SPI Slave to PWM Generation

April 2011 Reference Design RD1107

Introduction

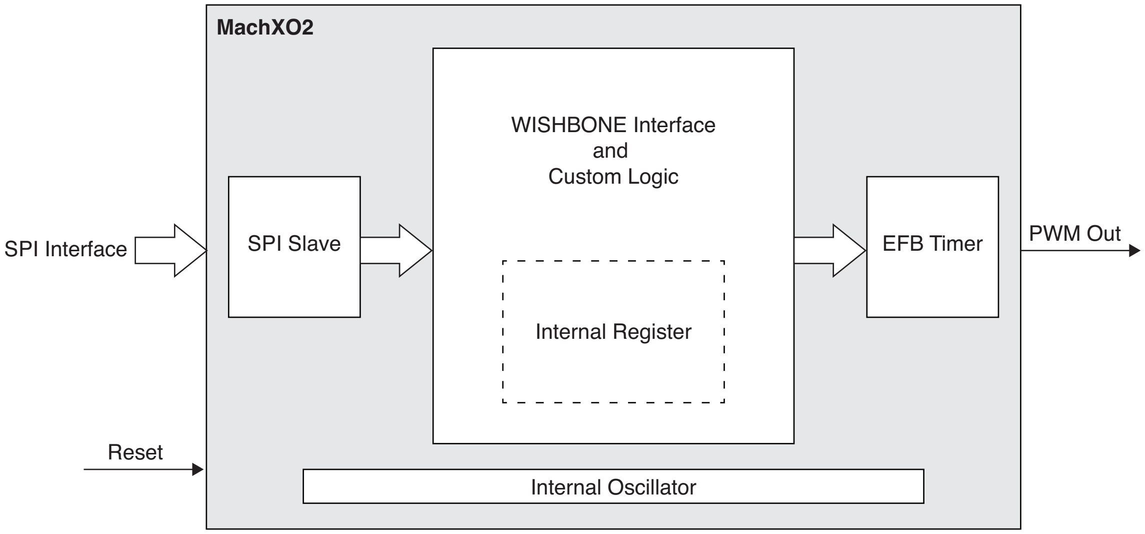
Pulse-width modulation (PWM) uses a rectangular pulse wave whose pulse width is modulated resulting in the variation of the average value of the waveform. Every PWM signal is a continuous succession of high and low pulses. The length of each pulse is defined by the desired duty cycle and frequency.

In mobile phones and other consumer electronic products, the Light Emitting Diode (LED) is increasingly being used as a display backlight. PWM offers an ideal solution for LED controllers as the dimming intensity of the LED can be controlled by changing duty cycle and frequency of the pulse.

This design provides a bridge between a microprocessor and a PWM generator. The SPI slave interface is used to receive command and data from an external SPI master. The command and data in turn are used to set the frequency and duty cycle of the PWM. In this design the Embedded Function Block (EFB) in the MachXO2™ device is used to generate the PWM signal.

A typical application of this design includes interfacing a SPI compliant on-board microprocessor and a LED device. This design can also be used as a reference to generate PWM for analog dimming.

Figure 1. Top-Level Block Diagram



Features

This design provides a bridge between the microprocessor and PWM generator via a SPI slave interface. The features include:

- Programmable frequency varying from 1 KHz to 100 KHz
- Programmable duty cycle
- Built-in look up table (LUT) for frequency and duty cycle counter generation
- Internal oscillator to generate clock signal

Functional Description

This design consists of three programmable registers for PWM signal generation. These registers are programmed by the microprocessor via the SPI interface. The internal oscillator is used to generate the clock signal for the design.

Table 1 lists the I/O ports of the design.

Table 1. Pin Descriptions

Signal	Width	Type	Description
SPI Interface			
sclk	1	Input	Serial clock
mosi	1	Input	Serial data in
miso	1	Output	Serial data out
SSn	1	Input	Slave select
PWM Interface			
Pwm_out	1	Output	PWM signal
Rst_n	1	Input	Master reset

Design Description

The design has multiple sub-modules as shown in Figure 1. These modules include logic for SPI Slave, WISHBONE interface, internal registers, and embedded timer block. The SPI slave can operate with different CPOL, CPHA and DIRECTION settings. The top-level generic sets these parameters. The SPI slave receives a command from the external SPI master. The custom logic then programs the appropriate register in timer block.

Table 2 lists the internal registers of the design. These are WRITE only registers.

Table 2. Internal Register

Internal Register	Address	Width	Access	Description
Command	0x00	8 Bits	W	Control register
Frequency	0x01	8 Bits	W	Frequency for the PWM signal
Duty cycle	0x02	8 Bits	W	Duty Cycle for the PWM signal

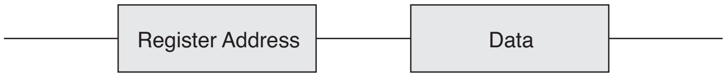
Table 3 lists descriptions of the internal register bits.

Table 3. Internal Register Bits

Name	Address	Width	R/W	Purpose
Command	0x00	8 Bits	W	The purpose of this register is to enable or disable the EFB Timer block. Bit[7]: Enable timer block Bit[6]: Disable timer block Bit [5-0]: Reserved for future purpose
Frequency	0x01	8 Bits	W	The purpose of this register is to divide the clock frequency by the programmed value. Bit[7]: Reserved Bit[6:0]: Frequency value 0,1- Frequency will be set to 1 KHz >100- Frequency will be set to 100 KHz Value 1- 99 will correspond to frequency between 1 KHz to 100 KHz
Duty Cycle	0x02	8 Bits	W	The purpose of this register is to set duty cycle for the PWM signal. Bit[7]: Reserved Bit[6:0]: Duty cycle Value >100 will correspond to duty cycle of 100%

SPI Command Format

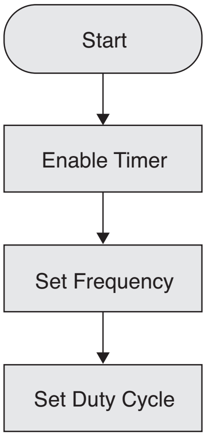
The SPI slave command format is quite simple. An external SPI Master needs to send the 8-bit register address followed by the data. There are no protocol overheads such as start, stop, status or acknowledgement token/packets associated with SPI transfer. All internal registers are write-only so there is no data present on the MISO line.



Common Operation Sequence

The design operation sequence is quite simple as well. Using the SPI interface, sequentially program the three internal registers of the design. The design will start driving the PWM output according to the programmed frequency and duty cycle.

Figure 2. SPI Slave to PWM Generation Flow



HDL Simulation and Verification

The SPI slave based PWM controller design is simulated using a standard SPI master model. The model programs the internal register of the timer-counter block. This model contains tasks for programming various registers in the design.

The following timing diagrams show the major timing milestones in the simulation.

Figure 3. Programming of the Control Register

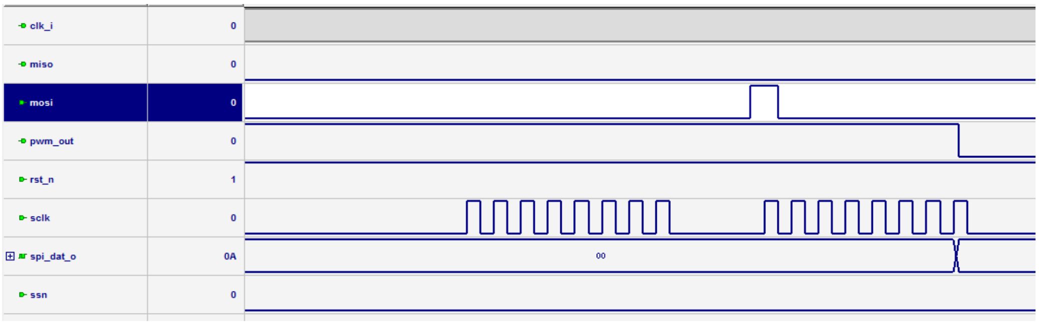


Figure 4. Programming of the Frequency Register

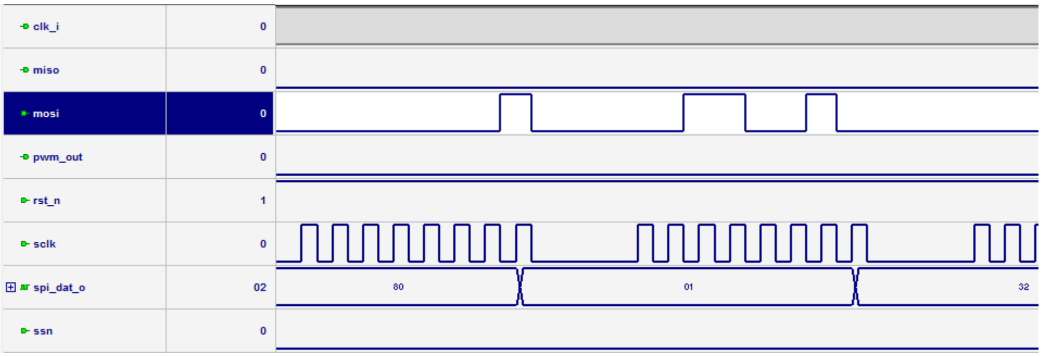


Figure 5. Programming of the Duty Cycle Register

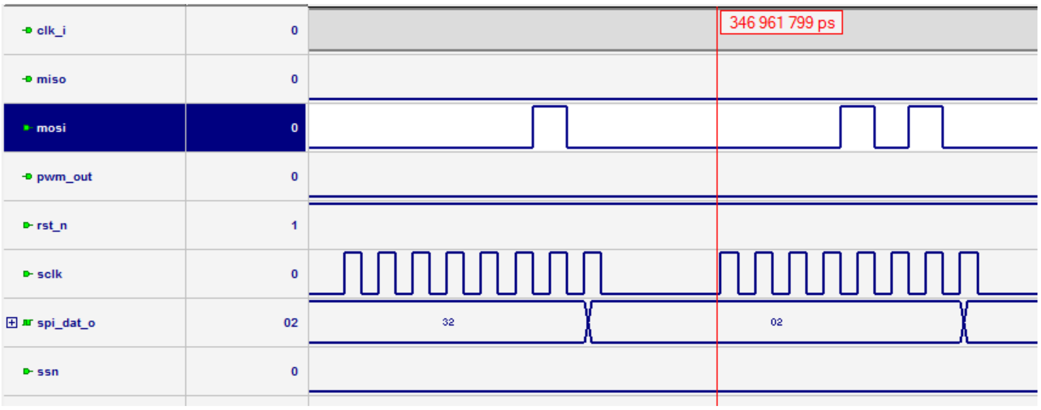
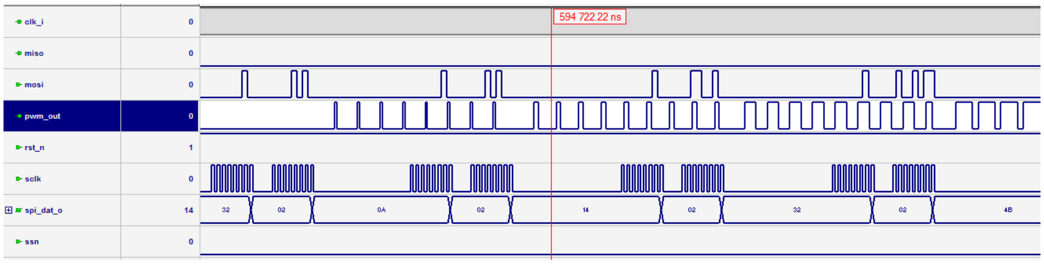


Figure 6. PWM Output with Varying Duty Cycle



Implementation

This design is implemented in Verilog. When using this design in a different device, density, speed, or grade, performance and utilization may vary. Default settings are used during the fitting of the design.

Table 4. Performance and Resource Utilization

Family	Language	Speed Grade	Utilization	f_{MAX} (MHZ)	I/Os	Architecture Resources
MachXO2[1]	Verilog	-4	389 LUTs	>10	6	EFB

[1] Performance and utilization characteristics are generated using LCMXO2-640HC-4TG100C, with Lattice Diamond™ 1.2 design software.

References

- SPI Specification from Freescale Semiconductors
- WISHBONE interface from OpenCores
- MachXO2 Family Data Sheet

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2011	01.0	Initial release.