CS25410 Computer Architecture and Hardware Assignment

Development of Digital Circuits

Richard Addicott

ria4@aber.ac.uk

Introduction

The task at hand for this assignment was to implement a digital circuit that includes the capabilities of counting forwards and backwards between 0 to 9, where the current value is displayed on a 7-segment display. The digital circuit should also contain the function to reset the circuit and display to 0, and also have the ability to specify the direction of the counting. The counting of the digital circuit should be triggered by a clock, allowing the option for the tick frequency to be changed. As of starting this assignment, my knowledge of implementing logic gates into digital circuits was fairly limited and my intention is to have vastly improved my understanding by the end of the project.

Overview of Design

The design of choice for me on this assignment, was majorly on the use of the AND and OR logic gates. The use of these particular logic gates allowed me to produce correct output throughout the digital circuit and eventually combining the logic to produce an output on the 7-segment display. Through the usage of the AND logic gates, it allows me assign a 'number' to each gate and the output would carry through and this would enable me to manipulate the power in the circuit to the desired output.

The choice of flip-flop will be described in full later on in the document, however for this assignment there were 3 main choices: JK, D and SR flip-flops. The planning phase of my circuit design was fairly confusing with the new concepts of circuit designs and complex logic gates. The planning largely consisted of some detailed research of the possibilities that could be used to implement a solution, as part of my planning – I created a simple digital circuit using JK flip-flops to output a loop to turn a series of LEDs on and off. This helped to form a basis for the choice of flip-flop and design of my actual digital circuit. The digital circuit uses a clock to trigger changing of the flip-flops to in turn produce different outputs and eventually change the 7-segment display. The reset feature of the circuit, is implemented using a simple AND gate that takes four inputs from the circuitry and then acts as one of two inputs in an OR gate. This along with an 'r' input which is needed to be triggered by the user is then fed into the OR gate which connects to the left south pin of the JK flip-flops to produce a reset. This action changes the output of all of the JK flip-flops to 1 and in doing so, turns every segment of the 7-segment display active and produces the '0' output. The overall output of my digital circuit is that when the clock is triggered (via Simulate -> Ticks Enabled) – it iterates through the various complex circuitry to change the output on the 7-segment display to change it from 0, 1, 2, ….., 9 and then it will continue back to 0 until further user input (Reset input).

As part of my design planning process, I undertook research into the 7-segment display and how the binary inputs of my circuit design will affect the display outputs and produced the below table (Figure 1) to document the changes.

Binary Input				Decoder Output							7-Segment Display Output
1	**2**	**4**	**8**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	
1	1	1	1	1	1	1	1	1	1	0	**0**
0	1	1	1	0	1	1	0	0	0	0	**1**
1	0	1	1	1	1	0	1	1	0	1	**2**
0	0	1	1	1	1	1	1	0	0	1	**3**
1	1	0	1	0	1	1	0	0	1	1	**4**
0	1	0	1	1	0	1	1	0	1	1	**5**
1	0	0	1	1	0	1	1	1	1	1	**6**
0	0	0	1	1	1	1	0	0	0	0	**7**
1	1	1	0	1	1	1	1	1	1	1	**8**
0	0	0	0	1	1	1	1	0	1	1	**9**

Fig 1. 7-Segment Display Decoder

The binary inputs are from the JK flip-flops in the circuit and each different combination produces a number of changes in the digital circuit. In the circuit, each differing input to the 7-segment display will activate/deactivate a particular segment and in turn – produce the display output which is shown to the right. The inputs are reversed in this table as to what they would presume to be in a standard binary representation, however this is how the digital circuit works – if I were to spend more time on this project, I would work on changing this. For ease of use, I have no produced an input to the DP segment of the display for this particular project.

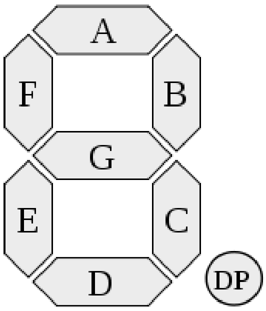

Fig 2. 7-Segment Display [1]

Choice of Flip-Flops

The choice of flip-flop was an issue of much deliberation for me during the implementation of my digital circuit. The decision was took after extensive research[2] into the subject to familiarise myself with all of the possible information about the different choices. The JK flip-flop[3] was the eventual choice of mine, as I decided that it was the most versatile of choices considering the lack of knowledge on my part and complexity of the problem at hand. This type of flip-flop is also very useful for counters and has the ability to easily create an oscillating signal, and therefore extremely helpful for the solution of the assignment brief as it will provide me with the feature to count from 0 to 9 and vice versa. The planning behind my usage of the JK flip-flops was to use four (A, B, C, D) to represent a 4-bit binary output. The eventual result of the circuitry would produce a demonstration of binary addition, the intention of my flip-flop planning is to produce an output similar to the following basic diagram.

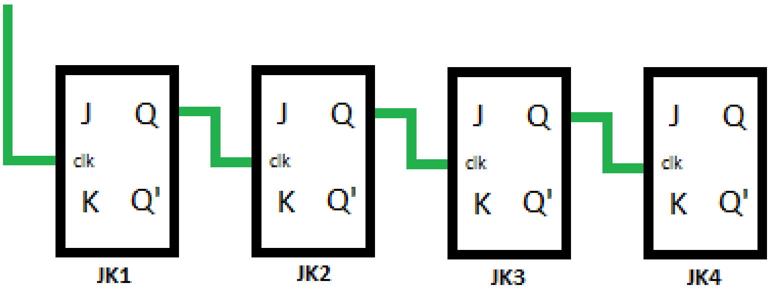

Fig 3. JK Flip-Flop Plan

The implementation of this JK flip-flop formation would mean that if correctly applied to the digital circuit, then the basis of the solution to the assignment brief would be created. The four JK flip-flops should represent binary numbers, so to speak (JK1 = 1, JK2 = 2, JK3 = 4, JK4 = 8) – this will enable me to implement a circuit to act as a sort of addition system. This potentially is not what the aim is – however, this is the best way that I can begin to understand a solution to the problem. This flip-flop design plan will also enable me to produce the digital output on the 7-segment display[4] using an input of a constant variable (1) and then I can produce circuitry from the JK flip-flop outputs.

In my personal opinion, there were not particularly any other comparable choices that would work as efficiently as the JK flip-flops. I took a significant time in attempting to implement a SR flip-flop (Set/Reset) in order to act as the reset input for the circuit, however after much experimentation, this was not a feasible solution as the SR flip-flop output would not trigger the correct circuit changes in order to produce the desired output. On paper, the SR flip-flop seemed like a competent choice as they can be driven by a clock, and the mechanism of the schematic leads to thinking that the logic of the flip-flop would fit the solution. However, in implementation this was not possible due to the complexity of my current circuit which made the addition of the new flip-flop very problematic. As use as part of the representation of the 4-part binary output, none of the other flip-flops were as convenient as the JK flip-flops. My particular way of implementing the JK flip-flops requires an input of a constant variable which is input into the J and K parts of the flip-flop and this would not be possible in the SR flip-flop due to the set and reset inputs. The only other viable option for the choice of flip-flop was the D flip-flop which would allow the constant variable to the data input and also can include a clock input for the counter.

Implementation

The general implementation of the task was difficult due to the complexity and concept of digital circuits and all of the components related to them. In addition to this, I had never used the program Logisim previously and therefore I was faced with a number of issues getting to grips with the features of the program.

There was a degree of difficulty for the implementation of this circuit, this was in part to the scale of the circuit. As previously stated, there was problems in producing a plan for the design of the circuit. This made the implementation more time consuming than expected, whilst having a basic idea of the outcome that I would have liked to produce – the implementation had some parts of trial and error in order to produce a correct solution. The debugging of my problems was made harder by the fact of my inexperience with the Logisim program, however I was able to make use of the feature of Logisim that allowed me to show route of specific currents which enabled me to focus my debugging easier. In amongst the plethora of circuit, this allowed me to see where my errors were and eventually I was able to produce a solution to the problem. There were also a few issues with parts of the 7-segment display, represented as segments (a -> g) and I used Figure 1 as shown earlier in the Design Overview to help debug the segment problems. At points of the implementation, I was missing one part of the segment display and I was able to alter the binary input as shown in the table and then change the circuitry to change the decoder output. This problem was fairly simple to solve with some trial and error of moving the circuitry around to produce the relevant 7-segment display output. I am disappointed not to be able to implement a solution to enable the count backwards feature and the inclusion of a direction input. However, due to the lack of time and partly knowledge on my part I was unable to produce the solution correctly. Despite, the exclusion of this feature of the project I am still impressed with my progress, gained knowledge and final output of the digital circuit.

Conclusion

In review of my assignment, I am fairly pleased with the outcome that has been produced. During the planning of the circuit, I was skeptical of whether I would be able to produce a functioning circuit and 7-segment display. I am only to presume that there would be more efficient way of solving the problem at hand, and this would be something that I would strive for in future. In addition to this, I would also prefer to compact the circuit – I have tried to align the logic gates to improve the aesthetics of my digital circuit as this helps me to understand the logic on show. The concept of circuit design on a computer program was also relatively new to me, and therefore this presented a minor challenge for me to learn the components of the project and the ways in which they can be manipulated. I have no doubt that if presented with another task of similar difficulty, the solution could be implemented with more efficiency.

There is of course the possibility for different decisions of the types of logic gates and flip flops that could be used to improve the efficiency and overall design of the circuit. However, due to this being my first implementation of a digital circuit in Logisim – I am pleased at the final output taking into account the difficulties of implementing the assignment brief. This project has also been extremely helpful in substantially improving my knowledge of this module of the different concepts involved in the planning and implementation of digital circuits.

References

[1] 7-Segment Display Output Image - *http://fpga-tutorials.blogspot.co.uk/2011/11/7-segment-displays.html*

[2] Flip-Flop Information - *http://www.ee.surrey.ac.uk/Projects/Labview/Sequential/Course/06-FlipFlops/*

[3] JK Flip-Flop Information - *http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/jkflipflop.html*

[4] 7-Segment Display Output Information - *http://www.thelearningpit.com/lp/doc/7seg/7seg.html*