FSM Problem Solutions

Madhav P. Desai

February 20, 2018

Implement an FSM with the following specification:

- The set of input symbols is $\Sigma = \{a,\ b\}$.
- The set of output symbols is $\Lambda = \{Y, N\}$.

The behaviour of the FSM is as follows:

- The FSM outputs a Y at time instant k only if the last 4 inputs were either

  ```
  abab
  baba
  ```

  (that is, either $x(k) = a, x(k-1) = b, x(k-2) = a,\ x(k-3) = b$ or $x(k) = b, x(k-1) = a, x(k-2) = b,\ x(k-3) = a$). Otherwise the FSM outputs N.

Suppose you are given the following building blocks:

- Inverters, NAND2, NOR2 gates each with a delay of 2 units.
- D-flipflops with $clock \rightarrow q$ delay of 3 units, set-up time of 2 units and hold-time of 2 units.

1. Design an abstract Mealy machine (identify a potential set of states and next-state, output functions) which implements the required behaviour.

   - We introduce states RST, Sa, Sb, Sab, Sba, $Saba$, $Sbab$. We add a reset symbol $sreset$ to the set of in put symbols to put the machine into RST at the beginning. The next state and output functions are then

Present-state	Input	Next-state	Output
_	sreset	RST	dont-care
RST	a	Sa	N
RST	b	Sb	N
Sa	a	Sa	N
Sa	b	Sab	N

Sb	a	Sba	N
Sb	b	Sb	N
Sab	a	Saba	N
Sab	b	Sb	N
Sba	a	Sa	N
Sba	b	Sbab	N
Saba	a	Sa	N
Saba	b	Sbab	Y
Sbab	a	Saba	Y
Sbab	b	Sb	N

2. Encode the input, output and state symbols using binary encoding.

   - Try two state encodings: the one-hot encoding and a compact encoding using $log_2|Q|$ state variables.
   - We illustrate the binary encoding. Encode the inputs using two variables *reset*, x as

	variables	
	reset	x
symbol		
sreset	1	dont-care
a	0	0
b	0	1

   the output symbols by one variable u as

	variables
	u
symbol	
Y	1
N	0

   and the states by using three variables $s0$, $s1$, $s2$ as

	variables		
	s2	s1	s0
symbol			
RST	0	0	0
Sa	0	0	1
Sb	0	1	0
Sab	0	1	1
Sba	1	0	0
Saba	1	0	1
Sbab	1	1	0

3. Complete the logic network for the FSM based on your encodings. Don't forget to add the reset signal. You may assume that the output is a dont-care when the reset signal is applied.

- We need to express the output variable u in terms of the input and state variables. The K-map for u (ignoring $reset$ since u is a dont-care when $reset = 1$) is

```
      x,s2
           00 01 11 10
s1,s0
   00
   01             1
   11          d  d
   10          1
```

  which is minimized to $u = x.s2.s0 + \overline{x}.s2.s1$. The next state functions $ns2$ can be written as $\overline{reset}.f2$, where $f2$ is 1 only in the following cases

```
s2 s1 s0 x
0  1  0  0  (Sb,a) -> Sba
0  1  1  0  (Sab,a) -> Saba
1  0  0  1  (Sba,b) -> Sbab
1  0  1  1  (Saba,b) -> Sbab
1  1  0  0  (Sbab,a) -> Saba
```

  which leads to the following map for $f2$:

```
      s0,x
           00 01 11 10
s2,s1
00
01         1          1
11         1       d  d
10             1   1
```

  so that $f2 = \overline{x}.s1 + s2.s1.x$. Thus, $ns2 = \overline{reset}.(\overline{x}.s1 + s2.s1.x)$. The other next-state functions can be calculated in a similar fashion.

4. Confirm, using a VHDL model, that your implementations are correct. Check the following test-case

```
k   0      1 2 3 4 5 6 7 8 9
x   reset  a b b a b a b a a
y   _      N N N N N Y Y Y N
```

5. Characterize the timing of your FSM implementations (find the minimum and maximum values of the input to flop, flop to flop, flop to output and input to output delays).

   - At each wire in the circuit, find the earliest and latest time instant within the clock-period at which the wire can change its value. To start, the outputs of the flip-flops will change exactly at 3 units after

the rising edge of the clock. At a wire w driven by a gate G of the circuit, the latest time at which the wire can change is

$$t_w^{max} \;=\; \max_{u \in inputs(G)} \; t_u^{max} + delay_G^{max}$$

and the earliest time at which the wire can change is

$$t_w^{min} \;=\; \min_{u \in inputs(G)} \; t_u^{min} + delay_G^{min}$$

In this manner, calculate the times at which every wire can change. Now for the circuit to work correctly, confirm that for each flip-flop F with input $d(F)$:

$$\begin{aligned} t_{d(F)}^{max} &\leq T_{clock} - Setup(F) \\ t_{d(F)}^{min} &\geq Hold(F) \end{aligned}$$