Experiment 8

Push Button: Datapath and Controller

In this lab, we are going to design datapath and controller for a push-button interface.

Overview of Push Buttons

Nexys A7 100-T has seven push-buttons out of which six are available to the user. One of them is the CPU RESET button and the rest five push-buttons are arranged in a plus-sign configuration. We will be using one of the five arranged in plus-sign configuration. They generate a low output when they are at rest, and a high output only when they are pressed. (Fig. 8.1). Mechanical push-buttons cause unwanted bounce on the signal when toggled. Therefore for using push-buttons as inputs, we are required to use (Fig. 8.2):

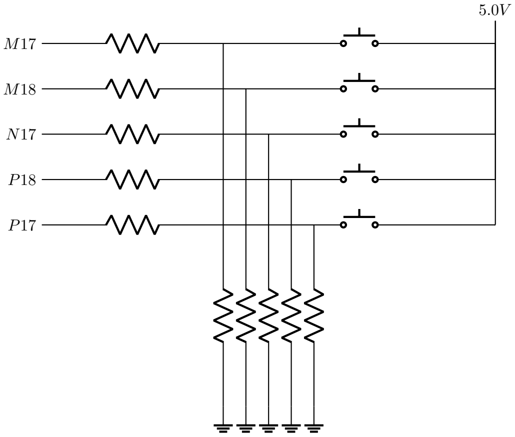

Fig. 8.1: Push Buttons in Nexys A7 100T

- A switch debouncing module, which removes unwanted bounce from the signal.
- A level-to-pulse converter, which generates a single pulse for one push of the button.

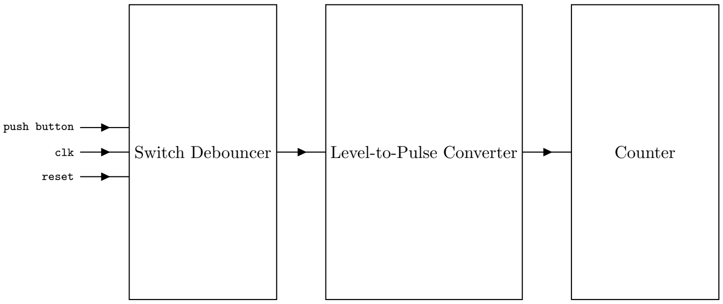

Fig. 8.2: Lab task modular circuit.

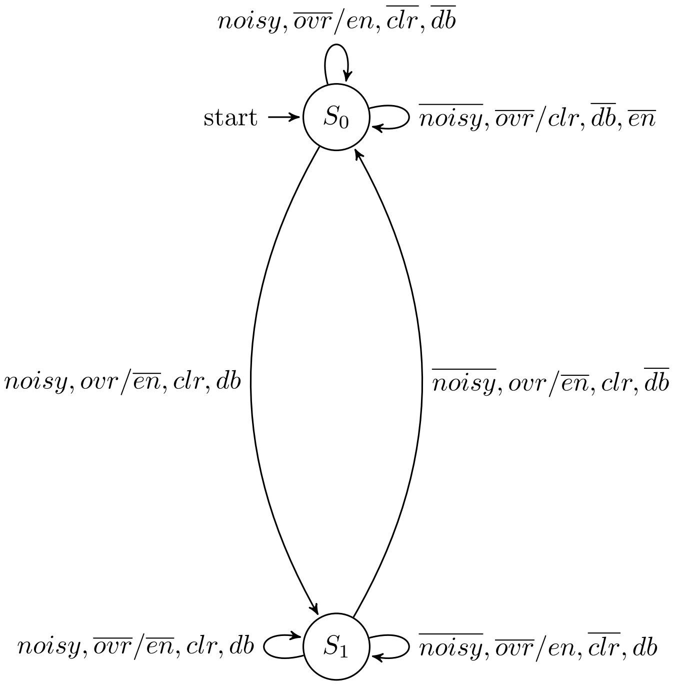
$noisy, \overline{ovr}/en, \overline{clr}, \overline{db}$
start
S_0
$\overline{noisy}, \overline{ovr}/clr, \overline{db}, \overline{en}$
$noisy, ovr/\overline{en}, clr, db$
$\overline{noisy}, ovr/\overline{en}, clr, \overline{db}$
$noisy, \overline{ovr}/\overline{en}, clr, db$
S_1
$\overline{noisy}, \overline{ovr}/en, \overline{clr}, db$

Fig. 8.3: State Transition Graph for Debouncer

Where noisy is the input from the push button and db is the debounced output. En, clr are the enable and clear inputs to a counter which counts till the debouncing time (100ms - 300ms) of the push button and generates an overflow (ovr).

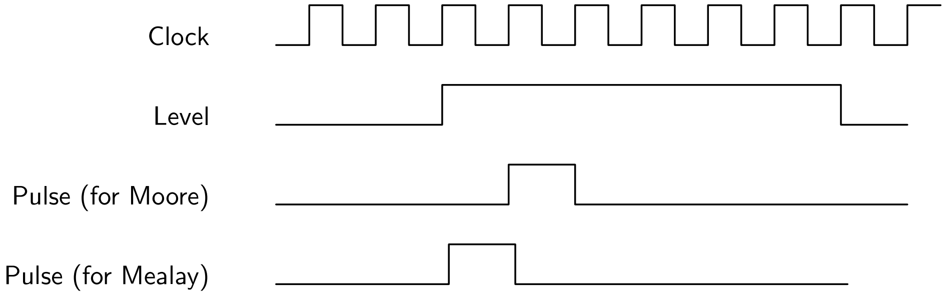

Fig. 8.4: Timing diagram of Level to Pulse Converter

Tasks

For this lab, you are required to design a datapath and controller for the debouncing module according to (Fig. 8.3), a level to pulse generator by designing its State Transition Graph and a counter that counts the number of times the push button is pressed. Display the count on the seven segment display.

Note: Count till a maximum of 99 so use only two of the seven segment displays. Also 10 will be displayed as 10 not A.

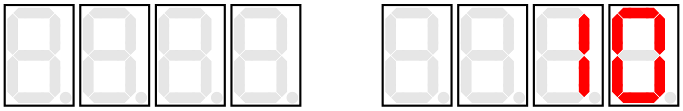

Fig. 8.5: Sample output for the Lab Task when push button has been pressed ten times.

Use of latches are not allowed in the design of synchronous circuits for the lab. Synchronous circuits must be designed with flip-flops.

Deliverables

1. A report containing the following items:
   - (a) State Transition Graph and Tables of level to pulse converter.
   - (b) The designed circuit for task module in the diagram above with hand-sketched design partition of the system.
   - (c) Resource Utilization Report: Read the synthesis report and identify how many resources in the FPGA such as lookup tables (LUTs), input/output (IOs), etc., has been utilized.
2. Simulation of your designed module on Questasim or Xcelium. Show the simulation to the instructor.
3. FPGA Implementation: Synthesize the circuit for Nexys A7 FPGA available in the lab. Tie input to one of the push button and outputs to the seven-segment displays available on the board.

The collaboration between students is encouraged, but blind code sharing/copying is not allowed. If you are unable to explain anything in your code, it will be assumed you have copied it. COPIED MANUALS GET ZERO MARKS. So make sure you know every thing you have written in your code. We are least concerned about how you have learnt something as long as you have learnt it well.

Acknowledgements

The manual has been written by Dr. Ubaid Ullah Fayyaz and Ms. Shehzeen Malik.