Solutions to Quiz 2

There were two versions of each question. The values and the 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 [7:0] x ;
logic [3:0] y ;
```

and that `x` has the value `8'h38` (or `8'hA6`) and that `y` has the value `4'b0101`. 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
x[3:0]	4'h8 (or 4'h6)
{ x >> 4, y == 5 }	
x ^ y ^ x + 1	

Answers

For `x=8'h38` and `y=4'b0101`:

expression	value
x[3:0]	4'h8
{ x >> 4, y == 5 }	9'h7
x ^ y ^ x + 1	32'h4

and for `x=8'ha6` and `y=4'b0101`:

x[3:0]	4'h6
expression	value
{ x >> 4, y == 5 }	9'h15
x ^ y ^ x + 1	32'h4

Question 2

A digital circuit using a 100 MHz clock generates a 10 (or 20) ms delay by decrementing a counter once per clock cycle. The count starts at $N-1$ and reaches 0. What is N? Show your work.

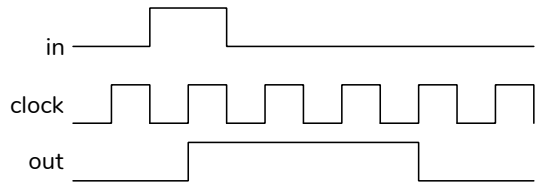
Answers

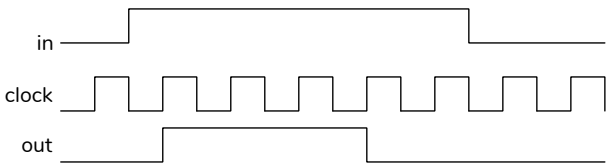
A clock frequency of 100 MHz corresponds to a clock period of $T = 1/100\,\text{Mhz} = 10\,\text{ns}$. The required delay is $NT = 10\,\text{ms}$ (or 20 ms) so $N = 10\,\text{ms}/10\,\text{ns} = \boxed{1\times10^6}$ (or $\boxed{2\times10^6}$).

Question 3

A "pulse width normalizer" circuit has two one-bit inputs named **`reset`** and **`in`**, a one-bit output named **`out`**, and a clock. It is a Moore state machine: the output only changes on the rising edge of the clock. The circuit outputs pulses with a duration of three clock periods for every input pulse.

The following diagrams show examples of the required behaviour when the **`in`** pulse is shorter than 3 clock periods and when the **`in`** pulse is longer than 3 clock periods:





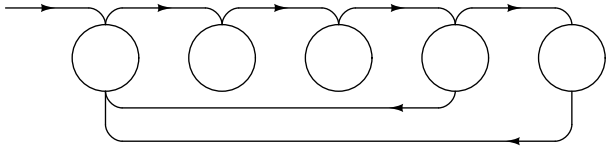
The output behaves as follows:

1. if **`reset`** is high (1) then **`out`** goes low (0), otherwise

2. while **in** is low, **out** stays low,
3. when **in** goes high, **out** goes high for exactly three clock periods, even if **in** goes low during that time,
4. after the output pulse:
    - if **in** is low, **out** goes low until **in** goes **hi** again (as in 2 above), otherwise
    - if **in** is high, **out** goes low and remains low until **in** has gone low and then stays low until **in** goes **hi** again (as in 2 above),

You may ignore cases where the input pulse duration is less than one clock period or where the time between input pulses is less than five clock periods.

The state transition diagram below implements this state machine. Fill in the circles that describe each state and label the edge (arrows) with the state transition conditions. Follow the conventions used in the lecture notes. Use a **one-hot (or binary)** encoding for the states.

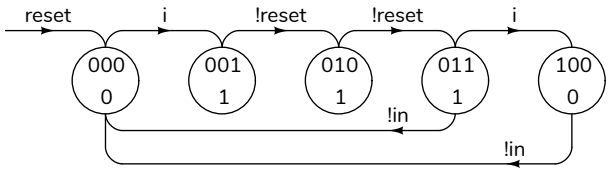


Answers

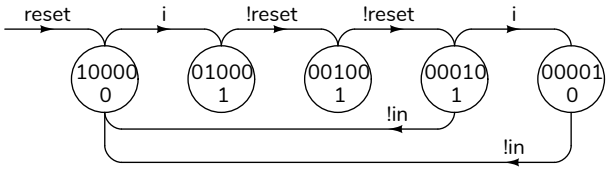
The solutions below have a "reset" state, three states where the output is 1 and one state that sets the output low if the duration of the input pulse is longer than three clock periods.

The state "bubbles" show both the state encoding value and the output for that state. Other values could be used for each state.

The state transitions are labelled with Verilog expressions involving the inputs. Marks were not deducted if the reset input was only used in the transition to the leftmost state.



i: in && !reset



i: in && !reset