Key Design Features

- VHDL source code of a PWM generator component
- Configurable duty cycle resolution
- Configurable number of outputs/phases
- Configurable PWM frequency
- Modulation around the center of the pulse
- PWM inverse outputs

Applications

- Motor Control
- Telecommunications
- Power delivery
- Voltage regulation

Pin-out Description

Pin name	*I/O*	*Description*	*Active state*
clk	in	Synchronous clock	rising edge
reset_n	in	Asynchronous reset	low
ena	in	Enable	high
duty[M..0][1]	in	Duty cycle	data
pwm_out[N..0][2]	out	PWM output pin/pins	data
pwm_n_out	out	Inverted pwm output pin/pins	data

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
sys_clk	System clock frequency	integer	≥1
pwm_freq	PWM switching frequency	integer	≥1
bits_resolution	Bits of resolution setting the duty cycle	integer	≥1
phases	Number of output pwms and phases	integer	≥1

[1] M is the duty cycle's specified resolution in bits, set by the bits_resolution generic.
[2] N is the specified number of outputs (and phases), set by the phases generic.

Block Diagram

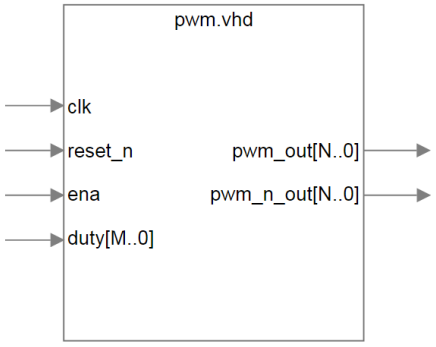

Figure 1: PWM architecture

General Description

This details a pulse width modulation (PWM) generator component for use in CPLDs and FPGAs, written in VHDL. The component outputs PWM signals based on the duty cycle set by user logic. The center of each pulse occurs at the PWM frequency, and the pulse width varies around the center. If set to multiple phases, the component generates one PWM signal for each phase, evenly spaced. For example, when set to three phases, it generates three PWM outputs 120° out-of-phase with one another. The component was designed with Quartus II, version 13.1 and tested with ModelSim-Altera 10.1d. Resource requirements depend on the implementation.

Theory of Operation

The system clock divided by the PWM frequency equals the number of system clock pulses in one PWM period. Counters define this PWM period for each phase. There is one counter for each PWM phase, with their values offset by the phase. Each counter increments on each system clock and clears once it reaches the end of its period.

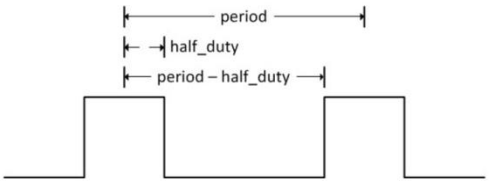

Figure 2: Waveform of a Pulse In-Phase with the PWM Period

The duty cycle determines the points during the period when the PWM signal's rising and falling edges occur. Figure 2 illustrates the basic concept used to determine these positions. The signal's falling edge happens at ½ duty cycle, and its rising edge happens at the end of the period minus ½ duty cycle. Once the counter reaches each of these positions, the PWM signal is toggled as appropriate. Since a

half duty cycle can never exceed a half period, the falling edge always occurs before the rising edge.

Configurable Parameters

The PWM generator is configured using four GENERIC parameters, set in the ENTITY. The PWM generator does not require a specific input clock, so long as the user sets the sys_clk parameter to the clock frequency provided. The parameter pwm_freq corresponds to the PWM frequency. The bits_resolution determines the resolution of the pulse width. For example, a value of 8 provides 8 bits of resolution. Therefore, the pulse width's resolution is 28 or 256, so in this case, the finest possible pulse width adjustment is the period (i.e. 1/pwm_freq) divided by 256. The parameter phases sets the number of outputs and their relation to one another. The number of PWM outputs is phases, and these outputs are 360°/phases out-of-phase with one another.

Since the PWM period is defined in system clocks as sys_clk/pwm_freq, this ratio also affects the duty cycle resolution. A duty cycle resolution is not achievable if it exceeds the number of system clocks in the PWM's period. Similarly, the achieved duty cycle is subject to single bit rounding errors if the period is not an integer multiple of the resolution.

Controlling the Duty Cycle

User logic can control the duty cycle by latching in new duty cycle values on the duty port. The PWM generator latches in values on this port on all system clocks when the ena port is set to '1'. Figure 3 shows a ModelSim simulation changing the duty cycle. Note when changing duty cycles that there is one pulse of intermediate width between the original and new pulse widths. This is due to the new duty cycle taking effect in the center of the pulse, so that pulse's width is half of the old pulse width plus half of the new pulse width.

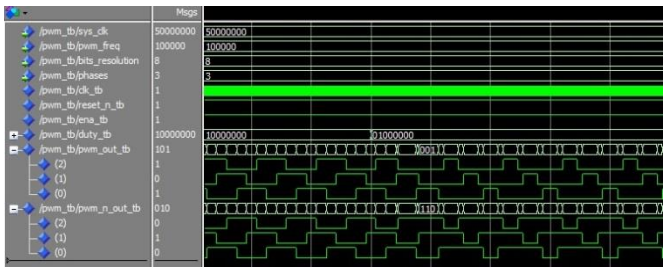

Figure 3: Simulation of Changing the Duty Cycle

Reset

The reset_n input port must have a logic high for the PWM generator component to operate. A low logic level on this port asynchronously resets the component. During reset, the component clears the period counters and sets both the PWM outputs and PWM inverse outputs to '0'.

Functional Timing Diagrams

none

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision	27/02/2017