Digital Vector Modulator with Diagnostic Circuit for Particle Accelerator

Szymon Tarnowski, Adam Piotrowski
Department of Microelectronic and Computer Science
Technical University of Łódź
ul. Wolczanska 221/223
90-924 Łódź, Poland
email: szymon@dmcs.pl, komam@dmcs.pl

***Abstract*—Modern particle accelerators[8] have very strict requirements for controlling signal. To generate signal controlling field in cavities there is a possibility to use digital vector modulator and analog upconverter[6]. Hardware circuit consists of FPGA device, digital to analog converter and low pass filter. For experiments there is also a need to design and develop measurement circuit. Same circuit could be used in final application as diagnostics module. This enables an option for constant long term measurement and fault detection. This paper presents construction of such module.**

***Index Terms*—Vector modulation, Digital modulation, Diagnostics**

I. Introduction

Modern particle accelerators have very strict requirements for controlling signal[2]. To satisfy this demands there is a need to develop new kind of controlling system. In field of generation high frequency signal there are some new possibilities[7][4] but they have very high cost or other disadvantages (like signal stability, temperature instability) disqualify using them in commercial project. New research in this area shows there is an option to generate low frequency signal with digital modulator and then use upconversion to generate appropriate frequency. In Fig. 1 is shown analog part block diagram. It is single digital to analog converter (DAC) and upconversion mixer. This circuit needs a another reference signal (LO) with frequency higher (or lower) than desired frequency. Because operating frequency of this circuit is about 10MHz in low frequency part and about 1.3GHz in high frequency part there is a need to design measurement and diagnostics circuit.

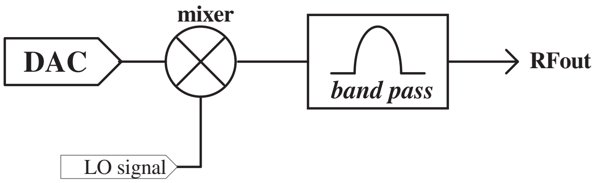


Fig. 1. Schematic diagram of analog part of analog upconverter

II. Vector Modulator

Vector modulator (VM) is a device or module which generate sinusoidal high frequency signal based on delivered parameters. As input parameters amplitude and phase can be used, but for many control systems using complex number makes arithmetic operation easier. Two values, real and imaginary part of input vector combine to IQ vector. Vector modulator can be build based on analog components, by digital signal synthesis or by low frequency synthesis with analog upconversion. To convert low frequency to high frequency there is needed difrential frequency usually called by local oscillator (LO). In XFEL considered LO frequency is master oscillator (MO) plus 9MHz or MO+13MHz. To generation of 9MHz or 13MHz signal with appropriate precision there is a need to clock circuit with 100MHz and more. Digital vector modulators for principle operation need to calculate samples using invert Fourier transform. This operation needs many multiplication so in FPGA implementation many multipliers are needed. Hardware multipliers have limited operating frequency, so new construction need to be developed[5]. Simple digital vector modulator design built with counter, sample look-up table and digital multipliers have limited operating frequency. The slowest part are multipliers. This problem can be defeated with new calculation scheme, all the samples must be calculated before signal is generated and then sent in correct order to DAC. Each time IQ vector changes new recalculation is needed. This will add some delay between IQ vector change and output signal update. Calculation of this delay will be shown later. For faster operation lookup table with calculated samples is constructed as circular buffer. When new data is ready multiplexer select new data, after loading time, circular buffer is operating normally.

A. Block diagram and state diagram

Block diagram shown in Fig. 2 consists of following components:

- Machine, Finite State Machine (FSM), control and timing block
- Comparator, compares current IQ vector with input IQ vector
- Multiplexer, selects which data load into shifter (new or old)

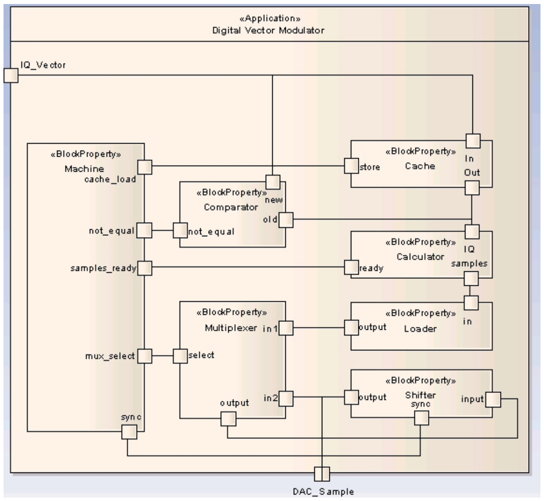


Fig. 2. Block Diagram

- Calculator, calculate new values of samples from IQ vector and stored internally sine and cosine tables
- Loader and Shifter, shifting registers
- Cache, temporary register for storing current IQ vector

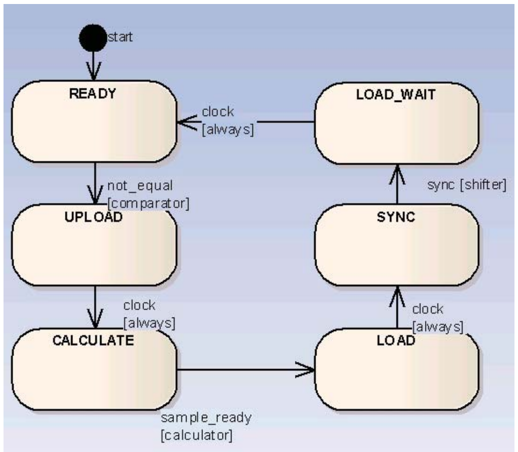


Fig. 3. State Diagram

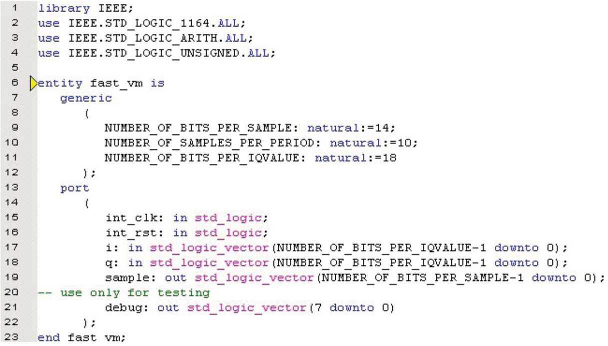
State diagram is shown in Fig. 3, when the vector modulator starts, waits for new IQ vector. When value of current IQ vector is different than IQ vector stored in cache register (at the start reseted to zero), calculation process begins. The sample update process consists of following steps:

1) **UPLOAD** New IQ vector is stored in cache register, takes 1 cycle
2) **CALCULATE** Calculation of samples, in parallel all samples are calculate, takes at least 1 cycle, depends on multipliers in FPGA
3) **LOAD** New samples are loaded parallel into Loader register, takes 1 cycle
4) **SYNC** FSM waits for synchronization between shifter current state and first sample of output signal period, time variable, takes at least 1 cycle, up to to number of samples per period cycles
5) **LOAD_WAIT** FSM waits until all samples from loader are stored in shifter, takes number of samples per period cycles
6) **READY** FSM ends operation, and waits for new data, data to DAC is sent via circular buffer, state is stable until IQ Vector update

III. VM IP Core Features

Vector modulator was written in VHDL language and prepared as complete module (IP core).

A. Flexible configuration


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity fast_vm is
   generic
      (
         NUMBER_OF_BITS_PER_SAMPLE: natural:=14;
         NUMBER_OF_SAMPLES_PER_PERIOD: natural:=10;
         NUMBER_OF_BITS_PER_IQVALUE: natural:=18
      );
   port
      (
         int_clk: in std_logic;
         int_rst: in std_logic;
         i: in std_logic_vector(NUMBER_OF_BITS_PER_IQVALUE-1 downto 0);
         q: in std_logic_vector(NUMBER_OF_BITS_PER_IQVALUE-1 downto 0);
         sample: out std_logic_vector(NUMBER_OF_BITS_PER_SAMPLE-1 downto 0);
-- use only for testing
         debug: out std_logic_vector(7 downto 0)
      );
end fast_vm;
```

Fig. 4. VM IP Core definition

VHDL entity is internal configurable. VHDL entity definition is shown on Fig. 4 There are 3 parameters:

- **NUMBER_OF_SAMPLES_PER_PERIOD**, number of samples per period of output signal
- **NUMBER_OF_BITS_PER_SAMPLE**, number of bits per one sample (sample width)
- **NUMBER_OF_BITS_PER_IQVALUE**, number of bits per one variable of vector (I or Q)

B. VHDL sample value calculation

VM IP core during VHDL synthesis calculate itself samples of sine and cosine. In synthesis output there is added for debugging dump of values in real, integer and binary format. Exemplary debug is shown on Fig. 5.

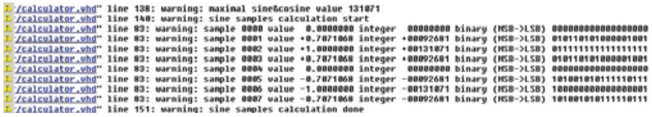

```
/calculator.vhd" line 138: warning: maximal sine&cosine value 131071
/calculator.vhd" line 140: warning: sine samples calculation start
/calculator.vhd" line 83: warning: sample 0000 value  0.0000000 integer  00000000 binary (MSB->LSB) 000000000000000000
/calculator.vhd" line 83: warning: sample 0001 value +0.7071068 integer +00092681 binary (MSB->LSB) 010110101000001001
/calculator.vhd" line 83: warning: sample 0002 value +1.0000000 integer +00131071 binary (MSB->LSB) 011111111111111111
/calculator.vhd" line 83: warning: sample 0003 value +0.7071068 integer +00092681 binary (MSB->LSB) 010110101000001001
/calculator.vhd" line 83: warning: sample 0004 value  0.0000000 integer  00000000 binary (MSB->LSB) 000000000000000000
/calculator.vhd" line 83: warning: sample 0005 value -0.7071068 integer -00092681 binary (MSB->LSB) 101001010111110111
/calculator.vhd" line 83: warning: sample 0006 value -1.0000000 integer -00131071 binary (MSB->LSB) 100000000000000001
/calculator.vhd" line 83: warning: sample 0007 value -0.7071068 integer -00092681 binary (MSB->LSB) 101001010111110111
/calculator.vhd" line 151: warning: sine samples calculation done
```

Fig. 5. Exemplary sine sample calculation

C. Update delay

Delay between new IQ vector and output signal change is variable, but can be calculate according equation $T_{update} = T_{ready} + T_{upload} + T_{calculate} + T_{load} + T_{sync} + T_{load\_wait}$ where:

- T_{ready} = 1 T_{cy}
- T_{upload} = 1 T_{cy}
- $T_{calculate}$ = 1 T_{mul}
- T_{load} = NUMBER_OF_SAMPLES_PER_PERIOD T_{cy}
- T_{sync} = 1 . . . NUMBER_OF_SAMPLES_PER_PERIOD T_{cy}
- T_{load_wait} = NUMBER_OF_SAMPLES_PER_PERIOD T_{cy}

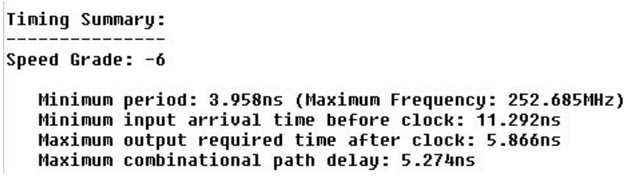
where:

T_{cy} equals time between clock cycles ($\frac{1}{Fclk}$)

T_{mul} equals time of asynchronously working multiplier time of calculation (depends on FPGA architecture)

D. Operating frequency

To test described project, there were made laboratory experiments using MEMEC Virtex2P test-board. For synthesis and implementation Xilinx ISE v9.0 was used. First test were made in very low clocking frequency (about 1Hz) to test machine operation. Input data (IQ vector) was set as table of 4 values which were selected asynchronously by dip-swiches. Circuit was checked with LED connected to debug output. State changes presented by LED were exactly as designed. For final test, clocking was from integrated oscillator multiplied by digital clock manager (DCM). Synthesizer timing comments are shown on Fig. 6. Synthesis was made for FPGA Virtex2P, which has internal multipliers capable of running at about 100MHz. This design is runing up to 250MHz. Implementation was tested up to 200MHz on Memec board. Project was not operating at 250MHz, probably due to DCM limitation or poor clock routing on the test board.



```
Timing Summary:
---------------
Speed Grade: -6

   Minimum period: 3.958ns (Maximum Frequency: 252.685MHz)
   Minimum input arrival time before clock: 11.292ns
   Maximum output required time after clock: 5.866ns
   Maximum combinational path delay: 5.274ns
```

Fig. 6. Xilinx ISE timing report

IV. Diagnostic Circuit

To test operation diagnostic circuit was designed. This subsystem can be divided into two parts. The first one is analog part and the second one is digital acquisition circuit. Before circuit design there were following requirements defined:

- High input impedance
- Capability to drive 50Ω input
- Monitoring output
- Low pass analog filters (about 13MHz) with bypass option
- Variable attenuator with bypass option
- Low frequency root mean square amplitude measurement (TrueRMS)
- High frequency power measurement (in external module)

V. Analog System

For experiments diagnostic circuit was integrated with analog filters on one board. Filters were designed with fast operational amplifiers (Analog Devices AD8014). Forth order low pass filter (two steps) was calculated as Butterworth filter. Values were tested in Spice program. Calculated cutoff frequency was set at 13MHz. Measured characteristic is shown in Fig. 7. To maximize amplitude characteristic of frequency mixer (below compression point) and reduce gain ripple there was a need to add variable attenuator in board. For this was selected digital potentiometer (MAXIM DS1267) controlled by system. This component has small bandwith, but if add some external components it is possible to extend bandwidth [1]. This will limit range of regulation, but in this circuit full range regulation is not necessary. Board consists also of output buffer for driving 50Ω input of mixer, relays bypassing filter and attenuator and also TrueRMS (Analog Devices AD637) converter with signal conditioning amplifier. Signal conditioning amplifier is needed because used TrueRMS converter is working near to its bandwidth limit. Block diagram of analog part is shown in Fig. 8.

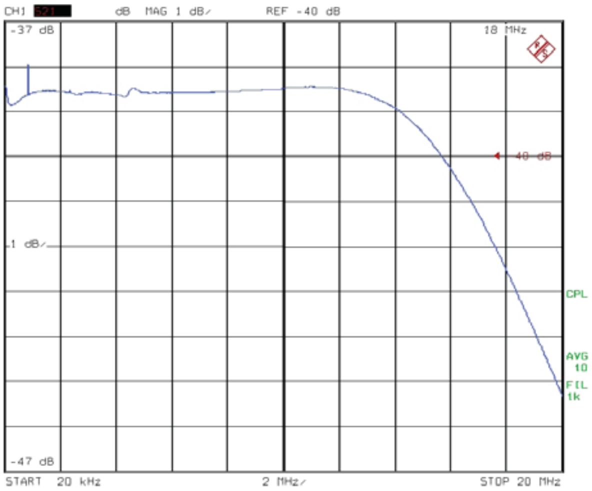


Fig. 7. Measured filter characteristics

VI. Control and Acqusition System

Acquisition system consists of two parts. The first one is hardware board with micro-controller. The second one is software for PC computer.

A. Hardware board

Hardware board is based of 8-bit micro-controller (Microchip PIC18F4523) with build in 12bit ADC. Communication between PC computer and board is made with EIA-232 interface. To achieve precision and recurrence external precision voltage reference source was added (Texas Instruments REF3040).

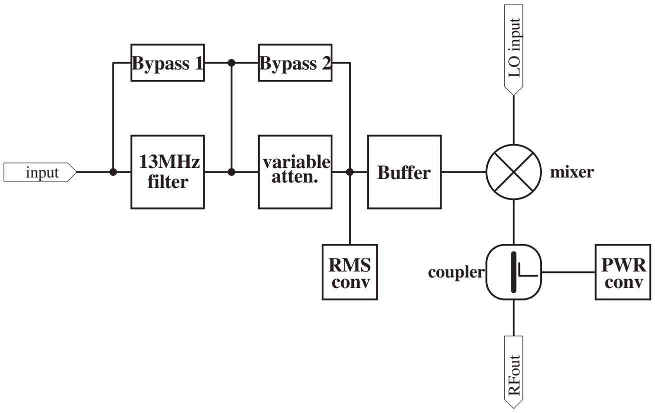


Fig. 8. Analog system block diagram

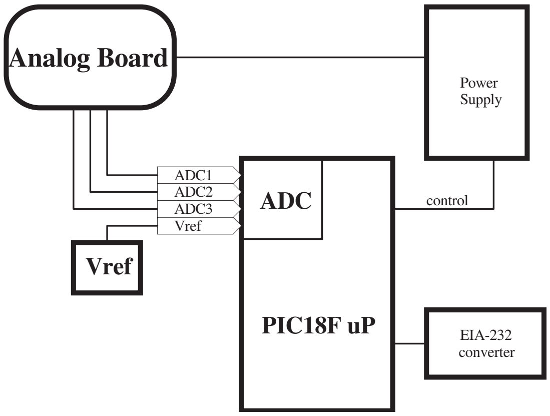
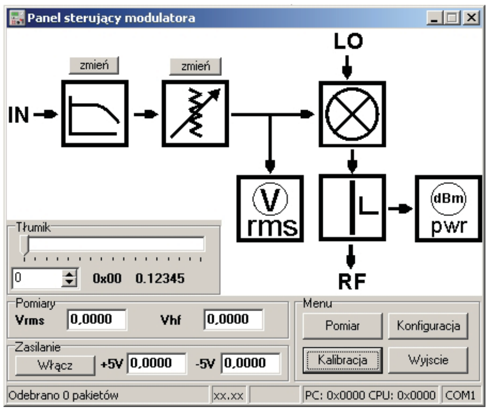


Fig. 9. Block diagram of daq board

B. PC application

To control device and read values from simple application was developed. Program was written under Borland Delphi environment. Application interface is shown in Fig. 10. To increase precision of measurement application makes multiple measurements and averages values.



Tłumik: 0 0x00 0.12345

Pomiary: Vrms 0,0000 Vhf 0,0000

Zasilanie: +5V 0,0000 -5V 0,0000

Odebrano 0 pakietów | xx.xx | PC: 0x0000 CPU: 0x0000 | COM1

Fig. 10. Application screen-shot (interface in Polish language)

VII. Conclusion

Presented solution was tested in laboratory and proved its operatibility. Proposed circuits could be implemented in final version of system integrated on ATCA[3] board or AMC module. Active filter based on discreete components does not have enough slope for this application. There are two possibilities how to deal with this problem. The one solution is to increase clocking frequency and the second one is to use passive ceramic filters with high Q-factor.

Acknowledgments

The research leading to these results has received funding from the European Commission under the EuCARD FP7 Research Infrastructures grant agreement no. 227579.

The author is a scholarship holder of project entitled ”Innovative education ...” supported by European Social Fund.

The work was supported by the Ministry of Science and Higher Education (grant agreement no. 2273/B/T02/2008/35).

References

[1] Maxim Integrated Products. Application Note 3081: How to Increase the Bandwidth of Digital Potentiometers 10x to 100x. 2004.

[2] S. Simrock. Achieving phase and amplitude stability in pulsed superconducting cavities. Technical report, Deutsche Elektronen-Synchrotron DESY, 2001.

[3] S. Simrock. Requirements for the ATCA based LLRF Evaluation System. In *Review of LLRF system based on ATCA standard*, 8-9 Nov 2007.

[4] Sz. Tarnowski. High frequency signal synthesis for particle accelerator control. *IX International PhD Workshop OWD'2007, 20-23 October 2007*.

[5] A. Piotrowski Sz. Tarnowski. Uniwersalny modulator wektorowy syntezowalny do ukadu fpga. *Reprogramowalne Ukady Cyfrowe, RUC 2008*, 2008.

[6] Sz. Tarnowski, A. Piotrowski. Application of precision low frequency signal generation as intermediate frequency for upconversion in particle accelerator field control. *MIXDES 2008*.

[7] Sz. Tarnowski, A. Piotrowski, A. Napieralski. Direct digital modulation system for LLRF. *IEEE-SPIE WILGA 2007*.

[8] Tesla collabolation members. A Proposal to Construct and Test Prototype Superconducting RF Structures for Linear Colliders. Technical report, CERN/DESY/FERMILAB, 1993.