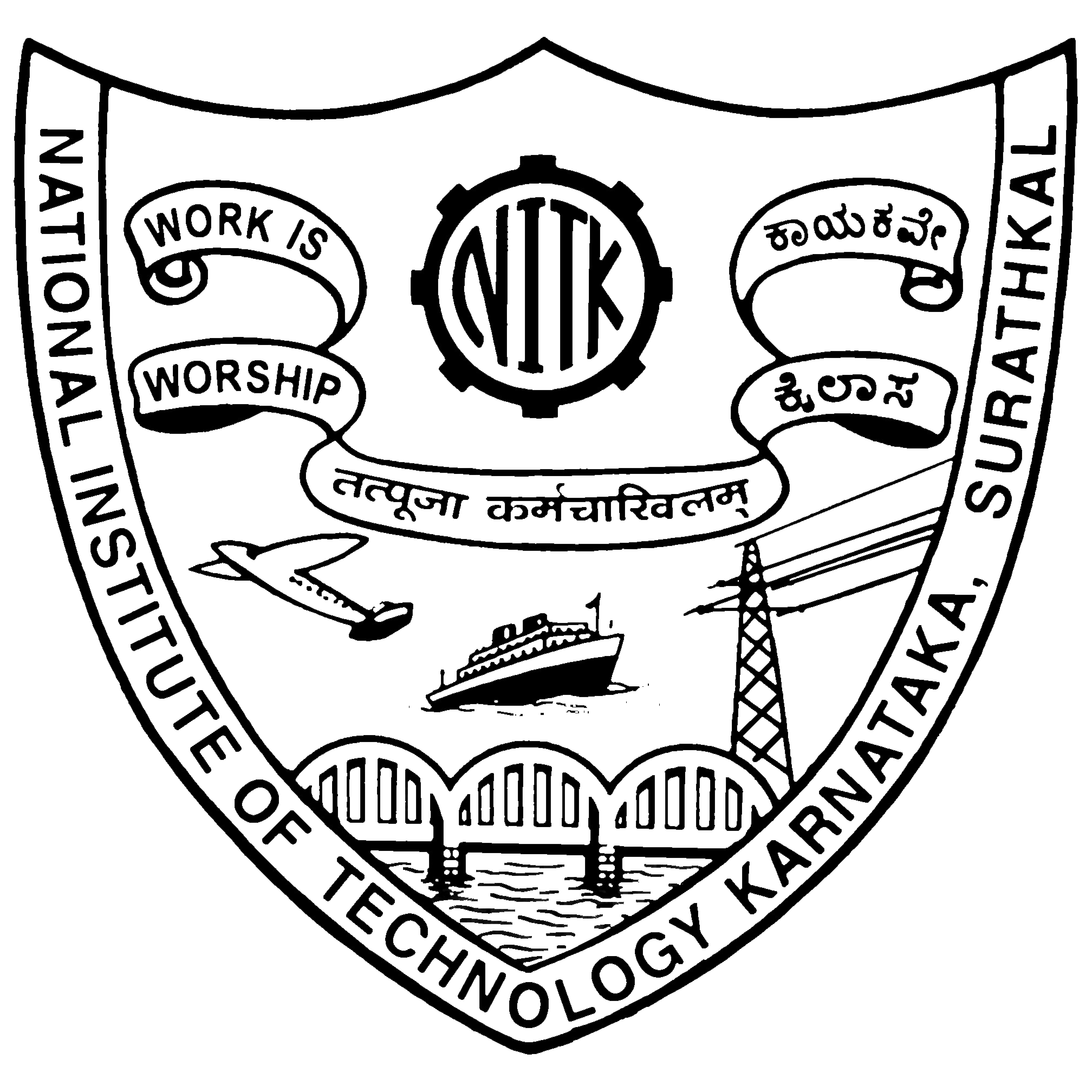
**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Digital system Project on**

**SIMULATION OF A DIGITAL CLOCK**

Submitted by: Submitted to:

Nikhil Chavan 11CO62 Ms Shilpa

Rahul Das 11CO72

Rahul K Bhojwani 11CO73

Shreyas G. 11CO88

**INTRODUCTION**

In this project, we have implemented the simulation of a digital clock that displays seconds, minutes and hours using the software Logism.

We know that 60 seconds equal to 1 minute and 60 minutes equal to 1hour. Hence the minute section is derived by second section and hour section derived by the minute section. Each of the minute and second section has been designed to give a count from 00 to 59 after which it resets to 00. And the hour section to give a count from 00 to 12 hours after which it resets to 01. For each cycle of 00 to 59 in second section the minute section increases its count by 1. Similarly for each cycle of 00to 59 in minute section the hour section increases its count by 1. In this way when the clock reaches 12hrs. 59mins. 59secs.each of the section resets to 01 giving us a display 01.00.00.

**OBJECTIVE AND SCOPE**

A clock is an instrument used to indicate, keep, and co-ordinate time. It is one of the oldest inventions by man. The earliest human clocks included sundials, water clocks, etc. We have come a long way since then to invent modern digital clocks, which are being improved upon even now. Today, the clock has become more than just a device that shows the time. You might actually be surprised when you learn of all the other uses for clocks that are out there today. They are commonly used to control other devices, such as a VDR or DVD players. When they are used in this manner, they allow you to go back and forth in the tape or disc to find a specific part of the movie that you want to watch. Computers actually depend on clocks in their internal parts to synchronize their processing, documents, and programs. While there are master computers in some networks that will control and keep the time on one master clock, normal computers use their own internal clocks to keep themselves running right. Without a navigation clock, many ships and pilots would be hopelessly lost. Atomic clocks actually use a radio signal to help keep our GPS' working and showing us where we are, where we should be going, or where someone needs to go to find you. These radio signals are bounced off of satellites in space to pinpoint exactly where people, cars, planes, and more are and need to be going. We then have seismic clocks that help to determine the arrival time of an earthquake, seismic wave, or other ground shaking happening. These clocks are used all over the world to ensure that accurate reporting can be made of any earthquake or seismic event. Clocks are also used as counters on explosive devices to allow the user to count down so that they have time to get out of the way. While there are many bad implications with this use of clocks, they are used in good ways - such as for demolition or for rocket launches. By counting down, it allows for maximum safety for any bystanders that might be around when used correctly. In the end, all these complicated clocks can be simplified to one simple thing – the digital clock. As mentioned above, like how the scope of this simple clock can be extended to even explosives, seeing the importance of clocks, this is what prompted us to make our project on the digital clock.

**COMPONENTS USED**

* J K Flip Flop
* Mod 6 Counter
* Mod 10 Counter
* Basic Gates (AND, OR, NOT)
* Multiplexers
* 7 Segment Display

**METHODOLOGY**

Digital clock is simulated by counting using basic counters and display each digit by seven segment display.

Basic JK Flip flop in connected as shown above to construct a 4 bit mod-6 and mod-10 counter

The 4 bit output of each counter at each clock cycle is given to BCD to Seven display converter

(Given in above diagram), the corresponding seven outputs from the above circuit is connected

accordingly to seven segment display such that when count is 1 it display digit 01 and next clock cycle it displays 02 and so on when it reaches 09 it resets the flip flops to zero.

Similarly Minute display is constructed but the circuit is enabled only when the second's clock resets itself to zero (mod 60).

Since hour clock resets to 1 not to zero, we cannot use mod-12 counter. Hence we design a special circuit using D-Flip flop which counts from 1-12 and resets back to 1. Count from 0 - 9 has to change it is given directly to BCD to seven segment converter when count increases from 10-12

A select line is used indicating increase in count beyond BCD using it in a multiplexer inputs for 10th place and unit place of hour clock is given by generating a function which produces a corresponding output as shown below

|  |  |  |  |
| --- | --- | --- | --- |
| STATE | 8 4 2 1 | INPUTS OF HOUR DISPLAY | |
| a b c d | Y1 | Y2 |
| 10 | 1 0 1 0 | 0001 | 0000 |
| 11 | 1 0 1 1 | 0001 | 0001 |
| 12 | 1 1 0 0 | 0001 | 0010 |

Function for select line S=a .(b+c) which is given as the input for LSB in Y1 and for Y2

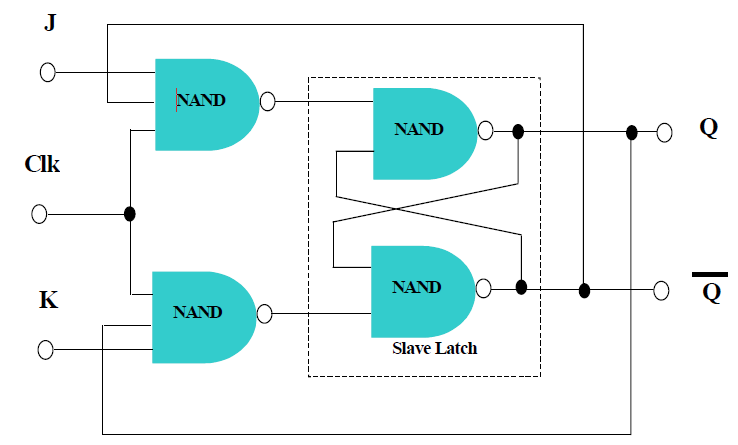
LSB is defined by q1=(c .d) and next significant bit q2=(a.b) while others are grounded.

Hence the hour display is done which is enabled only when minute wraps to zero after 60(mod 60).

**LOGIC CIRCUITS**

**JK FLIP FLOP**

The J-K flip-flop is the most versatile of the basic [flip-flops](http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/flipflop.html#c1). It has the input- following character of the clocked [D flip-flop](http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/dflipflop.html#c3) but has two inputs,traditionally labeled J and K. If J and K are different then the output Q takes the value of J at the next clock edge. If J and K are both low then no change occurs. If J and K are both high at the clock edge then the output will toggle from one state to the other. It can perform the functions of the [set/reset flip-flop](http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/setreset.html#c1) and has the advantage that there are no ambiguous states. It can also act as a [T flip-flop](http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/tflipflop.html#c1) to accomplish toggling action if J and K are tied together. This toggle application finds extensive use in binary counters.



**SYNCHRONOUS COUNTERS:**

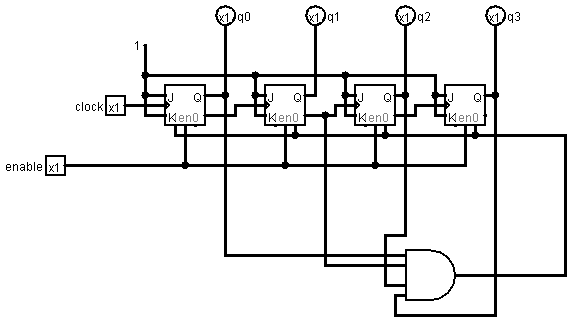
A 4-bit synchronous counter using JK flip-flops

A simple way of implementing the logic for each bit of an ascending counter (which is what is depicted in the image to the right) is for each bit to toggle when all of the less significant bits are at a logic high state. For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on.

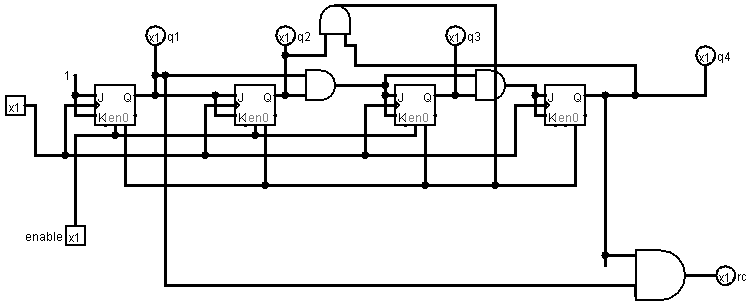
Synchronous counters can also be implemented with hardware [finite state machines](http://en.wikipedia.org/wiki/Finite_state_machine), which are more complex but allow for smoother, more stable transitions.

Hardware-based counters are of this type.

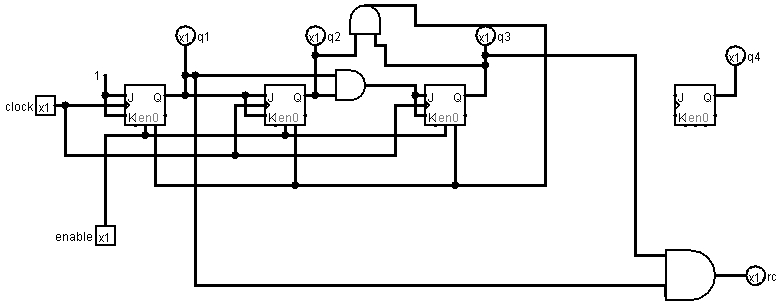
***MOD 12 COUNTER:***



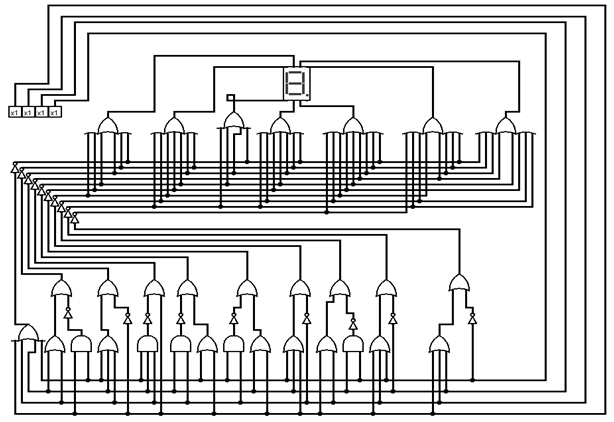
***MOD 10 COUNTER:***



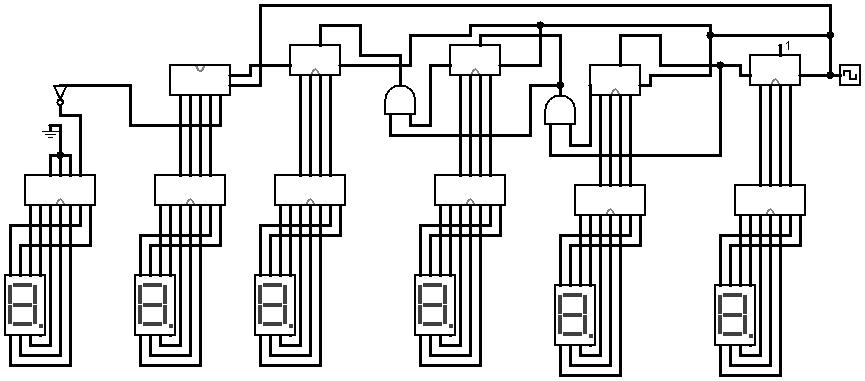
***MOD 6 COUNTER:***



**BCD TO SEVEN SEGMENT DISPLAY:**



**FINAL DIGITAL CLOCK:**



**CONCLUSION**

The circuit was purely designed with the basic knowledge on sequential circuit designing and with the software Logism. The advantage of our clock is that the clock tick frequency can be controlled and modified as per our wishes. This means that we can modify the time period of each second of the clock. We have only designed a simple digital clock. This can be further implemented in other devices like computers, mobile phones, etc. The clock is expected to operate normally with desired accuracy.

**ABOUT LOGISM**

Logism is a [logic simulator](http://en.wikipedia.org/wiki/Logic_simulation) which permits circuits to be designed and simulated using a graphical user interface. Released under the [GNU Public License](http://en.wikipedia.org/wiki/GNU_Public_License), Logism is [free software](http://en.wikipedia.org/wiki/Free_software) designed to run under the [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X), and [Linux](http://en.wikipedia.org/wiki/Linux) platforms. Its code is entirely in [Java](http://en.wikipedia.org/wiki/Java) using the [Swing](http://en.wikipedia.org/wiki/Swing_(Java)) graphical user interface library. The primary developer, Carl Burch, has worked on Logism since its inception in 2001.

The software is used most often by students in [computer science](http://en.wikipedia.org/wiki/Computer_science) classes to design and experiment with [digital circuits](http://en.wikipedia.org/wiki/Digital_circuit) in simulation. Circuits are designed in Logism using a graphical user interface similar to traditional drawing programs, an interface also found in many other simulators. Unlike most other simulators of Logism's sophistication, Logism allows the user to edit the circuit during simulation. The relative simplicity of the interface makes it work well for survey courses.[[1]](http://en.wikipedia.org/wiki/Logisim#cite_note-0) Design features for more sophisticated circuits, such as the "subcircuits" and "wire bundles" found in Logism, are available in few other open-source graphical tools.

While users can design complete CPU implementations within Logism, the software is designed primarily for educational use. Professionals typically design such large-scale circuits using a [hardware description language](http://en.wikipedia.org/wiki/Hardware_description_language) such as Verilog or [VHDL](http://en.wikipedia.org/wiki/VHDL). It is unable to accommodate [analog components](http://en.wikipedia.org/wiki/Analog_circuit).

**LIST OF REFERENCES**

* en.wikipedia.org/wiki/Logisim
* <http://electronics.howstuffworks.com/gadgets/clocks-watches/digital-clock.htm>
* <http://www.ustudy.in/node/3021>
* <http://www.google.co.in>
* http://www.wikipedia.org
* <http://ezinearticles.com/?Clocks-Arent-Just-For-Telling-Time---Other-Uses-For-Clocks-Around-The-World&id=3121429>
* Digital Design and Computer Design, by Morris Mano