Solutions to Midterm 1

There were two versions of each question. The questions and answers for both versions are given below.

Question 1

Fill the table below with the value of each expression as a Verilog numeric literal including the correct width and the correct value in *hexadecimal* base. Assume the following declarations:

```
logic [3:0] x ;
logic [7:0] y ;
```

and that `y` has the value `8'h83` (or `8'h5A`) and that `x` has the value `4'b0110`. The first row has been filled in as an example. You need not show your work or draw another box around the answer.

expression	value
`y[3:0]`	`4'h3` or `4'hA`
`y[5:0] + x * x`	
`x && ! ( 8 & y ) ^ 1`	
`~ y & 1 ? ~ ( y & 1 ) ? 0 : 1 : 2`	

Answers

For `y = 8'h83`:

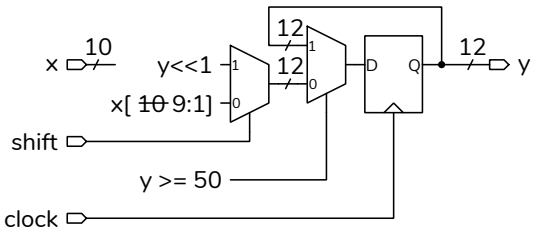
expression	value
`y[3:0]`	`4'h3`
`y[5:0] + x * x`	`6'h27`
`x && ! ( 8 & y ) ^ 1`	`1'h0`
`~ y & 1 ? ~ ( y & 1 ) ? 0 : 1 : 2`	`32'h2`

and for `y = 8'h5a`:

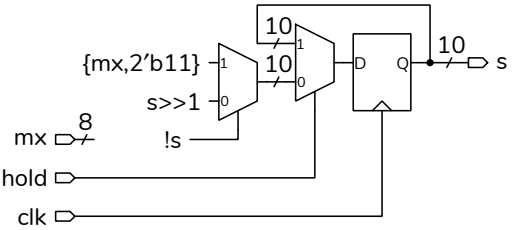
expression	value
`y[3:0]`	`4'ha`
`y[5:0] + x * x`	`6'h3e`
`x && ! ( 8 & y ) ^ 1`	`1'h1`
`~ y & 1 ? ~ ( y & 1 ) ? 0 : 1 : 2`	`32'h0`

Question 2

Write a Verilog module named `adj` (or `ctrl`) that implements the following block diagram. The diagram follows the course conventions for block diagrams. Follow the course coding guidelines but omit comments.



or:



Answers

```
module adj
  ( output logic [11:0] y,
    input logic [9:0] x,
    input logic shift, clock ) ;

   always_ff @(posedge clock)
      y <= y >= 50 ? y : shift ? y << 1 : x[9:1] ;
```

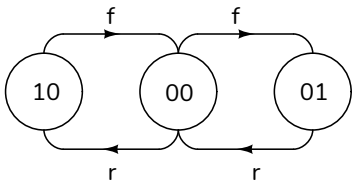
```
endmodule

module ctrl
  ( output logic [9:0] s,
    input logic [7:0] mx,
    input logic hold, clk ) ;

   always_ff @(posedge clk)
     s <= hold ? s : !s ? {mx,2'b11} : s>>1 ;

endmodule
```

Question 3

First Version A state machine with two bits of state and three states controls a motor. The states are:

`00` (stopped), **`10`** (reversed) and **`01`** (forward). The output of the state machine is the two bits of state. There are two bits of input: **`fwd`** (forward) and **`rev`** (reverse).

Write the truth table corresponding to the following state transition diagram:

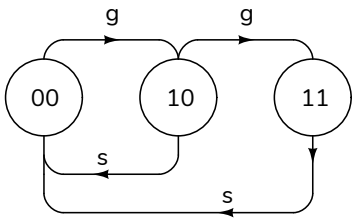


f: fwd && !rev
r: !fwd && rev

Second Version A state machine with two bits of state and three states controls a motor. The states are: **`00`** (stopped), **`10`** (slow) and **`11`** (fast).

The output of the state machine is the two bits of state. There are two bits of input: **`go`** (to speed up) and **`stop`** (to stop).

Write the truth table corresponding to the following state transition diagram:



s: stop && !go
g: !stop && go

Your table must include the state, input and next state. You may, but are not required to, use any of the simplification conventions described in the lecture notes.

Answers

First Version We can list all possible combination of state and input and show the next state:

state	fwd	rev	next state
00	0	0	00
00	0	1	10
00	1	0	01
00	1	1	00
01	0	0	01
01	0	1	00
01	1	0	01
01	1	1	01
10	0	0	10
10	0	1	10
10	1	0	00
10	1	1	10

But using the simplification that there are no state changes if none of the conditions in the table match, this can be written using one row per possible state transition:

state	fwd	rev	next state
00	0	1	10
00	1	0	01
01	0	1	00
10	1	0	00

Second Version As in the previous version of the question, the simplified truth table only needs one row per transition arrow in the diagram:

state	fwd	rev	next state
00	1	0	10
10	1	0	11
10	0	1	00
11	0	1	00