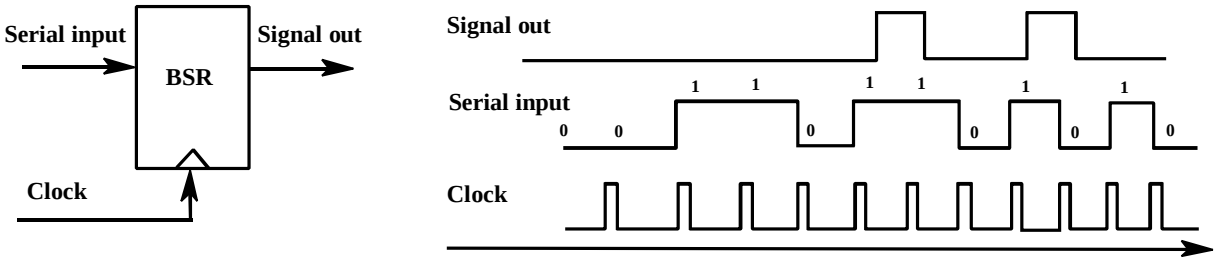
Department of Computing
Course 112 - Hardware
Tutorial 6

This tutorial contains exercises that are assessed. Make certain that your name, initials and group are clearly filled in. Do all your work on this sheet and hand it in at the end of the tutorial session.

LAST NAME_____**INITIALS**_____**GROUP**_____

PROBLEM DEFINITION

In this tutorial a **Bit Sequence Recogniser (BSR)** circuit will be designed. The **BSR** circuit has one data input, one signal output and one clock input line:



When a given sequence of bits has "arrived" on the **Serial Data** input line, the output becomes **1** otherwise the output is **0**. The sequence may be allowed to "**overlap**" or not allowed to overlap, depending on the problem definition. For our problem we have a four-bit sequence: **1101**. The input/output trace for **overlapped recognition** is shown in the timing diagram. For this problem you may assume that the serial input signal is synchronised with the same clock as the circuit; therefore, the input changes exactly at the same time as the output (i.e. at the rising or leading edge of the clock). The problem can be solved by using a Moore finite state machine with five states. The states indicate that the circuit: "has received a **1** already", has received two **1**s", "received the sequence **110**", or the full sequence **1101**. There is also an idle state:

State	State Type
1	Idle
2	has received 1
3	has received 11
4	has received 110
5	has received 1101

Problem 1.

Draw the finite state machine representation of the circuit

.

Problem 2
For the **BSR** fill in the transition table, assign flip-flop bits to each state, fill in the Karnaugh maps, minimize, and design the circuit.

Input	State	State Type	Q3	Q2	Q1	State	D3	D2	D1
0	1	Idle				1			
0	2	. . 1							
0	3	. . 11							
0	4	. . 110							
0	5	. . 1101							
0	6					X	X	X	X
0	7					X	X	X	X
0	8					X	X	X	X
1	1	Idle							
1	2	. . 1							
1	3	. . 11							
1	4	. . 110							
1	5	. . 1101							
1	6								
1	7								
1	8								

D1

IQ3 \ Q2 Q1	00	01	11	10
00				
01				
11				
10				

D2

IQ3 \ Q2 Q1	00	01	11	10
00				
01				
11				
10				

D3

IQ3 \ Q2 Q1	00	01	11	10
00				
01				
11				
10				

D3 =
D2 =
D1 =

Problem 3
Design the output logic which will provide the correct **Signal** output for each state (Q3Q2Q1).

Problem 4
Check what happens to your circuit when it starts in an unused state.

Input	Current State	Q3	Q2	Q1	Next State	D3	D2	D1
0	6							
0	7							
0	8							
1	6							
1	7							
1	8							