Logic

Andrew Mark Allen - 05370299

March 4, 2012

Abstract

NAND gates and inverters were used to construct several different logic gates whose operations were investigate under various inputs. Then the operation of a JK flip-flop was examined under a range of inputs, and finally both synchronous and asynchronous counters were constructed from JK flip-flops.

Aims

- To investigate the behaviour of logic gates
- To investigate the behaviour of JK flip-flops
- To construct a simple counter from JK flip-flops and logic gates

Introduction and Theory

Logic Gates

Logic gates are electronic devices that perform a logical operation on one or more inputs. Inputs are defined by voltage levels, if the voltage present is over a certain level it counts as logical 1 (True), otherwise logical 0 (False). From this it can be seen that for a logic gate with two inputs and one output there can be 16 possible rules defining the output of the logic gate. Some of these are trivial gates such as ones that always return 1 or 0, or the first input, or the second input. Of the rest, the most important are the AND, OR, NAND and NOR gate. Also of fundamental importance is the NOT gate, which takes one input and returns the opposite. The operations of the AND and OR gates are defined in the table below. It can be seen that the AND gate returns a 1 output when both the first input *and* the second input are true, while the OR gate returns 1 when either the first *or* the second output is true. The NAND and NOR gates are simply defined as the inverse of the AND and OR, i.e. where AND returns 1, NAND returns 0.

Boolean Algebra

(Idealised) logic gates are physical implementations of Boolean algebra, that is to say the output of any sequence of logic gates can be interpreted as and

X	Y	X AND Y	X OR Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

computed by a corresponding expression in Boolean algebra. Boolean algebra is an algebra with two operations, $\cdot$ (AND) and $+$ (OR). Ideas from Boolean algebra can be useful to help understand and to simplify digital circuits. One useful theorem from Boolean algebra is De Morgan's Law, $\overline{A \cdot B} = \overline{A} + \overline{B}$. This can be easily proved with a truth table:

A	B	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

Flip-flops

Flip-flops are similar to logic gates in that they are digital circuit components with some number of inputs and outputs. However they differ in that they maintain an internal state, the output being defined by a function of the inputs and the internal state.

Flip-flops can also have clock inputs which control when the device 'reads in' the inputs, this can be useful if we wish to change two or more of the inputs 'simultaneously'. For instance, without a clock input, if we wished to invert two of the inputs, it would be impossible to invert them such that they both changed exactly simultaneously, and there would be an undesirable brief period where one had changed and the other had not. This is known as a *race condition* and can be avoided with a clock pulse, if both signals change before the next rising edge of a clock pulse, then the behaviour would be as if they both changed instantaneously with the clock.

JK flip-flops are a type of flip-flop that have two normal inputs, J and K, a clock input, and two outputs, Q and its inverse $\bar{\mathrm{Q}}$. At any time Q is either 1 or 0. This is the state of the flip-flop. Inputs are read by the flip-flop on the downward cycle of the clock pulse (as the clock goes from 1 to 0). If both J and K are 0, the state does not change. If both J and K are 1, then the next state is the inverse of the current state. if J is 1 and K is 0, the next state is 1, the J is 0 and K is 1, the next state is 0. This behaviour is summarised in the table below:

Flip-flops can be used to easily construct more complicated circuits with predictable behaviour, such as as counters.

J	K	Q
0	0	Q (no change)
0	1	0 (reset)
1	0	1 (set)
1	1	$\bar{Q}$ (invert state)

Experimental Method

The apparatus for this experiment consisted of a board powered by a five volts power supply, with twelve sockets into which were inserted 7400 series integrated circuits, and five switches which can be set at logical one or logical zero (+5V or 0V respectively). The board also has a set of indicator lamps which light when connected to logical one. The integrated circuits contained either NAND gates (both two and three input types), inverters, or JK flip-flops.

NAND and AND gates

Using the SN7410 chip, the switches on the board and the indicator lights, the operation of the NAND gate was investigated and a truth-table was compiled.

OR and NOR Circuits

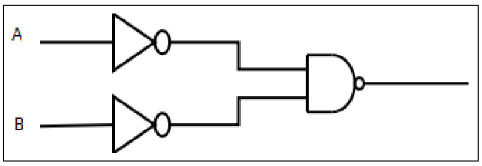

Figure 1: OR circuit created from NAND gates and inverters

Using Figure 1 the circuit for an OR gate was created and its operation investigated and a truth-table compiled.

Exclusive OR and Exclusive NOR Circuits

The circuits in 2 were wired shown to be equivalent implementations of an XOR gate, a gate whose output will be logical 1 if one or the other, but not both of the inputs, are logical 1. The operation of an XNOR (comparator) was then found by inverting the output of the XOR.

Flip-Flops

Using the SN7476 the JK flip flop is wired up and the effect of various inputs was recorded.

Counters

The counters in 3 and 4 were wired up and their operation was recorded.

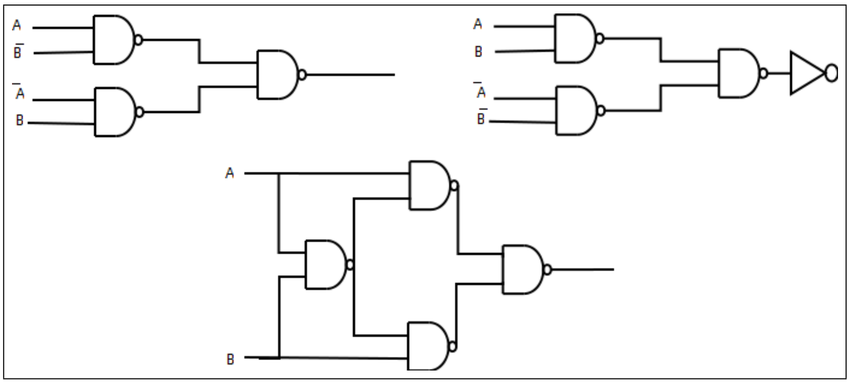

Figure 2: Three equivalent XOR circuits

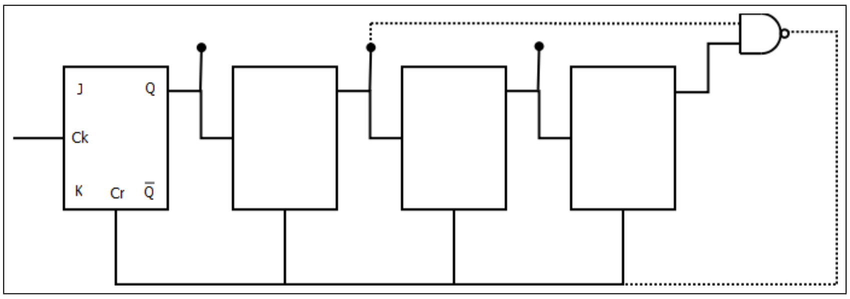

Figure 3: Asynchronous counter (scale of 16 with conversion to scale of 10 with dotted line)

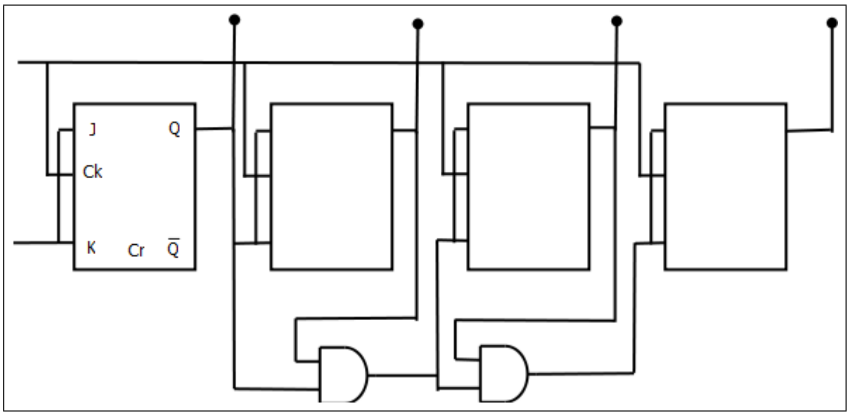

Figure 4: Synchronous counter (scale of 16)

Results and Analysis

The effect of the NAND gate was as expected, as shown in the table below:

The effects of the OR gate is shown below:

The effect of the XOR gate and its inverse the XNOR gate can be seen below. Looking at the results it can be seen that the 'comparator' is a more

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

intuitive name than 'exclusive nor' since its output is 1 if and only if both the inputs are equal to eachother.

A	B	A XOR B	A XNOR B
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

The behaviour of the JK flip-flop was observed and was the same as that described in the Theory section.

The asynchronous counting circuit has the result shown in the table below. The effect was for the lights to count up with each clock pulse in binary as far as fifteen, at which point the cycle would repeat itself. When the circuit was wired for the scale of ten, the cycle would repeat every ten clock pulses rather than sixteen, due to the clear signal sent when the 2^3 and 2^1 ($8 + 2 = 10$) outputs were true.

Pulse	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Discussion and Conclusion

The logic gates all produced the results expected mathematically. It was shown that any logic gate could be produced from a combination of NAND gates and inverters (or simply NAND gates, since any input NANDed with itself is inverted). After examining flip-flops, it was found that simple synchronous circuits running predictably on clock pulses could be created from asynchronous circuit components such as logic gates such as NAND gates. It was seen how simple counters could be created easily from such circuits.