ECE – 590

DIGITAL SYSTEM DESIGN USING HARDWARE DESCRIPTION LANGUAGE

HOMEWORK - 1

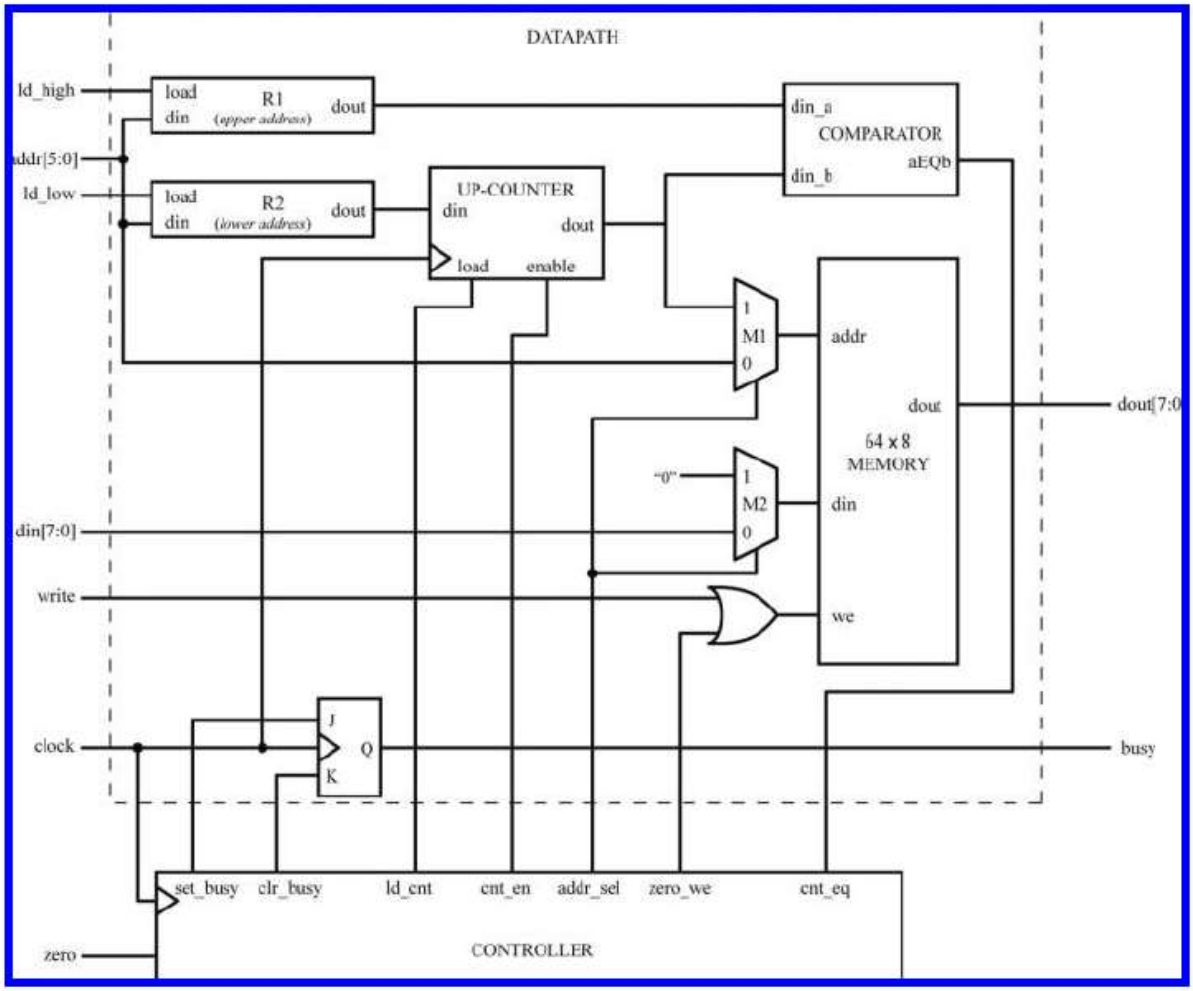
MEMORY CONTROLLER AND DATAPATH

GANKIDI VASU – 920361483, email: vasu@pdx.edu

VISHWAS PULUGU – 935084168, email: vishwas@pdx.edu

Memory Controller and Datapath:

The controller for the datapath operates in two modes: “normal” and “zero”. In normal mode memory performs write or read operation to specified address depending on the write signal. In zero mode system writes zeroes to all addresses in the range specified by registers R1 and R2. These registers are loaded in Normal mode. The system should assert busy in Zero mode.



Datapath:

Inputs of the datapath:

ld_high,ld_low : 1 bit inputs to registers R1 and R2 according to which value in addr is loaded into one of the registers.

addr[5:0]: 6 bit input address for the memory.

din[7:0]: 8 bit data to be written into the memory.

Write: 1 bit signal which activates write process to the selected address.

Clock: synchronising signal for controller and datapath.

Ouputs of datapath:

aEQb: 1 bit signal indicating the completion of range of addresses between R1 and R2

dout[7:0] : gives the value of data in memory at given address.

Busy: signal indicating that the system is in zero mode and any inputs will be ignored.

Function of each block:

Registers:

It has two registers R1 and R2. R1 holds higher address and R2 holds lower address. Both the registers are loaded with value in addr[5:0] input depending on the values of ld_high and ld_low inputs. Both the registers have dout, which holds the address in that register as output. dout of R1 passes the address in R1 to comparator while dout of R2 passes the address in R2 to up counter module.

Up-Counter:

The functioning of Up-Counter is controlled by ld_cnt and cnt_en signals from controller. When ld_cnt is asserted the address from R2 is loaded into the counter. When cnt_en is asserted the up_counter starts incrementing the address value and passes it to comparator each clock cycle.

Comparator:

It gets high address value from R1 and incremented address value from Up-Counter each clock cycle. Whenever both the values are equal, it asserts aEQb output signal.

Multiplexers:

there are two multiplexers M1 and M2. M1 multiplexes the input address to datapath and the address from up_counter. One of these two is selected by the addr_sel signal from the controller. M2 multiplexes input data to datapath and zero bits. It is also controlled by addr_sel signal from controller. In zero mode addr_sel value is 1 and it selects address from up-counter and data will be bits of 0. In normal mode addr_sel value is 0 and Multiplexers select address, data from datapath input.

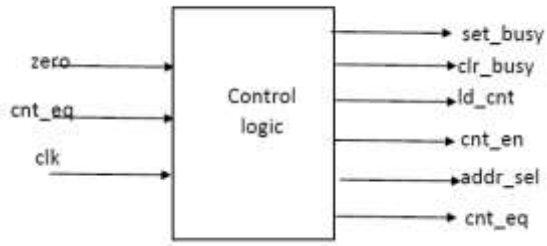
Memory:

Memory has addr, din and we as inputs and dout as output. When we is asserted, the value of din is written to the memory location pointed by addr and dout holds this data. If we is not asserted, dout displays the data at address location pointed by addr.

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.

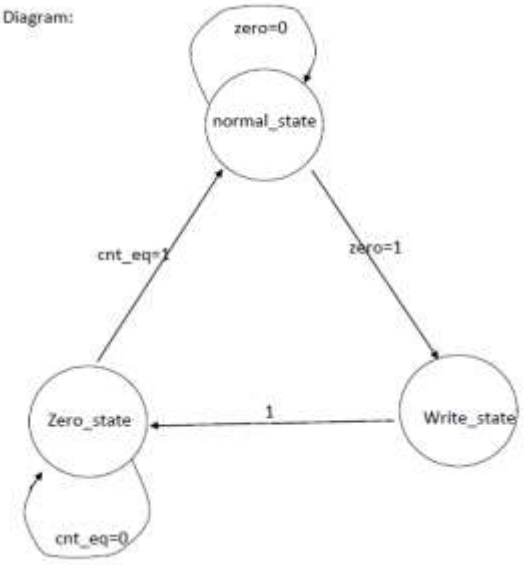
Zero_we – is 0 in the normal mode which enables the memory to read and write the data based on the write signal value in the datapath.

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)

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

State Diagram:



State outputs:

OUTPUTS\STATES	NORMAL_STATE	LOAD_STATE	WRITE_STATE
SET_BUSY	0	1	1
CLR_BUSY	1	0	0
LD_CNT	0	1	0
CNT_EN	0	0	1
ADDR_SEL	0	1	1
ZERO_WE	0	1	1

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-

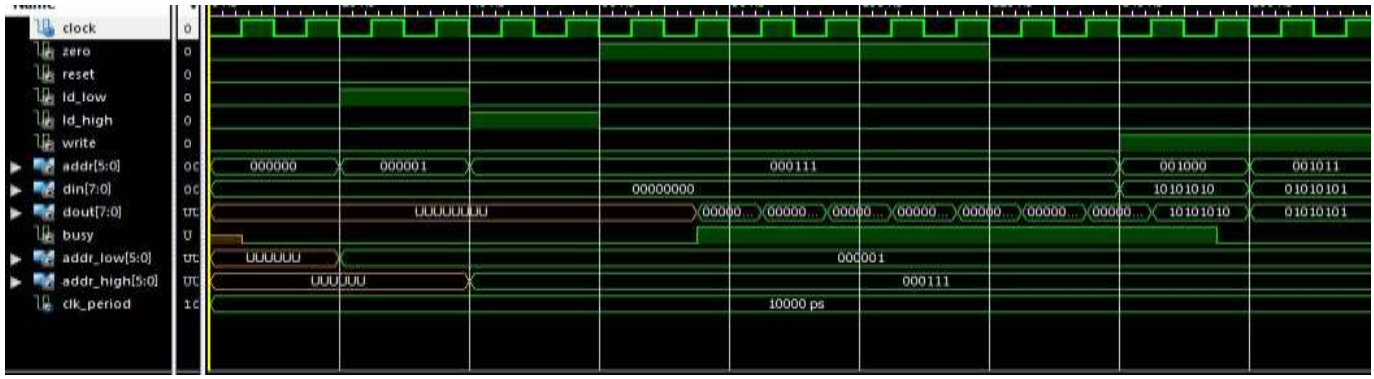
```
set_busy <='1';
clr_busy<='0';
ld_cnt<='1';
cnt_en<='0';
addr_sel<='0';
zero_we<='0';
```

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';
```

Testing:

Bothe the modes of system are verified. First R1 and R2 are loaded with high and low addresses. Then zero asserted. It resulted in system writing zeroes through the range of addresses between given input addresses.



For normal mode, zero is deasserted and different data was written to two different address locations by asserting write signal. Then by deasserting the write signal same addresses were given as input and the data read from these locations was as expected.