Islamic University of Gaza

Faculty of Engineering

Computer Engineering Department

VHDL

Washing Machine Controller

Prepared by:

Eng. Baker Raed Daher

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 **0s** 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.
        - 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.

Remark:

First the 50 MHZ takes a long time with simulation, so we assign 50kHz rather that it, we converted the CLK to 100 HZ and divided it to 10 parts and every part acts as a cycle.

Example:

For Cotton: Wash takes 4 cycles, SPIN 3 cycles,
RINSE 3 cycles.
So the sum of all is 10

And so on for each userProg selection.

The approach for implementing this controller involves multiple components. Here is a brief description of the approach:

1. Clock Divider: The clock divider module takes an input clock signal (MCLK) and divides it to generate a slower clock signal (HUNDREDHZCLOCK). It uses a counter to count the number of input clock cycles and toggles the output clock signal when the counter reaches a specific value.

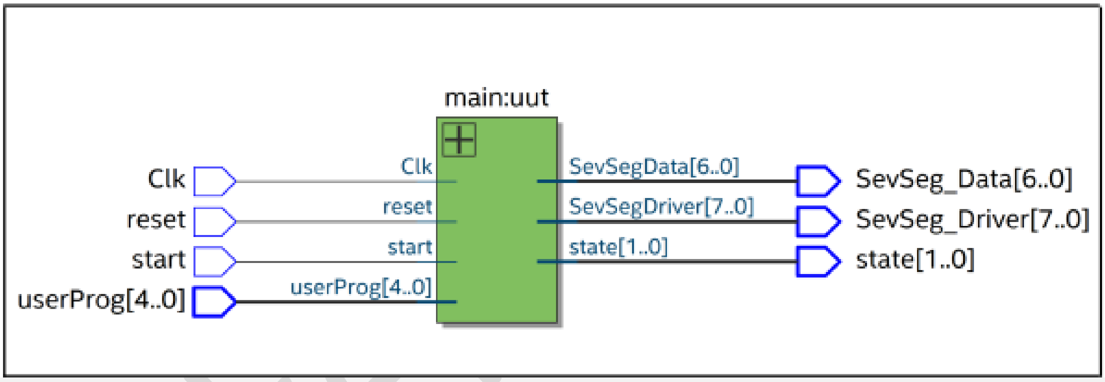
2. Seven-Segment Driver: The seven-segment driver module is responsible for driving the seven-segment display based on the selected washing mode (userMode). It uses a counter (COUNTER) to cycle through different segments of the display, and based on the current mode and counter value, it assigns the appropriate data to be displayed on the seven-segment display.

3. Seven-Segment Decoder: The seven-segment decoder module takes a binary input (INPUT) representing a digit or character and decodes it to generate the corresponding seven-segment display output (SEVSEG_BUS).

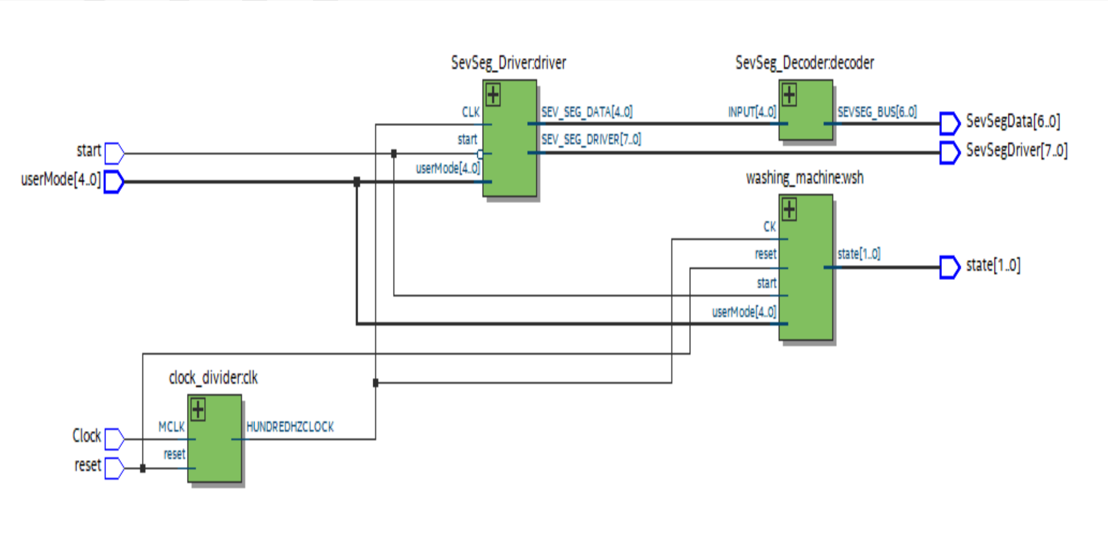
4. Washing Machine: The washing machine module is the main controller that manages the operation of the washing machine. It receives inputs such as the clock signal (CK), start signal, reset signal, and user-selected mode (userMode). It uses internal signals and variables to keep track of the current state and the countdown values for washing, spinning, and rinsing. It transitions between states based on the clock signal and the start signal, and updates the countdown values accordingly. The state output represents the current state of the washing machine.

Overall, the approach involves dividing the clock signal, driving the seven-segment display based on the selected mode, decoding the input to the seven-segment format, and implementing the main controller to manage the washing machine's operation.

Component



In Main Component



References:

- https://en.wikipedia.org/wiki/Seven-segment_display#Implementations
- https://github.com/hasancaslan/ELEC204-VHDLWashingMachineController/tree/master/elec204_final_project