Design Specification

PWM Module

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

Revision history

Date	Author	Description
2023-05-01	Kent Abrahamsson	First edition
2023-05-09	Kent Abrahamsson	Renamed Reset Handler to Reset Ctrl and clarified top level block diagram.

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

1 Table of Contents

1 TABLE OF CONTENTS 3

2 GENERAL 5

2.1 Introduction 5

2.2 Adaption guidelines 5

3 TOP LEVEL IMPLEMENTATION 6

3.1 Block Diagram 6

3.2 Top level external interface 7

3.2.1 Generics 7

3.2.2 Signals 7

3.2.3 General 8

3.2.4 Constraints 8

3.2.5 PLL / Reset Ctrl instantiation option 8

4 SUB-COMPONENT SPECIFICATIONS 9

4.1 PLL 9

4.1.1 Block diagram 9

4.1.2 Specification 9

4.1.3 Error handling 9

4.2 Reset Ctrl 9

4.2.1 Block diagram 9

4.2.2 Specification 9

4.2.3 Error handling 9

4.3 Key ctrl 9

4.3.1 Block diagram 9

4.3.2 Specification 9

4.3.3 Error handling 10

4.4 Serial UART 10

4.4.1 Block diagram 10

4.4.2 Specification 10

4.4.3 Error handling 10

4.5 Serial Ctrl 10

4.5.1 Block diagram 10

4.5.2 Specification 10

4.5.3 Error handling 10

4.6 PWM Ctrl 11

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

4.6.1 Block diagram 11
4.6.2 Specification 11
4.6.3 Error handling 11
4.7 DC Disp Ctrl 12
4.7.1 Block diagram 12
4.7.2 Specification 12
4.7.3 Error handling 12
5 TESTBENCH 13
5.1 Block Diagram 13
5.2 Testbench usage 13
5.3 Test cases 13
5.4 Testbench Specification 14
5.4.1 p_clk_gen 14
5.4.2 Serial UART 14
5.4.3 p_test_main 15
5.4.4 p_uart_send 15
5.4.5 p_uart_recv 15
5.4.6 p_pwm_check 15
5.4.7 p_7seg_check 15
5.4.8 Other checks 15
6 ABBREVIATIONS 16

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

2 General

2.1 Introduction

This component shall be used to control a PWM output. The target application is LED light intensity control. The LED intensity shall be able to be controlled by using active low push buttons (the key push buttons on the DE1 Altera development board). The intensity shall also be controlled by serial data input (RS232 communication).

2.2 Adaption guidelines

The PWM module shall be designed to be used as a FPGA top level implementation for the Cyclone II FPGA on the Altera DE1 development board. If another target HW is used the design will need updates in the PLL parts of the component.

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

3 Top Level Implementation

3.1 Block Diagram

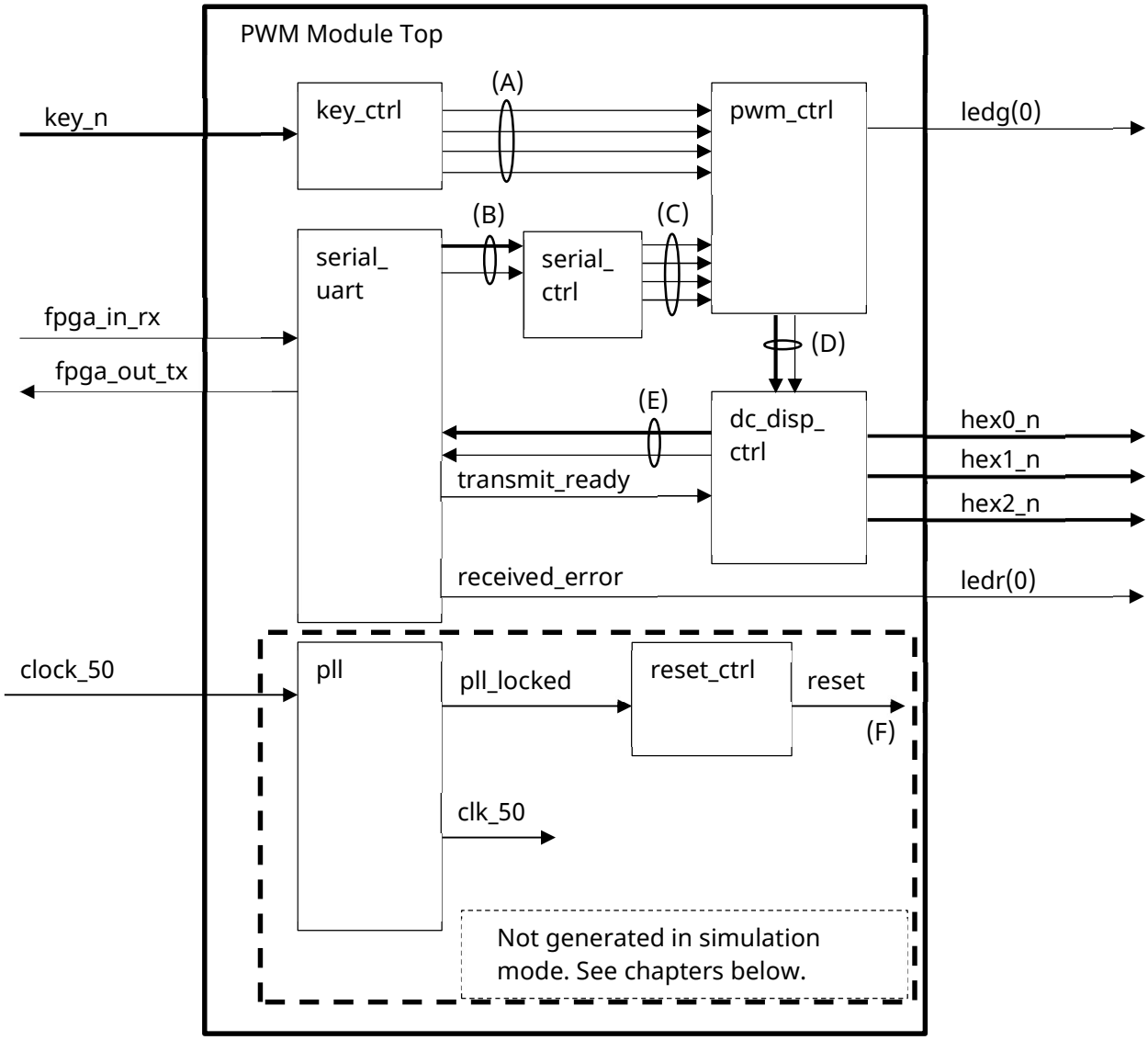

Figure 1 Top Level Block Diagram (all external signals are not shown)

Figure ref.	Source	Signal name	Description
A	Key Ctrl	key_on, key_off, key_up, key_down	Duty cycle control interface from key push buttons.
B	Serial UART	received_data, received_data_valid	Received serial data.

Figure ref.	Source	Signal name	Description
C	Serial Ctrl	serial_on, serial_off, serial_up, serial_down	Duty cycle control interface from serial data.
D	PWM Ctrl	current_dc, current_dc_update	Current duty cycle and update indication signal.
E	DC Disp Ctrl	transmit_valid, transmit_data	Data to transmit on serial interface.
F	Reset Ctrl	reset	Active high reset, routed to all components sub-components.

Table 1 Top level internal signals, referenced in Figure 1.

3.2 Top level external interface

3.2.1 Generics

Name	Type	Range	Description
g_simulation	boolean	true/false	Simulation generic, used to remove PLL and reset control component. (Speeds up simulation time for free Modelsim licence)

Table 2 Generics for top level entity

3.2.2 Signals

Name	Direction	Type	Description
clock_50	in	std_logic	Input 50 MHz clock signal, connected to internal PLL.
key_n	in	4 bit std_logic_vector	Key push buttons, active low.
fpga_in_rx	in	std_logic	Serial input data.
fpga_out_tx	out	std_logic	Serial output data.
ledg	out	8 bit std_logic_vector	Output vector for Green LEDs.
ledr	out	10 bit std_logic_vector	Output vector for Red LEDs.
hex0_n	out	7 bit std_logic_vector	7 segment display output digit 0.
hex1_n	out	7 bit std_logic_vector	7 segment display output digit 1.
hex2_n	out	7 bit std_logic_vector	7 segment display output digit 2.
hex3_n	out	7 bit std_logic_vector	7 segment display output digit 3.

Table 3 Signals for top level entity

3.2.3 General

The top level shall instantiate a number of sub-components, each sub-component shall have its own chapter specifying detailed functionality. The output signals that are unused by the application shall be driven to an inactive state in the top level architecture. I.e. unused LEDs and the fourth seven segment display shall be inactivated.

3.2.4 Constraints

The Design shall be fully timing constrained to ensure that the design passes all timing requirements. The timing requirements shall be written in the sdc_constraints.sdc file and be added to the design project.

3.2.5 PLL / Reset Ctrl instantiation option

To save simulation time when using free Modesim-Altera license the PLL / Reset Ctrl components shall be left out, i.e. not instantiated, when the top level generic g_simulation is set to true. When the top level generic g_simulation is set to true the PLL and Reset Ctrl shall be replaced with a simple process which models the PLL and Reset Ctrl functionality in a simplified way.

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

4 Sub-component specifications

4.1 PLL

4.1.1 Block diagram

None.

4.1.2 Specification

The PLL shall be connected to the 50 MHz input clock signal. When the PLL has locked on to the input clock the locked output is set high. This locked output shall be connected to the Reset Ctrl input.

The PLL shall output phase aligned 100 MHz, 50 MHz and 25 MHz clocks. The 50 MHz clock shall be used in all other components, the other clock outputs shall be left open.

4.1.3 Error handling

None.

4.2 Reset Ctrl

4.2.1 Block diagram

None.

4.2.2 Specification

The Reset Ctrl component shall have two reset inputs, one active high and one active low. The Reset Ctrl shall have two synchonous reset output signals one active high and one active low.

Both reset outputs shall be asynchronously set active and an internal couter shall be asynchonously reset if any of the reset inputs are set active. If both reset inputs are inacive a counter shall start counting up every clock cycle. When the counter has reaced its maximum (set by a generic value) the reset outputs shall be deasserted into inactive state.

4.2.3 Error handling

None.

4.3 Key ctrl

4.3.1 Block diagram

None.

4.3.2 Specification

The Key ctrl component shall double synchronize the active low key inputs.

The outputs from the compoent shall be set high one clock cycle if the inputs are detected to be low. If the inputs are held low the outputs shall be pulsed high one clock cycle every 10th millisecond.

The key_n input vector shall be mapped to the outputs in the following way:

key_n(0) shall control key_off output
key_n(1) shall control key_on output

key_n(2) shall control key_down output
key_n(3) shall control key_up output
Key_n input bits 3, 2 and 1 shall be ignored if key_n(0) is pushed down.
No pulses on key_up or key_down shall be generated if both key_n(2) and key_n(3) is pushed down simultaneously.

4.3.3 Error handling

None.

4.4 Serial UART

4.4.1 Block diagram

None

4.4.2 Specification

This component receives and transmits data over a serial interface. The component is an existing design with generic settings found in Lab 5 in the VHDL course. The generic settings shall be set as follows.

g_reset_active_state	: '1'
g_serial_speed_bps	: 115200
g_clk_period_ns	: 20
g_parity	: 0

The internal 50 MHz clock shall be used as clock signal.

4.4.3 Error handling

The received_error output shall be connected to the red LED ledr0 on the Altera DE1 development board.

4.5 Serial Ctrl

4.5.1 Block diagram

None

4.5.2 Specification

The Serial ctrl component shall receive data from the Serial UART component. The data received is expected to be ASCII Characters. When the ASCII Character 'U' or 'u' is received a one clock cycle long pulse shall be generated on the serial_up output. The serial_down signal shall be controlled in the same way when ASCII character 'D' or 'd' is received. The serial_off signal shall be pulsed high when the number '0' is received. And the serial_on signal shall be pulsed when the ASCII Character for number '1' is received. All other received data shall be ignored without any action.

4.5.3 Error handling

None.

4.6 PWM Ctrl

4.6.1 Block diagram

None

4.6.2 Specification

The PWM ctrl component implement a PWM function. The PWM output frequency shall be 1000 Hz. If any of the serial_* or key_* inputs are set high the PWM duty cycle shall be updated.

The PWM duty cycle in % shall be output towards the DC Disp Ctrl component on the current_dc vector. A one clock cycle high pulse on the current_dc_update signal shall be sent towards DC Disp Ctrl component when the duty cycle changes.

The on / off / up / down input signals shall be equally handled, independent of the source. The function shall be according to the following table.

Input signal	Function
serial_on / key_on	Go back to previous on state duty cycle (minimum 10%). Reset sets previous duty cycle to 100%.
serial_off / key_off	Set current duty cycle to 0%
serial_up / key_up	Increase duty cycle with 1%, maximum 100%, minimum 10%. I.e. the duty cycle shall be set to 10% if current state is off state and up command is received.
serial_down / key_down	Decrease duty cycle with 1%, maximum 100%, minimum 10%. This input shall be ignored if PWM output is in off state (at 0%).

Table 4 input signal functionality in PWM Ctrl component

In the (unlikely) case when both serial_* and key_* signals are activated at the same time the key_ inputs shall have priority.

The duty cycle of the output signal shall be updated at the end of each period (every ms).

The PWM period is defined to start with a rising edge on the PWM output and end (+start) with the next rising edge of the PWM output. The timing of the falling edge of the PWM output defines the duty cycle.

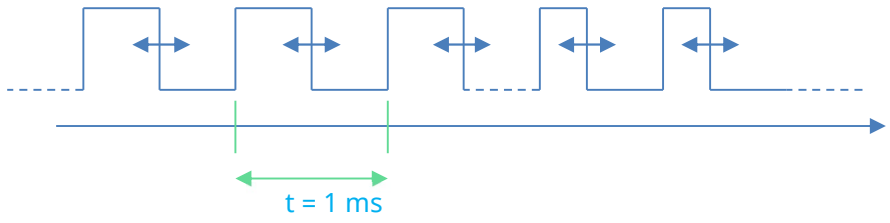

Figure 2 PWM period and Duty cycle control strategy

The output PWM duty cycle shall be off (0% after reset).

4.6.3 Error handling

None.

4.7 DC Disp Ctrl

4.7.1 Block diagram

None

4.7.2 Specification

The DC disp ctrl component shall output the currect duty cycle in % on three 7 segment digits. The numbers shall be between 0 and 100. Using an BCD converter is probably the easiest approach to do this.

The duty cycle shall also be trasmitted on the serial interface whenever the duty cycle is updated. The transmitted data shall be five bytes of data. Three ASCII characters representing the duty cycle between 0 and 100 followed by a '%' character, followed by a carrage return.

In the case of a duty cycle between 10 and 99 the first character shall be replaced with a space. And in the case when the duty cycle is between 0 and 9 the first two characters shall be space.

In the case of a new duty cycle update have been reported before the current duty cycle information have been fully transmitted on the serial interface the serial send shall be directly started again when finished in order to update the serial interface with the latest information.

Reset shall initialize a serial transmission sending 0% duty cycle and display a 0 on the seven segment display.

4.7.3 Error handling

None.

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se

5 Testbench

5.1 Block Diagram

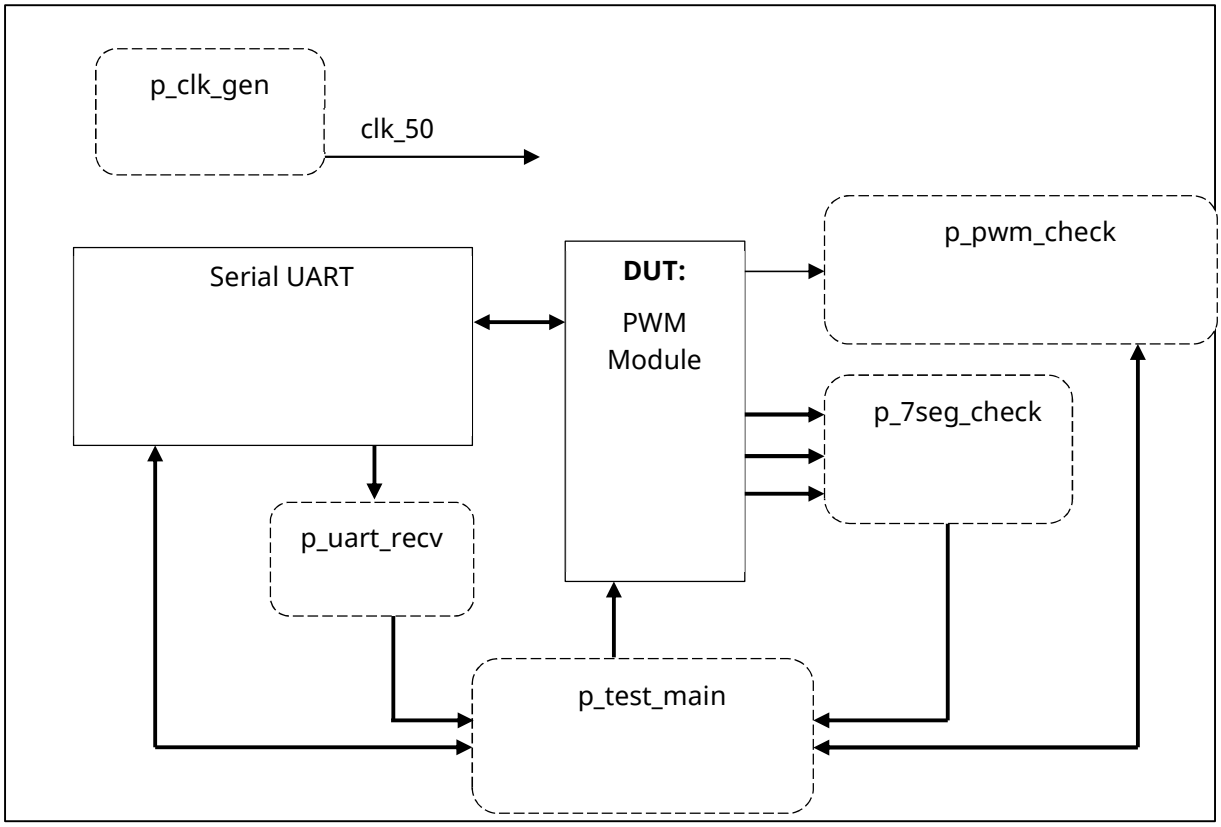

Figure 3 Testbench Block Diagram

5.2 Testbench usage

Start Modelsim and compile the testbench_pkg.vhd and testbench_top.vhd files.

If no compile errors exist start the simulation for the testbench_top entity.

Drag in signals of interest into the wave window (if any) and run the testbench with the command:

```
run -all
```

The testbench should report progress and end automatically when all tests have finished.

5.3 Test cases

Test case number	Description
0	Check that reset conditions are as expected. 0 % duty cycle reported everywhere.
1	Control key input to switch PWM output on, check that expected result is given on serial UART, PWM output and 7 segment display. (100%)
2	Control key input to switch PWM output off, check that expected result is given on serial UART, PWM output and 7 segment display. (0%)

Test case number	Description
3	Control key input to increase PWM duty cycle (short pulse on up key), check that expected result is given on serial UART, PWM output and 7 segment display. (10%)
4	Check PWM frequency (1000 Hz).
5	Hold key input to increase PWM duty cycle (100 ms long pulse on up key), check that expected result is given on serial UART, PWM output and 7 segment display. (19 - 21%)
6	Control key input decrease PWM duty cycle (short pulse on down key), check that expected result is given on serial UART, PWM output and 7 segment display. (One percent lower than test 4 result.)
7	Hold key input to decrease PWM duty cycle (120 ms long pulse on down key), check that expected result is given on serial UART, PWM output and 7 segment display. (stop at 10%)
8	Send serial data corresponding to character ‘0’. Check that expected result is given on serial UART, PWM output and 7 segment display. (0%)
9	Send serial data corresponding to character ‘1’. Check that expected result is given on serial UART, PWM output and 7 segment display. (10% - remembered from test 7)
10	Send serial data corresponding to character ‘u’. Check that expected result is given on serial UART, PWM output and 7 segment display. (11%)
11	Send serial data corresponding to character ‘U’. Check that expected result is given on serial UART, PWM output and 7 segment display. (12%)
12	Send serial data corresponding to character ‘D’. Check that expected result is given on serial UART, PWM output and 7 segment display. (11%)
13	Send serial data corresponding to character ‘d’. Check that expected result is given on serial UART, PWM output and 7 segment display. (10%)
14	Send two bytes of serial data corresponding to character ‘d’. Check that expected result is given on serial UART, PWM output and 7 segment display. (10% - no change)

Table 5 Test cases

5.4 Testbench Specification

5.4.1 p_clk_gen

The p_clk_gen process generates the 50 MHz system clock and the reset signal for the testbench. The system clock shall be able to be stopped by a kill_clock signal from p_test_main process to enable “run -all” command in Modelsim.

5.4.2 Serial UART

The serial UART is a component of the same type as used in the DUT, generics shall be configured for the testbench and to allow communication with the UART specified in 4.4.

5.4.3 p_test_main

The p_test_main process controls all tests that shall be done. It receives status for the PWM output signal, 7 segment displays and UART receive process. The p_test_main process controls the key_n inputs towards the PWM controller and the send interface on the Serial UART. This process shall also write all the test status and results.

5.4.4 p_uart_send

The p_uart_send process shall control the send interface of the serial UART component, it shall be controlled from the p_test_main process.

5.4.5 p_uart_recv

The p_uart_recv process shall receive data from the serial UART compoent and decode the received characters into a natural number presented to the p_test_main process. Any unexpected characters received shall result in an error assertion.

5.4.6 p_pwm_check

The p_pwm_check process shall sample the PWM output signal (ledg(0)) from the DUT. The current duty cycle shall be presented to p_test_main process as a rounded natural number. The frequency of the output signal shall also be calculated and sent to p_test_main process.

5.4.7 p_7seg_check

The p_7seg_check process shall check the 7 segment display outputs from the DUT and decode these into a natural number presented to the p_test_main process.

5.4.8 Other checks

An error assertion shall be made if the ledr0 output is set high (representing an error in received serial UART data in the DUT).

It would be a good idea to set the generic g_simulation to false once and see that the internal PLL and reset functionality works as it should before downloading into hardware.

6 Abbreviations

DC	Duty Cycle
DUT	Design Under Test
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
UART	Universal Asynchronous Receiver-Transmitter

Address:
Mor Kerstins väg 27
511 53 Kinna
Sweden

Tel:
+46-703-748429

Registration no:
559037-6603

E-mail:
kent@elekteknik.se