In the name of Allah

UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department

Digital Logic Design

Naeem Esfahani 810180158
naeem@ece.ut.ac.ir

Abstract: Logic circuits are developing fast and they will have important role in our life so the improvement in designing methods will be unavoidable. I'll try to cover *design hierarchy, abstraction levels, design divide and conquer, top down design and the use of simulation and synthesis tools in design* in my paper.

What is Digital System? Long before analog systems and computers were used. Analog systems are systems which their data is represented and manipulated in continues manner in the other hand we have Digital systems, where data is represented discreetly. Nowadays Digital computers are more economic because they have wide spectrum of use. It ushers lower cost so they are good replacement for fixed circuits. So the paradigm of Digital Design will have important role in our life.

Divide and conquer: When we want to deal with something huge we must divide it into several layers and make logical relations between them. If we use the old methods which used to design old simple digital systems it'll be confusing and it'll be difficult for us so we divide our task into some subtasks and then we attack to subtasks and if subtasks are still huge divide them more into subtasks. We have an English proverb that says: *"Divide and then conquer."* When they wanted to become big brother of other countries they were using it. It means if there is a big problem break it to parts and then deal with each part alone. We use it in design; we break it into parts so our labor will become less and our work will become easier.

Digital system design hierarchy and abstraction levels: We'll design our digital system in several different layers of abstraction, from full behavioral model which has the highest level of abstraction to physical level which has the least. In the full behavioral model no hardware details is applied but we'll specify structures of physical materials in the physical level.

Levels of abstraction are:

1. **The System and Register Levels**; it is also named Register Transfer Level (RTL) which is in the highest level. Our view from system is one or more interacting function- able modules. The behavior of each module is described without specifying implementation details.
2. **The gate level** is the lower level. The behavior of a digital system is specified as a set of logic equations from switching algebra. We have two set of gates Combinational logic circuits; the gates which have no memory and Sequential logic circuits; the gates have memory.
3. **Transistor and Physical Design Levels:** The behavioral definition of logic circuits will be completely determined by the gates. Gates

are constructed by low level transistors which their base is semiconductors. The transistors will be dealt with in Electronics. Transistors are used in gates because they are tiny on/off switches. So we can make gates with any other switches but transistors are chosen because they are tiny, faster and more reliable. Before transistors mechanical switches, relays and electron tubes were used. After transistors we have ICs which is combination of many transistors. The power of an IC is determined by number of gates it contains. Small Scale Integration (SSI) refers to ICs with 1 to 10 gates, Medium Scale Integration (MSI) corresponds to 10- to 100-gate ICs, Large Scale Integration (LSI) to 100 to 10,000 gates, and Very Large Scale Integration (VLSI) to ICs with more than 100,000 gates.

Top-down design: In this approach you divide design paradigm into several levels in which the ability of system is fixed. More down you go the ability decreases and you must deal with details. For example, you want to make a calculator then you sat OK we have something to get inputs, something to show results, something to… in this level you pretend displaying is something meaningful but when you want to construct display part the word display is no longer meaningful and you must deal with details. How much you go down is depending on the technology you have. I mean if you have CAD tools (in hardware), compilers (in software) and …

The use of simulation and synthesis tools in design:

Computer Aided Design (CAD) tools enable us to work in higher level of abstraction they will automatically make the lower level so we'll become less busy and our attendance will become designing instead of time consuming details.

When we work with CAD tools we must communicate with tool so we model our problem in the way that computer will understand us by modeling this model can be in any range from behavioral to structural. Then we want something to translate our model into full detailed implementation. The mean we use for this task is a Hardware Description Language (HDL). Model can be pure behavioral which is in the highest level of abstraction or pure structural which is in the least abstraction level or mixed-mode model which has both of them.

CAD tools help us in two way synthesis and simulation:

-Synthesis: We convert high level design to low level implementation so that the design level can move some steps higher. The final output of synthesis is a logic circuit netlist which indicate the elements used and their connection which will be obtained from flattening process. Netlist is called to textual representation of logic diagram. In this process some testing tasks apply but all of them refer to design rule errors (against basic logic circuit design rule). But logical errors wont discovered until simulation process.

-Simulation: We test circuit to see if circuit works in the way we desire and if its results satisfy us. First we test if the whole system is work correctly by applying test vectors (an ordered list of ones and zeroes) and checking the output to see if it is compatible with the source from which the model was developed. In this stage we pretend the gates have no delay. That means we simulate ideal circuit with ideal gates.

In the second stage we become more reasonable and we consider delays. We see its effect on circuit performance in simulation. We can also have test cases that detect the fault component in acceptable little time with good percentage.

References:

[1] V. P. Nelson, H. T. Nagel, B. D. Carroll, J. D. Irwin, *Digital logic circuit analysis & design,* Prentice-Hall Inc., 1995.

[2] Z. Navabi, *Verilog digital system design,* McGraw-Hill Book Co, 1999.