INF2270, exercise in sequential logic

.

January 26, 2012

Abstract

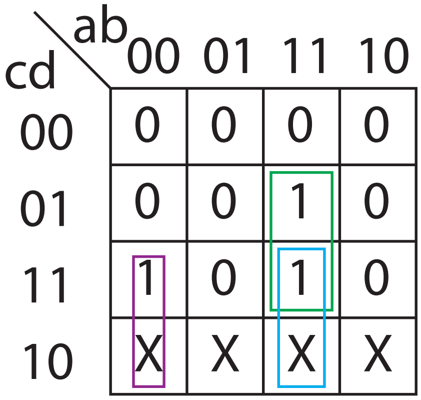
In this exercise you can test your skills in designing simple sequential logic

Task

Imagine you want to construct a simple control circuit for the floor heating in your bathroom. As input signals you have a clock signal with a clock period of 15 minutes, and a control switch with 5 different positions (e.g. a control *input* c coded as binary number, e.g. '000', '001', '010', '011', '100' or '00001', '00010', '00100', '01000', '10000'). The output signal is one bit where '1' means that the heating is on and '0' that the heating is off. If the control switch is in position 0, the floor heating should be switched off all the time, if it is in position 1, the floor heating should be turned on 15 minutes per hour, in position 2 30 minutes per hour, in position 3 45 minutes per hour, and in position 4 all the time. You may assume that the heat distribution in the floor is so slow that effectively this control mechanism results in 5 different floor temperatures.

1. Construct a Moore FSM to achieve this controller. Draw the state transition diagram and draw the sequentil logic circuit. Use D-flip-flops and/or T-flip-flops as memory cells. Hint: you will need 8 states, i.e. coding for each 15 minute period within an hour combined with the heating being on or off. It will be convenient to code the *state* with a two-bit time variable t and a one bit control variable h that turns on and off the heating. Note that the part of the FSM that controls t can conveniently be a 2-bit counter as presented in the lecture. So you need only to derive the combinational logic to govern the state transistions of state variable h.

2. Construct a Mealy FSM for the same task. Hint: you will only need 4 states since you can deduce the on/off signal for the heater from a combination of the control switch position and from which 15 minute period the controller is in. So you need only states to code for the three 15 minute periods within an hour.



cd \ ab	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	0	1	0
10	X	X	X	X

Figure 1: K-map with 'X's

On the Karnaugh maps for this exercise

There may be a new element to the Karnaugh maps of this exercise: Some possible input constellations are of no consequence for the FSM since they do not occure: If the input control signal c is encoded as a binary code '000' to '100', then the possible inputs $c = 101, 110, 111$ will never occure since the user control panel will not offer these. Thus, the output h for these cases can be both '1' or '0' with no consequence what so ever for the functioning of the FSM and the output in the K-map should thus be marked with an 'X', i.e. 'don't care'. Those 'X's become a kind of Joker in the K-map: you can use them just like '1's to make bigger groups of the '1's that are there. Check the example in figure 1!

Furthermore there might be K-maps with 5 inputs occuring in this exercise. They have been discussed in the lecture (but will not occure at the exam).