Logic

Andrew Mark Allen - 05370299

March 3, 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

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.

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

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 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 has two inputs, J and K, a clock input, and two outputs, Q and its inverse $\bar{\text{Q}}$. 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:

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

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

Experimental Method

Results and Analysis

Discussion and Conclusion