

**INSTRUCTION CYCLE SIMULATOR:  
A MODEL USED IN TEACHING COMPUTER  
ORGANIZATION SUBJECT IN SBCA  
Engr. Ma. Lea A. Martinez, MBM**

**Abstract**

A computer is an electronic equipment that can be programmed to manipulate symbols. It can perform simple to complex and repetitive procedures quickly, precisely and reliably; and it can also store and retrieve small or large amounts of data.

Computer Organization (IT105), a major subject taught for BSIT-1 students of SBCA covers this concept of a computer system and the architectural aspects which relate each of the computer's components. But considering the fast and innumerable developments on computers today, it would be way too complicated to study and explain the detailed operation of these newer computers. Hence, only the basic and standard operation can be introduced and simulated.

Instruction Cycle Simulator (ICS) is a program that executes instruction and performs arithmetic and logical calculations. This application operates in a very similar way to real computers but is much easier to understand. It can generally teach the users on how the computer works.

**Key Words:** Instruction Cycle Simulator (ICS), computer system, architectural aspects

**Introduction**

A computer is a device that accepts information and manipulates it for some result based on a program or sequence of instructions on how the data is to be

processed. Most histories of the modern computer begin with the Analytical Engine envisioned by Charles Babbage following the mathematical ideas of George Boole, the mathematician who first stated the principles of logic inherent in today's digital computer. Babbage's assistant and collaborator, Ada Lovelace, is said to have introduced the ideas of program loops and subroutines and is sometimes considered the first programmer. Apart from mechanical calculators, the first really useable computers began with the vacuum tube, accelerated with the invention of the transistor, which then became embedded in large numbers in integrated circuits, ultimately making possible the relatively low-cost personal computer.

Modern computers inherently follow the ideas of the stored program laid out by John von Neumann in 1945. Essentially, the program is read by the computer one instruction at a time, an operation is performed, and the computer then reads in the next instruction, and so on. Recently, computers and programs have been devised that allow multiple programs (and computers) to work on the same problem at the same time in parallel. With the advent of the Internet and higher bandwidth data transmission, programs and data that are part of the same overall project can be distributed over a network and embody the Sun Microsystems slogan: "The network is the computer."

Contemporary computers contain millions of elementary electronic components. How, then, can one clearly describe them? The key is to recognize the hierarchical nature of most complex system. A hierarchical system is a set of interrelated subsystem, each of the later, in turn, hierarchical in structure until we reach some lowest level of elementary subsystem is reached.

The hierarchical nature of complex systems is essential to both their design and their description. The designer needs only to deal with a particular level of the system at a time. At each level, the system consists of a set of components and their interrelationships. The behavior at each level depends only on a simplified, abstracted characterization of the system at the next lower level. At each level, the designer is concerned with structure and function:

Structure is the way in which the components are interrelated. Meanwhile, function is the operation of each individual component as part of the structure.

In general terms, there are four main functions of a computer:

- Data processing
- Data storage
- Data movement
- Control

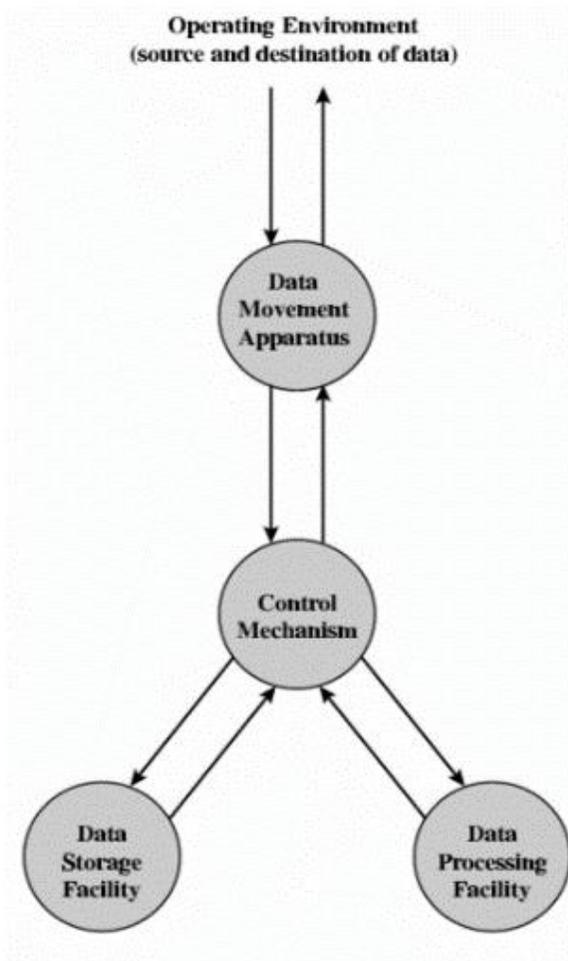


Figure 1: The Computer

The figure is the simplest possible depiction of a computer. The computer is an entity that interacts in some fashion with its external environment. In general, all of its linkages to the external environment can be classified as peripheral devices or communication lines.

- **Central Processing Unit (CPU):** Controls the operation of the computer and performs its data processing functions. Often simply referred to as processor.

- Main Memory: Stores data.
- I/O: Moves data between the computer and its external environment.
- System Interconnection: Provides communication among CPU, main memory, and I/O.

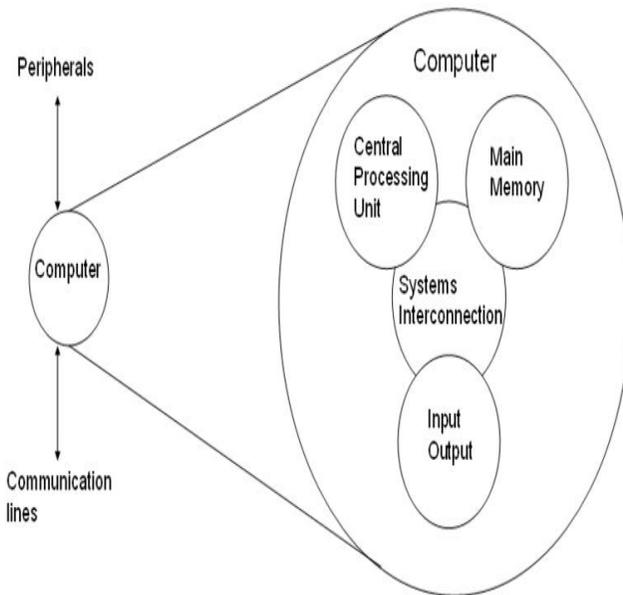


Figure 2: The Von Neumann Machine

The programming process is facilitated if the program can be represented in a form suitable for storing in memory alongside the data. Then, a computer can get

its instructions by reading them from memory, and then a program is altered by setting the values of a portion of memory.

This idea, known as the Stored-program concept, is usually attributed to the ENIAC designers, most notably the mathematician John von Neumann, who was a consultant on the ENIAC project. The first publication of the idea was in a 1945 proposal by von Neumann for a new computer, the EDVAC (Electronic Discrete Variable Computer).

In 1946, von Neumann and his colleagues began the design of a new stored-program computer, referred to as the IAS computer, at the Princeton Institute for Advanced Studies. The IAS computer, although not completed until 1952, is the prototype of all subsequent general-purpose computers. It consists of:

- A main memory, which stores both data and instructions.
- An arithmetic-logical unit (ALU) capable of operating on binary data.
- A control unit, which interprets the instructions in memory and causes them to be executed.
- Input and Output (I/O) equipment operated by the control unit.

The Instruction Cycle Simulator (ICS) is an application that is based on the concept of Von Neumann architecture which explains or shows the basic functionality of a personal computer. It also helps to assist in the understanding of the fetch/execute cycle through the visualization of the elements of computer that would normally be hidden. A computer architecture is a requirement specifying how a set of software and hardware technology standards interact to form a computer system or platform. In short, computer

architecture refers to how a computer system is designed and together with the determination of technologies it is compatible with.

## Method

The proposed physical layout of the Instruction Cycle Simulator is shown.

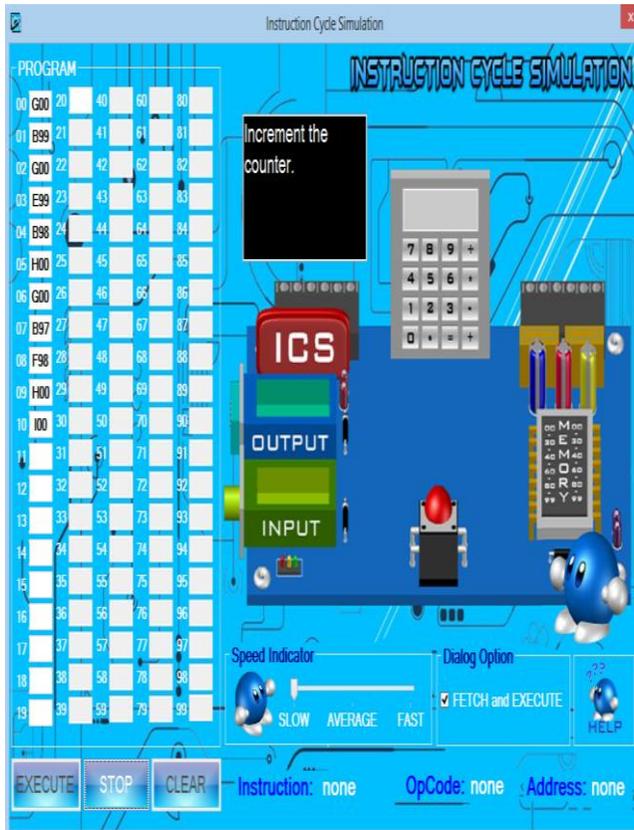
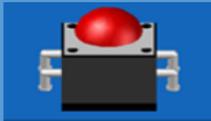
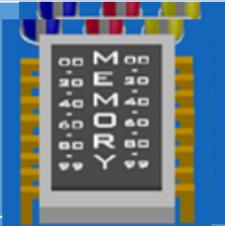
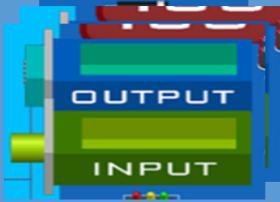


Figure 3: Instruction Cycle Simulator

**Table 1: ICS (Virtual) Hardware Components**

Instruction Cycle Simulation	
 <p>Chipboy</p>	<p>The character of the program where all the important work inside the application by following a process known as the 'Fetch and Execute Cycle', which involves oddly enough, fetching instruction, and executing them. On a real computer, this would be the Control Unit (CU)'.</p>
	<p>Keeps track of the location where Chipboy must look for the next instruction. It always start by getting the number in the counter, then incrementing the counter.</p>
	<p>Memory can be used to store programs, and data. In the fetch phase of the fetch cycle we are only interested in the fact that we can get instructions from the memory.</p>
	<p>These are what we (the user) use to communicate with Chipboy. We provide him with data through the Input tray, and when he's ready, he provides us with data through the Output tray.</p>
	<p>On a real computer, this would be the Arithmetic Logic Unit (ALU). The Arithmetic Logic Unit (ALU) where numbers are temporarily stored and used in arithmetic operations. The basic operations it can do are: addition, subtraction, multiplication and division.</p>

## The ICS and a Real Computer

The relationship between the ICS model and a physical computer is as follows:

1. The **Mailboxes** represent the Random Access Memory (RAM) which is used for loading programs and data for immediate use by the CPU.
2. The **Calculator** represents the Arithmetic Logic Unit (ALU) of the CPU.
3. **Chipboy** himself is like the Control Unit (CU) of the CPU.
4. The **Input Tray** represents the various inputs connected to the CPU, which includes keyboard, mouse, hard drive connections and so on.
5. The **Output Tray** represents the various outputs connected to the CPU, which include monitors, speakers and so on.

On the part of ICS there are objects that never existed in some models. These are:

Output Dialog – Displays the value of the calculator to the output tray.

Stop Button – Stops the program.

Clear Button – Clears' all the program mailbox.

Table 2: Instruction Set for the ICS

ICS	Operation	Meaning
Axx	Load	Load number from memory location (xx) into calculator.
Bxx	Store	Store the number from calculator to memory location (xx).
Cxx	Add	Add the number in memory location (xx) from the calculator.
Dxx	Subtract	Subtract the number in memory location (xx) from the calculator.
Exx	Multiply	Multiply the number in memory location (xx) from the calculator.
Fxx	Divide	Divide the number in memory location (xx) from the calculator.
G00	Input	Get the input from the input tray and load it in the calculator.
H00	Output	Display the number in the output tray available at the calculator.
I00	Halt	End / Stop the program.
Jxx	Jump	Get the address (xx) and enter to the program counter then jump to that address (xx).
K00	Skip -	Skip the next instruction if the number in the calculator is negative.
K01	Skip 0	Skip the next instruction if the number in the calculator is zero.
K02	Skip +	Skip the next instruction if the number in the calculator is positive.
Lxx	Loop	Get the value from the address (xx) and then reduce to 1 until the value from the address (xx) become 0.
LEN	End Loop	The terminator of the loop.

## Example of Operations- Store Instruction

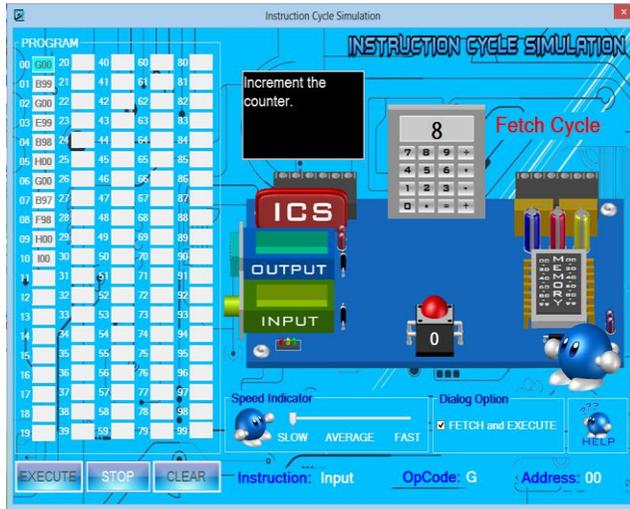


Figure 4: Chipboy identifies the program counter needs to increment.

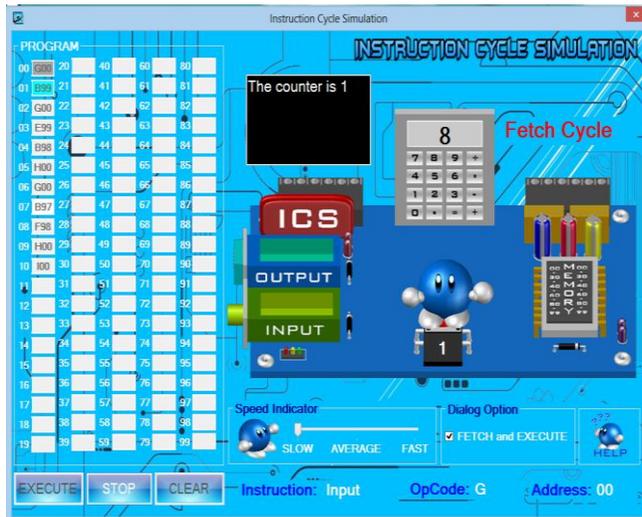


Figure 5: Chipboy identifies the ICS code stored in memory address 01.

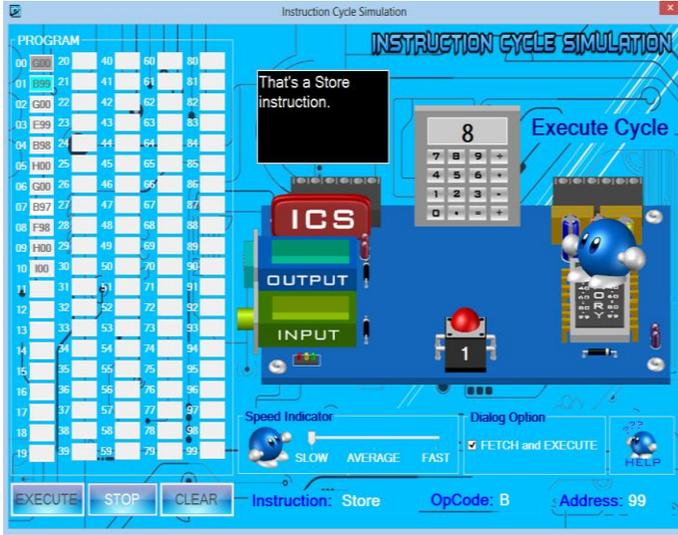


Figure 6: Chipboy interprets and displays the instruction located in memory address 01.

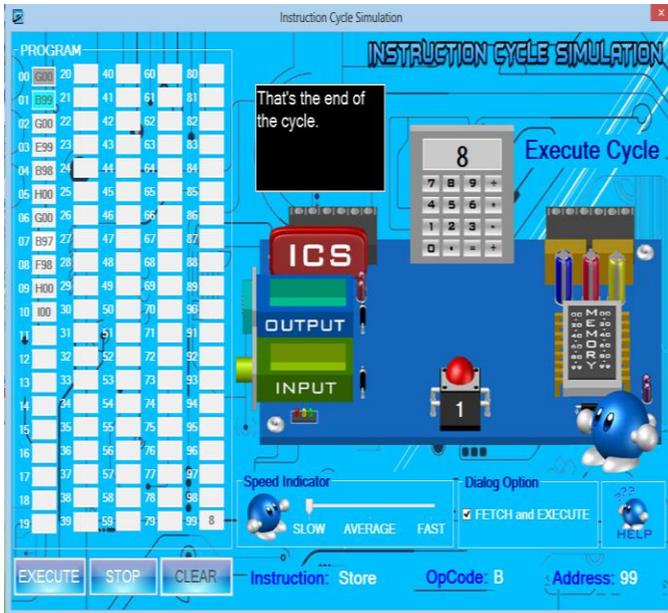


Figure 7: The Navy blue man ends the cycle.

## Results

The Instruction Cycle Simulator (ICS) will be a good teaching model to explain and visualize how the computer fetch and execute the instructions given by a user. It offers some basic functionality features of a personal computer.

- The mailbox address is numbered from 00 to 99. Each mailbox is designed to hold a combination of one character and a 2-digit decimal number. It can also hold a 3-digit decimal number.
- The calculator can accept up to 3-digit numbers.
- A 2-digit location/program counter that indicates what address is being executed.
- The input and output will be delivered to the input tray and output tray respectively.
- The calculator can be used to show the value in the input/output tray, the value temporarily stored or retrieved from the memory box with to add, subtract, multiply and divide functions.
- The ICS can perform the following processes: Addition, Subtraction, Multiplication, Division, Loop, Jump, Conditionals (e.g. Skip if Positive number, Skip if Negative number and Skip if Zero), Load, Store, Input, Output, and Halt/Stop.

## **Conclusion**

The Instruction Cycle Simulator (ICS) is a teaching model of a computer which will be used to communicate to students the concept of computer organization. It models a simple von Neumann architecture computer which has all of the basic features of a modern computer. ICS can also help to assist in the understanding of the fetch/execute cycle through the visualization of the elements of a computer which are normally hidden.

## References

[www.dictionary.reference.com](http://www.dictionary.reference.com)  
[www.searchwindowserver.techtarget.com](http://www.searchwindowserver.techtarget.com)  
[www.techopedia.com](http://www.techopedia.com)  
[www.cnx.org](http://www.cnx.org)