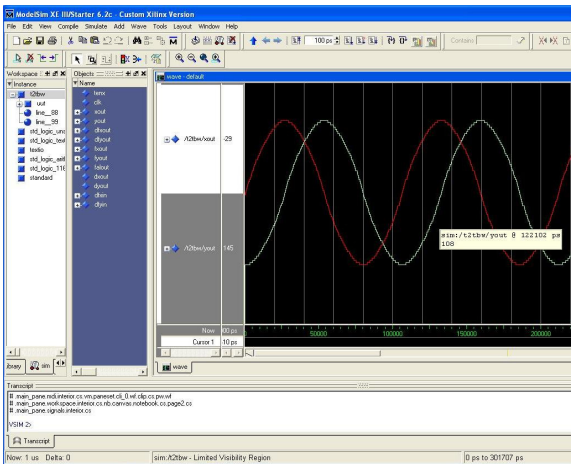

Fig.4 PicoBaze simulation of multi frequency signal generated using the amplitude based DDFS architecture.

VI. DISCUSSION

An extension of the Xilinx PicoBlaze soft processor with timers for digital signal synthesis applications targeting FPGA implementations is presented. The novel direct digital synthesis (DDS) amplitude sequencing architecture is used.

The Xilinx picoBlaze soft microprocessor assembly code for direct digital synthesis algorithm was verified using the graphical pBlazIDE tool.

The synthesis results clearly indicate the small number of circuit resources necessary.

The results of simulations for two sample applications are presented. One simple solution identified to generate two frequencies is to use the same clock and leveled up the amplitude by truncation to the smaller one.

A second sample simulated was a stored chain of one bit step directions binary encoded. It was found that the picoBlaze running at 40 MIPS can generate signal sample frequency in the 10 Ms/s range.

Future work needs to concentrate on selectively calculating samples in a cycle to accommodate high frequency applications over the 1 MHz range.