⚡ ELECTRICAL ENGINEERING

Implementing FSM with Sequential Logic
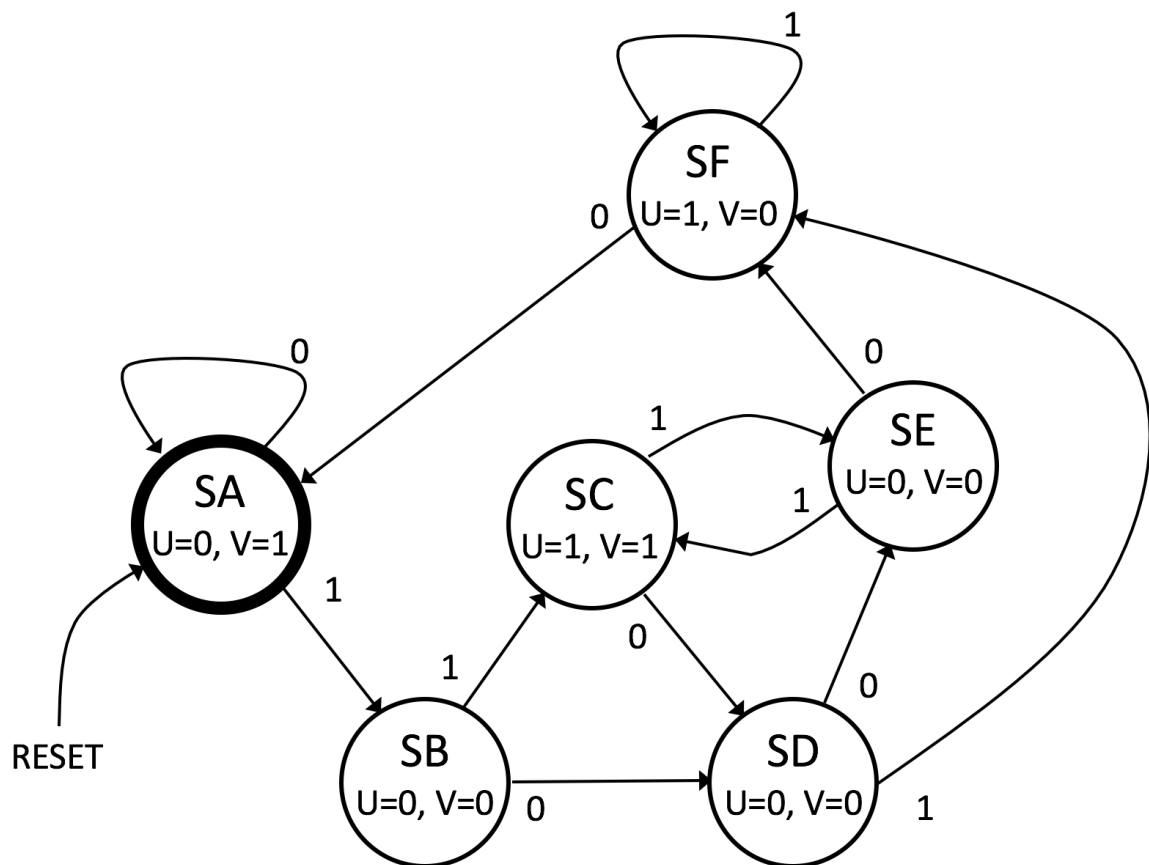
Asked 11 months ago　　Modified 11 months ago　　Viewed 95 times
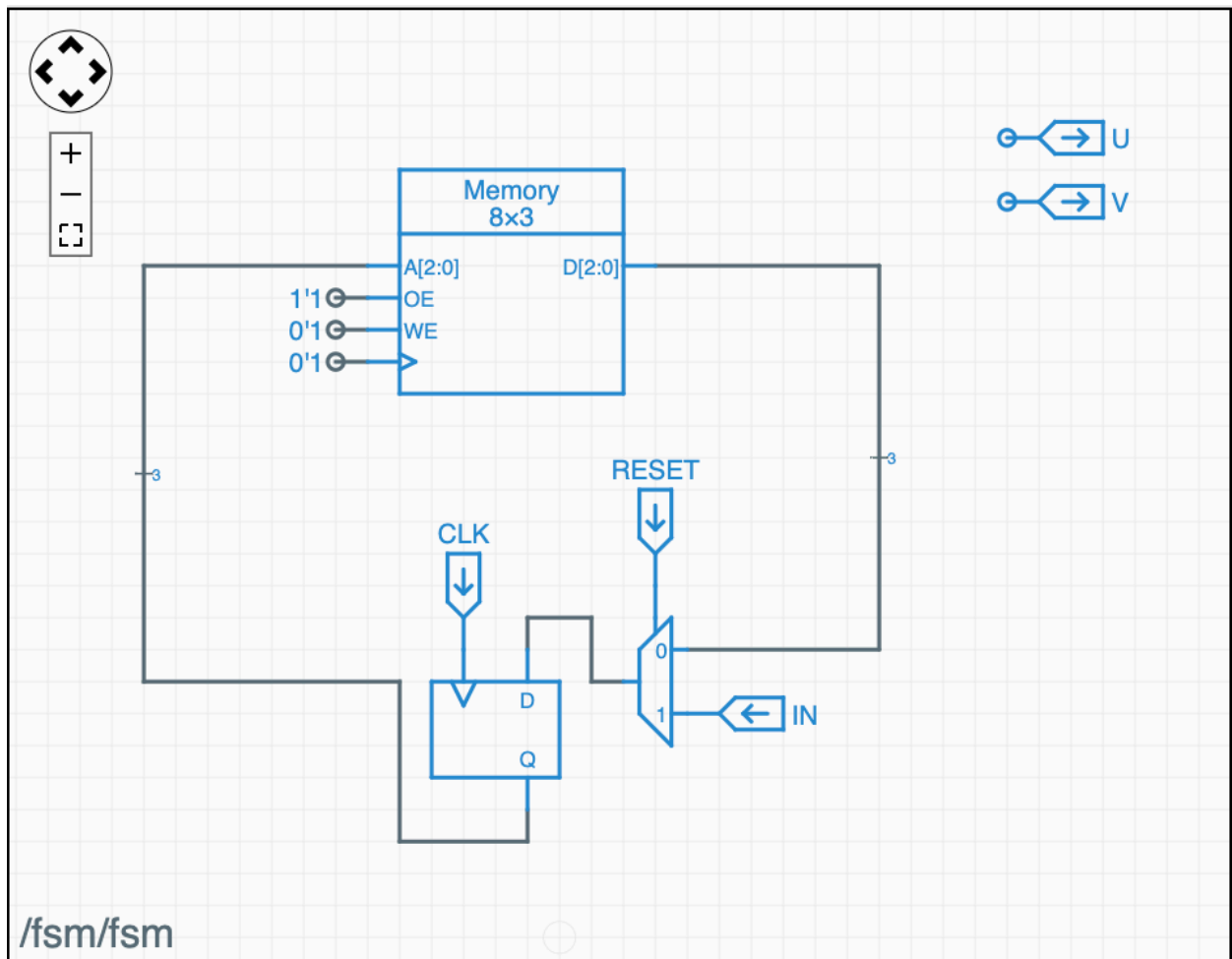
1

I am trying to complete Problem 4 of this MIT Computation Structures lab, which requires you to use a circuit simulator called Jade to build a sequential logic circuit that implements the below FSM:
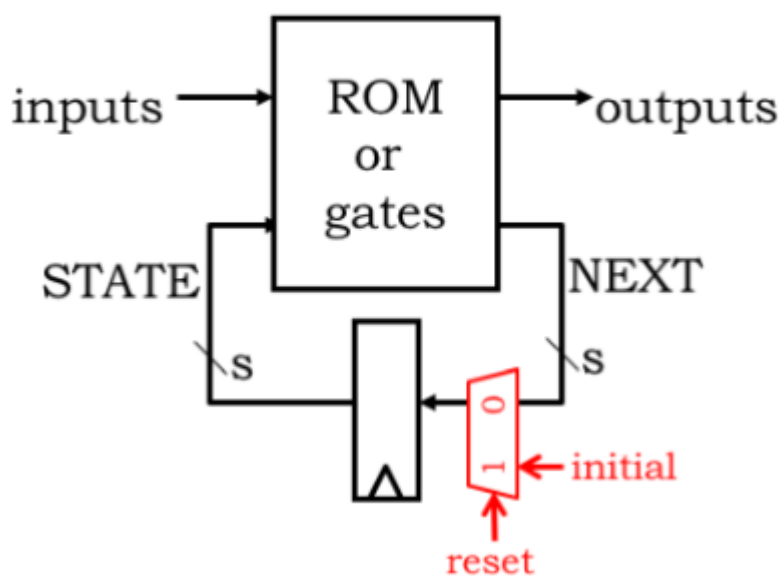

So far, the below is what I have:

/fsm/fsm

I'm trying to implement this using the ROM/register combo that is referenced in the problem (and on the lecture slides the problem references).

However, I'm not sure how to connect the U and V outputs. The ROM/register combo shown in lecture looks a bit different, as it has a separate spot for inputs and outputs:

Does anyone have any thoughts on how I might be able to implement that in Jade?

register   rom   sequential-logic

Share Cite Edit Follow Flag

asked Jun 3, 2023 at 17:28

Coder1913
25   2

I don't know Jade and I'm certainly not going to go learn how to use it (tools, semantics and syntax, etc) just to solve this problem for you. Luckily, you just seem stuck on the U and V parts. Why not include those as part of the ROM and simply capture them with DFFs along with the next state at the rising edge of the clock? – periblepsis Jun 4, 2023 at 3:56

The bottom diagram shows inputs&outputs connected to the lookup table. The top one does not (depending on whether IN is *initial* or *input*). – greybeard Jun 4, 2023 at 7:38

Coder, Since you are non-responsive, I'll just hand you an answer but one that doesn't use a ROM. If you help us help you, I may write more in an answer. – periblepsis Jun 5, 2023 at 1:39

Appreciate the responses. Very busy day today so haven't had a chance to take another crack at this problem today but hope to have some time tomorrow. – Coder1913   Jun 5, 2023 at 3:14

@Coder1913 Okay. When you get a moment, then... – periblepsis Jun 5, 2023 at 21:31
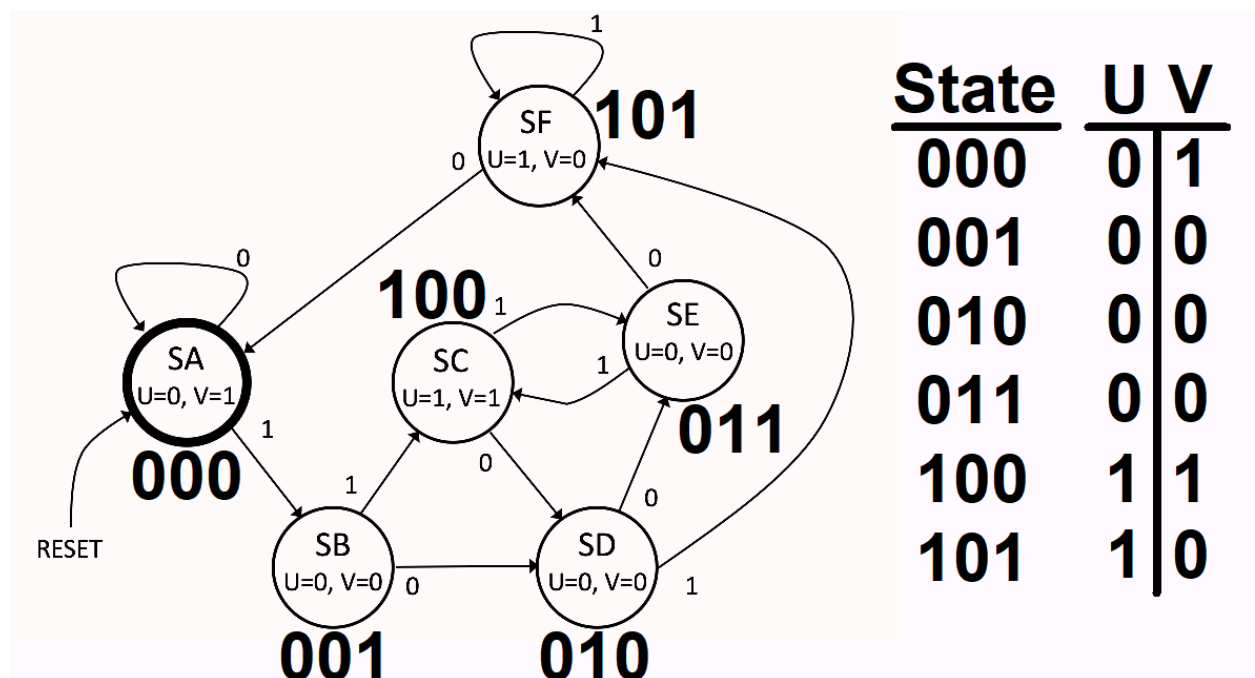
|

1 Answer

Sorted by: Highest score (default) ⇕

0

Let's start by assigning state numbers to each state:

My reasoning here is that U is just a copy of the most significant DFF. That makes U easy to produce. Next, I decided that $V = 1$ when both of the less significant DFFs are zero.

That's also easy to produce.

From here it is now possible to create a table:

Q_C	Q_B	Q_A	IN	Q_C	Q_B	Q_A
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	x	x	x
1	1	0	1	x	x	x
1	1	1	0	x	x	x
1	1	1	1	x	x	x

(Header: "State" spans Q_C Q_B Q_A; "IN"; "New State" spans Q_C Q_B Q_A.)

That table should be very easy to stick into a ROM along with the associated U and V values. You could either latch them from the ROM output at the time of the clock-driven state transition or else derive them from the current address presented at the ROM address bus prior to the next clock.

Alternately, you avoid the ROM and instead just produce the Karnaugh maps:

$Q_C\ D$	$\overline{Q_A}\ \overline{IN}$	$\overline{Q_A}\ IN$	$Q_A\ IN$	$Q_A\ \overline{IN}$
$\overline{Q_C}\ \overline{Q_B}$	0	0	1	0
$\overline{Q_C}\ Q_B$	0	1	1	1
$Q_C\ Q_B$	x	x	x	x
$Q_C\ \overline{Q_B}$	0	0	1	0

$Q_B\ D$	$\overline{Q_A}\ \overline{IN}$	$\overline{Q_A}\ IN$	$Q_A\ IN$	Q_A
$\overline{Q_C}\ \overline{Q_B}$	0	0	0	
$\overline{Q_C}\ Q_B$	1	0	0	
$Q_C\ Q_B$	x	x	x	
$Q_C\ \overline{Q_B}$	1	1	0	

$Q_A\ D$	$\overline{Q_A}\ \overline{IN}$	$\overline{Q_A}\ IN$	$Q_A\ IN$	$Q_A\ \overline{IN}$
$\overline{Q_C}\ \overline{Q_B}$	0	1	0	0
$\overline{Q_C}\ Q_B$	1	1	0	1
$Q_C\ Q_B$	x	x	x	x
$Q_C\ \overline{Q_B}$	0	1	1	0

Get updates on questions

And work out the logic required for tha   and answers   mizing logic from those tables.

Share Cite Edit Follow Flag

answered Jun 6, 2023 at 0:58

periblepsis
9,475 1   4   18

▲ Thank you for the very detailed and helpful response. This definitely has me on the right track. I think now the main thing is just digging into Jade and figuring out how to implement it there. It's not the most intuitive simulator, but I think I'm close. – Coder1913 Jun 6, 2023 at 2:39

▲ @Coder1913 Best wishes. I hope it does help you get where you need to go with it. :) You might consider posting up the resulting Jade code so that I can learn something about Jade from you. ;) – periblepsis Jun 6, 2023 at 2:42 ✎