**ECE – 590**

**DIGITAL SYSTEM DESIGN USING HARDWARE DESCRIPTION LANGUAGE**

**HOMEWORK - 1**

**MEMORY CONTROLLER AND DATAPATH**

**GANKIDI VASU – 920361483, email:** [**vasu@pdx.edu**](mailto:vasu@pdx.edu)

**VISHWAS PULUGU – 935084168, email:** [**vishwas@pdx.edu**](mailto:vishwas@pdx.edu)

**CONTROLLER:**

Controller operates in two modes normal mode and zero mode. In normal mode the memory can perform a write operation to the specified address (if write is asserted). And in zero mode the system writes zeros to all address in the range specified by registers R1 and R2. These registers are loaded in normal mode. The system asserts busy when it’s busy (in zero mode), and not respond to inputs during that time.

The controller in this digital module has four inputs and seven outputs. The input signals are clock, zero, reset, cnt\_eq. and the output signals are cnt\_en ,set\_busy ,clr\_busy ,ld\_cnt ,addr\_sel ,zero\_we. A single clock is used to both the controller and datapath.

Inputs:

**Zero** – The input zero is used to decide the mode of the controller, if zero = 1 then the controller is used to write zero’s to the memory in the specified address range (loaded into the registers R1 and R2). if zero = 0 then the system is in normal mode in which the controller is used to write and read data to/from specified address.

Reset – reset’s the controller to initial state. i.e. normal mode.

**Cnt\_eq** - if cnt\_eq = 1 this implies that the write operation has been completed in the zero mode and returns to the initial state.

Outputs:

**Cnt\_en** – is used to enable the up counter.

**Set\_busy** and **clr\_busy** – set\_busy is high and clr\_busy is low in the zero mode which makes the busy flag of the controller high, which indicates the inputs given to the controller in zero mode are not considered. Whereas the set\_busy is low and clr\_busy is hign in the normal mode which makes the busy flag go low, which indicates the registers can be loaded and data can be written and read from memory i.e. normal mode of operation.

**Ld\_cnt** – loads the value of the register R2 into the UPCOUNTER.

**Addr\_sel** – in the normal mode the value of Addr\_sel is set to 0 so that the data can be read/write from/to a memory location. In the zero mode the value of Addr\_sel is set to 1 so that the we can write zero’s to the chunk of memory specified by the registers R1 and R2.

**Zero\_we** – is 0 in the normal mode which enables the memory to read and write the data based on the write signal value in the datapath.

Zero\_we is 1 in the zero mode which enables writing only zero’s to the memory i.e. it doesn’t support read operation in zero mode.

Finite State Machine: Moore state machine is

There are three states in the FSM **normal, load and zero**.

**Normal** – until the value of the input zero is one the controller is in this state. In state the registers R1 and R2 are loaded and write and read from/to memory can be done. The outputs of the controller in this state are-

set\_busy <= '0';

clr\_busy<='1';

ld\_cnt<='0';

cnt\_en<='0';

addr\_sel<='0';

zero\_we<='0';

**load –**  in this state the ld\_cnt is set to one, such that the upcounter is loaded to with the value of the value of the register R2 and the busy flag is set to 1. Such that the data is not written to the memory or read from memory in this state. The outputs of the controller in this state are-

set\_busy <='1';

clr\_busy<='0';

ld\_cnt<='1';

cnt\_en<='0';

addr\_sel<='0';

zero\_we<='0';

**Zero –** this state is immediately followed by load state without any input condition for the transition. In this state the zero\_we is one which is the input to multiplexer which enables writing zero to the range specified by registers R1 and R2. The outputs of the controller in this state are-

set\_busy <= '1';

clr\_busy<='0';

ld\_cnt<='0';

cnt\_en<='1';

addr\_sel<='1';

zero\_we<='1';