Types of logic gates philosophy essay

Logic circuit is one that behaves like a switch, i. e. a two-positive devise with ON or OFF states. This is termed as binary device, in which the ON state is represented by 1 and the OFF state by 0.

We require devising a logic statement which can be expressed in only one of two forms. For instance

As you may know, computers store information in gigabytes, megabytes, and so on. The fundamental unit of storage is the bit or binary digit, which is similar to a switch. Like a switch, which has two states on or off, the bit has two states 1 and 0 (sometimes represented as true and false. We use this abstraction to represent numbers with the binary number system.

In order to perform calculations, the computer manipulates bits by means of logical operations, which use inputs to yield a particular output based on the input bits' states. To explicitly define the logic, we use truth tables which express the outputs in terms of all combinations of inputs.

Examples of truth tables for logical operations are given below:

Logical " AND" (i. e. A. B= C)

Logical " OR" (i. e. A+B= C)

Logical " NOT" (i. e. A')

A useful way of representing these logical operations is by means of logic gates, which are pictorial representations of the logic. The most basic logic gates, which illustrate the Boolean logic of the above truth tables, are

AND logic gate C= A. B

OR logic gate C= A+B

NOT logic gate B= A’

Using the logic gates above, we can create schematics of more complicated circuits. These combinational logic circuits involve feeding the output of one gate to the input of another gate. This allows us to create useful circuits that function according to our needs. Examples are provided below:

F= X. Y. Z

F= X+Y+Z

F= X+Y. Z

We probably know that calculators and computers store decimal (0-9) numbers as long strings of zeros and ones in a form called binary code. Each number is stored using microscopic electronic switches called transistors. It's easy to store binary numbers simply by switching transistors on and off. Switching on a transistor stores a number one; switching it off stores a zero. So storing numbers is easy. But how can you add, subtract, multiply, and divide using nothing but electric currents? Calculators and computers do these using clever electronic circuits called logic gates.

Fig: 1. 1

Photo: A gate can keep we out or let we into a field. In the same way, a microscopic logic gate is a barrier in an electronic circuit that can let electricity through or stop it flowing altogether. Put lots of logic gates together and you make a machine that's capable of basic mathematical " reasoning".

TYPES OF LOGIC GATES:

Logic gates: circuits that compare

A logic gate might sound horribly complex, but it's simply an electric circuit with two inputs and an output. It receives two incoming electric currents, compares them, and sends on a new, outgoing electric current depending on what it finds. A logic gate is a bit like a doorman or bouncer who is allowed to let people into a nightclub only if they pass certain tests. There are quite a few different types of logic gate, the most common of which are called AND, OR, NOT, XOR (Exclusive Or), NAND (NOT AND), and NOR (NOT OR). Let's look at the three simpler ones, AND, OR, and NOT:

AND

Suppose we went to a nightclub where the doorman's job is to enforce a simple rule: " Everyone in our group must wear a tie to come in". We went along with a friend one night. If we're both wearing ties, you'll get in. If only one of we is wearing a tie, or if neither of us is, neither of you will get in. An AND logic gate works the same way with two electrical inputs. If both inputs are switched on (that is, carry a number 1), the output will be 1 as well. Otherwise the output will be 0. In electronics, we can represent an AND gate with this little symbol. Three ways in which the gate can work are shown below.

OR

We're not wearing a tie, so you go to another club further down the street. Here, the person on the door is enforcing a different rule: " A group of people can come in if any one of them is a member". If either we or our friend is a member, or if you both are members, we can both come in. If neither of we is a member, you're both left out in the cold. An OR logic gate works this way with two electrical inputs. If either input is switched on (that is, carries a number 1), the output will be 1 as well. Otherwise the output will be 0. In electronics, we represent an OR gate with a different symbol. Three ways in which it can work are shown beneath:

NOT

So far, we've failed to get into either of the clubs. But there's one last hope: you know a friend is having a party a few streets away. The only trouble is, there's a really argumentative and contrary person on the door. He talks to each person in turn as they approach him. If you're nice and polite, he

shouts abuse at you and turns you away. But if you're rude to him, he likes that for some reason and lets you in. In other words, he does exactly the opposite of what you'd expect! In electronics, there's a logic gate that works in the same, contrary way and it's called a NOT gate or invertor. Unlike AND and OR gates, it has only one input and one output. The output is exactly the opposite of the input, so if the input is a 0, the output is a 1 and vice versa. Here's how we represent a NOT in electronics. Two ways it can work are shown beneath.

The other three common logic gates are variations on these three. XOR (Exclusive OR) is like an OR, but it switches off if both the inputs are switched on. NAND is just like AND, only the end result is swapped over (so where AND produces an output of 1, NAND produces an output of 0). NOR is like OR with the end result swapped over in the same way.

Calculators can do all the things they need to do using different combinations of logic gates. It's logic gates that control how the display works in a calculator and more logic gates that figure out the results of calculations. Let's take a closer look...

NAND gate

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

NOR gate

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high.

The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

EXOR gate

The ‘ Exclusive-OR’ gate is a circuit which will give a high output

if either, but not both, of its two inputs are high. An encircled plus sign () is used to show the EOR operation.

EXNOR gate

The ‘ Exclusive-NOR’ gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

The NAND and NOR gates are called universal functions since with either one the AND and OR functions and NOT can be generated.

Note:

A function in sum of products form can be implemented using NAND gates by replacing all AND and OR gates by NAND gates.

A function in product of sums form can be implemented using NOR gates by replacing all AND and OR gates by NOR gates.

IMPLEMENTATION OF LOGIC GATES

FIG: 1. 2 “ 130 ELECTRONIC CALCULATOR”

The Friden 130 uses diode-resistor “ OR” and “ AND” logic gates, with transistor-based inverter, buffer, and flip-flop devices. It performs math operations in bit-serial form, using the magnetostrictive delay line as the medium for storing its working registers. Logic levels are 0 Volts representing logic 1, and -12 Volts (nominally) representing logic 0. The delay line input transducer is driven with a pulse of approximately 20V, and by the time the signal makes it to the other end of the delay line, the voltage induced in the transducer is approximately 35mV, or 35/1000th’s of a volt. Digits are stored within the delay line as a series of pulses arranged in groups for each digit. Zero pulses represent a zero, and nine pulses represent a nine, with the numbers in-between represented by a number of pulses matching the number. As the pulses exit the delay line, they are amplified and fed into the counters (the A and/or D counters), which count the number of pulses in the digit to form a unique five-bit identifier that represents the number. The counter registers are not configured as counters in the usual binary sense. They are instead configured as five stage switch-tail shift registers, such that they count in a sequence of shifting 1’s. For example; 0 is represented as 00000; 1 as 10000; 2 as 11000; 3 as 11100; 4 as 11110; 5 as 11111; with 6 as 01111, and ending with 9 as 00001. With five flip flops, each counter can represent the numbers zero through nine as unique combinations of bit patterns.

In terms of applying logic gates to real world applications, many gates can be used in

Figure1`. 3CEP-1114

Piezo Buzzer

Frequency processing. For example, when dealing with piezo audio transducers such as

the CEP-1114, an oscillating frequency signal is require to produce a sound pattern from

the buzzer. In order to accomplish this, a square-wave frequency signal from sort of

Figure 3: AND gate signal output

wave generator such as an oscilloscope or a programmed PIC

must be logically processed with a second frequency signal to

create a combination of high/low outputs which will turn the

Buzzer on and off creating a sound pattern. So if you used a

Square-wave signal from a PIC as one input into an AND gate

and a second varying frequency as the other input, it is proven

due to AND gate logic that when both signals are high the buzzer

will produce a sound, and any other combination will turn the

buzzer off.

Furthermore, as you can see from Figure 3 on the

left, the output is only high when both Input A and

Input B are high. If Input B was

changed or altered, this would alter the output

signal sent to the buzzer. Imagine that Input B

was stretched so that only one period was

observed in the time give rather than two periods.

This would extent the time that Input B is high

and would increase the time that the output is

high.

Additionally, if variation of the waveform inputs

is difficult or impossible, variation of the output

can still be accomplished by simply changing the

logic gate implemented. If the AND gate currently

being used was replaced with a NAND gate, the

output would be completely reversed. Every time the output was high using the AND

gate will now be low, and every time the output was low will now be high.

Conclusion

Logic gates come in all shapes and sizes, and whether they are used in combinations or

individually, they provide the user with many options and solutions to problems which

may appear difficult at first. Furthermore, these seven basic logic gates ease the

complexity of Boolean algebra and allow for simple application in electronics and circuit

analysis. These gates can be used in almost any situation such as comparison of

frequencies when developing filters in communication or in more mechanical settings

when using choppers and inverters which compare input and output currents to determine

modulating indexes.

Figure 2: CEP-1114

Piezo Buzzer

FLIP FLOPS

Flip flops are actually an application of logic gates. With the help of Boolean logic you can create memory with them. This is the most basic idea of a Random Access Memory [RAM]. If the logic gates are designed correctly, they will be helpful in remembering the input value given to them. A higher application of flip flops is helpful in designing better electronic circuits.

The main use of flip flops is the implementation of a feedback circuit. As a memory relies on the feedback concept, flip flops can be used to design it. Given below is a simple feedback circuit using two inverter logic gates. Take a look.

Though this circuit is not good for practical electronic circuits, it will help you to get a clear idea on what a feedback circuit is. When the value of the output ‘ Q’ happens to be 0, it always remains 0. If the output happens to be 1, it always remains 1

There are mainly four types of flip flops that are used in electronic circuits. They are

The basic flip flop or S-R Flip Flop

Delay Flip Flop [D Flip Flop]

J-K flip flop

T flip flop

1. S-R Flip Flop

The SET-RESET flip flop can be designed with the help of two NOR gates and also two NAND gates. These flip flops are also referred to as S-R Latch.

S-R Flip Flop using NOR Gate

The flip flop has mainly two inputs, called the SET [S] and RESET [R]. There are also two outputs, Q and Q'. The diagram and truth table is shown below.

S-R Flip Flop using NOR Gate

From the diagram it is evident that the flip flop has mainly four states. They are

S= 1, R= 0-Q= 1, Q'= 0

This state is also called the SET state.

S= 0, R= 1-Q= 0, Q'= 1

This state is known as the RESET state.

In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the value of S.

S= 0, R= 0-Q & Q' = Remember

If both the values of S and R are switched to 0, then the circuit remembers the value of S and R in their previous state.

S= 1, R= 1-Q= 0, Q'= 0 [Invalid]

This is an invalid state because the values of both Q and Q’ are 0. They are supposed to be compliments of each other. Normally, this state must be avoided.

S-R Flip Flop using NAND Gate

The circuit of the S-R flip flop using NAND Gate and its truth table is shown below.

S-R Flip Flop using NAND Gate

Like the NOR Gate S-R flip flop, this one also has four states. They are

S= 1, R= 0-Q= 0, Q’= 1

This state is also called the SET state.

S= 0, R= 1-Q= 1, Q’= 0

This state is known as the RESET state.

In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the compliment value of S.

S= 0, R= 0-Q= 1, & Q’ = 1 [Invalid]

If both the values of S and R are switched to 0 it is an invalid state because the values of both Q and Q’ are 1. They are supposed to be compliments of each other. Normally, this state must be avoided.

S= 1, R= 1-Q & Q’= Remember

If both the values of S and R are switched to 1, then the circuit remembers the value of S and R in their previous state.

Clocked S-R Flip Flop

It is also called a Gated S-R flip flop.

The invalid state can only be removed by using a bistable SR flip-flop that can change outputs when certain invalid states are met regardless of the condition of either the Set or the Reset inputs. For this, a clocked S-R flip flop is designed by adding two AND gates to a basic NOR Gate flip flop. The circuit diagram and truth table is shown below.

Clocked S-R Flip Flop

A clock pulse [CP] is given to the inputs of the AND Gate. In the beginning the outputs of both the AND Gates remain 0 until the value of CP is 0. When a pulse is given the value of CP turns 1. This makes the values at S and R to pass through the NOR Gate flip flop. But when both the S and R values turn 1, the HIGH value of CP causes both of them to turn to 0 for a short moment. As soon as the pulse is removed, the flip flop state becomes intermediate. Thus either of the two states may be caused, and it depends on whether the set or reset input of the flip-flop remains a 1 longer than the transition to 0 at the end of the pulse. Thus the invalid states can be eliminated.

2. D Flip Flop

The circuit diagram and truth table is given below.

D Flip Flop

D flip flop is actually a slight modification of the above explained clocked SR flip-flop. From the figure you can see that the D input is connected to the S input and the complement of the D input is connected to the R input. The D input is passed on to the flip flop when the value of CP is 1. When CP is HIGH, the flip flop moves to the SET state. If it is 0, the flip flop switches to the CLEAR state.

To know more about the triggering of flip flop click on the link below.

TAKE A LOOK : TRIGGERING OF FLIP FLOPS

TAKE A LOOK : MASTER-SLAVE FLIP FLOP CIRCUIT

3. J-K Flip Flop

The circuit diagram and truth-table of a J-K flip flop is shown below.

J-K Flip Flop

A J-K flip flop can also be defined as a modification of the S-R flip flop such that it is a refinement at the indeterminate state of the SR flip flop.

The inputs J and K behave just like the S and R inputs of the S-R flip flop. The letter J stands for SET and the letter K stands for CLEAR.

When both the inputs J and K have a HIGH state, the flip-flop is designed to switch to its complement state. So if the value of Q was 1, it switches to Q= 0 and if the value of Q was 0 it switches to Q= 1.

Two 3-input AND Gates are used in the circuit. The output Q of the flip flop is given as a feedback to the input of the AND along with other inputs like K

and clock pulse [CP]. So, the flip flop gets a CLEAR signal when the value of CP is 1 only if the value of Q was earlier 1. Similarly output Q’ of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP]. So the output becomes SET when the value of CP is 1 only if the value of Q’ was earlier 1.

The output may be repeated in transitions once they have been complimented for J= K= 1 because of the feedback connection in the JK flip-flop. This can be avoided by setting a time duration lesser than the propagation delay through the flip-flop. The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction.

4. T Flip Flop

This is a much simpler version of the J-K flip flop. Both the J and K inputs are connected together and thus are also called a single input J-K flip flop. When clock pulse is given to the flip flop, the output begins to toggle. Here also the restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction. Take a look at the circuit and truth table below.

T Flip Flop

Conclusion

Logic gates come in all shapes and sizes, and whether they are used in combinations or

individually, they provide the user with many options and solutions to problems which

may appear difficult at first. Furthermore, these seven basic logic gates ease the

complexity of Boolean algebra and allow for simple application in electronics and circuit

analysis. These gates can be used in almost any situation such as comparison of

frequencies when developing filters in communication or in more mechanical settings

when using choppers and inverters which compare input and output currents to determine

modulating indexes