Virtual laboratories
The future for quality on line Engineering Education

Report submitted in Partial fulfillment of the requirement for

degree of

B.Sc.

In

Electrical and Electronic Engineering

Under the Supervision of

Dr. Idris Ahmed Ali

By

Aya Abouzaid Jalal El-dean

To

Department of Electrical and Electronic Engineering

University Of Khartoum

July 2005
Dedication:-

To:

my father,

my mother,

my sister,

friends,

department staff,

my family,

and for those who give me the advice and like to see me in the right way…..
Acknowledgment: -

I would like to acknowledge my supervisor; Dr. Idris Ahmed Ali for his patience and suggestions.

Also, I would like to thank my colleague Mai Safi Eldeen for her nice accompany.

Besides, I would like to thank Ustaz Nazar Hashim and our colleague Taha Yousif for their support and to all who give me advice and suggestion.
Contents

Chapter 1: Introduction .................................................................................................................1

Chapter 2: Literature review ..........................................................................................................4

1- Introduction .................................................................................................................................4

2- Advantages of Virtual Laboratories ..........................................................................................4

3- Categories of Available Systems ...............................................................................................6

3.1- Accessibility and GUI ..............................................................................................................6

3.2- Types of Applications .............................................................................................................6

4- Types Of Virtual Labs ................................................................................................................7

4.1- Simulated Virtual Labs ..........................................................................................................7

4.1.1- Advantages of Simulation ..................................................................................................7

4.1.2- Disadvantages of Simulation ..............................................................................................8

4.1.3- Classification of Simulated Systems ...................................................................................8

4.1.4- Simulation Models ...............................................................................................................8

4.1.4.1- Continuous Simulation ....................................................................................................8

4.1.4.2- Discrete Simulation .........................................................................................................9

4.1.4.3- Monte Carlo Simulation .................................................................................................9

4. 2- Remote Virtual Labs ..............................................................................................................9

4.2.1- Advantages of Remote Lab ...............................................................................................10

5- Approaches For Constructing Virtual Labs ...........................................................................10

5. 1- Pure HTTP Approach ..........................................................................................................10

5. 2- Mobile Agent Approach .......................................................................................................11

5. 2. 1- Advantages of Mobile Agent Approach ...........................................................................11

5. 2. 2- Disadvantages of Mobile Agent ......................................................................................13

5. 3- Telnet Approach .....................................................................................................................13

5. 3. 1- Advantages of Telnet ........................................................................................................15
Chapter 3: Design & Analysis .......................................................... 18
1- Design Requirements ......................................................... 19
   1.1- HTTP Approach ......................................................... 19
   1.2- Java Language .......................................................... 19
      1.2.1- Java For Simulation ............................................ 19
      1.2.2- Java For Programming Over The Networks ............ 20
2- Design Procedure ............................................................ 21

Chapter 4: Results & Discussion ............................................... 23
1- Basic Logic Gates Experiment ............................................. 24
2- Half Adder Experiment ..................................................... 24
3- The Full Adder experiment ............................................... 25
4- Decoder Experiment ......................................................... 26
5- Multiplexer Experiment ................................................... 26
6- DeMultiplexer Experiment .............................................. 26
7- Seven Segment Display .................................................... 28

Chapter 5: Conclusion ............................................................. 29

References .............................................................................. 33

Appendix: Code ........................................................................ 35
Abstract:-

This project is concerned with the main concepts of the virtual laboratories which are special type of laboratories that can be accessed and done from remote areas using the web and designed by special type of software programs, and in some cases a combination of software and hardware.

The project gives a description of virtual labs: their application, advantages and disadvantages for the virtual reality and real world, and available approaches for designing such a system.

The project also include analysis and procedure by which an implementation of digital lab was made using the web and written in java programming language. Also, some results of the lab and the procedure by which the user can do the experiment had been included.
المستخلص:

يهدف هذا المشروع إلى فهم المفاهيم الأساسية للمعامل الإفتراضية والتي تمثل نوع خاص من المعامل يتم دخولها والعمل فيها من خلال البحث عن طرق الشبكة ومن دون الحاجة إلى التواجد داخل مبنى المعمل. يتم تصميمها باستخدام برامج خاصة أو برامج وعتاد. يتم المشروع تفصيل عن هذه المعامل الإفتراضية:

- استخداماتها، الفائدة المستخلصة منها، وطرق المستخدمة لتصميم مثل هذه النظم.
- أيضاً يحتوي هذا البحث على الطرق المتباينة في التحليل والتصميم لتصميم معامل افتراضي لمقرر النظام الرقمي، وذل ذلك عن طريق المحاكاة باستخدام لغة الجافا. أيضاً تم إرفاق بعض النتائج لعمل وتشغيل هذا المعامل الإفتراضي إضافة إلى شرح وتوضيح لطريقة عمل هذا المعامل الإفتراضي.
CHAPTER [1]
INTRODUCTION

The internet plays an important role in our life. It provides a connection to nearly every part of the world. In the last few years internet had been used for educational purposes in what we call "distance learning" or "e-learning". Distance learning is a recently developed reality in which students and teachers in different areas can interact with each others by accessing remote resources available in the internet. It also represents the way by which students can learn and join virtual universities and e-learning institutions using the web as a teaching medium to make the students able to learn at their own place. The Internet is very popular as it is not limited by political, geographic boundaries or cultural barriers. Learners can save their time by scheduling progress of learning according to their own styles. This is very effective and useful facility which will widely increase the chance for people in remote areas to learn from their home using the internet. Also, it can be redirected to identifying and helping learners who are having problems.

Some times, as in engineering education, there is emergency need for laboratories and practical exercises for theoretical taught materials. The traditional laboratories are inefficient in this case since it requires that the user should be available inside the lab. Thus, there is a need for designing lab that would be available in the virtual reality. Thus labs should be available in the virtual universities as long as the other parts of the course, such as lectures, homeworks, tests, exams, and assignments.

Virtual laboratory thus is the lab that is entered and used by the students using the web and from its location with no need for the student to be available in the same place of the lab. There are many approaches by which virtual labs can be constructed. Virtual labs can be either a simulated or remotely controlled labs.

Simulation is a proper way to complement e-learning and virtual realities but in general, virtual laboratories cannot replace experiments on real labs because it does not give the user the same feeling as in real, physical, laboratories. On the other hand remote laboratories could reduce the operating costs and give a better use of instruments and many other benefits which make it great internet application. Thus, students will be able to do and see the experiments using virtual environment. Another objective of a virtual laboratory is to provide hands-on lab activities to enhance online courses.

This project is about virtual laboratories; the advantages, application, the design, and implementation of virtual labs. In chapter 2 of this project the reader will find a discussion concerned with the previous work of virtual laboratories and the available approaches of designing such a laboratory.
In this project a simulated digital virtual lab was designed using the web based approach and java programming language. The lab is now available in the net and on the university local area network and every student can use it at any time. In the next chapter the reader will find the design methods and analysis which were used for designing this virtual lab. In the last chapter the results were presented.
Chapter [2]

LITERATURE REVIEW

INTRODUCTION:

Laboratories and experimentations are very important educational tools; by which the students can gain experience by practice. A virtual laboratory system is a computing system that allows the share of physical resources available in a laboratory with remote users connected on the internet. Another definition of the term "Virtual Laboratory" is the interactive environment for creating and conducting simulated experiments: a playground for experimentation. It consists of domain-dependent simulation programs, experimental units called objects that encompass data files, and tools that operate on these objects. The objects are organized, accessed and manipulated using domain independent Virtual lab (Vlab) system programs [1].

Few years ago, the internet was only used for purposes of information searching, e-mails, and entertainments. But today, the internet has many applications and it can be used to conduct actual experiments to physical devices available in remote locations or to enable the student to interact with a simulated form of virtual labs. Virtual lab is a very powerful technology; it gives the students the ability for doing the experiments at any time and from a remote area with equipments that are not physically available this is a very effective technology especially in the case of engineering education where concepts taught through lectures must followed by laboratory experimentation. Such a technology highly improves the education techniques, training, and collaborative investigations. The development of new communication infrastructures supports the extension of the function offered by distance learning to real laboratories whose physical processes can be remotely controlled [2].

Virtual labs have main functions and roles which when done will largely enhance the distance learning and those are: It must be a pre-practice for physical lab. With the aid of improved technology the virtual lab should replace the physical labs. It can also be as a post-analysis of physical lab [3]. For more convenient, the lab resources should provide alternative explanations or examples that will be valuable for the students. Moreover, the resource should be engaging and interactive and should meet high standards of access.

2- ADVANTAGES OF VIRTUAL LABORATORIES:

Virtual laboratories have many advantages in education especially in engineering e-learning. It let the student observe the dynamic phenomena that are often difficult to explain by written material. It covers the
limit of time and distance since the user could use the lab at any time and from any where using the internet or mainly networks. Also, in our quickly developing world there is always need for new lab infrastructure which cost very much, but when using virtual labs we decrease the number of used hardware and equipments thus the cost is reduced drastically. Besides, a single virtual lab can be used by a larger group of people and shared between different universities and thus it becomes more efficient when we need a wider access to expensive or rare equipment. Add to these, the sharing of geographically distributed equipments. It is also useful in case of education or training with equipment that is not easily accessible or the one that can be damaged by a novice. More over, it incorporates other computer tools, such as intelligent tutoring systems, in order to provide feedback for the learning and training processes. Virtual labs bring people working in different aspects together to share their experiences and ideas, and to demonstrate their developments by providing access to their virtual laboratories. Also, the students are allowed to access resources they do not have, and they can run experiments for much longer time and repeat it many times rather than typically allowed use in real world laboratories. Students can also visit several virtual laboratories available in the world, and then compare the different approach and solutions for the same problem and the different techniques by which a unique experiment can be done which is very effective way for developing students' skills. An amazing property of virtual labs is that the scientists and technicians can cooperate together by running joint experiments from their own office on large and expensive equipment, reducing travel costs and allowing observation of results to other people in real time universities.

Adding to these advantages, which were related to the performance improvement, virtual labs have several economical benefits. In fact, the classical educational model requires the usage of a large amount of resources, in terms of people, space, travel, and time for teaching and in certain disciplines, especially those related to engineering. Moreover, shared devices are more protected against inappropriate usage. In fact, a suitable user interface can disable dangerous operations, or bring the system in a safe state when it detects critical conditions that might cause any damage. Virtual labs are also useful for industrial purposes since manufacturing engineers can update product line equipment and correct manufacturing line problems from their office and the support engineers can troubleshoot and configure products without visiting the site, once again saving time and expense while speeding up customer service. Also, they can be used to explore places that cannot normally be reached such as the space, oceans, volcanoes…etc.[4,5,6]. The virtual laboratories are particularly useful when some experiments involve equipment that may cause harmful effects for humankind. Recall, practical laboratory experiment is required to complete the course of study and corroborate the theoretical parts with direct experience. In the absence of real laboratories, or when the available resources are not sufficient for satisfying the actual demand of students, experiments are performed via simulation through specific software tools. Where, a real laboratory experience offers the student the possibility to know the actual behavior of a system, including non linearity and noisy data a virtual laboratory environment does not preclude a student to face with these problems. A study at East Carolina University found that virtual laboratories help the students to understand concepts and theory in online courses although
there is not direct interaction with the remote system, the student can send input data to the real system and receive back all output data produced during the experiment. Different kinds of experiments can be implemented using the virtual laboratory approaches.

3- CATEGORIES OF AVAILABLE SYSTEMS: -

The previous reported work in virtual labs can be categories using two features: accessibility and graphical user interface, and the type of experimentation.[7]

3.1- Accessibility and graphical user interface (GUI):

The practical systems implementation shows two kinds of virtual lab depending upon the standpoint of access to these systems and there GUIs. These two types are; labs that can be accessed using local area networks LAN with dedicated software. The others are labs that can be accessed via the internet with web browsers and there GUIs.

3.2- Types of Applications:

The types of application or experiment that appears in the previous works are mainly limited to the following categories. The setting systems parameters; such as robotic arms movement, video camera photo, and sound transferring. Another type is the process and/or software simulations as logic circuit and nuclear explosion simulations. Beside, there are the status reading applications; such that can be found in weather reading,, vending machines, and networks status. Many of virtual labs operate with the use of software simulators. But, the remained systems use built virtual labs. These categories and there sub categories can be illustrated using figure 1.

![Figure 1: Simplified taxonomy of virtual labs](image-url)
4- TYPES OF VIRTUAL LABS:

The virtual labs can be divided into two main types from design point of view; remote virtual labs and simulated labs [8].

4.1- Simulated Virtual Labs:

A simulated lab has computer models of the equipment and its environment, and the user interacts with these models to perform the required lab or experiment using the provided graphical user interface. The simulation process can be defined as the technique used for the design of a real system model and it can be the way by which we conduct experiments with this model for the purpose of understanding the behavior of the system or evaluating various strategies for the operation of the system [9]. The increased technologies show that there will be more and more complex systems and unless using simulators the understanding of the way by which such a system work becomes very difficult. Unfortunately, the design complexity of the simulation system and its implementation is increased with the increase of implementation level of the systems being simulated. As a trend to reduce this undesirable effect, the software community has made a concerted effort to apply the lastest advancements in software technology in an attempt to counteract this ever increasing complexity. Such an interface can be used to show the operation of the system under simulation this give the ability to observe the fundamental behavior of elementary systems.

4.1.1- Advantages of Simulation:

Simulation is the way by which the designer can provide the user with practical feedback of a real world system. This allows the designer to determine the efficiency and qualified the performance of a design before it is actually constructed. Besides, by simulators the user can explore the merits of alternative designs without actually physically building the systems to find the best design. As a result, the overall cost of system building diminishes significantly. An example of those is the circuit simulator which is able to provide the designer with information and performance of alternate designs. After operating several trials for the alternate designs, the best circuit then may be fabricated. Another advantage, of simulators is there ability to let system designers to study a problem at several different levels of abstraction. By approaching a system at a higher level of abstraction, the designer must understand the behaviors and interactions of all the high level components within the system and is therefore better equipped to counteract the complexity of the overall system. This complexity may simply overwhelm the designer if the problem had been approached from a lower level. As the designer better understands the operation of the higher level components through the use of the simulator, the lower level components may then be designed and subsequently simulated for verification and performance evaluation. The entire system may be built based upon this top-down technique. Moreover, simulators can also be used for teaching purposes to students using computer graphics and animations. The simulators thereby give the user, student, a good idea and a well understanding for the nature of the system. Such a simulator should provide a full control for the user to speed up, slow down, stop or even
reverse a simulation as a means of aiding understanding. In experimentation the simulation is very powerful since it permits hard or impossible experimentation. Simulation some times is a technique for gaining time e.g. it can show the behavior of a system that requires a period of a year in only a few minutes.

4.1.2- Disadvantages of Simulation:

Like any other used technique simulation has also some disadvantages many of these can be related to the computationally intensive processing required by some simulators. The simulation some times may be a time consumption technique this is seen in case of complex simulation where a result that may take some seconds in real time will take up to few hours which becomes very boring. These delays may be due to an exceedingly large number of entities being simulated or due to the complex interactions that occur between the entities within the system being simulated. However, this problem is becoming with less attention since more powerful platforms and improved simulation techniques become available. Also, there are some methods used to reduce the simulation time delay as to use simplifyed assumptions but some times it may gives its users a false sense of security regarding the accuracy of the simulation results. Another method to reduce the computational complexity is the employment of hierarchical design for the simulator. But, this technique may also result on other problems as the problem of nonconvient results that appears with the use of too high abstraction level. This is because of the negligible lower level details of the system and the actual construction of the system will not be able to occur until the user provides low level information concerning subcomponents of the system.

4.1.3- Classification of Simulated Systems:

There are two main types or classifications of simulated systems. This depends on the degree of randomness associated with the behaviour of the system in the simulation environment. Those are: stochastic systems, and deterministic simulation systems. The stochastic system is such one that relies upon random behaviour and simulates the duration of each transaction. The results generated from a stochastic system are typically analyzed statistically in order to make conclusions regarding the behaviour of the system. An example of this type is a telephony network simulation. In the other side, a deterministic simulation system is the one that has no random behaviour. Thus, the results of simulation for a given set of inputs will always be identical. An example of this type is circuit simulation systems.

4.1.4- Simulation Models:

Another division of simulator is the one that depends on the simulation models. Simulators are designed using either continuous or discrete event techniques to simulate a given system.
4.1.4.1- Continuous Simulation:

The continuous simulators are those characterized by the extensive use of mathematical formulae which describe the response of a simulated component with various conditions. But, it is not suitable in case of huge number of interconnected elements because of its intensive computation and very complex mathematical equations used. In such a case, continuous simulators may be slow and only useful in case of simulating a small number of components which are described at a low level of abstraction.

4.1.4.2- Discrete Simulation:

Discrete event simulation [10] is used to simulate components that normally operate at a higher level of abstraction than components simulated by continuous simulators. Here, the event defined by the incident which causes the system to change its state. The main different between the discrete event simulation and continuous event simulation is that in a discrete event simulator events can occur only during a distinct unit of time and they are not permitted to occur in between time units during the simulation. Discrete event simulation is generally more preferable than continuous simulation because it is usually faster than continuous simulation and it is also has more accurate approximations.

4.1.4.3- Monte Carlo Simulation:

Monte Carlo simulator works by generating random number which is then used to simulate the desired system. It lightly defferes from the discrete simulors by the property of using probability and nondeterminism plays a major role to form the best model of systems. As such, Monte Carlo simulators are commonly used to model stochastic systems whereby the discrete simulators are used to model deterministic systems. The relationship between simulations models is displayed in Figure2.

![Figure 2: Simulation Models](image)

Hierarchical simulation, although not a simulation type by itself, may be used in conjunction with continuous or discrete even simulators to simplify the simulation process. Hierarchical simulation is a process whereby higher order components delegate behavioural responsibility to its composite subcomponents. The higher
level components are responsible for activating their respective child components in a meaningful sequence so as to model the correct behaviour of the system. This is used to cope with the complexity associated with a given system. [11, 12].

4. 2- Remote Virtual Labs:

In the remote virtual lab the user interacts with equipment that is actually in another place using sensors and actuators, a computer, and a communication network. Remote control of laboratory experiments is a new concept which avoids the disadvantage of inconvenience experiment integration used in distance education laboratory. The term remote access refers to the way by which a user can access a special computer on a network from a remote location. Remote access connections often limit the types of tasks the user can perform with the computer. An extension of remote access is the remote experimentation which is the action of performing experiments from a remote location this allows the user to interact with the physical world instruments. This is effected by electronic control and monitoring systems that located external to the computer but controlled by it.

4.2.1- Advantages of Remote Lab:

The remote labs have many advantages some of them are mentioned here. It allows the students to login and carry out experiments from any place of the world i.e. reduce the limit distance. The remote labs also provide extended access to expensive and/or highly specialized devices. The remote labs provide real lab experience and give the students a more powerful experimentation capability the one that is given from simulations. Beside, the remote labs give students the opportunity to work in the remote mode, which will eventually become important in engineering. An amazing advantage of the remote experimentation is that unique or expensive equipment can be shared between different universities. Since a wider range of laboratory resources can be made accessible, the students have the choice between more experiments. An example of remote lab is the Bochum university laboratory, which is used in order to set up a prototype experimental environment. The students will have access via Internet to various experiments in control engineering, which are physically situated in the control laboratories. Computers in the Virtual Lab are then connected to engineering test equipment like oscilloscopes and function generators. When students log in to these machines over the internet, they are able to control both the computer and the equipment. A video camera can also be used to see what happen in the real world. It does not matter if the student is in a nearby dorm room or on the other side of the world. Remote control and reconfiguration of instrumentation will become an increasingly common event in the workplace. [13, 14].
5- Approaches For Constructing Virtual Labs:

There are many approaches by which virtual labs can be constructed. Available approaches for the design of virtual labs are: the pure HTTP approach, the mobile agent, and the telnet approach. The next section presents these three types in details.

5.1- Pure HTTP Approach:

This is the approach that uses the web and the web browsers. It uses the web for connection and the web browsers for interface. It is very useful approach because of the locations of both user and server machines are transparent. Also, it provides an unlimited access time for the user such that it can be available for the whole 24 hours. Let us now discuss the terminology and methods used in this approach. The HTTP term is stands for Hyper Text Transfer Protocol and it is the set of rules that control the transfer of files. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols which is the basic protocols for the Internet. Thus, when the web user opens his/her web browser, the user is indirectly make use of HTTP. It is an application-level protocol for distributed, collaborative, hypermedia information systems. Thus, it is widely used in the World-Wide Web (WWW). There are various document format provided by the WWW such as text, graphic images, sound, video, and other multimedia files. The function of HTTP is to define how messages are formatted and transmitted, and what actions web servers and browsers should take as a response to various commands. For example, when the user write a Uniform Resource Location URL in browser address bar and hit enter, an HTTP command is sent to the web server directing it which is then fetches and transmits the requested web page. HTTP is also used as a generic protocol that communicates between user agents and proxies/gateways to other internet systems. In this way, HTTP allows basic hypermedia access to resources available from diverse applications. The HTTP protocol is then a request/response protocol. There is another main standard which controls how the World Wide Web works and it is known as hypertext metalanguage (HTML). It gives the way by which web pages are formatted and displayed. HTTP is called a stateless protocol because each command is executed independently from the previous commands. For this reason there are some difficulties for implementing web sites that react intelligently to user input such as the virtual lab systems. But, this backwardness of HTTP is being addressed in a number of new technologies such as ActiveX, Java, JavaScript and cookies. With this approach it is often desirable to design a general purpose, distance experimental system with efficient interactive/multi-user operations. It also preferable to have an independent Machine/platform and secure operations. Beside, a compatible graphical user interface, and a high processing bandwidth with true real-time capabilities are needed. The web based virtual labs use user authorization if needed using the user ID and password. This is always done by the administrator who is responsible for adding all users with there corresponding passwords.
and IDs. However, there are some disadvantages for the web based approach. The approach is not suitable for the experiments that need long processing. It also has a weak security reside with the use of the internet. Moreover, it has a limited number of collaborative groups of users. Besides, this approach, as mentioned previously, has the stateless nature and thus it doesn't support perfect real time application because of the retrieval delay in the web which is depends on the network latency [7, 15, 16].

5. 2- Mobile Agent Approach:

The mobile agent can be defined as a piece of computer software, typically written in script language, which has the ability of transports itself from one computer to another independently and executes on the destination computer. It is a powerful tool for implementing distributed applications in a computer network because it is not bounded to the host where it begins execution. A mobile agent moves to a host that contains an object within which the agent wants to interact, and thus it takes advantage of being in the same host as the object. The amazing property of the mobile agents is that they can operate across both dynamic and heterogeneous systems and networks. Figure 3 describes the main structure for the mobile agent system:

![Figure 3: Mobile agent Paradigm](image)

5. 2. 1- Advantages of Mobile Agent Approach:

The mobile agent reduces the network load because of the decreased number of network resources used [17]. It also overcomes network latency, because it carries most of its required resources with it [18]. This is very useful in case of critical real time systems such as robots in manufacturing processes need to respond to changes in their environments in real time. Besides, it encapsulates protocols such as that evolve to accommodate new efficiency or security requirements. Mobile agents are able to move to remote hosts in order to establish communication channels based on proprietary protocols. Moreover mobile agent is very
flexible and can execute either asynchronously or autonomously. Another benefit is that the mobile agents dynamically adapt because they have the ability to sense their execution environment and then react to changes. Adding to those, the ability of multiple mobile agents to distribute themselves among the hosts in the network in such a way as to maintain the optimal configuration (for solving a particular problem). They are robust and fault-tolerant since they have the ability to react dynamically to undesirable situations and events which makes it easier to build robust and fault tolerant distributed systems. Also, if a host is being shut down, all agents executing on that machine will be warned and given time to dispatch and continue their operation on another host in the network. Sometimes, it is more efficient to use the mobile agent approach than the web approach because of the limitation that minimize the web based approach such as its long execution time, and the problem of data loss in case of server failure.

5.2.2 Disadvantages of Mobile Agent:

Unfortunately, mobile agent technique has also some disadvantages as it suffers from security problem since it is an open system. Also, it needs a dedicated mobile agents environments to be existed in every host that mobile agent will visit. Beside, it has the problem of delay when there is a widespread access at least outside the education and research institutions.

The mobile agent approach has many advantages over the web approach. For example in the web approach if the server is down for any reason, the user will have to wait for the server to restart which results on a huge waste of time. Where, in mobile agent system the mobile agent will be stored until the target machine is available. Duplication or lost mobile agent could not happen. In web based virtual labs the access to the lab must be done while the user is online while in mobile agent the user does not have to be online during the lab session to perform the experiment and if he is offline, the mobile application will stay in the system until the user get back online. Also, we can let the user specifies the time that the experiment will take place and the time he wants mobile application to travel to the destination and the time to get the result. Another advantage is that in the web based approach if the server fails for any reason, all of the steps in the session should be done all over again. This is not occurring in the mobile agent. Moreover, the mobile agent is more flexible for switching from one lab to another in case of multi virtual lab, having the same set of equipment and setup, since it must be done manually by the user. While in mobile agent technique if more than one identical virtual lab are provided then the mobile application can be easily designed to interacts with any one of them either transparent or not to the user. For a long time execution the web based approach is not suitable since the connection between the user browser and the server will be disconnected after reset time. Also, the problem of long server processing time does not exist in the mobile agent approach [19].

5.3 Telnet Approach:

Telnet stands for Telecommunications Network; it is a protocol that provides a connection from the
clients to servers over the Internet [4]. In most cases, Telnet is used to communicate with a remote login service since it provides a full duplex connection between the user's terminal and a remote service. Thus Telnet client performs two tasks simultaneously. The client must read characters that the user types on the keyboard and send them across a TCP connection to the remote service and also must read characters that arrive from the TCP connection and display them on the user's terminal screen. On the Internet, the ability to connect with another machine is made possible by the Transmission Control Protocol (TCP), which enables two machines to transmit data, and the Internet Protocol (IP), which provides a unique address for each machine connected to the network. Telnet is used under TCP/IP networks. The telecommunications application built over these capabilities provides the local terminal with the means to emulate a terminal compatible with the remote computer. The Telnet TCP connection is established between the users port and the server port and it is maintained for the duration of the Telnet session, which can remain alive for hours, days, or even weeks at a time. Since a TCP connection is full-duplex and identified by the pair of ports, the server can engage in many simultaneous connections involving its port and different user ports. The Telnet program works as follow the program runs on the computer and connects it to a server on the network. Then, the commands are entered by key presses through the Telnet program and transmitted to the host computer which processes the event. They will be executed as if they were entered directly on the server console. These commands are then transmitted directly to the remote machine and the response from the remote machine is displayed on the users monitor screen. An interactive connection is also known as remote login. In order to remote login the users computer must have the ability to establish a connection to another machine, emulate a terminal compatible with the remote machine, regulate the flow of data from the users terminal to remote machine, and vice versa. This allows the control of the server and also enables the communication with other servers on the network. The Telnet session, is protected by authorization information and it does not start unless a valid user name and password are entered. The Telnet is a common way to remotely control web servers thus it can be defined as the way by which you access someone else's computer and it is often used as a mean of configuring a device over the network by remotely inter to device and change its configuration. More technically, Telnet is a user command and an underlying TCP/IP protocol for accessing remote computers. This differs from the web protocols, which give the ability for just request specific files from remote computers, because it gives the ability of actually be logged on as a user of this specific computer. Telnet is most commonly used by program developers and in case of using specific applications or data located at a particular host computer. Figure 4 illustrates the basic structure of the Telnet and describes the path of data in a Telnet remote terminal session as it travels from the user's keyboard to the remote operating system.
In order to use the Telnet and open a telnet session the knowledge of telnet address of the remote computer and the login procedure, if available, is required. Any computer in a network have an address, which is a number that distinguishes it from other computers. Every Telnet site has two addresses one composed of alphabetic, that are easier for people to remember, and the other one is numerical address which is better suited for computers. An example of alphabetic domain name is (infogate.lib.utah.edu) and for the numeric IP address is (128.110.40.179). Once connected, you will require using the commands of the remote site. These connections will be text based i.e. use the keyboard for entering data.

5.3.1 Advantages of Telnet:

The main advantages of Telnet are: The Telnet is very easy to use and it allows you to access databases, library catalogs, and other information resources over the Internet. It also can provide errors free connection. The telnet protocol gives the user the ability to connect to a machine, by giving commands and instructions interactively to that machine, thus creating an interactive connection. In such a case, the local system becomes transparent to the user, who gets the feeling that he is connected directly to the remote computer. For more convenient results, the TCP guarantee that data is received reliably and in order, and ensures that data is not sent at too high rate for either client or server. Thus a machine offering Telnet service can support multiple simultaneous sessions with different users, keeping each distinct by identifying it using
the IP address and port number of the client. Add to these, the convenience of Telnet for poor line quality and remarkable stability.

5. 3. 2- Disadvantages of Telnet:

It has a main disadvantage that is it doesn't support graphics property. Beside, there are three main problems related to security that make Telnet a bad choice for modern systems. One of them is that Telnet does not encrypt any data send over the connection. The second one is that Telnet has some weakness in the authentication scheme that makes it possible to ensure that communication is carried out between the two desired hosts, and not intercepted in the middle. These two problems means that anybody who has access to any router, switch, or gateway located on the network between the two hosts where telnet is being used can intercept the telnet packets passing by and easily obtain login and password information. The third one is that many of used telenet daemons have several vulnerabilities discovered over the years and more been discovered with time. Thus, in environments where security is important Telnet should not be used. Thus at the coming time and with the exponential rise in the number of people access to the Internet and the number of people attempting to crack into other people's servers, Telnet should generally not ever be used on networks with Internet connectivity.

5. 3. 3- Telnet Concepts:

Telnet is based on three main principles. The concept of the Network Virtual Terminal (NVT) the concept of negotiations, and a symmetrical view of terminals and processes. These concepts illustrated briefly in the next session:

5. 3. 3.1- Network Virtual Terminal (NVT):

The Network Virtual Terminal is a character based device used by Telnet to enable a local computer to communicate with a remote machine. NVT can also be defined as a concept used by Telnet to represent both computers in a connection. So, it implements client-server architecture. A Telnet client transfers characters between the user terminal and a remote service. On one side, it uses the local operating system functions to interacts with the user terminal. while, it uses a TCP connection when it communicates with the remote service on the other side. The Telnet protocol defines the character set for the virtual terminal. The chief advantage of using a Network Virtual Terminal is that it permits clients from a variety of computers to connect to a service. The main objectives of the Network Virtual Terminal, NVT are: It enables Telnet to interoperate between heterogeneous systems. It defines how data and commands are transferred across the internet. Both the client and server translate data and commands from operating system format to NVT format, and vice versa. And thus they must support the concept of an NVT. Both ends of a Telnet connection map their own terminal device characteristics to and from the NVT.
5. 3. 3. 2- CONCEPT OF NEGOTIATIONS:

The option negotiation mechanism allows the user to set terminal parameters to values other than the default or to negotiate more sophisticated facilities. The Telnet protocol negotiation mechanism can be initiated by either side. Besides, negotiating whether or not a particular option is to be in effect. The mechanism also allows one to specify, when appropriate, which side is to perform the function.

5. 3. 3. 3- A SYMMETRICAL VIEW:

The third concept of Telnet is one of symmetry in the negotiation syntax. This symmetry allows either the client or server ends of the connection to request a particular option as required, thus optimizing the service provided by the other party. A terminal protocol should not only allow a terminal to interact with an application process on a host, but it should also allow process-process and terminal-terminal interactions [7, 20, 21, 22].

Beside, there are two other possible realizations of interactive virtual laboratories: using software with powerful programming development environment for data acquisition and control, data analysis, and data presentation. The second method is based on the Internet Server Application Programming Interface (ISAPI) extensions. The first method is adopted to demonstrate interactive laboratory where virtual instruments with front panel user interface that may contain numerical displays, meters, charts, and advance graphs are used. An example of this virtual instrument is LabVolts virtual instrument for electrical engineering. In the second method, advanced features of modern web browsers and servers can be utilized that allows data transmission in several suitable formats for presentation and analysis of virtual labs. With ISAPI filters more useful tasks related to security, queuing, and logging may be added. [23].
Chapter [3]  
Design & Analysis

The objective of this project is to design a virtual lab. In our project we use a simulated virtual lab with the HTTP, or web-based, approach and the used programming language is java language. In the design of our lab we choose some basic experiments from digital laboratory. There were many factors which lead us to use this approach and the java language for designing the lab. The simulation was the most logical choice for us since remote controlled labs need a very complex software and hardware connection. With our simulation the web based was the most reasonable approach.

1- Design Requirements:-

The reasons for which the HTTP approach and java language were used can be illustrated in the next section:

1.1- HTTP Approach:

Telnet approach was not used here because it does not support graphics, which is a main objective in our project. Telnet provides just a text interface while the HTTP has a very friendly interface. Then when we look to the mobile agent, the approach has friendly user interface but it is more complex and needs a higher degree of effort in development than the effort required by the other two approaches. Thus the better technique to be use is the web-based approach. But, unfortunately there are also some disadvantages that decreased the degree and efficiency for the virtual labs designed with web based approach such as the need for connection which is critical in HTTP approach while this is not necessary for the mobile agent approach. Also it does not support failure tolerance which is enabled in mobile agent and it also doesn't support long time processing. This approach has also many advantages such as the unlimited usage of the virtual lab. That is the lab was uploaded to work online using the internet then it can be used widely not only by the students but also for any external users. The lab was designed to be a digital virtual lab for its ease of implementation.

1.2- Java Language:

In our virtual digital lab the java programming language was used. It is more convenient because of its high power with graphics and user interface. Java also has many other perfect properties which lead us to use it. Java language has the ability to run over any platform using Java-capable browser that is relative to the independency of java applet from the underlying hardware and software and this is the reason for which java is preferable in the internet and web pages.
1.2.1- Java For Simulation: -

The Java programming language is an object-oriented language and it has many features that make it improves the process of computer simulation. It is constructed from classes, one or more, and this is a very interesting feature since the class may be shared by different programs. It also gives an infinite number of variables declarations since each class component is an object and thus it can be used later by any other part of the program. Java, as mentioned previously, is very flexible and efficient language for the implementation of user interface. It consists of special objects which are used with the user interface. The most significant feature of java language is its simple and straightforward approach with multi-threading programs. So, java is not just an underlying programming language for simulation but java is the simulation language. Just as a traditional programming language provides fundamental behaviors such as, do if start read write a Java-based simulation adds simulation-specific entity behaviors. All simulation components could be designed as individual classes and these classes share a common set, packet, and any one class can use and call any other class or shared function in this packet. Java's internal support for distributed network programming provides the remainder of the necessary infrastructure to achieve distributed simulation experiment processing without making the simulation dependent on the operating system or hardware platform. The ability to execute within the Internet browser is the feature of Java that will enable Java-based simulation to realize its long sought potential as an online demonstration and training tool.

1.2.2- Java For Programming Over The Network:

The java is a perfect language for using in the internet because the following features: java is a simple and familiar programming language, it looks quite similar to C and C++ languages, but with many of the elements which sometimes prove troublesome (pointers, memory deal location) removed. Object-oriented: a program created in Java is essentially the weaving together of various objects (mixtures of code and data) which are instantiations of various "classes. Because it is an interpreted language, it requires only an interpreter or just in time compiler otherwise known as the java Virtual Machine VM in order to run. So, as long as you have the VM for the machine you want to run it on, then all Java programs should theoretically run properly. Because of the above the code is written once and then it should work on all machines. Java is a distributed language i.e.: a program can create objects from classes stored on different web servers and only download the ones appropriate for a particular group of actions at a particular time. Besides, java is as secure language that the whole layer of Java's architecture is devoted to security, so that it formalizes the way to deal with possible security problems (since you are allowing a program to run on your machine from an internet site which may cause infection by viruses). The java foundation classes are made up of several technologies: AWT (Abstract Windowing Toolkit), swing, accessibility and Java 2D. These technologies are the core of Java’s user interface support. The swing component extends the original AWT by adding a comprehensive
set of graphical interfaces class libraries that is completely portable and it is delivered as part of the java platform. The swing components improve the GUI development providing high quality GUI components which are peerless or lightweight, look and feel pluggable, customizable and transparent. From the design point of view Web-based simulations can be categories into: simulations programs that can be accessed remotely through the web browser on the server, those which are downloaded from servers and run on the client machine, and those which show web-based execution on the client side (such as java applets which is a self launching application that runs in a web browsers and publication). Java applets can enable many actions when attached to web pages as: expert graphics calling, real-time interaction with users, live information updating, and instant interaction with servers over the network. Java applets are downloaded from any server and run safely on any platform. They provide internet sites with a huge number of functionality such as: animation, live updating, and two-way interaction. Java applets can be embedded right into HTML pages. To access an applet, the browser needs a special tag that is written in the HTML page. This tag tells the browser the name of the applet, its parameters, and everything other values that the browser needs to know to run the applet. When the browser encounters an applet tag in a HTML page, it requests the applet code (the <applet>.class file) from the server then the browser retrieves this binary form of the applet and executes it with the associated parameters. Thus, all the GUI intelligence that resides on the server is loaded on demand by the browser. There are some great advantageous for java such as: platform independency, client side execution, and the powerful programming language. But, java is an emerging technology, a general-purpose programming language, not a scientific or engineering tool. This is why there are no facilities related to the scientific calculations, algorithm implementations or other simulations. The only real way to achieve this is to write a dedicated program that interacts with the user on one end and communicates with the server on the other end [24].

2- Design Procedure: -

As said previously the simulation was made for the digital lab. It is the better from the programming point of view since it often use an identical value, binary values zero and one. The lab was design by simulating the basic element and circuit that illustrate digital design. Java programming language is used. Codes were then compiled and tested and they give good results. Each compiled code, which represents an experiment, was made in a different page the associated applet tag for each compiled program( called the class) were inserted in the HTML code to enable the browser to open the desired simulation. The site was made using front page, HTML editor, software. Each page, actually each experiment, was preceded by a brief description of the associated digital circuit its function, implementation, its truth table, and one or more possible design. The experiments that we used in the virtual lab design were: the Basic logic gate, the half adder circuit, Full adder, the Decoder, Multiplexer (MUX), Demultiplexer, and Seven Segment Display. The constructed experiments were uploaded firstly to the Local Area Network LAN, using server software for
serving requests. Here the student can enter the lab using internet browser and write the IP address for the server which was: 172.16.14.238 but the user could not retrieve the site unless the server program, we use Abyss web server, is set to RUN. The next step was done by loading the site to World Wide Web WWW. This done by registering in a site, 50megs.com, which give us a 5o mega site. Then we upload our HTML pages and the classes and all needed pictures. These pages are found in the following link: http://www.divlab.50megs.com.

The page can be assessed by any one it does not need user name or password. When the user visits the site he/she will find the home page which contains hyperlinks for all of the available labs and theory. To use this lab the user has to download the java environment in his computer in order to see the simulations which its source is found in the net for free.

The code for each experiment was written, compiled and the uploaded to the web page, HTML page, as mentioned previously. The main problem that we found was how to let the action that the user take to be seen by the program and to let it then deals with it. Then the solution found from the java action listener objects and action performed function and the program work properly. The code for each one of these simulated virtual labs was written using the flow chart in figure 6:

![Flow chart for simulation programs](image)

**Figure 5: Flow chart for simulation programs**
Chapter [4]

Results and Discussion

The programs were compiled and then executed in the HTML page. The program works as follow: (for example take the half adder experiment) when the classes are full loaded to the page the user see a display for one possible design for the half adder with its two inputs being in the zero state and the zero corresponding outputs. The half adder has two output, Sum and Carry, which are displayed by using LED where the zero case represented by LED with dark red, OFF state, and shine red color for the ON case or 1 volt case. Beside, there are two buttons for each one of the inputs. The user can select the desired combination of the inputs, two in case of half adder, by selecting the binary value related to each button. The output is then can be detected from the state of LED. By this the user can learn how the circuit works and what is the output for each case and this will increase his/her skill. Recall that all programs were made to work in HTML pages and each page, which represents one experiment, contains link for the remaining pages and experiments for more convenient. There is also a brief description for each circuit or logic gate and its function. The truth table for each design was also included in the site. The display of each experiment's web page is shown below.

1- Basic Logic Gates Experiment:

This simulates the main basic logic gates which are: AND gate, OR, XOR, and NOT or inverter all in one applet. Here we use choice boxes for input selection. The program run exactly as expected and gives results similar to theoretical values. The output from any gate is independent of the other gates. The AND gate gives an output equal to 1, LED being in the ON state, if the two inputs were equal to one. In any other case the LED goes OFF. For OR gate there is one case for the zero, OFF state, which appears when both inputs are equal to one where in the other cases the LED has a shine red color, i.e. the LED is ON. For the XOR gate the output is displayed by OFF state for the LED when the two inputs are equal i.e. they equal 11 or 00. The inverter gate has one input and when we entered zero it gives one and the LED turns on and when its input is 0 the output was 1. The display of this applet and some combination of inputs and corresponding outputs are shown in figure 6.

2- Half Adder Experiment:

This has two inputs and two outputs and when we select the input combination the output is then displayed by LEDs for each one of the two outputs. The Sum output represents the summation of the two inputs and it equals 1, i.e. the LED is ON, if any one (but not both) of
the input is equal one. The other output, the carry, is ON only when both of the inputs are equal 1. One of these results is shown in figure 7.

3- The Full Adder experiment:
This is similar to the half adder it just differs from it by the number of inputs. Here there are three inputs. When it is full loaded to the page the applet is displayed with a zero inputs and zero corresponding outputs. A comment is then appears to tell the user that he/she should use the mouse click for selecting the inputs values. When the inputs are selected then the outputs are automatically displayed. Figure 8 shows one of the possible inputs and the corresponding outputs.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>Sum</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 8: FullAdder applet

4- Decoder Experiment:

Here a 3-8 decoder, which has 3 inputs and 8 output line, was simulated as example for the decoder circuits. Outputs from the decoder are retrieving correctly after the On radio button is selected. The selected output line is detected by a LED with ON state. There was only one input active at any time. The output from the decoder is directly proportional to the input for example if the user selects the combination: 011, which is 3 in decimal, the output will come at the output line D3. This is shown in the figure 9.

5- Multiplexer Experiment:

The results of this applet were displayed as follows: the user selects one of the available combinations for the input then the output is displayed. The output is the value of the selected line, either A or B. An example of these results is displayed in figure 10.

6- DeMultiplexer Experiment:

Here the one LED goes turns on when the input data, IN, equals to one and it is displayed in the address line relative to the inputs. There was only one LED ON at any time and this happen when the input IN is one. For example, when the IN input is 1 and A equals 1 and B equals 0 the output is displayed by D2 LED ON and all the other LEDs are OFF. This can be seen from figure 11.
Figure 9: The 3-8 decoder simulation

Figure 10: One possible result from the Mux applet
7-Seven Segment Display:

This works as follow: when On button is selected the circuit displays the input data in a decimal form for example if the input was 0100, i.e. decimal 4, the output is then displayed as the decimal number four. This is Shown in figure 12.

Figure 12: Preview of Sevensegments applet
Chapter 5
Conclusion

E-learning and virtual universities are effective recently increased technologies that largely improve the education. This leads to more and more chances for the people to learn and also more economical cost reduction.

The improvement of virtual universities and e-learning requires that all the equipments of real world universities be available. One of the most important of these equipments, specially in engineering, is the laboratories. The term virtual laboratory is given for such a system that gives the same behavior of real labs in the virtual realities. Virtual laboratories let the users, students, able to do the experiment from there places.

Virtual laboratory is a very effective technology which drastically reduces the cost of infrastructure used and it also imposes the barrier of cost and distance rather than many other benefits that they have.

Virtual laboratories can either be simulated or remote controlled one. There are many approaches and technologies are available to be used for design and development of virtual laboratories. They are: web-based or HTTP approach, mobile agent approach, and Telnet approach. The advantages and disadvantages and other relative topics were discussed in this project research.

In this project a design for virtual laboratory was made. The simulation was done for the digital lab. This is mainly for students of electrical and electronics engineering faculties. The language used for the simulation was java language. Analysis and design requirement for this laboratory was made and been discussed in this research. The digital virtual lab works properly and gives the expected values.

The simulated lab consists of seven experiment and it runs by the web browsers that enables java. The results from this lab were good and similar to actual results and thus it can be used for educational purposes. The lab was uploaded to the internet and to the LAN of university of Khartoum.

Future Extension:

More digital experiments can be added to this lab. Also, the other labs related to electrical and electronics engineering courses can be added as: electronic and telecommunication labs to make all laboratories available in a virtual form the thing that can be used as a preparation for students before they will enter the real laboratory.

Effects of noise and other errors was not taken into account in our simulation. Thus, for more effective one remote controlled laboratory can be added to let the user feel the actual behavior of instruments and deal with the different types of error.
REFERENCES


[4] K. C. Chu "What are the benefits of a virtual laboratory for student learning?" Department of Electrical & Communications Engineering Hong Kong Institute of Vocational Education (Tsing Yi)

[5] George S. Georgiev, Hubert Roth, Silvia Stefanova, Georgi T. Georgiev, Emil Stoyanov & Otto Rösch "How and why to build and use virtual laboratories" Rousse University, Rousse, Bulgaria University of Siegen, Siegen, Germany

[6] Giuseppe Carnevali and Giorgio Buttazzo "A VIRTUAL LABORATORY ENVIRONMENT FOR REAL-TIME EXPERIMENTS" University of Pavia, Italy

[7] Saleh A.Al-Shehri, Mohmmed Aloqeely "Design And Implementation Of Virtual Labs" Department of computer engineering college of computer and information sciences, king Saud University, Riyadh, Saudi Arabia


http://www.ece.cmu.edu/~stancil/virtual-lab/concept.html


[15] Venkata N. Padmanabhan (University of California -- Berkeley) and Jeffrey C. Mogul (Digital Equipment Corporation Western Research Laboratory), "Improving HTTP Latency".
http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/mogul/HTTPLatency.html

[16] "Hypertext Transfer Protocol -- HTTP/1.1" http://www.w3.org/protocols/rfc2616/rfc2616.html


[23] Sheikh, Sharif Iqbal, Junaid Siddiqui. Assistant professor, Electrical Engineering Department, KFUPM. "Web Based Engineering Education" Lecture, ADC, KFUPM.

Appendix

The codes for each experiment's simulation had been attached in this section:

1-Basic gates code:

```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;
import java.applet.*;
public class Digital extends JApplet
{
    // Variables declaration
    private JLabel jLabel1;
    private JLabel jLabel3;
    private JLabel jLabel4;
    private JLabel jLabel5;
    private JComboBox jComboBox1;
    private JComboBox jComboBox2;
    private JComboBox jComboBox3;
    private JComboBox jComboBox4;
    private JLabel xorcct;
    private JLabel xorled;
    private JLabel not;
    private JLabel notled;
    private JComboBox xorb;
    private JComboBox xora;
    private JComboBox nota;
    private JPanel contentPane;
    private int anda,andb,ora,orb,xoraa,xorbba,noa;
    private ImageIcon andicon,ledoficon,ledonicon,oricon,xoricon,noticon;
    // End of variables declaration
    public Digital()
    {
        super();
        initializeComponent();
        this.setVisible(true);
    }
    private void initializeComponent()
    {
        jLabel1 = new JLabel();
        jLabel3 = new JLabel();
        jLabel4 = new JLabel();
        jLabel5 = new JLabel();
        jComboBox1 = new JComboBox();
        jComboBox2 = new JComboBox();
        jComboBox3 = new JComboBox();
        jComboBox4 = new JComboBox();
        xorcct = new JLabel();
        xorled = new JLabel();
        not = new JLabel();
        notled = new JLabel();
        // Variables initialization
        // Code body...
    }
}
```
xorb = new JComboBox();
xora = new JComboBox();
nota = new JComboBox();
anda=0;andb=0;ora=0;orb=0;xoraa=0;xorbb=0;noa=0;
contentPane = (JPanel)this.getContentPane();
// jComboBox1
jComboBox1.addItem("0");
jComboBox1.addItem("1");
jComboBox1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox1_actionPerformed(e);
    }
});
// jComboBox2
jComboBox2.addItem("0");
jComboBox2.addItem("1");
jComboBox2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox2_actionPerformed(e);
    }
});
// jComboBox3
jComboBox3.addItem("0");
jComboBox3.addItem("1");
jComboBox3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox3_actionPerformed(e);
    }
});
// jComboBox4
jComboBox4.addItem("0");
jComboBox4.addItem("1");
jComboBox4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox4_actionPerformed(e);
    }
});
// xorb
xorb.addItem("0");
xorb.addItem("1");
xorb.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        xorb_actionPerformed(e);
    }
});
// xora
xora.addItem("0");
xora.addItem("1");
xora.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        xora_actionPerformed(e);
    }
});
xora_actionPerformed(e);
}
});
// nota
nota.addItem("0");
nota.addItem("1");
nota.addActionListener(new ActionListener() {
publi...
private void jComboBox3_actionPerformed(ActionEvent e)
{
    System.out.println("jComboBox3_actionPerformed(ActionEvent e) called.");
    Object o = jComboBox3.getSelectedItem();
    System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");
    if(o=="0")
        ora=0;
    else ora=1;
}

private void jComboBox4_actionPerformed(ActionEvent e)
{
    System.out.println("jComboBox4_actionPerformed(ActionEvent e) called.");
    Object o = jComboBox4.getSelectedItem();
    System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");
    if(o=="0")
        orb=0;
    else orb=1;
}

private void xorb_actionPerformed(ActionEvent e)
{
    System.out.println("xorb_actionPerformed(ActionEvent e) called.");
    Object o = xorb.getSelectedItem();
    System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");
    if(o=="0")
        xorbb=0;
    else xorbb=1;
}

private void xora_actionPerformed(ActionEvent e)
{
    System.out.println("xora_actionPerformed(ActionEvent e) called.");
    Object o = xora.getSelectedItem();
    System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");
    if(o=="0")
        xoraa=0;
    else xoraa=1;
}

private void nota_actionPerformed(ActionEvent e)
{
    System.out.println("nota_actionPerformed(ActionEvent e) called.");
    Object o = nota.getSelectedItem();
    System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");
    if(o=="0")
        noa=0;
    else noa=1;
}

// The following method controls circuit behaviour:
public void paint(Graphics g)
{
    super.paint(g);
    andicon = new ImageIcon(getImage(getDocumentBase(),"Andcct2.JPG");
    ledoficon= new ImageIcon(getImage(getDocumentBase(),"tafia.JPG");
    ledonicon= new ImageIcon(getImage(getDocumentBase(),"fatha.JPG");
    oricon= new ImageIcon(getImage(getDocumentBase(),"Or.JPG");
    xoricon= new ImageIcon(getImage(getDocumentBase(),"Xor.JPG");
    noticon= new ImageIcon(getImage(getDocumentBase(),"Invert.JPG");
jLabel1.setIcon(andicon);
jLabel3.setIcon(ledoficon);
jLabel4.setIcon(oricon);
jLabel5.setIcon(ledoficon);
xorct.setIcon(xoricon);
xorled.setIcon(ledoficon);
not.setIcon(noticon);
notled.setIcon(ledoficon);

//And gate:
if ((anda==0)&&(andb==0))
{
jLabel3.setIcon(ledoficon);
}
if ((anda==0)&&(andb==1))
{
jLabel3.setIcon(ledoficon);
}
if((anda==1)&&(andb==0))
{
jLabel3.setIcon(ledoficon);
}
if((anda==1)&&(andb==1))
{
jLabel3.setIcon(ledonicon);
}

//Or gate
if ((ora==0)&&(orb==0))
{
jLabel5.setIcon(ledoficon);
}
if ((ora==0)&&(orb==1))
{
jLabel5.setIcon(ledonicon);
}
if((ora==1)&&(orb==0))
{
jLabel5.setIcon(ledonicon);
}
if((ora==1)&&(orb==1))
{
jLabel5.setIcon(ledonicon);
}

//Xor gate
if ((xoraa==0)&&(xorbb==0))
{
xorled.setIcon(ledoficon);
}
if ((xoraa==0)&&(xorbb==1))
{
xorled.setIcon(ledonicon);
}
if((xoraa==1)&&(xorbb==0))
{
xorled.setIcon(ledonicon);
}
if((xoraa==1)&&(xorbb==1))
{
xorled.setIcon(ledonicon);
}
40

// Inverter gate
if (noa==0)
{
 notled.setIcon(ledonicon);
}
if(noa==1)
{
 notled.setIcon(ledonicon);
}
} repaint();
} } 
// End of Digital.

2-Half Adder code:

import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.applet.*;
import java.swing.*;

public class HalfAdder extends JApplet {

// Variables declaration
private JLabel cctlabel;
private JLabel sumled;
private JLabel carryled;
private JRadioButton frstzero;
private JRadioButton frstone;
private JRadioButton scndzero;
private JRadioButton scndone;
private JPanel contentPane;
private ButtonGroup frstinput;
private ButtonGroup scndinput;
private int a,b;
private ImageIcon ccticon,ledicon,ledonicon ;
// End of variables declaration
public HalfAdder()
{
 super();
 initializeComponent();
 this.setVisible(true);
}
private void initializeComponent()
{
cctlabel = new JLabel(ccticon);
sumled = new JLabel(ledicon);
carryled = new JLabel(ledicon);
frstzero = new JRadioButton();
frstone = new JRadioButton();
scndzero = new JRadioButton();
scndone = new JRadioButton();
contentPane = (JPanel)this.getContentPane();
frstinput = new ButtonGroup();
scndinput = new ButtonGroup();
a=0;b=0;
// cctable
cctable.setHorizontalTextPosition(SwingConstants.CENTER);
cctable.setBackground(new Color(255, 255, 255));
cctable.setForeground(new Color(51, 51, 0));
//cctable.setIcon(ccticon);
cctable.setIconTextGap(5);
cctable.setText("half adder circuit");
cctable.setVerticalTextPosition(SwingConstants.BOTTOM);
cctable.setOpaque(true);
cctable.setToolTipText("One possible design for Half adder circuit");
cctable.setBackground(new Color(255, 255, 255));
sumled.setOpaque(true);
sumled.setToolTipText("indicator of the sum output");
sumled.setBackground(new Color(255, 255, 255));
//sumled.setIcon(ledicon);
carryled.setOpaque(true);
carryled.setToolTipText("indicator for carry output");
carryled.setBackground(new Color(255, 255, 255));
//carryled.setIcon(ledicon);
// frstzero
frstzero.setBackground(new Color(255, 255, 255));
frstzero.setForeground(new Color(95, 95, 95));
frstzero.setText("A=0");
frstzero.setSelected(true);
frstzero.setToolTipText("click to select 1 or 0");
frstzero.addItemListener(new ItemListener() { public void itemStateChanged(ItemEvent e) { frstzero_itemStateChanged(e); }});
// frstone
frstone.setBackground(new Color(255, 255, 255));
frstone.setForeground(new Color(95, 95, 95));
frstone.setText("A=1");
frstone.setSelected(true);
frstone.setToolTipText("click to select 1 or 0");
frstone.addItemListener(new ItemListener() { public void itemStateChanged(ItemEvent e) { frstone_itemStateChanged(e); }});
// scndzero
scndzero.setBackground(new Color(255, 255, 255));
scndzero.setForeground(new Color(95, 95, 95));
scndzero.setText("B=0");
scndzero.setSelected(true);
scndzero.setToolTipText("click to select either 1 or 0");
scndzero.addItemListener(new ItemListener() { public void itemStateChanged(ItemEvent e) { scndzero_itemStateChanged(e); }});
scndone.setBackground(new Color(255,255,255));
scndone.setForeground(new Color(95, 95, 95));
scndone.setText("B=1");
scndone.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        scndone_itemStateChanged(e);
    }
});
// contentPane
contentPane.setLayout(null);
contentPane.setBackground(new Color(255, 255, 255));
addComponent(contentPane, crctable, 45, 75, 286, 231);
addComponent(contentPane, sumled, 330, 59, 57, 82);
addComponent(contentPane, carryled, 331, 159, 57, 82);
addComponent(contentPane, frstzero, 0, 95, 48, 18);
addComponent(contentPane, frstone, 0, 117, 48, 18);
addComponent(contentPane, scndzero, 0, 168, 48, 21);
addComponent(contentPane, scndone, 0, 191, 48, 19);
// frstinput
frstinput.add(frstzero);
frstinput.add(frstone);
// scndinput
scndinput.add(scndzero);
scndinput.add(scndone);
private void addComponent(Container container, Component c, int x, int y, int width, int height) {
    c.setBounds(x, y, width, height);
    container.add(c);
}
private void frstzero_itemStateChanged(ItemEvent e) {
    a=0;
    // set the value to 0 for the first input
}
private void frstone_itemStateChanged(ItemEvent e) {
    a=1;
    // set the value to 1 for the first input
}
private void scndzero_itemStateChanged(ItemEvent e) {
    b=0;
    // set b value to 0
}
private void scndone_itemStateChanged(ItemEvent e) {
    b=1;
    // set b value to 1
}
// The following method controls circuit behaviour :
public void paint(Graphics g)
{ super.paint(g);
ccticon = new ImageIcon(getImage(getDocumentBase(),"circuit.gif"));
ledicon = new ImageIcon(getImage(getDocumentBase(),"off.gif"));
ledonicon = new ImageIcon(getImage(getDocumentBase(),"cone.gif"));
cretlable.setIcon(ccticon);
if ((a==0)&&(b==0))
{
  sumled.setIcon(ledicon);
carryled.setIcon(ledicon);
}
if ((a==0)&&(b==1))
{
  sumled.setIcon(ledonicon);
carryled.setIcon(ledicon);
}
if((a==1)&&(b==0))
{
  sumled.setIcon(ledonicon);
carryled.setIcon(ledicon);
}
if((a==1)&&(b==1))
{
  sumled.setIcon(ledicon);
carryled.setIcon(ledonicon);
}
repaint();
}
// End of Half Adder code.

3-FullAdder code:

import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.applet.*;
import javax.swing.*;
public class FullAdder extends JApplet
{
  // Variables declaration
  private JLabel addercircuit;
  private JLabel sumled;
  private JLabel carryled;
  private JRadioButton zeroa;
  private JRadioButton onea;
  private JRadioButton zerob;
  private JRadioButton oneb;
  private JRadioButton zeroc;
  private JRadioButton onec;
  private JLabel hint;
  private JPanel simulationarea;
  private ButtonGroup frstin;
  private ButtonGroup scndin;
  private ButtonGroup thrdin;
  private ImageIcon ccticon,ledicon,ledonicon,zeroicon,zeronicon,oneicon,oneonicon ;
private String cct = new String("fulladr.jpg");
private String led = new String("of.jpg");
private String ledon = new String("on.JPG");
private String zero = new String("ovolt.gif");
private String zeron = new String("ovoltslected.gif");
private String one = new String("fivevlt.gif");
private String oneon = new String("fivevltselected.gif");
private int a,b,c;
// End of variables declaration
public FullAdder()
{
  super();
  initializeComponent();
  this.setVisible(true);
}
private void initializeComponent()
{
  addercircuit = new JLabel(ccticon);
  sumled = new JLabel(ledicon);
  carryled = new JLabel(ledicon);
  zeroa = new JRadioButton();
  onea = new JRadioButton();
  zerob = new JRadioButton();
  oneb = new JRadioButton();
  zeroc = new JRadioButton();
  onec = new JRadioButton();
  hint = new JLabel();
  simulationarea = (JPanel)this.getContentPane();
  frstin = new ButtonGroup();
  scndin = new ButtonGroup();
  thrdin = new ButtonGroup();
  a=0;
  b=0;
  c=0;
  addercircuit.setToolTipText("One possible design for full adder circuit");
  sumled.setToolTipText("this led show the sum output");
  carryled.setToolTipText("this led show the carry output");
  //zeroa
  zeroa.setBackground(new Color(255, 255, 255));
  //zeroa.setText("A=0");
  zeroa.setSelected(true);
  zeroa.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
      zeroa_itemStateChanged(e);
    }
  });
  // onea
  onea.setBackground(new Color(255, 255, 255));
  //onea.setText("A=1");
  onea.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
      onea_itemStateChanged(e);
    }
  });
}
zerob.setBackground(new Color(255, 255, 255));
zerob.setText("B=0");
zerob.setSelected(true);
zerob.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
    zerob_itemStateChanged(e);
    }
});

oneb.setBackground(new Color(255, 255, 255));
oneb.setText("B=1");
oneb.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
    oneb_itemStateChanged(e);
    }
});

zeroc.setBackground(new Color(255, 255, 255));
zeroc.setSelected(true);
zeroc.setToolTipText("click to select 5 or 0");
zeroc.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
    zeroc_itemStateChanged(e);
    }
});

onec.setBackground(new Color(255, 255, 255));
onec.setText("C=1");
onec.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
    onec_itemStateChanged(e);
    }
});

hint.setText("Click to set the inputs either to 0 or 1 & see the outputs .");

simulationarea.setLayout(null);
simulationarea.setBackground(new Color(255, 255, 255));
addComponent(simulationarea, addercircuit, 61, 8, 473, 299);
addComponent(simulationarea, sumled, 529, 35, 64, 134);
addComponent(simulationarea, carryled, 531, 169, 60, 135);
addComponent(simulationarea, zeroa, 22, 33, 30, 18);
addComponent(simulationarea, onea, 22, 51, 29, 18);
addComponent(simulationarea, zerob, 24, 104, 30, 18);
addComponent(simulationarea, oneb, 24, 122, 29, 18);
addComponent(simulationarea, zeroc, 24, 180, 30, 18);
addComponent(simulationarea, onec, 24, 197, 29, 18);
addComponent(simulationarea, hint, 4, 317, 345, 55);

// frstin
frstin.add(zeroa);
frstin.add(onea);
// scndin
scndin.add(zero);
scndin.add(one);
// thrdin
thrdin.add(zero);
thrdin.add(one);
// FullAdder
}

private void addComponent(Container container, Component c, int x, int y, int width, int height) {
    c.setBounds(x, y, width, height);
    container.add(c);
}

private void zeroa_itemStateChanged(ItemEvent e) {
    a = 0;
}

private void onea_itemStateChanged(ItemEvent e) {
    a = 1;
}

private void zerob_itemStateChanged(ItemEvent e) {
    b = 0;
}

private void oneb_itemStateChanged(ItemEvent e) {
    b = 1;
}

private void zeroc_itemStateChanged(ItemEvent e) {
    c = 0;
}

private void onec_itemStateChanged(ItemEvent e) {
    c = 1;
}

// The following method controls circuit behaviour:
public void paint(Graphics g) {
    super.paint(g);
    ccticon = new ImageIcon(getURL(cct));
    ledicon = new ImageIcon(getURL(led));
    ledonicon = new ImageIcon(getURL(ledon));
    zeroa.setIcon(new ImageIcon(getURL(zero)));
    zeroa.setSelectedIcon(new ImageIcon(getURL(zeron)));
    onea.setIcon(new ImageIcon(getURL(one)));
    onea.setSelectedIcon(new ImageIcon(getURL(oneon)));
    zerob.setIcon(new ImageIcon(getURL(zero)));
    zerob.setSelectedIcon(new ImageIcon(getURL(zeron)));
    oneb.setIcon(new ImageIcon(getURL(one)));
    oneb.setSelectedIcon(new ImageIcon(getURL(oneon)));
    zeroc.setIcon(new ImageIcon(getURL(zero)));
    zeroc.setSelectedIcon(new ImageIcon(getURL(zeron)));
    onec.setIcon(new ImageIcon(getURL(one)));
    onec.setSelectedIcon(new ImageIcon(getURL(oneon)));
    ccticon = new ImageIcon(getURL(cct));
    ledicon = new ImageIcon(getURL(led));
    ledonicon = new ImageIcon(getURL(ledon));
    zeroa.setIcon(new ImageIcon(getURL(zero)));
    zeroa.setSelectedIcon(new ImageIcon(getURL(zeron)));
    onea.setIcon(new ImageIcon(getURL(one)));
    onea.setSelectedIcon(new ImageIcon(getURL(oneon)));
    zerob.setIcon(new ImageIcon(getURL(zero)));
    zerob.setSelectedIcon(new ImageIcon(getURL(zeron)));
    oneb.setIcon(new ImageIcon(getURL(one)));
    oneb.setSelectedIcon(new ImageIcon(getURL(oneon)));
    zeroc.setIcon(new ImageIcon(getURL(zero)));
    zeroc.setSelectedIcon(new ImageIcon(getURL(zeron)));
    onec.setIcon(new ImageIcon(getURL(one)));
    onec.setSelectedIcon(new ImageIcon(getURL(oneon)));
}
zeroc.setSelectedIcon(new ImageIcon(getURL(zeron)));
onec.setIcon(new ImageIcon(getURL(one)));
onec.setSelectedIcon(new ImageIcon(getURL(oneon)));
addercircuit.setIcon(new ImageIcon(getURL(cct)));
if ((a==0)&&(b==0)&&(c==0))
{
    sumled.setIcon(new ImageIcon(getURL(led)));
carryled.setIcon(new ImageIcon(getURL(led)));
}
if ((a==0)&(b==0)&&(c==1))
{
    sumled.setIcon(new ImageIcon(getURL(ledon)));
carryled.setIcon(new ImageIcon(getURL(led)));
}
if ((a==0)&&(b==1)&&(c==0))
{
    sumled.setIcon(new ImageIcon(getURL(ledon)));
carryled.setIcon(new ImageIcon(getURL(led)));
}
if ((a==0)&&(b==1)&&(c==1))
{
    sumled.setIcon(new ImageIcon(getURL(ledon)));
carryled.setIcon(new ImageIcon(getURL(ledon)));
}
repaint();
}

protected URL getURL(String filename) {
    URL codeBase = this.getCodeBase();
    URL url = null;
    try {
        url = new URL(codeBase, filename);
    } catch (java.net.MalformedURLException e) {
        System.out.println("Couldn't create image: badly specified URL");
        return null;
    }
    return url;
4-Decoder code:

```java
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.applet.*;
import javax.swing.*;
public class Decoder3_8 extends JApplet {
    // Code Variables
    private JLabel jLabel1;
    private JTextArea comment;
    private JRadioButton onof;
    private JRadioButton off;
    private ButtonGroup switch;
    private JLabel d0led;
    private JLabel D1led;
    private JLabel D2led;
    private JLabel D3led;
    private JLabel D4led;
    private JLabel D5led;
    private JLabel D6led;
    private JLabel D7led;
    private JComboBox xinput;
    private JComboBox yinput;
    private JComboBox zinput;
    private JPanel contentarea;
    private int x,y,z,OFF;
    ImageIcon ccticon,ledicon,ledonicon;
    private Decoder3_8() {
            super();
            initializeComponent();
            this.setVisible(true);
    }
    private void initializeComponent() {
        jLabel1 = new JLabel();
        comment = new JTextArea(2,18);
        onof = new JRadioButton();
        off = new JRadioButton();
        d0led = new JLabel();
        D1led = new JLabel();
        D2led = new JLabel();
        D3led = new JLabel();
        D4led = new JLabel();
        D5led = new JLabel();
        D6led = new JLabel();
        D7led = new JLabel();
        xinput = new JComboBox();
        yinput = new JComboBox();
        zinput = new JComboBox();
    }
```
switch = new ButtonGroup();
contentarea = (JPanel)this.getContentPane();
x=0;
y=0;
z=0;
OFF=1;
//setting comment area
comment.setBackground(new Color(255, 255, 255));
comment.setText("Firstly, you must \n turn the decoder ON");
comment.setEditable(false);
// setting labels
jLabel1.setIcon(ccticon);
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
// the switch onof radio button
onof.setText("ON");
onof.setToolTipText("click here to select turn it ON");
onof.setBackground(new Color(255, 255, 255));
onof.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        onof_itemStateChanged(e);
        validate();
    }
});
// off
off.setText("OFF");
off.setSelected(true);
off.setBackground(new Color(255, 255, 255));
off.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        off_itemStateChanged(e);
        validate();
    }
});
// xinput
xinput.addItem("0");
xinput.addItem("1");
xinput.setToolTipText("click to select X value");
xinput.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        xinput_actionPerformed(e);
        validate();
    }
});
// yinput
yinput.addItem("0");
yinput.addItem("1");
yinput.setToolTipText("click to select Y value");
yinput.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        yinput_actionPerformed(e);
        validate();
    }
});

// zinput
zinput.addItem("0");
zinput.addItem("1");
zinput.setToolTipText("click to select Z value");
zinput.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        zinput_actionPerformed(e);
        validate();
    }
});

// contentarea
contentarea.setLayout(null);
contentarea.setBackground(new Color(255, 255, 255));
addComponent(contentarea, onof, 4, 9, 50, 22);
addComponent(contentarea, off, 4, 33, 50, 22);
addComponent(contentarea, comment, 3, 60, 99, 48);
addComponent(contentarea, JLabel1, 103, 2, 470, 510);
addComponent(contentarea, d0led, 572, 4, 33, 56);
addComponent(contentarea, D1led, 572, 66, 31, 52);
addComponent(contentarea, D2led, 571, 129, 33, 53);
addComponent(contentarea, D3led, 571, 191, 34, 58);
addComponent(contentarea, D4led, 572, 248, 33, 56);
addComponent(contentarea, D5led, 571, 304, 34, 54);
addComponent(contentarea, D6led, 571, 362, 39, 52);
addComponent(contentarea, D7led, 571, 420, 35, 51);
addComponent(contentarea, xinput, 60, 155, 43, 22);
addComponent(contentarea, yinput, 60, 239, 43, 22);
addComponent(contentarea, zinput, 60, 336, 43, 22);

// switch
switc.add(onof);
switc.add(off);
x=0;
else x=1;
}  
private void yinput_actionPerformed(ActionEvent e)  
{  
Object o = yinput.getSelectedItem();  
if (o=="0")  
y=0;  
else y=1;  
}  
private void zinput_actionPerformed(ActionEvent e)  
{  
Object o = zinput.getSelectedItem();  
if (o=="0") z=0;  
else z=1;  
}  
// The following method controls circuit behaviour:  
public void paint(Graphics g)  
{  
super.paint(g);  
ccticon = new ImageIcon(getImage(this.getCodeBase(),"Decoder3_8.jpg"));  
ledicon = new ImageIcon(getImage(this.getCodeBase(),"NOTSELECTED.JPG"));  
ledonicon = new ImageIcon(getImage(this.getCodeBase(),"ISSELECTED.JPG"));  
jLabel1.setIcon(ccticon);  
if (OFF==0){  
comment.setText("Change inputs to\nselect the output");  
if((x==0)&&(y==0)&&(z==0))  
{  
comment.setText("inputs 0 0 0\noutputs:10000000");  
d0led.setIcon(ledonicon);  
D1led.setIcon(ledonicon);  
D2led.setIcon(ledonicon);  
D3led.setIcon(ledonicon);  
D4led.setIcon(ledonicon);  
D5led.setIcon(ledonicon);  
D6led.setIcon(ledonicon);  
D7led.setIcon(ledonicon);  
}  
if ((x==0)&&(y==0)&&(z==1))  
{  
comment.setText("inputs 0 0 1\noutputs:01000000");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==0)&&(y==1)&&(z==0))  
{  
comment.setText("inputs 0 1 0\noutputs:00100000");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==1)&&(y==0)&&(z==0))  
{  
comment.setText("inputs 1 0 0\noutputs:00010000");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==1)&&(y==1)&&(z==0))  
{  
comment.setText("inputs 1 1 0\noutputs:00001000");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==1)&&(y==0)&&(z==1))  
{  
comment.setText("inputs 1 0 1\noutputs:00000100");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==0)&&(y==0)&&(z==1))  
{  
comment.setText("inputs 0 0 1\noutputs:00000100");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
if ((x==1)&&(y==1)&&(z==1))  
{  
comment.setText("inputs 1 1 1\noutputs:00000010");  
d0led.setIkon(ledonicon);  
D1led.setIkon(ledonicon);  
D2led.setIkon(ledonicon);  
D3led.setIkon(ledonicon);  
D4led.setIkon(ledonicon);  
D5led.setIkon(ledonicon);  
D6led.setIkon(ledonicon);  
D7led.setIkon(ledonicon);  
}  
}  
else  
{  
   comment.setText("Wrong inputs, try again");  
   }  
}
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
}
if((x==0)&&(y==1)&&(z==1))
{
    comment.setText("inputs 0 1 1 
outputs:00010000");
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
}
if ((x==1)&&(y==0)&&(z==0))
{
    comment.setText("inputs 1 0 0 
outputs:00001000");
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
}
if ((x==1)&&(y==0)&&(z==1))
{
    comment.setText("inputs 1 0 1 
outputs:00000100");
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
}
if((x==1)&&(y==1)&&(z==0))
{
    comment.setText("inputs 1 1 0 
outputs:00000010");
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
}
if((x==1)&&(y==1)&&(z==1))
{
    comment.setText("inputs 1 1 1 
outputs:00000001");
d0led.setIcon(ledicon);
D1led.setIcon(ledicon);
D2led.setIcon(ledicon);
D3led.setIcon(ledicon);
D4led.setIcon(ledicon);
D5led.setIcon(ledicon);
D6led.setIcon(ledicon);
D7led.setIcon(ledicon);
} else {
    comment.setText(" Firstly, you must \turn decoder ON");
    d0led.setIcon(ledicon);
    D1led.setIcon(ledicon);
    D2led.setIcon(ledicon);
    D3led.setIcon(ledicon);
    D4led.setIcon(ledicon);
    D5led.setIcon(ledicon);
    D6led.setIcon(ledicon);
    D7led.setIcon(ledicon);
} repaint();
//End of Decoder code.

5-Multiplexer code:

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Mux extends JApplet
{

    // Variables declaration
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JComboBox jComboBox1;
    private JPanel contentPane;
    private String a;
    private ImageIcon muxicon,ledoficon,ledonicon,commicon;
    // End of variables declaration

    public Mux()
    {
        super();
        initializeComponent();
        this.setVisible(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     */
    private void initializeComponent()
    {
        jLabel1 = new JLabel();
        jLabel2 = new JLabel();
        jLabel3 = new JLabel();
    }
jComboBox1 = new JComboBox();
contentPane = (JPanel) this.getContentPane();
a = "0 0 0";
// JLabel3
jLabel3.setText("click on the choice box\n to select your input");
// JComboBox1
jComboBox1.addItem("X A B");
jComboBox1.addItem("0 0 0");
jComboBox1.addItem("0 0 1");
jComboBox1.addItem("0 1 0");
jComboBox1.addItem("0 1 1");
jComboBox1.addItem("1 0 0");
jComboBox1.addItem("1 0 1");
jComboBox1.addItem("1 1 0");
jComboBox1.addItem("1 1 1");
jComboBox1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox1_actionPerformed(e);
    }
});
// contentPane
contentPane.setLayout(null);
contentPane.setBackground(new Color(255, 255, 255));
addComponent(contentPane, jLabel1, 87, 78, 367, 174);
addComponent(contentPane, jLabel2, 450, 141, 60, 137);
addComponent(contentPane, jLabel3, 1, 10, 124, 75);
addComponent(contentPane, jComboBox1, 18, 90, 64, 22);
}

private void addComponent(Container container, Component c, int x, int y, int width, int height) {
    c.setBounds(x, y, width, height);
    container.add(c);
}

// Event Handling Methods
private void jComboBox1_actionPerformed(ActionEvent e) {
    System.out.println("jComboBox1_actionPerformed(ActionEvent e) called.");
    Object o = jComboBox1.getSelectedItem();
    System.out.println(">>> + ((o==null)? "null" : o.toString()) + " is selected.");
    // handling code here for the particular object being selected
    a = o.toString();
}

// The following method controls circuit behaviour:
public void paint(Graphics g) {
    super.paint(g);
    muxicon = new ImageIcon(getImage(getDocumentBase(), "mux3.JPG"));
    ledoficon = new ImageIcon(getImage(getDocumentBase(), "maiof.JPG"));
    ledonicon = new ImageIcon(getImage(getDocumentBase(), "muxon.JPG"));
    JLabel2.setIcon(ledoficon);
jLabel1.setIcon(muxicon);
// making the circuit under the control of the truth table:
if (a.equals("X A B"))
{
    jLabel2.setIcon(ledoficon);
}
if (a.equals("0 0 0"))
{
    jLabel2.setIcon(ledoficon);
}
if (a.equals("0 0 1"))
{
    jLabel2.setIcon(ledonicon);
}
if (a.equals("0 1 0"))
{
    jLabel2.setIcon(ledoficon);
}
if (a.equals("0 1 1"))
{
    jLabel2.setIcon(ledonicon);
}
if (a.equals("1 0 0"))
{
    jLabel2.setIcon(ledoficon);
}
if (a.equals("1 0 1"))
{
    jLabel2.setIcon(ledonicon);
}
if (a.equals("1 1 0"))
{
    jLabel2.setIcon(ledoficon);
}
if (a.equals("1 1 1"))
{
    jLabel2.setIcon(ledonicon);
}
repaint();
}

6-Demultiplexer code:

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Demux extends JApplet
{
    // Variables declaration
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JLabel jLabel4;
    private JLabel jLabel5;

private JComboBox jComboBox1;
private JComboBox jComboBox2;
private JComboBox jComboBox3;
private JPanel contentPane;
private String a,b,d;
private ImageIcon ccticon,ledicon,ledonicon;
// End of variables declaration

public Demux()
{
    super();
    initializeComponent();
    this.setVisible(true);
}

private void initializeComponent()
{
    jLabel1 = new JLabel();
    jLabel2 = new JLabel();
    jLabel3 = new JLabel();
    jLabel4 = new JLabel();
    jLabel5 = new JLabel();
    jComboBox1 = new JComboBox();
    jComboBox2 = new JComboBox();
    jComboBox3 = new JComboBox();
    contentPane = (JPanel)this.getContentPane();
a="0";
b="0";
d="0";
// jLabel1
jLabel1.setIcon(ccticon);
// jLabel2
jLabel2.setIcon(ledicon);
// jLabel3
jLabel3.setIcon(ledicon);
// jLabel4
jLabel4.setIcon(ledicon);
// jLabel5
jLabel5.setIcon(ledicon);
// jComboBox1
jComboBox1.addItem("0");
jComboBox1.addItem("1");
jComboBox1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        jComboBox1_actionPerformed(e);
    }
});
// jComboBox2
jComboBox2.addItem("0");
jComboBox2.addItem("1");
jComboBox2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e)
    {
        jComboBox2_actionPerformed(e);
    }
});
jComboBox3.addItem("0");
jComboBox3.addItem("1");
jComboBox3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox3_actionPerformed(e);
    }
});

contentPane.setLayout(null);
contentPane.setBackground(new Color(255, 255, 255));
addComponent(contentPane, jLabel1, 52, 32, 306, 435);
addComponent(contentPane, jLabel2, 357, 160, 55, 70);
addComponent(contentPane, jLabel3, 357, 231, 54, 72);
addComponent(contentPane, jLabel4, 357, 310, 60, 74);
addComponent(contentPane, jLabel5, 356, 383, 60, 72);
addComponent(contentPane, jComboBox1, 10, 205, 42, 22);
addComponent(contentPane, jComboBox2, 10, 52, 42, 22);
addComponent(contentPane, jComboBox3, 10, 129, 42, 22);
}
private void addComponent(Container container, Component c, int x, int y, int width, int height) {
    c.setBounds(x, y, width, height);
    container.add(c);
}

// Event Handling Methods
private void jComboBox1_actionPerformed(ActionEvent e) {
    Object o = jComboBox1.getSelectedItem();
    d = o.toString();
}
private void jComboBox2_actionPerformed(ActionEvent e) {
    Object o = jComboBox2.getSelectedItem();
    a = o.toString();
}
private void jComboBox3_actionPerformed(ActionEvent e) {
    Object o = jComboBox3.getSelectedItem();
    b = o.toString();
}

// The following method controls circuit behaviour:
public void paint(Graphics g) {
    super.paint(g);
    ccticon = new ImageIcon(getImage(getCodeBase(), "dec1.jpg"));
    ledicon = new ImageIcon(getImage(getCodeBase(), "of1.jpg"));
   ledonicon = new ImageIcon(getImage(getCodeBase(), "on1.jpg"));
    jLabel1.setIcon(ccticon);
    if (d.equals("0"))
        jLabel2.setIcon(ledicon);
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;
public class Sevensegments extends JApplet
{
    // Variables declaration
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JRadioButton jRadioButton2;
    private JRadioButton jRadioButton3;
    private JComboBox jComboBox1;
    private JComboBox jComboBox2;
    private JComboBox jComboBox3;
    private JComboBox jComboBox4;
    private JPanel contentPane;
    private ButtonGroup working;

    //End of Demux code.

7-Sevent Segment Display code:

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;
public class Sevensegments extends JApplet
{
    // Variables declaration
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JRadioButton jRadioButton2;
    private JRadioButton jRadioButton3;
    private JComboBox jComboBox1;
    private JComboBox jComboBox2;
    private JComboBox jComboBox3;
    private JComboBox jComboBox4;
    private JPanel contentPane;
    private ButtonGroup working;

    //End of Demux code.
private int a, b, c, d; // binary inputs
private int f; // switch
private ImageIcon ccticon, oficon, zero, one, two, three, four, five, six, seven, eight, nine;
// End of variables declaration
public Sevensegments()
{
    super();
    initializeComponent();
    this.setVisible(true);
}
/**
 * This method is called from within the constructor to initialize the form.
 */
private void initializeComponent()
{
    JLabel1 = new JLabel();
    JLabel2 = new JLabel();
    jRadioButton2 = new JRadioButton();
    jRadioButton3 = new JRadioButton();
    jComboBox1 = new JComboBox();
    jComboBox2 = new JComboBox();
    jComboBox3 = new JComboBox();
    jComboBox4 = new JComboBox();
    contentPane = (JPanel) this.getContentPane();
    working = new ButtonGroup();
a = 0;
b = 0;
c = 0;
d = 0;
f = 0;
jRadioButton2.setBackground(new Color(255, 255, 255));
jRadioButton2.setText("on");
jRadioButton2.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        jRadioButton2_itemStateChanged(e);
    }
});
// jRadioButton3
jRadioButton3.setBackground(new Color(255, 255, 255));
jRadioButton3.setText("off");
jRadioButton3 setSelected(true);
jRadioButton3.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        jRadioButton3_itemStateChanged(e);
    }
});
// jComboBox1
jComboBox1.addItem("0");
jComboBox1.addItem("1");
jComboBox1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox1_actionPerformed(e);
    }
});
// jComboBox2
jComboBox2.addItem("0");
jComboBox2.addItem("1");
jComboBox2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox2_actionPerformed(e);
    }
});
// jComboBox3
jComboBox3.addItem("0");
jComboBox3.addItem("1");
jComboBox3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox3_actionPerformed(e);
    }
});
// jComboBox4
jComboBox4.addItem("0");
jComboBox4.addItem("1");
jComboBox4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jComboBox4_actionPerformed(e);
    }
});
// contentPane
contentPane.setLayout(null);
contentPane.setBackground(new Color(255, 255, 255));
addComponent(contentPane, jLabel1, 103,-13,284,390);
addComponent(contentPane, jLabel2, 384,128,101,111);
addComponent(contentPane, jRadioButton2, 54,331,45,14);
addComponent(contentPane, jRadioButton3, 54,313,41,16);
addComponent(contentPane, jComboBox1, 57,18,45,22);
addComponent(contentPane, jComboBox2, 56,95,46,22);
addComponent(contentPane, jComboBox3, 55,178,47,22);
addComponent(contentPane, jComboBox4, 56,249,46,22);
// working
working.add(jRadioButton3);
working.add(jRadioButton2);
// Sevensegments
private void addComponent(Container container,Component c,int x,int y,int width,int height) {
    c.setBounds(x,y,width,height);
    container.add(c);
}
// Handling Methods
private void jRadioButton2_itemStateChanged(ItemEvent e) {
    System.out.println("jRadioButton2_itemStateChanged(ItemEvent e) called.");
    System.out.println(">>" + ((e.getStateChange() == ItemEvent.SELECTED) ? "selected":"unselected"));
    // switch is ON
    f=1;
}
private void jRadioButton3_itemStateChanged(ItemEvent e)  
{  
System.out.println("jRadioButton3_itemStateChanged(ItemEvent e) called.");  
System.out.println(">>>" + ((e.getStateChange() == ItemEvent.SELECTED) ? "selected":"unselected");  
// Switch is OFF  
f=0;  
}  

private void jComboBox1_actionPerformed(ActionEvent e)  
{  
System.out.println("jComboBox1_actionPerformed(ActionEvent e) called.");  
Object o = jComboBox1.getSelectedItem();  
System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");  
if(o=="0")  
a=0;  
else a=1;  
}  

private void jComboBox2_actionPerformed(ActionEvent e)  
{  
System.out.println("jComboBox2_actionPerformed(ActionEvent e) called.");  
Object o = jComboBox2.getSelectedItem();  
System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");  
if(o=="0")  
b=0;  
else b=1;  
}  

private void jComboBox3_actionPerformed(ActionEvent e)  
{  
System.out.println("jComboBox3_actionPerformed(ActionEvent e) called.");  
Object o = jComboBox3.getSelectedItem();  
System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");  
if(o=="0")  
c=0;  
else c=1;  
}  

private void jComboBox4_actionPerformed(ActionEvent e)  
{  
System.out.println("jComboBox4_actionPerformed(ActionEvent e) called.");  
Object o = jComboBox4.getSelectedItem();  
System.out.println(">>>" + ((o==null)? "null" : o.toString()) + " is selected.");  
if(o=="0")  
d=0;  
else d=1;  
}  

// The following method controls circuit behaviour:  
public void paint(Graphics g)  
{  
super.paint(g);  
ccticon = new ImageIcon(getImage(getDocumentBase(),"sevencct.jpg"));  
oficon =  new ImageIcon(getImage(getDocumentBase(),"initial.jpg"));  
zero  = new ImageIcon(getImage(getDocumentBase(),"0.jpg"));  
one   = new ImageIcon(getImage(getDocumentBase(),"1.jpg"));  
two   = new ImageIcon(getImage(getDocumentBase(),"2.jpg"));  
three = new ImageIcon(getImage(getDocumentBase(),"3.jpg"));  
four  = new ImageIcon(getImage(getDocumentBase(),"4.jpg"));  
five  = new ImageIcon(getImage(getDocumentBase(),"5.jpg"));  
six   = new ImageIcon(getImage(getDocumentBase(),"6.jpg"));  
}
seven = new ImageIcon(getImage(getDocumentBase(),"7.jpg"));
eight = new ImageIcon(getImage(getDocumentBase(),"8.jpg"));
nine = new ImageIcon(getImage(getDocumentBase(),"9.jpg"));
javaLabel1.setIcon(ccticon);
javaLabel2.setIcon(oficon);
// making the circuit under the control of the truth table:
if (f==1)
{
    if ((a==0)&&(b==0)&&(c==0)&&(d==0))
        javaLabel2.setIcon(zero);
    else if ((a==0)&&(b==0)&&(c==0)&&(d==1))
        javaLabel2.setIcon(one);
    else if ((a==0)&&(b==0)&&(c==1)&&(d==0))
        javaLabel2.setIcon(two);
    else if ((a==0)&&(b==0)&&(c==1)&&(d==1))
        javaLabel2.setIcon(three);
    else if ((a==0)&&(b==1)&&(c==0)&&(d==0))
        javaLabel2.setIcon(four);
    else if ((a==0)&&(b==1)&&(c==0)&&(d==1))
        javaLabel2.setIcon(five);
    else if ((a==0)&&(b==1)&&(c==1)&&(d==0))
        javaLabel2.setIcon(six);
    else if ((a==0)&&(b==1)&&(c==1)&&(d==1))
        javaLabel2.setIcon(seven);
    else if ((a==1)&&(b==0)&&(c==0)&&(d==0))
        javaLabel2.setIcon(eight);
    else if ((a==1)&&(b==0)&&(c==0)&&(d==1))
        javaLabel2.setIcon(nine);
    else
        javaLabel2.setIcon(oficon);
}
else
    javaLabel2.setIcon(oficon);
repaint();
}

// End of Seven Segment code.