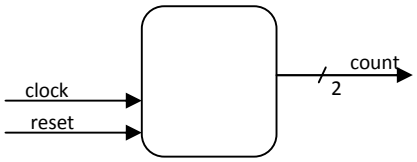
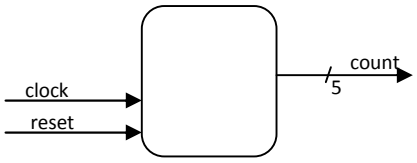
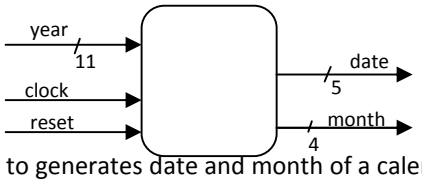
| Assignment4 | IIIT-H<br>VLSI Architectures | | | | |
|---|---|---|---|---|---|
| | Date of upload | 10Feb2014 | Date of submission | 19Feb2014 | |

1. Draw the functional block diagram of each problem using Vectored Mux/Comparators/Adder.
2. Use Pencil to draw all the diagrams
3. Name all internal signals appropriately ,
    a. sig1,sig2 .. for one bit signals and
    b. bus1,bus2,bus3 for greater than one bit signals
4. Write Verilog code (Design.v) and TestBench (tb.v)
5. Use DataFlow model style of Verilog Coding. (using “assign” statements)
6. Make appropriate assumptions if needed and mention the same.

| SlNo | Questions | |
|---|---|---|
| 1 | Sandwich counter.<br>  <br><br>Design a counter which follows the following sequence.<br><br>0,3, 1,2,2,1,3,0<br>Counter should start with “0” on reset<br>All odd locations it should increment by unity (observe the black colored entries, they differ by unity and are incrementing)<br>All even locations should decrement, starting with “3”<br>The counting sequence should repeat the same forever.<br><br>Please follow the following naming style only for consistency between every student’s solution<br>module student ( input wire clock,reset,<br>output reg [1:0] count);<br>//Your code starts here.<br>….<br>….<br>endmodule | |

| | | |
|---|---|---|
| 2 | Collapsing counter.<br>  <br><br>Design a counter which starts with two initial values say 25 and 8 and reduces to zero "0" by generating the next value, which is an absolute difference between the two latest counts.<br>The counting sequence should be<br>25,8,17,9,8,1,7,6,1,5,4,1,3,2,1,1,0,0,0,0,0,0<br>Step 1: Generate 25 on first clock ( on reset)<br>Step2: Generate 8 on the second clock<br>Step3: The next value is a modulus subtraction of 25 and 8, which is 17<br>Step 4: The next value is modulus subtraction of the latest two counts, 8 and 17 which is 9<br>Step5: Continue Step4 until you reach "0".<br>Step6: After reaching "0", retain zero for ever.<br><br>Please follow the following naming style only for consistency between every student's solution<br>module student ( input wire clock,reset,<br>output reg [4:0] count);<br>//Your code starts here.<br>….<br>….<br>Endmodule | |
| 3 |  <br><br>Design a counter to generates date and month of a calendar as follows<br>Example: If "year" input is 2014 then the output should be<br>Month 1: Date: 1,2,3,….. 31,<br>Month 2: Date: 1,2,3,……28,<br>Month3 : Date: 1,2,3,……31,<br>…. etc<br>Guidelines<br>1. All the months Jan,Mar,May,July,Aug,Oct,Dec should contain 31days<br>2. All the months Apr,Jun,Sep,Nov should contain 30 days<br>3. Feb month should contain 29 days on a leap year and 28 otherwise.<br>4. Leap year should be calculated from a 11 bit "year" input as follows<br>All years which are divisible by 4 are leap years, except the ones which are divisible by 100 (hundred)<br>Please follow the following naming style only for consistency between every student's solution<br>module student ( input wire clock,reset,<br>input wire [10:0] year,<br>output reg [4:0] date,<br>output reg [3:0] month);<br>//Your code starts here.<br>….<br>….<br>endmodule | |