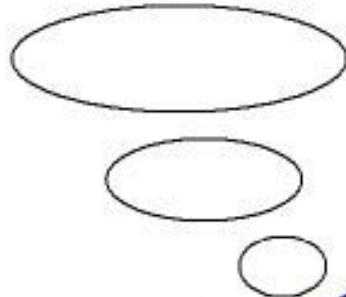
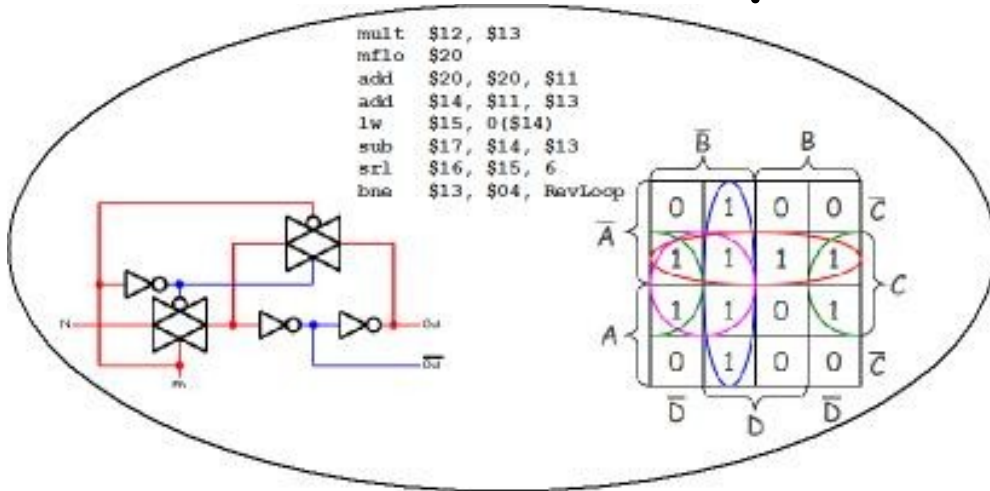


# Introduction to Computer Design



# Introduction to Computer Design

Our world is full of computer-based systems: smartphones, laptops, tablets. And new, useful and attractive products are appearing regularly. How are they designed? What are they made of? Today's high tech products contain dozens of subsystems, each composed of many components. Some are specialized, like a color display or a wireless transceiver. But the *computing* in a computing system is general and programmable. Interestingly, while computing provides the system "smarts", it is built using two very simple components: switches and wire.

So how does one create a smart product out of dumb components? First, these components are extremely versatile. Wire-connected switches can be used to build functional elements (e.g., perform arithmetic operations), memory elements (e.g., store values and program instructions), and control elements (e.g., sequence program instructions in an execution datapath). These elements can form a programmable computer that can exhibit a very smart behavior.

Second, technology offers the ability to integrate millions or even billions of wire-connected switches in an extremely compact integrated circuit (IC). This large number of switches supports the construction of a powerful programmable computer and sophisticated real-world interfaces. The low cost of these devices allows them to be incorporated in a wide range of products.

Designing systems with millions of components is a challenge. A successful strategy is a *design hierarchy*. Use several simple elements to produce a more complex component. Then design the next level using these more complex components. The design hierarchy below spans from switches and wire to assembly programming.

Assembly Language		
Instruction Set		
Memory	Datapath	Controller
Storage	Functional Units	State Machines
Building Blocks		
Gates		
Switches and Wire		

While there are other ways to describe the organization of programmable computers, this hierarchy illustrates the design path presented in the following chapters. Some chapters address specific levels of the hierarchy (designing with switches). Others focus on design techniques (simplification) and mathematics (Boolean algebra). All chapters emphasize design; how do you build a computer system.

## Chapters

1. Introduction to Computer Design
2. Switches and Wire
3. Boolean Algebra
4. Gate Design
5. Simplification
6. Building Blocks
7. Number Systems
8. Arithmetic
9. Latches and Registers
10. Counters
11. State Machines
12. Memory
13. Datapath
14. Controller and Instruction Set
15. Assembly Programming

**CompuCanvas:** There is a free, open-source tool that can help you try out many of the ideas and techniques describe in these chapters. It is written in the Python programming language (available at [www.python.org](http://www.python.org)); so it runs on almost everything. Find out more about it at the CompuCanvas website [www.compucanvas.org](http://www.compucanvas.org).