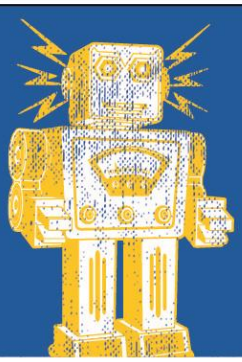
Elektronik



Digital System Design

Distance Learning Letter
Streamlined Counter Project
Part: I/O Control Unit

Version: 0.2
Author: A. Puhm

Copyright Notice

Introduction

This document describes the IO control unit of the counter project. The IO control unit handles all I/O ports (except the clock and reset signal). It includes the multiplexer needed for the 7-segment digits, debounces the switches and push buttons and makes the debounced signals available for FPGA-internal logic.

Overview

The I/O Control unit shall:

- [SHALL] Synchronize inputs
- [SHALL] Debounce inputs
- [SHALL] Prepare the synchronized and debounced inputs for the interface with the counter sub unit
- [SHALL] Encode the internally generated counter values for the seven segment display
- [SHALL] Multiplex the encoded counter values for the seven segment bus

I/O Control Unit

General Interfacing

Table 1:I/O control unit generic map

Generic Name	Type	Description
G_CNT_UP_BUTTON_ID	Integer 0 … 3	Exemplary generic
G_DOWN_UP_BUTTON_ID	Integer 0 … 3	Exemplary generic
G_CNT_HOLD_BUTTON_ID	Integer 0 … 3	Exemplary generic
G_CNT_RESET_BUTTON_ID	Integer 0 … 3	Exemplary generic
G_CNT_MODE_BIT_0_SW_ID	Integer 0 … 15	Exemplary generic
G_CNT_MODE_BIT_1_SW_ID	Integer 0 … 15	Exemplary generic
G_CNT_SPEED_BIT_0_SW_ID	Integer 0 … 15	Exemplary generic
G_CNT_SPEED_BIT_0_SW_ID	Integer 0 … 15	Exemplary generic
G_CNT_SPEED_DIV_SIM	Integer	may be used to shorten the 10kHz timer period for simulation

Table 2: I/O Control unit port map

Port Name	Direction	Description
clk_i	In	System clock (100 MHz)
reset_i	In	Asynchronous high active reset
cntr_1_i(3:0)	In	Digit 0 (from internal logic)
cntr_10_i(3:0)	In	Digit 1 (from internal logic)
cntr_100_i(3:0)	In	Digit 2 (from internal logic)
cntr_1000_i(3:0)	In	Digit 3 (from internal logic)
sw_i(15:0)	In	16 switches (from FPGA board)
btnu_i	In	Basys3 button BTNU
btnd_i	In	Basys3 button BTND
btnl_i	In	Basys3 button BTNL
btnr_i	In	Basys3 button BTNR
ss_o(7:0)	Out	Seven segment LEDs CA…CG and DP
ss_anode_o(3:0)	Out	Anodes: AN0…3
cntr_hold_o	Out	if ‘1’, holds the counter
cntr_reset_o	Out	If ‘1’, resets the counter to 0

Port Name	Direction	Description
cntr_up_o	Out	the counter will count up, if set to ‘1’
cntr_down_o	Out	the counter will count down, if set to ‘1’
cntr_mode_o(2:0)	Out	determines, if the counter operates in octal/decimal/hexadecimal mode
cntr_speed_o(3:0)	Out	Determines, if the counter operates at 1Hz/10Hz/100Hz/1000Hz

[MAY] The generics, as specified in Table 1 may be used for the implementation. If new generics are added, they have to be added in all relevant design files of the counter project!

[SHALL] The port map, as specified in Table 2, shall be used for implementation.

[SHALL] No ports shall be added or removed!

Port clk_i is connected to the top-level entity clk_i port.

Port reset_i is connected to the top-level entity reset_i port, which is further connected to BTNC as depicted in Figure 1.

Ports sw_i, btnu_i, btnd_i, btnl_i, btnr_i, ss_o, ss_anode_o are connected to the similar named top-level ports.

Ports cntr_* are connected to the counter sub-unit.

[SHALL] The designer shall decide which buttons (btn*_i)) and switches (i.e., sw_i) he/she uses to generate the necessary control information for the counter sub unit, as specified in the chapter “Interfacing with the counter sub-unit”.

Figure 1 shows an extract of the circuit diagram of the Basys3 FPGA development board. The 16 LEDs are not used here.

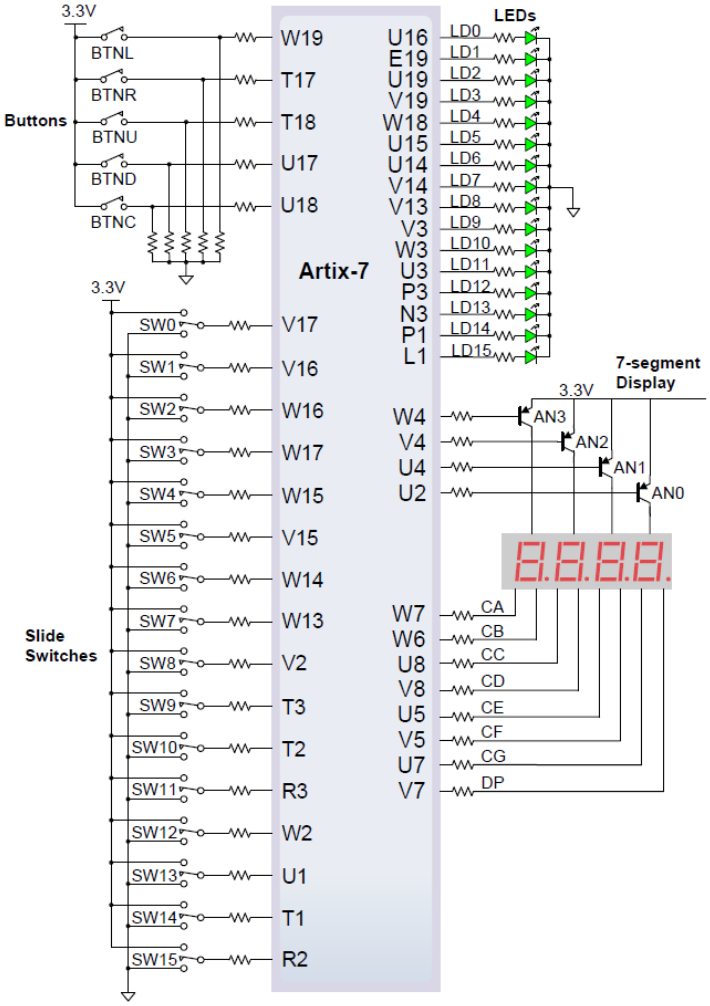


Figure 1: Schematic part for button, switches, leds, and seven segment display of the Basys3 board

Synchronization

Inputs to a digital design have to be synchronized with the clock used for those parts of the design which utilize these inputs. This has to be done before the input is used in any other part of the design. It avoids non-deterministic behavior in the design, refer to [1], chapter 1.8 Metastability for more information. To synchronize an input, it is usually sufficient to apply two or three shift register stages as depicted in [1], chapter 1.9.1.

Debounce

Push buttons and switches often generate spurious open/close transitions when pressed, due to mechanical and physical issues. These transitions may be read as multiple presses in a very short time fooling the application. Thus, the IO control unit must use a debounce mechanism for the 16 switches and 4 push buttons of the Basys3 board in order to eliminate the unwanted transitions. Note, that only the four buttons BNTU, BTND, BTNL and BTNR are handled by the IO control unit while button BTNC is used as the global asynchronous reset signal for the whole design.

Interfacing with the counter sub unit

The I/O control unit has the following ports, which are connected to the counter sub unit and control the counter operation:

- cntr_hold_o:
  [SHALL] If this is set to '1', the counter sub unit shall not increase or decrease its counter values.
  [SHALL] If this is set to '0', the counter shall operate as indicated by the other control inputs.
- cntr_reset_o:
  [SHALL] If this is set to '1', the counter sub unit shall reset its internal counter values to 0.
  [SHALL] If this is set to '0', the counter shall operate as indicated by the other control inputs.
- cntr_up_o:
  [SHALL] If this is set to '1', the counter shall increase its counter values in steps of 1.
- cntr_down_o:
  [SHALL] If this is set to '1', the counter shall decrease its counter values in steps of 1.
  [SHALL] cntr_up_o and cntr_down_o shall never be set to '1' or '0' at the same time.
- cntr_mode_o:
  [SHALL] cntr_mode_o(0) … if set to '1', the counter shall operate in octal mode (min: 0, max: 7)
  [SHALL] cntr_mode_o(1) … if set to '1', the counter shall operate in decimal mode (min: 0, max: 9)
  [SHALL] cntr_mode_o(2) … if set to '1', the counter shall operate in hexadecimal mode (min: 0, max: F)
  [SHALL] One, and only one, index of this port shall be set to '1' at all times.

- cntr_speed_o:
  [SHALL] cntr_speed_o(0) … if set to ‘1’, the counter lowest digit shall count at a speed of 1Hz
  [SHALL] cntr_speed_o(1) … if set to ‘1’, the counter lowest digit shall count at a speed of 10Hz
  [SHALL] cntr_speed_o(2) … if set to ‘1’, the counter lowest digit shall count at a speed of 100Hz
  [SHALL] cntr_speed_o(3) … if set to ‘1’, the counter lowest digit shall count at a speed of 1000Hz
  [SHALL] One, and only one, index of this port shall be set to ‘1’ at all times.

Output encoding and multiplexing

The Basys3 board contains one four-digit 7-segment LED display. Eight data signals (CA, CB … CG, DP) and four control signals (AN0-AN3) are used to control the 7-segment display. Each digit consists of eight LEDs which are connected to a common anode as well as individual cathodes. The common anode signals AN0-AN3 are used to enable the four digits. The cathodes of all four digits are connected to eight signals labeled CA, CB, … CG through DP. Figure 2 shows the connections of a single digit. To illuminate a segment, the anode is driven high while the cathode is driven low. Since the FPGA board uses transistors to drive enough current into the common anode, the anode enables are inverted. Therefore, both the AN0-AN3 and the CA..G/DP signals have to be driven low when a LED shall be turned on.

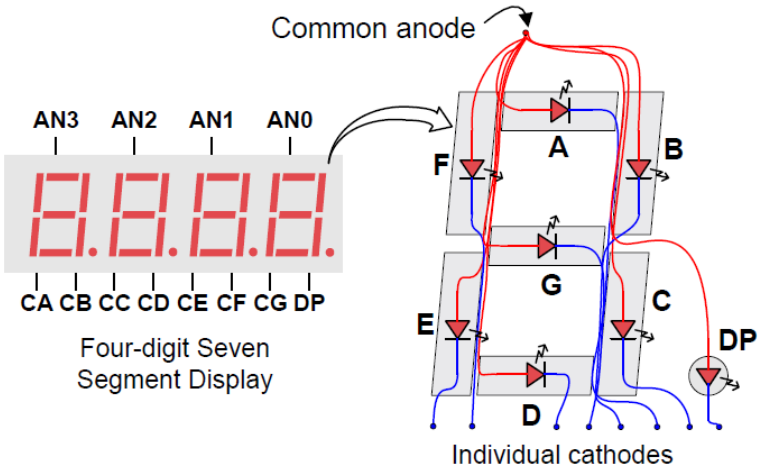


Figure 2: 7-segment display

A scanning display controller circuit can be used to display a four-digit number. This circuit drives the anode signals and corresponding cathode patterns of each digit in a repeating, continuous succession at an update rate that is faster than the human eye can detect. Each digit is illuminated just one-fourth of the time, but because the eye cannot perceive the darkening of a digit before it is illuminated again, the digit appears continuously illuminated. For each digit to appear bright, a refresh frequency of 10 kHz can be used. This frequency range is also suitable for debouncing buttons and switches.

Abbreviations

VHDL	Very High Speed Integrated Circuit Hardware Description Language
LED	Light Emitting Diode
FPGA	Field Programmable Gate Array

References

[1] Distance Learning Letter: Synchronous Design Methodology, Roland Höller, 01/2016, v0.3

Version

Version 0.1, 2018-09-14	Modified the document for a streamlined counter project (based on counter/io ctrl v0.7 DLL)
Version 0.2, 2018-09-20	Updated Introduction, clarified v0.1 note

If you find errors or inconsistencies, please report them to the supervisors of this lecture via email. Thank you!