FUNCTION

PWM modulator

VHDL File

pwmmod.vhd

Applicable Devices

Spartan 3A DSP, Spartan 6, Kintex 7, Zynq

Xilinx primitive used

DSP48A

Sub modules used

ldivision.vhd
pwmshape.vhd

Execution time

49 clocks for 3-phases

36 clocks for 2-phases

Introduction

This block realizes a PWM modulator to control stepper and three-phase motors. The modulator uses the intersecting method between two waveforms to generate the PWM modulation. The reference waveform is user programmable (in the IP core there are two default waveforms), while the other waveform is a triangular wave. The IP Core has 4 tables to store the user defined waveforms.

Detailed Description

The IP Core implements a PWM (pulse width modulation) modulator that can drive three-phase motors or bipolar motors. The method used to generate the PWM is the intersecting method, which requires two waveforms; one is the reference waveform the other is the modulation waveform. The first can be a sine, cosine or any other waveform defined by the user and stored in a table, the second is a triangular wave generated by the IP Core with a frequency that depends on the clock and the ***pwm_presc[16:0]*** parameter.

PWM parameters

The calculation of a new PWM waveform begins when the ***start*** signal is driven high for 1 clock cycle; the end of the calculation of the new modulation is signaled by the process when the ***finish*** signal is set high for 1 clock cycle. At this point the modulator generates a PWM using the new input parameters.

The ***dc_link[16:0]*** UNSIGNED17 parameter is the motor power supply voltage value from the A/D. The unit weight depends on the A/D acquisition system.

The ***table[1:0]*** parameter selects a waveform between four waveforms that are stored in the DPR. These tables can be overwritten by the user. In the default configuration at the addresses 0x0 and 0x2 there is a cosine waveform (see Figure 1); at the addresses 0x1 and 0x3 there is a cosine waveform mixed with a harmonic so that a zero-signal-insertion modulation can be generated when using a three-phase motor (see Figure 2).

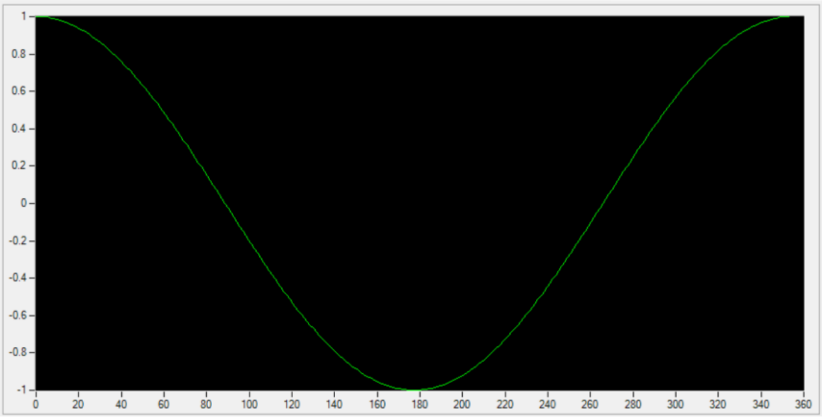

Figure 1: Cosine waveform stored in the table at address 0[1]

[1] The figure is only a quality representation of the table

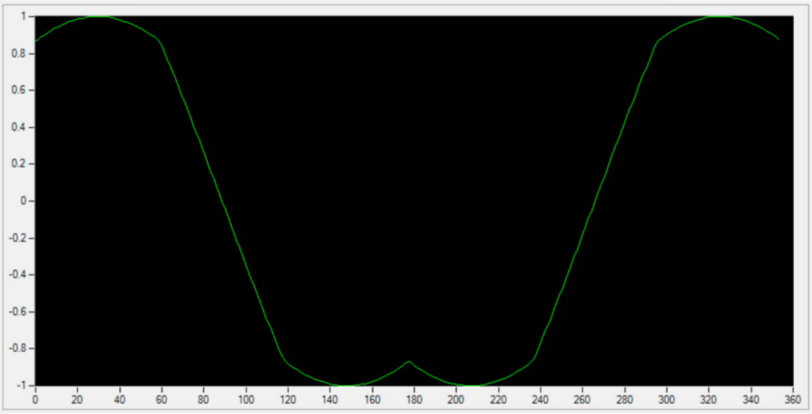

Figure 2: cosine+3rd harmonic waveform stored in the table at addresses 1[1]

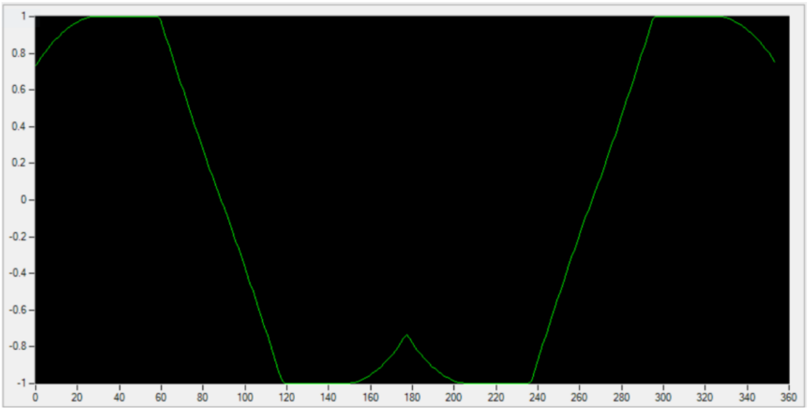

Figure 3 - Discontinuous mode DPWM3 is stored in the table at address 2

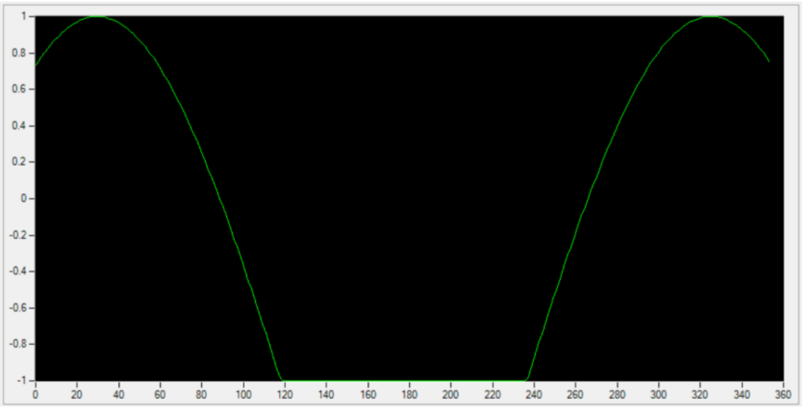

Figure 4 - Discontinuous mode DPWMin is stored in the table ad address 3

The waveforms stored in tables 1,2 and 3 are space vectors so they emit about 15% more voltages than table 0 that is pure Cosine waveform.

The tables 2 and 3 are both discontinuous. Comparing with continuous space vector they produce less switches to the gate units. The drawbacks is higher THD.

The ***is3phases*** input flag selects the type of motor to drive.

is3phase value	Motor type
0	stepper motor
1	three-phase motor

Table 1: motor type selection

The ***p3cdvx [2:0]*** PWM output is the coil driver signal used for 3-phases motors. The p2cdv1[1:0]and p2cdv2[1:0] PWM output is the coil driver signal used for 2-phases bipolar stepper motors.

Mode	p3cdvx (0)	p3cdvx (1)	p3cdvx (2)
three-phase motor	Drive phase 1 (A=0) Phi=0	Driver phase 2 Phi=240	Driver phase 3 Phi=120

Table 2: drive of the coils 3-phase bldc/pmsm

Mode	p2cdv1 (0)	p2cdv1 (1)	p2cdv2 (0)	p2cdv2 (1)
stepper motor	Drive-high of coil-A Phi[2]=0	Drive-low of coil-A Phi=180	Drive-high of coil-B Phi=270	Drive-low of coil-B Phi=90

Table 3: drive of the coils 2-phases bipolar stepper

The ***pwm_presc[16:0]*** UNSIGNED17 sets the PWM frequency (f_{PWM}). The IP Core runs at the ***clock*** rate (f_{clock} is the clock frequency), so the PWM frequency depends both from the ***clock*** and the PWM prescaler parameter. The value can be changed at any time.

$$f_{PWM} = \frac{f_{clock}}{2 \cdot (pwm\_presc[16:0] + 1)}$$

Equation 1: PWM frequency

$$pwm\_presc[16:0] = \frac{f_{clock}}{2 f_{PWM}} - 1$$

*Equation 2: calculation of **pwm_presc[16:0]***

Example: f_{clock}=50MHz, f_{PWM}=20KHz, pwm_presc[16:0] = 50,000,000/(2*20,000)-1=1249

The output flag ***pwm_mdovf*** indicates the modulator overflow (“1”=overflow).

[2] The *phi* (phase offset) refers to input argument ***angle[24:0]*** that represents the stator reference angle.

The ***angle[24:0]*** UNSIGNED25 input parameter sets the desired electrical output angle. To transform the parameter in degrees the Equation 3 can be used.

$$\theta° = \frac{360}{2^{25}} \cdot angle[24:0]$$

*Equation 3: **angle[24:0]** to degrees transformation*

The parameter ***modulo[16:0]*** UNSIGNED17 set the desired output module. This parameter is multiplied for the parameter ***pwm_kmod[31:0]*** UNSIGNED32

The ***pwm_mdmax[16:0]*** UNSIGNED17 argument limits the modulation value. If modulation value is greater than limit, the value is reduced to the specified limit and the ***pwm_mdovf*** (modulation overflow indicator) is set to "1".

The ***pwm_mdval[16:0]*** UNSIGNED17 output report the modulation value in the range 0..***pwm_mdmax[16:0]*** and ***pwm_presc[16:0]*** is 100%.

$$pwm\_mdval[16:0] = \frac{modulo[16:0] \cdot pwm\_kmod[31:0]}{dc\_link[16:0] \cdot 2^{14}}$$

Equation 4: pwm_mdval[16:0] calculation

The ***pwm_mhovf*** output flag indicates an internal math overflow in a temporary variable.

The PWM value is defined by the following algorithm:

```
tmp = ((pwm_kmod / dc_link) modulo)  * 211
if tmp > pwm_mdmax then
        tmp = pwm_mdmax;
        pwm_mdovf = 1;
else
        pwm_mdovf = 0;
end
pwm_mdval = tmp
```

Common mode offset in 3-phase mode

The input signal ***cmmofs*** can be used to split residual for modulation index 50% to floor/ceil.

The adding of common mode increases the minimum output pulse width reducing stress to power stage. The feature works in 3-phase mode only and for modulation index less than 100%.

Waveforms consideration and programming

The waveform entries are normalized to UNSIGNED17 (range 0..131071) and in case of stepper motors (***is3phases***=0) the signed is used to drive the bipolar coil.

The PWM value is evaluated by: pwm_value = (Pwm_mdval * waveform_entry) / 131072. The resulting value does not exceed Pwm_mdval.

An internal (initialized) DPR memory is used to hold up to four modulator waveforms. One port of DPR is used (read-only) by PWM modulator to generate coil commands. The other port of DPR can be optionally connected to a Microblaze processor and overwrite the default waveforms on the fly.

The signals used to interface the modulator DPR to Microblaze are:

Mem_en: access enable when '1'.
Mem_we: write access when '1' else read access.
Mem_addr[9:0]: UNSIGNED10 DPR address. The two MSB's refer the modulation table and the others eights LSB's refer to desired value.
Mem_din[17:0] : SIGNED18 data input used on write cycle to overwrite DPR location.
Mem_dout[17:0] : SIGNED18 data output loaded with selected DPR location content.

PARAMETERS

Parameter	Type	Values	Default	Description
C_FAMILY	string	spartan3adsp spartan6 kintex7	spartan3adsp	Xilinx FPGA Family name

SIGNALS

Signal[3]	I/O	Description
clock	IN	Clock (rising edge).
reset	IN	Reset. Active high.
Dual port ram control interface signals		
mem_en	IN	Memory enable. Active high.
mem_we	IN	Memory write enable. Active high.
mem_addr[9:0]	IN	Memory address. Bit [9:8] select the modulation table; bit [7:0] select the memory address inside the table.
mem_din[17:0]	IN	Input data to write in the selected DPR memory address. SIGNED18.
mem_dout[17:0]	OUT	DPR data output.
PWM control parameters/ouputs		
dc_link[16:0]	IN	Motor power supply voltage from the A/D. UNSIGNED17.
pwm_kmod[31:0]	IN	Parameter used in the PWM duty cycle evaluation. UNSIGNED32.
pwm_presc[16:0]	IN	PWM prescaler setting. UNSIGNED17.
pwm_mdmax[16:0]	IN	Max output modulation. UNSIGNED17.
pwm_mdval[16:0]	OUT	PWM modulation value. Range is 0 to pwm_mdmax (prescaler=100%). UNSIGNED17.
pwm_minpw[16:0]	IN	Minimum pulse width. 0=disable
table[1:0]	IN	Select one of the 4 waveforms stored in the DPRAM.
cmmofs	IN	Common mode offset
is3phases	IN	Select Bipolar motor (low='0') or tri-phases (high='1') motor.
start	IN	Start calculation. The pulse width must be of 1 clock. Active high.
angle[24:0]	IN	Input angle. UNSIGNED25
modulo[16:0]	IN	Input module. UNSIGNED17
pwm_mdovf	OUT	Modulator overflow flag. Active high.
finish	OUT	End of calculation pulse. The pulse width is of 1 clock. Active high.
p3cdvx[2:0]	OUT	3-phases output coil driver
p3sync	OUT	3-phases output sync
p2cdv1[1:0]	OUT	2-phases coil-1 output driver
p2cdv2[1:0]	OUT	2-phases coil-2 output driver
p2sync	OUT	2-phases output sync

[3] The width of the signals is the default value (the width of some signals is user defined).

TIMING PERFORMANCE AND RESOURCE USAGE

This section provides data on the timing performance and resource utilization of the core. Performance has been obtained on one representative device from the Spartan-3 Generation and Spartan 6 families of FPGAs. The following tables lists the devices used for characterization.

Execution Time

output	input	clock cycles[4]
finish	start	52 (3-phases) 39 (2-phases)

[4] Unless otherwise noted.

Reference Documents

1. Xilinx LogiCORE IP DSP48 Macro V2.1 [DS754 March 1, 2011]
2. Xilinx LogiCORE IP Block Memory Generator V6.1 [DS512 March 1, 2011]

Support

QDESYS provides technical support for this LogiCORE product when used as described in the product documentation.

QDESYS cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

Ordering Information

For information on pricing and availability of QDESYS modules and software, please contact info@qdesys.com

Revision History

Date	Version	Description
17/08/2011	1.0	Initial QDeSys release
09/10/2011	1.1	Fixed reference to Table 2. Fixed the pwm_mdval formula.
20/11/2011	1.2	Extended signals width and removed configuration parameters. Removed pwm_dmod signal.
23/12/2011	1.3	Added parameters for FPGA family
12/05/2012	1.4	Added Kintex 7 and Zynq support
10/07/2013	1.5	Programming on the fly support
02/03/2015	1.6	Separate outputs for 3-phases and 2-phases
20/12/2017	1.7	Common mode offset for 3-phase output, update module signal interface
April 11, 2022	1.8	Minimum Pulse Width, synchronization of programming parameters, add two discontinuous PWM modes.

Disclaimer

In disclosing the information contained in this document QDeSys assumes no obligation to correct any errors herein contained, or to advise you of any corrections or updates. QDeSys expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the information.

THE DOCUMENTATION IS DISCLOSED TO YOU "AS-IS" WITH NO WARRANTY OF ANY KIND. QDESYS MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL QDESYS BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL

DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION