TÉCNICO LISBOA

Bologna Master Degree in Electronics Engineering

Design, Test and Reliability of Electronic Systems

1st Project

Group 2

Diogo Luís | 96922

Duarte Marques | 96523

Prof. Fernando Manuel Duarte Gonçalves

1st semester, P2

November 30th 2023

1 Introduction

The main objective of this project is to develop a **controller** with three inputs (CLK, RESET and START) and three outputs (OUT, BIST_END and RUNNING). This controller must be sensitive to the rising edges of the clock, which is defined to have a frequency of 10 MHz. It should produce a waveform with M sequences, in which each of these is characterized by OUT at '1' for N clock periods followed by OUT at '0' for one clock period. In the end, OUT must go HIGH again for another N clock periods. The values of these two parameters are indicated in Table 1. While a full sequence is being produced, the output RUNNING must go HIGH, whereas BIST_END goes HIGH whenever a full sequence has been completed and OUT remains at '0'. When the RESET signal is activated, OUT goes LOW and a subsequent rising edge in START (with RESET at '0') leads to a new full sequence in the output. Additionally, when the full sequence is complete, a 0→1 transition in the START signal is enough to restart it - i.e., the activation of RESET is not necessary in this case. However, all transitions of START during the generation of the full sequence are discarded.

Table 1: Parameters which characterize the full sequence produced in OUT.

Parameter	**N**	**M**
Value	6	11

2 Finite state machine

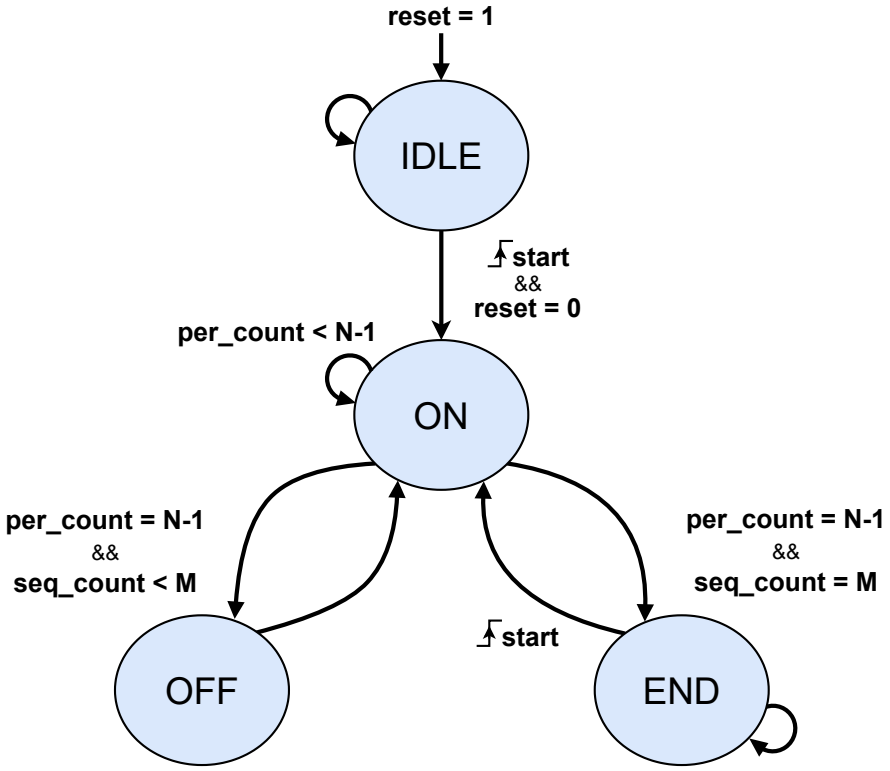

Figure 1: FSM state diagram.

2.1 IDLE state

The controller assumes this state is when RESET is active. While in IDLE, the counters used to track the number of periods and the number of sequences in OUT (PER_COUNT and SEQ_COUNT, respectively) are set to zero in the rising edge of the clock, since the only possible next state after IDLE (besides itself) is ON, where a brand new sequence should start. These counters are updated using the auxiliary values PER_RST and SEQ_RST (respectively). Additionally, the controller outputs RUNNING and BIST_END are both set to '0'. The controller remains in the current state until a 0→1 transition in START - registered with NEW_SEQ - occurs after RESET is set to '0'.

2.2 ON state

In this state, the output RUNNING goes HIGH and the signal OUT is set to '1' for N clock periods. The former serves as the enable for the period counter, leading to an increment in the value of PER_COUNT in each rising edge of the clock while in the ON state. Once a single sequence is completed (i.e., when the value of PER_COUNT equals N-1, since its initial value is zero), the controller changes to either OFF or END state. The first situation occurs when the desired M+1 sections with OUT at '1' have not yet been executed, which means that the value of SEQ_COUNT is lower than M. On the other hand, if SEQ_COUNT (which starts at zero) equals M, the controller transitions to the END state, which indicates that the full sequence has been generated in the output.

2.3 OFF state

One of the main functions of this state - in which OUT goes LOW - is to indicate that another single sequence has been generated, by adding one to the value of SEQ_COUNT. This is performed by setting its enable signal SEQ_EN to '1'. Additionally, the value of PER_COUNT is set to zero (by changing PER_RST to '1') in order to prepare the controller to start a new single sequence in the ON state, which necessarily follows the current state - unless RESET is active, of course.

2.4 END state

This state indicates that a full sequence has been generated, thus OUT and RUNNING go LOW, while BIST_END must go HIGH. Additionally, both counters PER_COUNT and SEQ_COUNT are set to zero in the rising edge of the clock, in order to prepare the controller for a new complete sequence, which occurs once it goes into ON state once again. For this to happen without setting RESET to '0' - which leads to the IDLE state first - a 0→1 transition in START (registered in NEW_SEQ) is enough.

Table 2: Values of relevant signals for each state.

	Internal signals			Output signals		
	PER_RST	**SEQ_RST**	**SEQ_EN**	**OUT**	**BIST_END**	**RUNNING**
IDLE	1	1	0	0	0	0
ON	0	0	0	1	0	1
OFF	1	0	1	0	0	1
END	1	1	0	0	1	0

3 Testbench

To validate the behaviour of the controller, a testbench was developed taking into account several test scenarios. Having defined a time unit of 10 ns, an arbitrary frequency of 10 MHz - corresponding to a period of 10 time units - was considered. This means that a full sequence in the output corresponds to 830 time units, taking into account the parameters defined in Table 1.

After initially deactivating the RESET signal, a rising edge in START leads to the generation of the full sequence, starting in the very first rising edge in the clock after the aforementioned change. In the second scenario, another full sequence is subsequently generated by applying another rising edge in START - this time, without using the RESET signal. Along the sequence, it is also tested that a rising edge in START does not affect this behaviour of the controller - as desired, while the full sequence is not completed, further transitions in the START signal must be ignored. Having forced a third rising edge in START, the full sequence is then interrupted midway by activating RESET. When this signal has been deactivated, another full sequence (with START permanently at '1') is then created only after the rising edge in START occurs, as desired. Finally, it is proven that, while RESET is active, the transitions in START are ignored - only after RESET goes LOW a 0→1 transition in START leads to a final full sequence.

Along this behavioral simulation, the expected variations in the outputs RUNNING and BIST_END were also observed. As seen in Fig. 2, the former remains at '1' only when a full sequence is being generated, while the latter changes to '1' only when a full sequence has been completed. The proper behaviour of the counters and the changes in the current state during a full sequence can be seen in more detail in Fig. 3. The quality of this testbench was verified by the attainment of a code coverage of 100% for the Block and FSM categories, using Xcelium.

Figure 2: Behavioral Simulation results displayed in waveforms.

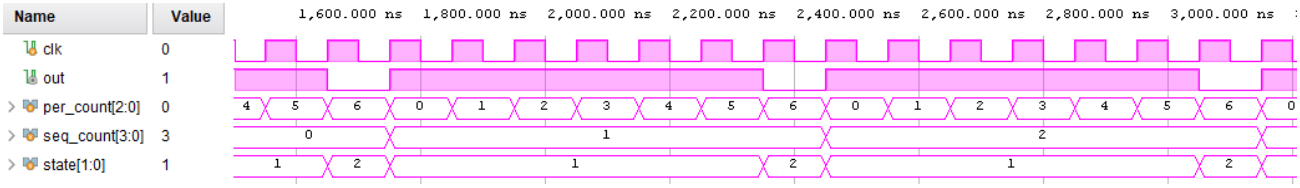

Figure 3: Zoomed-in waveforms during a full sequence in the output.