University of Nottingham
UK | CHINA | MALAYSIA

LTspice Simulation of Sequential Circuits

EEEE2044 – Digital Coursework

The aim of this lab is to look at circuits to detect patterns of bits and to study the effect of gate and flip-flop delays on this circuit. We will be using LTspice to analyse these circuits.

Sequential circuits have already been covered in the material for EEEE1004. During EEEE2044, this has been extended to cover counters with external inputs and circuits to detect specified sequences in a stream of bits. Although an analysis of the theoretical behaviour of such circuits has been looked at, the behaviour of the circuit in practice also needs to be analysed. Propagation delays occur in the flip-flops and in the combinational circuits built round them and these will determine the maximum clock frequency at which these circuits will operate.

Prerequisites

It is assumed that you have read and understood up to section 3.4 of the Digital Electronics notes and that you have completed the Sequential Circuits part of the Digital Electronics Example Sheet.

Assignment

At various points during this coursework you will be asked questions and the number of marks allocated will be indicated. Your lab submission should be word processed, however if preferred some diagrams can be hand drawn, with a scanned image/photo of the drawing included in the document – please ensure that this is legible. Certain graphs that are required to be added to your report are specified in this document. More detail of the report structure will be given on the final page of this document. The report should be submitted to Moodle before the deadline of 3pm 14 November 2022 (UK time).

This coursework is an individual piece of work. Please remember your Academic Misconduct training. If you are unsure about what constitutes Academic Misconduct, please ask!

University of Nottingham
UK | CHINA | MALAYSIA

Circuit to Recognise the Sequence 101 in a Stream of Bits

Definition of Operation of Circuit

The circuit to be designed should detect the presence of the sequence “101” in the last three bits of the input signal and to output a ‘1’ whenever this occurs.

Hence, if the input (X) is:

0 1 0 1 1 0 1 0

then the output (Z) is:

0 0 0 1 0 0 1 0

In this part of the coursework we shall look at the *Moore* design of this circuit.

States, State Diagram and State Table

For a Moore implementation, there should be *four* states as follows:

State	Description of state
S_0	Initial State/Last two Bits are ‘00’.
S_1	Detect ‘1’ in most recent bit.
S_2	Detect “10” in most recent two bits.
S_3	Detect “101” in most recent three bits.

Question 1:
Using the above states draw a State Diagram to detect the sequence “101”.
[4 marks]

Allocation of States and Logic Relations for Circuit

As there are 4 states, the circuit requires 2 flip-flops. Allocate the flip-flop outputs as follows:

$S_0 = 00$; $S_1 = 01$; $S_2 = 10$; $S_3 = 11$

We are going to use D-type positive edge triggered Flip-Flops.

Question 2:
Construct a set of appropriate state tables, both in terms of S_0, S_1 etc and also the flip-flop inputs/outputs.
[4 marks]

Question 3:
Show that the above allocation of states leads to the following logic relations for the Flip-Flop inputs and for the output from the circuit:

$$D_A = Q_B\bar{X} + Q_A\overline{Q_B}X$$
$$D_B = X$$
$$Z = Q_AQ_B$$

where X is the input, Z is the output, D_A (D_B) is the input to Flip-Flop A (B) and Q_A (Q_B) is the output from Flip-Flop A (B)

[6 marks]

Design and Simulation of the Circuit using LTspice

The circuit with the above logic relations will now be simulated using LTspice.

We will be using the digital components built into LTspice for this coursework. The components can be found in the 'Digital' folder when using the 'Select Component Symbol' window (shortcut to Window is F2). For the flip-flops, we will be using the one named *dflop*, as shown in Figure 1.

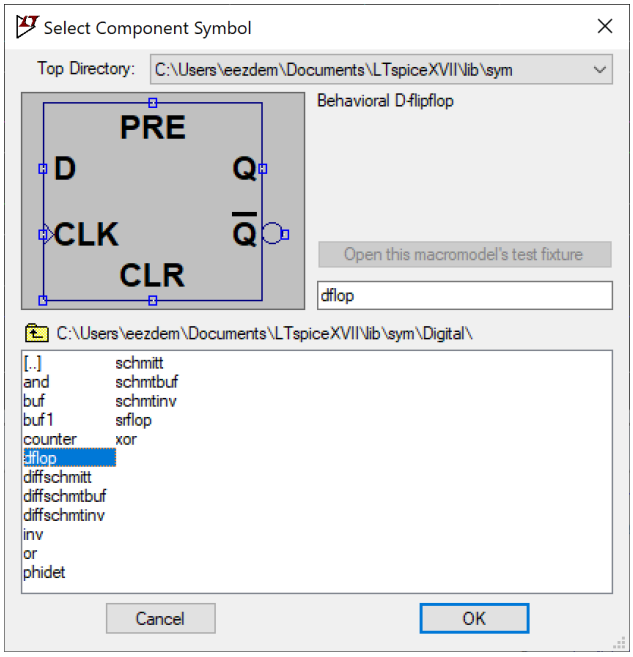

Figure 1: Select Component Symbol

For the purposes of this work you can ignore the PRE and CLR pins – leave them unconnected.

By default the digital elements in LTspice have the logic low at 0 V and the logic high at 1 V, this is what we will be using for this coursework.

The AND and OR gates have 5 input pins in LTspice, see Figure 2 for an example of the OR gate. Any pins that are not connected are ignored by LTspice, therefore to simulate a 2 input OR gate, you only need to connect to 2 input pins. You will also notice that the device has two outputs. The output denoted with the circle is inverted, therefore the device can be used as a NOR and OR gate at the same time.

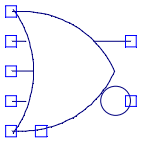

Figure 2: OR Symbol in LTspice

To build the circuit you will need (the names of the circuit elements in LTspice are given in brackets):

- 2 x D-type flip-flops (*dflop*)
- 2 x voltage supplies (*voltage*)
- 3 x AND gates (*and*)
- 1 x NOT gate (*inv*)
- 1 x OR gate (*or*)

In LTspice, the logic gates do not need powering. The voltage supplies are used for providing the clock to the CLK pins of the flop-flops and to provide the stream of bits for checking (X).

The circuit to implement the sequence detector is shown in Figure 3.

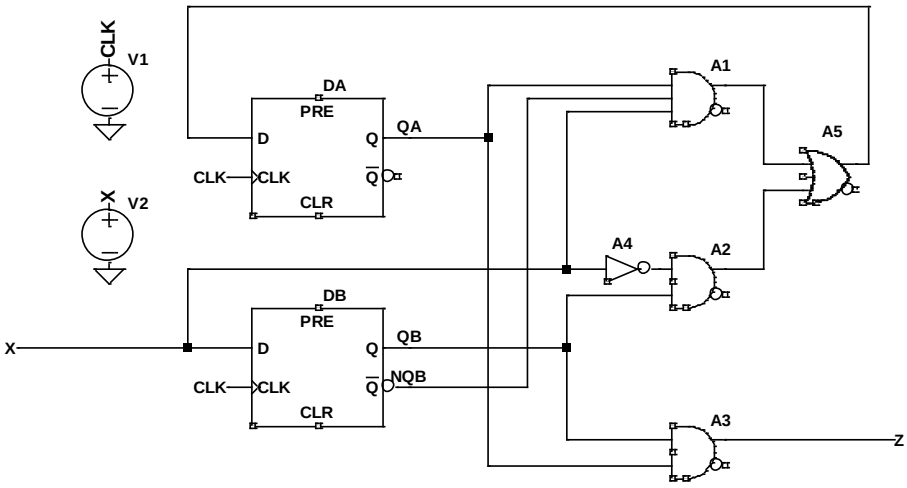

Figure 3: Circuit to implement detector of '101' sequence in input stream

Question 4:
Implement the circuit shown in Figure 3 in LTspice and export the image as an .emf file and include this in your document.

[2 marks]

Clock and Inputs

The starting clock input to the circuit should be a 1000 Hz square wave.

Right click on the voltage supply that you have connected to the CLK net, choose PULSE mode, and set:

Vinitial(V)	=	0
Von(V)	=	1
Tdelay(s)	=	0.5m
Trise(s)	=	1n
Tfall(s)	=	1n
Ton(s)	=	0.5m
Tperiod(s)	=	1m
Ncycles	=	1000

The input stream (X) will be defined by a piecewise linear (PWL) function. Here the input signal level at certain times is stated and LTspice will draw straight lines between the points to create the intermediate values.

The test signal will be the same as that stated at the top of the document, that is:

0 1 0 1 1 0 1 0

The length of each bit will be approximately (because of rise and fall times) equal to the clock period (that is 1 ms). The points for the test signal are shown in Table 1.

Table 1: PWL function for test signal

Time (s)	Voltage (V)
0m	0
0.999999m	0
1m	1
1.999999m	1
2m	0
2.999999m	0
3m	1
4.999999m	1
5m	0
5.999999m	0
6m	1
6.999999m	1
7m	0

LTspice can load the points for a PWL function from a simple text file. To create the file, each line of the file should look like each row of the Table 1. The time and voltage values can be separated with a comma (,) or a tab. The test signal above can be downloaded from the EEEE2044 Moodle page.

To attach a PWL file to a voltage source, right click on it, and use the “PWM FILE” option, see Figure 4. The PWL file for use in this simulation is provided on Moodle (101_testpwl.txt).

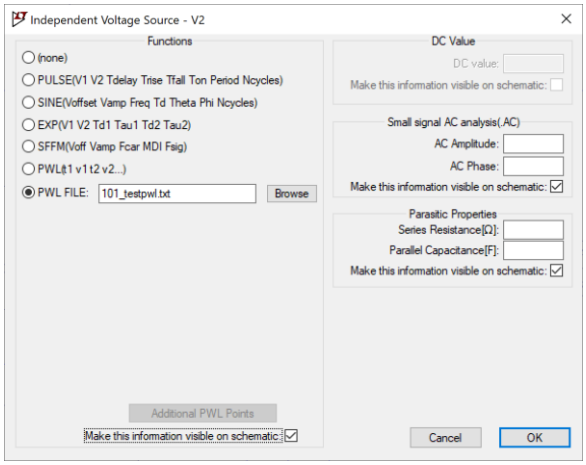

Figure 4: Attaching a PWL file to a voltage source

Initial Simulation of Circuit

We now wish to check that the circuit is working as it should do.

We shall be running a *Transient* simulation over a period of 8 ms (the length of the test signal). To set this up, open the '*Edit Simulation Command*' window (Simulate → Edit Simulation Command) and enter the '*Stop time*' as '*8m*' and click '*OK*'.

Run the simulation, by clicking on the running person icon or Simulate → Run.

Select the CLK, X and Z signals for plotting. Plot them on different panes to make it clearer (right click on the graph to add a pane using "Add Pane Plot", you can drag traces between panes, by dragging their names).

The simulation results should look similar to those shown in Figure 5.

You should immediately notice that there are no pulses on the output signal (Z). The circuit appears not to work. This is because by default LTspice simulates all components as ideal – there are currently no transition delays. This means that the outputs of the flip-flop change as soon as the inputs do which means the feedback signal put into the input of DA does not work as expected.

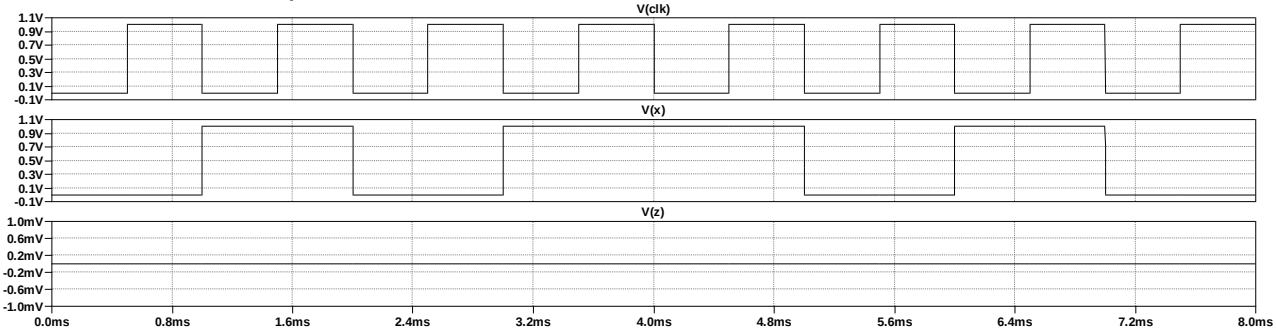

Figure 5: Initial simulation result

Transition delays can be added to the flip-flops, AND and OR gates by right clicking on the components and adding a Td= statement to the "SpiceLine", as shown in Figure 6.

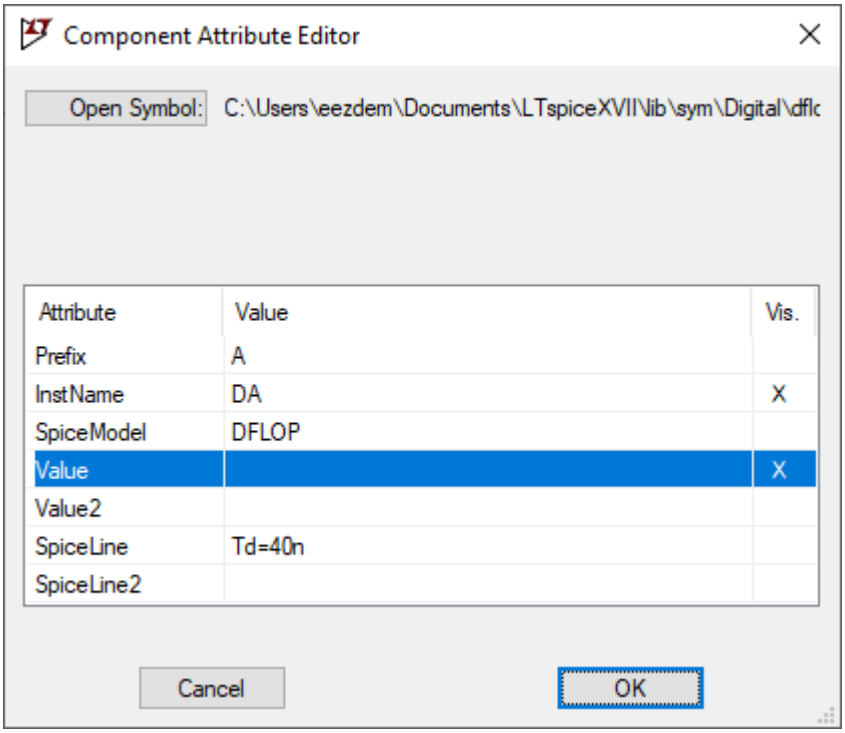

Figure 6: Component Attribute Editor

The value of Td for the different components are shown in Table 2.

Table 2: Td values for different components

Component	Td
dflop	40n
and	27n
or	22n
inv	22n

After updating the values of Td and simulating again, the results should look like those shown in Figure 7.

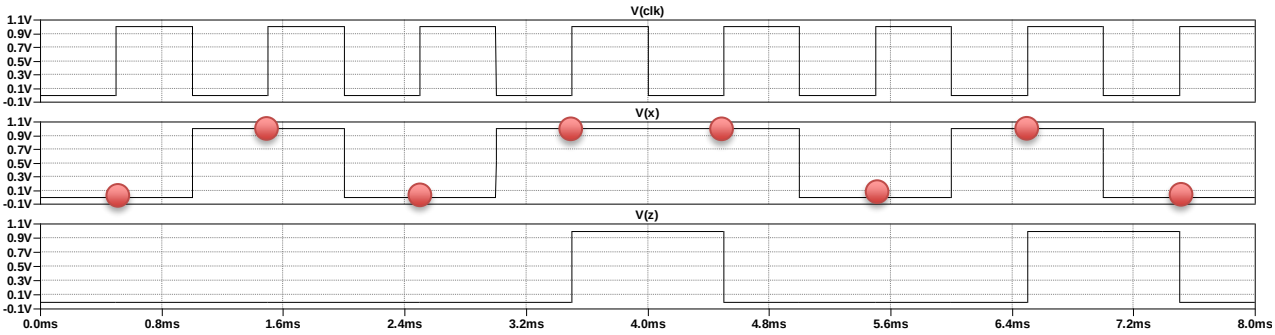

Figure 7: Simulation result after adding Td values to components

The flip-flops are clocked on the rising edge, the locations where the input signal has been sampled are marked with red circles in Figure 7. Note that the output for a Moore design does not change as soon as the signal is '101', it only changes on the next clock cycle after the final '1' has been received.

Question 5:
Using panes for each trace, as in Figure 7, generate a graph (export as .emf file) and import this into your document.

Ensure your output is correct.

[2 marks]

Estimating the Minimum Clock Period

So far in this module, we have not considered the effect of delays in a sequential circuit on the output signal. We will now do this and calculate the minimum period, t_{min}, between the rising edges of the clocks for the circuit to work correctly. The corresponding maximum clock frequency is given by:

$$f_{max} = \frac{1}{t_{min}}$$

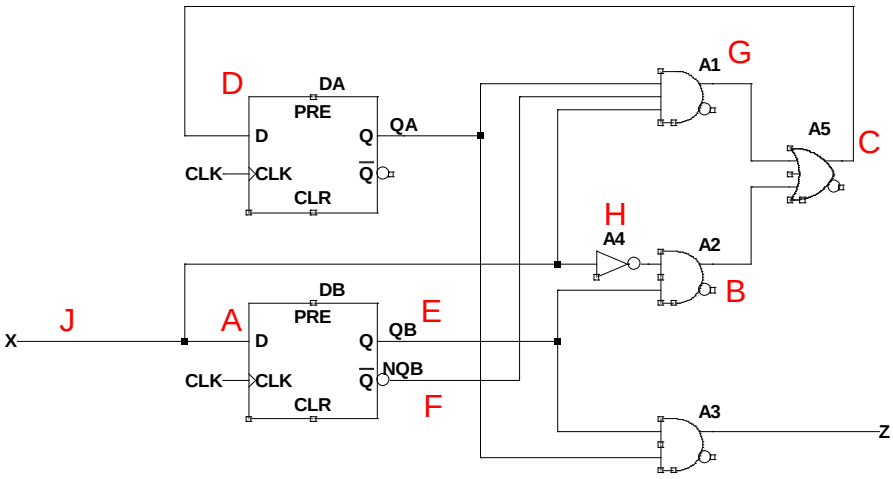

Figure 8: Circuit with annotated components for determining delays

For positive edge triggered flip-flops, delays are normally calculated by starting off on a rising clock edge. There are two types of path leading into the input (D) to a flip-flop in the circuit shown in Figure 8. One path originates from a logic circuit connected to the outputs of the flip-flops, the other path is from the external input X.

For example consider the delays occurring in the path AEBCD shown above which is the path travelled from the input to the lower Flip-Flop (DB) to the input to the upper Flip-Flop (DA). The delays occurring DURING A CLOCK PERIOD are as follows:

On the rising clock edge, the signal at A travels through the Flip-Flop to E. The time it takes to go through to the output (E) from the flip-flop input A is the *propagation delay*, t_{pd} (our value of Td from the simulation).

The signal then travels from E through the two input AND gate (A2) to B. This adds a further delay, t_{A2} through the gate.

The signal then travels to C through the two input OR gate (A5). There is a further delay t_{A5} through this gate.

The signal then travels to the input of the Flip-Flop DA. Now the signal needs to be stable at the input of this flip-flop for a time equal to at least the Flip-Flop's *set-up time* t_{su} for the signal to be correctly recognised as a 1 or a 0. LTspice does not simulate t_{su} and therefore we shall ignore this for this piece of work.

The sum of all these delays including the flip-flop set-up time is equal to

$t_{AEBCD} = t_{pd} + t_{A2} + t_{A5}$

Looking at Figure 8, there is another path for the input at A to travel, going from A through the flip-flop to the $\overline{Q}$ output to F then through the 3-input AND gate (A1) to G then through the two-input OR gate (A5) to C then to the input of the upper flip flop (DA).

Yet another path for which the delay must be computed is the signal going from point D in Figure 8, through the upper Flip-Flop (DA), via G and C back to D again.

Question 6:
Denoting the delay through the 3-input AND gate (A1) by t_{A1} derive an expression for the delay, t_{AFGCD}, in the loop AFGCD and the delay t_{DGCD} in loop DGCD.
[4 marks]

These are the only three paths where the signal travels through a flip flop and is then fed back into the same or another flip flop.

Now if the input X changes a sufficient time before the next clock edge, then the minimum time between the rising edges of the clock is given by the maximum of t_{AEBCD}, t_{AFGCD} and t_{DGCD}.

What do we mean by "if the input X changes a sufficient time before the next clock edge"?

Look at Figure 8 again. There are two paths JHBCD and JGCD, leading to the input to the upper flip-flop (DA) from the external input, X. The delay along JHGCD is given by

$t_{JHBCD} = t_{A4} + t_{A2} + t_{A5}$

where t_{A4} is the delay through the inverter (NOT gate). Note that this path bypasses the lower flip-flop (DB).

Question 7:
Derive an expression for the delay along the path JGCD.
[2 marks]

Now there is an additional requirement that if X changes from 0 to 1 or from 1 to 0 during a clock cycle it should occur at least a time t_X before the next rising edge of the clock, where t_X is the maximum of t_{JHBCD} and t_{JGCD}. If this transition occurs too late in the clock cycle, then, because of the propagation delay, the value of X at the input to the upper flip-flop may be wrong at the next clock edge leading to the wrong working of the circuit.

Question 8:
Using the data given in Table 2, determine the propagation delays t_{AEBCD}, t_{AFGCD} and t_{DGCD}. Hence, determine t_{min} which is the maximum of these three values.
[6 marks]

Question 9:
Also determine the individual delays in going from X to the input to the upper flip-flop: t_{JHBCD} and t_{JGCD}.
[4 marks]

University of Nottingham
UK | CHINA | MALAYSIA

Simulations

Question 10:
Change the clock period to 200 ns. You can do this by setting Ton = 100n, Tdelay = 100n and Tperiod[s] = 200n for the voltage supply acting as the clock input (right click on it to change the settings).

You also need to change the period of the test file or at least how LTspice interprets the file. This can be achieved by right clicking on the text that should say “PWL file=101_testpwl.txt” on the schematic next to the voltage source and adding “TIME_SCALE_FACTOR=0.0002” **between** “PWL” and “file=101_testpwl.txt”, so the line reads “PWL TIME_SCALE_FACTOR=0.0002 file=101_testpwl.txt”. This multiplies every time value in the PWL file by 0.0002, so for example, turns a 1 ms period into a 200 ns one.

Change the “Stop time” of the simulation to “1.6u” (8 ms multiplied by 0.0002) and perform the simulation.

Using panes for each trace (V(CLK), V(X) and V(Z)), as in Figure 7, generate a graph (export as .emf file) and import this into your document.

Verify that the output (Z) is correct.

[4 marks]

Question 11:
Change the clock period to 120 ns and set an appropriate value of TIME_SCALE_FACTOR to set the duration of each input bit to 120 ns.

Select a suitable “Stop time” for the simulation.

Using panes for each trace (V(CLK), V(X) and V(Z)), as in Figure 7, generate a graph (export as .emf file) and import this into your document.

The output is incorrect, despite 120 ns being greater than t_{min}. Explain why this is.

[6 marks]

Question 12:
Increase the clock delay (Tdelay[s]) by 10 ns, but keep the period the same.

Using panes for each trace (V(CLK), V(X) and V(Z)), as in Figure 7, generate a graph (export as .emf file) and import this into your document.

Is the output (Z) now correct? If so, why has shifting of the clock solved the problem?

[6 marks]

Recognising a long sequence in a Stream of Bits

Definition of Operation of Circuit

The circuit to be designed should detect the presence of a 5 bit sequence from a binary sequence that is unique to **your** Student ID number in the last five bits of the input signal received and to output a '1' whenever this occurs.

While you can test your circuit with any sequence of numbers, you will submit work based on a test stream (which is guaranteed to contain your sequence at least once).

A file that links your student ID to the sequence and test stream **you** should be using is provided on Moodle.

In this part of the coursework we shall look at the ***Mealy*** design of this circuit.

University of Nottingham
UK | CHINA | MALAYSIA

States, State Diagram and State Table

Question 13:
For a Mealy implementation of the state machine, determine the required number of states and what each of these states represents. You should present the answer in a table as follows:

State	Description of state
S_0	Description of state S_0
S_1	Description of state S_1
⋮	⋮
S_n	Description of state S_n

[6 marks]

Question 14:
Write down your sequence (5 bits) and test stream (25 bits). Using the states you have derived in Question 13, draw a State Diagram to detect your sequence.
[10 marks]

Allocation of States and Logic Relations for Circuit

Question 15:
Determine the number of flip-flops required to implement the state machine and allocate the flip-flop outputs to the states.
[2 marks]

Question 16:
Construct a set of appropriate state tables, both in terms of S_0, S_1 etc and also the flip-flop inputs/outputs.
[5 marks]

For this circuit, as before, we will be using D-type positive edge triggered flip-flops.

Question 17:
From your allocation of state tables, determine the minimised logic relations for the Flip-Flop inputs and for the output (Z) of the circuit. Show all your working.
[10 marks]

University of Nottingham
UK | CHINA | MALAYSIA

Design and Simulation of the Circuit using LTspice

Create the PWL file

Question 18:
Create a text file (using a text editor) that will be a binary representation of your 25 bit test stream. Use the bit period of 1 ms. The file should have similar to format to the provided 101_testpwl.txt, with a rise/fall time of 0.000001 ms (1 ns).

Name this file as mytestpwl.txt.

Copy and paste this text file into your report as **text** (do not screenshot).

The most significant bit (the bit on the left) of your test stream should enter the circuit first.

[3 marks]

Draw the circuit

Question 19:
Draw the schematic of the circuit required to detect your sequence. You can use the circuit from the 101 sequence detector as a starting point, if you wish.

Ensure that the Function of the input voltage source (which provides X) is set to the PWL file you created in Question 18. The voltage source that provides the clock should be set to 1 kHz (the same as the initial state of the previous 101 detector simulation). Set up the values of Td using the values from Table 2.

Part of this work will be checked automatically. Please ensure that you label the output net of your circuit as "Z", and your input net as "X".

[5 marks]

Simulate the circuit

Question 20:
Verify the function of your circuit using a transient simulation. Select an appropriate stop time so that the full 25 bits of your test steam are shown on the output graphs.

Place each of V(CLK), V(X) and V(Z) on a separate pane and export the .emf file. Add this graph to your document.

[5 marks]

University of Nottingham
UK | CHINA | MALAYSIA

Report Requirements

You should submit a report by the assigned deadline.

The report should be set out as follows:

Front page:

Include your Student ID and title of coursework

Answers:

Answer each question sequentially (make sure you make it clear which question number you are answering), including any requested image with your answer. For further clarity, add a caption to the image stating the question is relates to. Ensure that for your answers you show any working.

Total marks: **100**

What to submit

You are required to submit four (4) files, these are your:

1. Report (.doc/.docx/.pdf)
2. final .asc file (for the 101 detector) as it was for answering Question 12 – name this 101_detect.asc
3. final .asc file for your test sequence detector – name this mydetector.asc
4. the test pwl file containing your test stream – name this mytestpwl.txt