Islamic University of Gaza

Engineering Faculty

Computer Engineering Department

VHDL Project

Washing Machine Controller

May 5, 2023

Overview

Introduction

A washing machine is a commonly used appliance used to clean clothes. In this project, we aim to design and implement a washing machine controller using VHDL. The controller is responsible for controlling the washing machine's cycles such as the washing cycle, rinsing cycle, etc. The controller is designed to support multiple wash cycles and is pre-programmed with multiple washing programs.

Terminology

Wash Cycle: refers to the process of cleaning clothes in a washing machine. For example, washing clothes.
Wash Program: a pre-programmed sequence of wash cycles designed to clean specific fabrics or clothing.

Control Panel

The washing machine has a control panel with the following buttons:

- **Power:** On/Off button.
- **Dial:** controls the time added for each wash program.

System Requirements

The washing machine controller must meet the following requirements:

- The washing machine supports the following **wash cycles**:
    1. **WASH**: Wash the clothes.
    2. **SPIN**: Spinning the clothes to remove excess water.
    3. **RINSE**: draining the soapy water.
    4. **IDLE**: All cycles have finished.
- The washing machine must be pre-programmed with the following **wash programs**:
    - **COTTON**
    - **RAPID**
    - **SLOW**
    - **DRAIN**: Only rinse and spin.
    - **WHITE**
- Each program will go through cycles different times. **DRAIN** program should only go through **SPIN**, and **RINSE**.
- The **wash cycles** must have the following state code (**WASH: 00**, **SPIN: 01**, **RINSE: 10**, and **IDLE: 11**). Each cycle must be repeated multiple times depending on the **wash programs**. For Example: In the **COTTON** program, there must be 4 ***WASH*** cycles, 5 ***SPIN*** cycles, and 2 ***RINSE*** cycles.
    - To do this you must use a counter and a single internal output **ZERO** that will be 1 each time the counter goes down to 0.
    - The counter for ***WASH*** state starts counting down (S1S0 = 00). As soon as this counter reaches zero, ***ZERO*** becomes 1, and the system transitions to ***SPIN*** state. Then, the counter for ***SPIN*** state (S1S0 = 01) starts counting down. As soon as this counter reaches zero, ***ZERO*** becomes 1 again, and the system transitions to ***RINSE*** state. In this final operational state which is ***RINSE*** (S1S0 = 10), the counter for ***RINSE*** state starts counting down. As soon as this counter reaches zero, ***ZERO*** becomes 1 again, and the system transitions to ***IDLE*** state.

Design

The washing machine controller black box is described as follows:

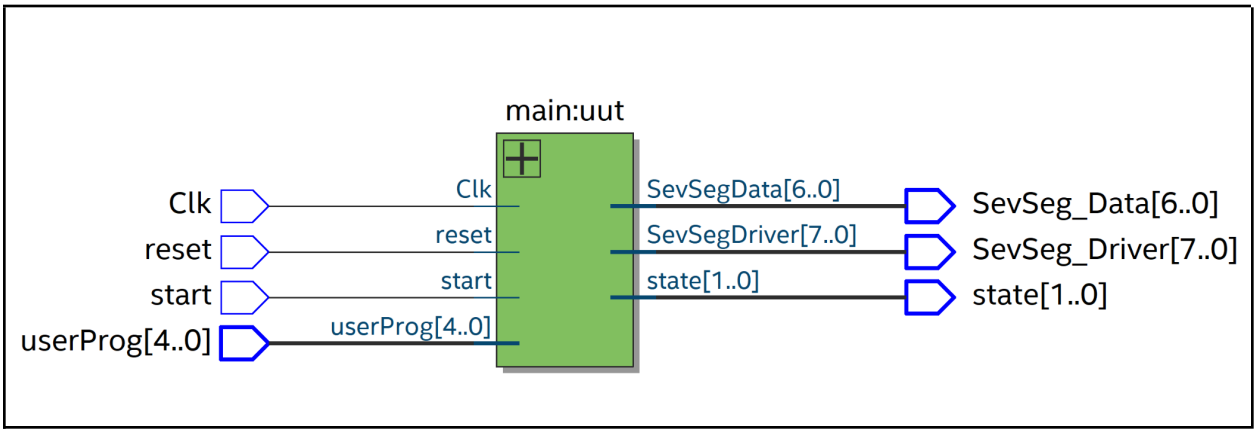
- Input:
    - **userProg**: the machine **dial** controls which program will be used to wash the clothes. The dial is an active-high component that will output ***1*** for the program that's being selected and ***0***s for the other program bits. For Example: If the user selects the **COTTON** program, the input value will be "10000". If **RAPID** is selected, the input value will be "01000", and so on.
    - **clk**: the clock of the component. This clock is programmed to run on 50 megahertz (MHz).
    - **reset**: An active-high bit that resets the state of the controller to ***IDLE***.
    - **start**: An active-high bit that starts the washing program.
- Output:
    - **sevseg_data**: contains 7-bit output for a seven-segment display that will show the current cycle (i.e. WASH, RINSE, etc.). If there's no cycle (i.e. the program hasn't yet started), the display should show the current chosen program (COTTON, RAPID, etc.).
        - *Tip:* A 7-segment display can only show one character at a time, you should use a counter to display all the characters in multiple displays.
    - **sevseg_driver**: an active low seven-segment driver that determines which display is enabled at a certain clock pulse. This is used to show one character in each display. See **"A 4-digit display"** Figure on this page.

      For example, to show the word "COTTON" in the first clock pulse, **sevseg_driver** must be "01111111" and **sevseg_data** must be the decoding for the letter 'C'. Next clock pulse **sevseg_driver** must equal "10111111", and **sevseg_data** be the letter 'O', and so on.

      Remember: This process must happen fast so that the human eye sees it all showing together.

    - **state**: two-bit output that shows the controller's current state.

Component



Report

Each group must submit a report for the project, explaining the approach to achieve this controller, including all entities, architectures, and test benches for each entity (including the main component), and the **references**, if any.

Submission

- Each group can be a maximum of 2 students contributing to the same project. Plagiarism **will not** be tolerated.
- The **deadline** for submitting the report on Moodle is May 31st.
- The **deadline** for discussing the project is the day before the final exam (June 13th).