International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:1, 2015

A Novel Approach to Asynchronous State Machine Modeling on Multisim for Avoiding Function Hazards

L. Parisi, D. Hamili, N. Azlan

Abstract**—**The aim of this study was to design and simulate a particular type of Asynchronous State Machine (ASM), namely a 'traffic light controller' (TLC), operated at a frequency of 0.5 Hz. The design task involved two main stages: firstly, designing a 4-bit binary counter using J-K flip flops as the timing signal and, subsequently, attaining the digital logic by deploying ASM design process. The TLC was designed such that it showed a sequence of three different colours, i.e. red, yellow and green, corresponding to set thresholds by deploying the least number of AND, OR and NOT gates possible. The software Multisim was deployed to design such circuit and simulate it for circuit troubleshooting in order for it to display the output sequence of the three different colours on the traffic light in the correct order. A clock signal, an asynchronous 4-bit binary counter that was designed through the use of J-K flip flops along with an ASM were used to complete this sequence, which was programmed to be repeated indefinitely. Eventually, the circuit was debugged and optimized, thus displaying the correct waveforms of the three outputs through the logic analyser. However, hazards occurred when the frequency was increased to 10 MHz. This was attributed to delays in the feedback being too high.

Keywords**—**Asynchronous State Machine, Traffic Light Controller, Circuit Design, Digital Electronics.

I. Introduction

Asynchronous state machines (ASMs) deploy sequential logic to follow set sequences and they can be used for detecting PIN numbers or setting alarm codes. The 'sequential logic' of ASMs allows them to remember which values have been inputted into the circuit both previously and currently, to follow a specific sequence such as typing in a PIN number in a shop or setting an alarm code. While synchronous circuits depend on the edge of a clock to change state, an asynchronous circuit does not refer to any clock signal used. Thus, the latter circuit progresses to the next programmed step immediately, theoretically resulting in faster, simpler, less power-demanding logic than those relying on a clock to control them. The TLC that was designed and tested throughout two laboratory sessions is an example of an ASM application in real life. The first traffic light system in the world was installed near the House of Commons in London in 1868. In the 20th century, several automatic electric traffic light systems were created, including a hanging four-way, red, green, and yellow light system [1]. At the present time, a TLC deploys microprocessors that allow it to control the lights more effectively.

L. Parisi, D. Hamili and N. Azlan are with Cardiff University, School of Engineering, Division of Medical Engineering, Trevithick Building, 14-17 The Parade, Cardiff, South Glamorgan CF24 3AA (e-mail: ParisiL@cardiff.ac.uk).

However, the modern TLC is still based on basic digital systems named as gates. The traffic light is controlled by two detected voltage levels, respectively of 0 and 5 V.

It feeds in a digital system for electrical signals whereby the binary values of 0 and 1, as well as their combination, are assigned, representing the outputs of such digital system. The main aim of these two laboratory sessions was to gain an appreciation of a digital system's TLC design, how the logic gates work and how such controller could be designed and tested using the software Multisim. In such TLC programme, when a time value was below a particular temporal threshold named as T_0, all three lights remained switched off. If the temporal threshold was greater than T_0, hence the green light was on; green and amber Light-Emitting Diodes (LEDs) were both switched on, when the time value was greater than T_1, red and amber were turned on when the time value was higher than T_2, whilst all three lights were switched on, when the time threshold was greater than T_3. Eventually, all three lights remained switched on until the time value was smaller than T_0. The minimum number of only nine gates among AND, OR and NOT logic gates was used [2].

II. Design and Discussion

The TLC needed to operate iteratively such that the amber LED would be switched on for 2 seconds, the red LED for 14 seconds, the red-amber transition LED for 2 seconds, whilst the green LED for 14 seconds. Since the traffic light was to be operated at 0.5Hz, i.e. each tick corresponded to 2 seconds, therefore, for one complete cycle to be completed, 16 counts needed to be deployed through a 4-bit counter, following the given sequence. Input logic was created to convert primary inputs to grey coded secondary inputs avoiding function hazards. The state machine diagram and primitive flow tables were created initially with seven states that were subsequently decreased using 'merging' to four states as a Moore machine. Thus, each output was determined only by the current state and not directly by the input. Output logic was created to generate secondary outputs from the primary outputs, to simplify the task of working out the state machine excitation equations. The cycle was programmed in order for it to be repeated indefinitely. Firstly, the timing clock is attained further to designing an asynchronous 4-bit binary counter deploying J-K flip flop, whilst the second stage of the design involved the design of the digital logic using ASM. A counter is a device that can store memory.

The asynchronous 4-bit binary counter used throughout these two laboratory sessions is composed of four J-K flip flops as shown in Fig. 1, cascaded to generate four binary outputs, whereas the output Q of one flip flop is connected to the clock of the next flip flop [3], [4].

The four-bit binary outputs are represented by T_0, T_1, T_2 and T_3, where T_0 is the least significant bit (LSB) and T_3 is the most significant bit (MSB). As stated, this counter acted as the digital clock to time traffic light transition states.

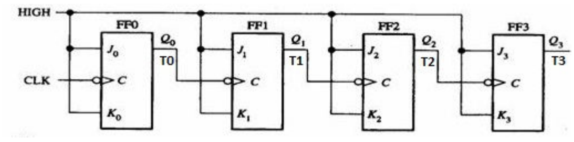

Fig. 1 4-bit binary counter using J-K flip flops

The primary inputs previously defined by T_3, T_2, T_1, T_0 can be described by secondary inputs Y_1, Y_0. A Karnaugh map was created to design the logic required to make this conversion that is show in the circuit diagram (Fig. 2).

III. Digital Logic Using ASM

The changes from state to state must be sequential and avoid multiple inputs changing simultaneously to prevent function hazards. A state diagram (Moore state diagram) is drawn to identify the primary inputs, the primary outputs, its current state display and to display flow of the transition of the light. Based on such state diagram, the primitive flow diagram can now be built. It was checked whether there were any race issues. The primitive flow table was subsequently 'merged' by using specific rules to simplify the state table and also avoid function hazards. As mentioned, with a Moore machine the outputs must be the same in order to merge states to create a Mealey machine. Whilst the outputs of a Moore machine are determined only by the current state and not directly on the input, the output of a Mealey Machine is computed by the current state and input. Output logic is created to convert grey coded secondary outputs, 'Y', from the outputs that correspond to the original problem.

IV. Excitation Tables for the State Machine and Circuit Design

In order to confirm the necessary minimum inputs required for the ASM, excitation tables needed to be calculated for Y_1Y_0, Y_1 and Y_0. These relate the secondary outputs, 'Y', in terms of the primary inputs 'T', and secondary inputs 'y'. The prime implicants of each table were overlapped, thus eliminating static logic hazards, which would occur when the output switches from one prime implicant to another. 'Don't cares' were also used to maximize the coverage of prime implicants and, therefore, decrease the complexity of the logic by reducing the excitation equations. The excitation tables are drawn taking the primitive flow and state diagram into account. This needs to be done in grey coding, where only one input changes at a time.

From the excitation table, excitation equations can be established for each output of Y_1 and Y_0 that would be fed into the input as secondary inputs. Using Karnaugh map methods and Boolean algebra the excitation tables for Y_1 and Y_0 were simplified into the following corresponding excitation equations:

$$Y_1 = T_2 \tag{1}$$

$$Y_0 = y_1'T_0 + y_0T_0' + y_0y_1' \tag{2}$$

$$y_1(T_0 + y_0) + y_0y_1' \tag{3}$$

Once Y_1 and Y_0 equations were calculated, the output table was plotted, leading to output equations for red, amber and green. Such simplification of Y_0 ensures that the number of gates deployed from three is reduced to two. The use of the asynchronous 4 bit binary counter in the circuit allowed to reduce the number of inverter logic gates down to one, as shown in Fig. 2. The T values come from the output of the 4-bit binary counter that was used to time the transitions between the light states. From these equations and the output equations, a circuit was designed using the Multisim software and simulated as shown in Fig. 2.

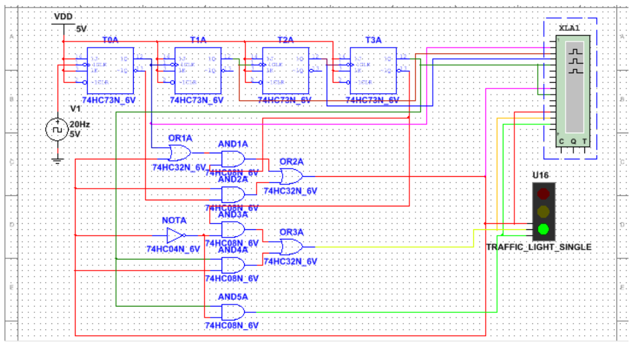

Fig. 2 4-bit binary counter using J-K flip flops

V. Simulation of the Circuit through Multisim

In this experiment, Moore state machine is applied, as the output of each state depends on its current state only and grey coding is applied to avoid any races [5]. An additional input, y_0y_1 is added to the equation of Y_0 to prevent hazards from occurring, since some of the prime applicants are adjacent to each other. Moreover, the equation could be simplified, as can be seen in (2) and (3), to reduce logic that can later decrease the number of gates used, thus lessening the cost of production. Simulation is done prior to testing the actual circuit to ensure the circuit would output the correct sequence of light.

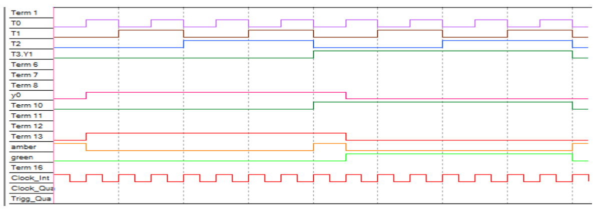

Fig. 3 Logic analyzer of circuit using Multisim, where 'term 13' represents the 'red'-related waveform

As expected, the logic circuit was successfully built, thus generating the required sequence of lights (Fig. 3). By manipulating the Clocks/Div and clock rate values, the signals shown below were correctly displayed through the logic analyzer.

With the initial excitation and output equations, it was found that the idea of hazards in Karnaugh maps was omitted, thus possibly leading to incomplete and inaccurate results due to incorrect equations. When undertaking logical network simplifications, the biggest map simplification would be ideal relatively to the excitation Y_1 and its corresponding equation but not Y_0, leading to hazards. A hazard is defined to be a circuit that may produce a glitch; this occurs when an output is 1 before and after the transition. In order to reduce the risk of hazards and glitches, redundant prime implicants were added to the relevant Karnaugh maps to ensure that all single-bit input changes were covered (i.e. an extra loop in the truth table). This modification resulted in a less simple but more stable system, as shown in the equations for Y_1 and Y_0 [6]. It was, then, possible to create a circuit that used only 9 gates (Fig. 2). Considering that the traffic light system has a total period of 32 seconds, if the clock-rate frequency was increased, the period of the system would be reduced. The output sequence would be highly affected with some lights turning on sporadically. It would be more preferable to add a DC coupling capacitor at each gate to eliminate the ripple. If a similar system were required, where the total period was of the order of microseconds, whose corresponding clock rate was set to 10 MHz, it would be so fast that it would cause promulgation delays on the output. Such hazards can lead to incorrect circuit function, for instance, not outputting the amber light at clock rate frequency of 10 MHz (Fig. 4) via the Multisim circuit simulator, rather than taking 32 seconds to make a complete cycle as it was expected to be.

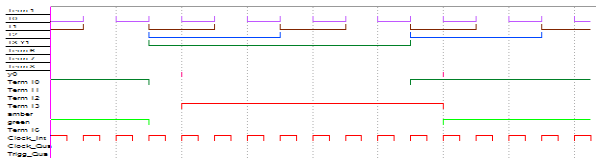

Fig. 4 Logic analyzer of circuit operated at 10MHz, where 'term 13' represents the 'red'-related signal

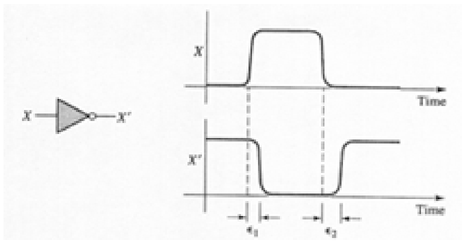

Fig. 5 The effect on propagation delay [8]

There was a delay from the input to the correct output as seen in Fig. 4. This is called propagation delay (Fig. 5) and it is defined by the length of time between the time an input to the circuit changes and the time in which such change propagates through the circuit, thus altering the output [7]. As the circuit deployed multiple inputs, the propagation delay was equal to the maximum delay of the circuit; this was due to the input travelling through different paths until it reached the output. Hence, the more complicated the circuit, the bigger the delay.

If any design constraints were provided, the circuit should have been redesigned using three-input logic gates. This would have reduced the number of logic gates required within the circuit, e.g. by using a complex programmable logic device (CPLD) to set up the sequence. CPLDs are deployed to create circuits with hundreds of logic gates and flip flops. They have predictable timing characteristics, rendering them suitable for critical control applications such as TLC [9]. If the timing were to go wrong, it could potentially endanger lives as lights would turn on and off intermittently, disrupting the sequence. The preferred method of production for the miniature traffic light would be to use a CPLD, as they are reliable and have fairly low power consumption. However, due to the small number of logic gates required for this circuit, using a two-input logic gate method would be cheaper. The three-input logic gate method would not be used as the propagation delay would be increased, resulting in higher error rates on the traffic light sequence [10]. If a traffic light control design was to be implemented, the preferred method would be using two-input gates in order to reduce errors. If a three-input gate system were deployed, the likelihood of errors would increase and this would cause a higher number of accidents.

The clock signal in this exercise was generated using a signal generator provided in the laboratory. However, if a traffic light were to be implemented without using the lab equipment, a 555 timer IC could be used as clock signal generator (Fig. 6). For this TLC, the 555 timer IC would be operated in a stable mode and it would generate a continuous series of pulses at a specified frequency [11].

In order to generate a clock signal operated at the frequency of 0.5Hz, the values of resistance R_1 and R_2 would need to be determined beforehand.

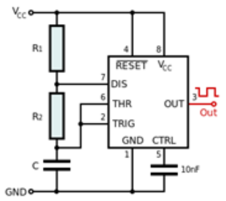

Fig. 6 555 Timer IC in a stable mode

If the value of R_1 is chosen to be 1 MΩ and R_2 is 2 MΩ, then the calculated value for C is 57.6 μF. These predetermined values of components would enable the 555 timer to generate pulses at the frequency of 0.5 Hz. Moreover, microprocessor, FGPA (Field-Programmable Gate Array) and ASIC (Application-specific integrated circuit) could be deployed to provide higher integration, performance, speed with less individual devices. This is due to the propagation delay of the logic gates being too slow for the input response. This causes the gates to have unwanted transitions as shown in Fig. 4 and is undesirable for controlling a traffic light system [12]. Fig. 3 shows the circuit responding to the input signal, as designed in Fig. 2. The primary outputs of the circuit appear to respond instantaneously to the change in primary inputs and output the desired values. However, this occurs when the inputs are changing at a rate of 120Hz. If the inputs were to change at a faster rate, for instance, at 10 MHz (Fig. 4), thus with a circuit having a period of the order of a microsecond, the gate delay would be evident. When the inputs arise from the starting position of all low, as shown in Fig. 4, a gate delay appears to be between T_0 going high and Y_0 going high. Looking at the circuit in Fig. 2, it can be seen that the number of gates going to each output is relative to the delays. The effect of such delays is shown more clearly in Fig. 4. The outputs are not stable in relation to the inputs as expected, whereas they oscillate with increasing magnitude, thus indicating that the gate delays in the feedback are too high, implying that a finite propagation delay of the inverter occurred. The simulation could have been extended to include the case of reducing the gate delays from those default in Multisim to zero, hence eliminating the oscillations; however, this would not represent a real circuit. A more realistic solution would be adding extra delays where required within the circuit to ensure that there are not any races between primary and secondary inputs.

VI. Conclusion

The aim of this laboratory exercise was fulfilled. A TLC was successfully designed and operated under the required conditions. Furthermore, through this exercise a better understanding was gained on how to deal with the process of designing counter and logic circuit.

When the inputs were changing relatively slowly, e.g. at a rate 120 Hz, the primary outputs of the circuit appeared to respond instantaneously to the change and in the predicted manner. When the inputs changed at a faster clock rate, e.g. of 10 MHz, a delay was generated, due to the relative number of gates feeding into each output, causing the outputs to oscillate with increasing magnitude. This indicates that the gate delays in the feedback were too high. The logic circuit simulated using the software Multisim matched all design requirements, thus displaying the traffic light sequence in the correct order and with the correct timing. Using the simulate trace function within Multisim, it was successfully shown that no hazards were present and that, after the sequence finished, the cycle was repeated.

Acknowledgment

The authors would like to thank Dr S Armstrong and Dr G Min, Senior Lecturers in Electronics Engineering for giving them the opportunity to carry out the present study at Cardiff University and, therefore, get the chance to develop the present work.

The first author Luca Parisi would like to thank SWIEET2007 and, in particular, Prof Denys Vernon Morgan for their contribution in making his participation at the conference possible, held in Paris, France, on January, 23rd and 24th, 2015.

References

[1] Mary Bellis, History of Traffic Lights. Available at: http://inventors.about.com/History-Of-Roads_3.htm..
[2] Ricardo O Rabinovich, (1994). Transition maps guide successful asynchronous state-machine design. Electronics Design Strategy News. Available at: http://www.edn.com/archives/1994/051294/10df2.htm. Last accessed on: December, 1st, 2014.
[3] Hayes, J. (1994). Digital Logic Design. Addison –Wesley. p. 443-446.
[4] Asynchronous Counters. Available at: http://www.allaboutcircuits.com/vol_4/chpt_11/2.html.
[5] Sequential Information. http://inst.eecs.berkeley.edu/~cs150/fa05/Lectures/07-SeqLogicIIIx2.pdf.
[6] Interface BUS, (2012). Glue Logic Timing Hazards. Available at: http://www.interfacebus.com/Design_Logic_Timing_Hazards.html. Last accessed on: December, 2nd, 2014.
[7] Strom, E. G., Parkvall, S., Miller, S. & Otterson, B. (1996). Propagation Delay Estimation in Asynchronous Direct-sequence code-division multiple access systems.. IEEE Transactions on Communicaitions., 44 (1), p. 84-93.
[8] ECE 152A - Digital Design Principles. (2012). Propagation Delay, Circuit Timing and Adder Design. Available at: http://www.ece.ucsb.edu/Faculty/Johnson/ECE152A/L4/%20Propagation%20Delay%20Circuit%20Timing%20Adder%20Design.pdf. Last accessed on: December, 2nd, 2014.
[9] Global Spec, (2007). Complex Programmable Logic Devices (CPLD) Information. Available at: http://www.globalspec.com/learnmore/analog_digital_ics/programmable_logic/complex_programmable_logic_devices_cpld. Last accessed on: November, 28th, 2014.
[10] Hewes, J. (2011). Multiple Input Gates. Available at: http://www.allaboutcircuits.com/vol_4/chpt_3/4.html. Last accessed on: December, 2nd, 2014.
[11] 555 timer IC. Available at: http://www.doctronics.co.uk/555.htm. Last accessed: December, 1st, 2014.
[12] Tinder, R. (2000). Chapter 9 - Propagation Delay and Timing Defects in Combinational Logic, Engineering Digital Design, Academic Press San Diego, p. 391-418.