MSEEVL - Master of Science in Electrical Engineering (VLSI DESIGN) 1/2

- A minimum cumulative grade point average (GPA) of 3.0 must be earned on all course work.
- Completing the normal requirements for the Master of Science in Electrical Engineering, with the following additional 4 required courses:
  (1) EE 536a; (2) EE 577a; (3) EE 577b or EE 536b; (4) EE 552.
If a student chooses to take EE 536b as well as EE 577b, the student may either count EE 536b as one of the courses for Area 2 or EE 577b as one of the courses for Area 1 or Area 3. (described below).
- The student must also take two courses from one of the following areas and one course from a second area:
  Area 1 “Tools”: CSCI 455x, EE 560, EE 577b (see above), EE 658, EE 680 and EE 681.
  Area 2 “Analog”: EE 448, EE 504L, EE 536b (see above), EE 537 and EE 630.
  Area 3 “Architectures”: CSCI 455x, CSCI 570, EE 557, EE 577b (see above), EE 659, EE 677.
MSEEVL - Master of Science in Electrical Engineering (VLSI DESIGN) 2/2

Thesis Option

The minimum requirement is 27 units; four of these units are to be thesis. At least 16 units, not including thesis, must be at the 500 level or higher, and at least 18 units must be in the major department. A total of not less than four nor more than eight units of 590 Research and 594ab Thesis must be included in the program. The minimum thesis requirement in 594a is two units; in 594b, two units.
VLSI System Design: Foundation, Tools & Processes

THE END RESULT

VLSI chip

THE PROCESS

MANUFACTURING

VERIFICATION

DESIGN

TEST

477
577ab

658

THE TOOLS

TCAD

CAD I, II, III

THE FOUNDATION: C, data structures, digital logic design

8/16/10

lasted edited by Mel Breuer
Simplified VLSI System Design Flow

1. Design specification
2. Behavioral description
3. RTL description (HDL)
4. Functional verification
5. Logic synthesis +DFT+BIST
6. Gate-level netlist
7. Logic verification
8. Floor planning, place and route
9. Physical layout
10. Layout verification and implementation
11. FAB
12. Testing
13. Sort/bin
14. Pass
15. Fail
16. Trash
17. Slow
18. Less capability

8/16/10 lasted edited by Mel Breuer
Simplified VLSI System Design Flow

Design specification → Behavioral description → RTL description (HDL) → Functional verification → Logic synthesis +DFT+BIST → Gate-level netlist → Logic verification → Floor planning, place and route → Physical layout → Layout verification and implementation → FAB → Testing → Sort/bin → Circuit/electrical aspects (Transistor level design, interconnect, crosstalk, noise, power distribution, clock distribution, ...) → Pass/Fail/Sort/Bin

- Power
- Reliability
- Area
- Performance
- Yield
System Synthesis

The first step in the design process is the human transformation of an implementation independent specification and the functionality and performance specs into a flow of both parallel and sequential conditional computational steps e.g., data and control flow graphs. Key tasks during this process are (1) the selection of a suitable architecture, (2) decomposing the required functionality over the components of this architecture including decision about which functionality is implemented in hardware and which in software (to be run on a general purpose or embedded processor), and (3) scheduling the various activities so as to meet the performance specs. Part of this process deals with determining and analyzing various algorithms for instantiating the desired behaviors. For example there are many ways of building a Viterbi decoder. In addition to the algorithmic solution, one must also consider the non-behavioral aspects of the spec, such as imposed constraints and performance objectives. Since the actual design space is quite vast, automation is often used to prune the space and to predict values of attributes such as gate count and power dissipation. Often languages such as hardware C or System C are used. The next step in the design process, namely behavioral synthesis, is highly intertwined with system synthesis.
Behavioral Level Synthesis

The next step is to take the system description as a collection of interacting machine descriptions or data and control flow graphs and produce the system architecture in terms of what components are used to perform required operations, how much sharing is desirable or possible, how are the various operations scheduled, where the intermediate results of computation are stored, how these components communicate with each other, etc.

The three main problems typically addressed in behavioral synthesis systems are allocation of resources, scheduling of processes, and binding of variables to hardware entities.

The binding of variables to hardware entities includes defining the bit-width of various elements. Some bit-widths may be determined at the higher behavioral level, but sometimes domain-expert algorithm designers work at the behavioral level and leave bit-width analysis to the hardware designers.

Often languages such as Verilog, VHDL and System C are used to describe the behavior of a chip.
Register Transfer Level Synthesis

System and behavioral synthesis are somewhat like programming in a high level language and applying compiler-like transformations to optimize the program (in terms of its size or its execution time); you have all the variables, memory and operators you want. But a chip consists of a finite number of gates, flip-flops and memory, though this finite number is getting to be in the billions. Thus a behavioral description must be transformed into one that relates variables, constants and operators with hardware entities such as registers, memories, busses and blocks of logic. This transformation process, which over the past 15 years has been partially automated, is referred to as register-transfer-level (RTL) synthesis. At this point various computer-aided design (CAD) tools can be employed that predict attributes of the final circuit if the design process were to be continued. Some of these attributes include power dissipation, gate count, area, yield and performance. So, at this point, a designer can make an intelligent decision as to whether to continue refining this design, or consider alternative behavioral and/or register solutions.

The inputs to this step are a set of operations described as transfer of values between registers and functional units through a hierarchy of interconnect structures including buses and multiplexers under the command of a hardware controller. The outputs of this step are blocks of combinational logic separated by sequential circuit elements (latches and flip-flops). An important task performed in this step is the encoding and realization of the controller block (specified as a finite state machine or a collection of interacting machines) in hardware. Here is where your knowledge of FSM’s is essential.
Logic synthesis

Logic synthesis is the process of taking what we normally think of as a logic design consisting of gates, flip-flops, and clusters of flip-flops referred to as registers, and manipulating this digital circuit so that it retains its functionality, but now satisfies various constraints and optimizes various objectives. First, you need to recognize that 2-level AND-OR on NAND-NAND logic is not encountered very often in real circuits. Most logic structures are multi-level, and include both primitive and complex gates. Area, testability and performance (delay) are of prime concern. Thus, logic synthesis, which also is a highly automated process, consists of operations such as re-timing, library mapping/binding, fan-out optimization and gate sizing.

If the area, performance, or power does not meet your specifications, then re-design at the RTL or behavioral level is needed.

Well, at this point we finally have a design in terms of things we understand, namely gates and flip-flops. Now lets go get out our soldering iron and build it in hardware.
Physical Design

- Physical design is the process of mapping the output from temporal and logic synthesis into a form that can be sent to a factory for manufacturing. The end product of physical synthesis is actually a set of masks in the form of a GDSII file. A mask is somewhat analogous to a film negative that has millions of black rectangles exposed on a transparent sheet. One main difference is that a set of masks costs several millions of dollars, so you do not get many chances to “capture that moment.” Physical synthesis consists of several processes that are mostly automated, usually highly inter-related, but for simplicity, described here as separate entities, namely floorplanning, placement and routing.
Testing

Gordon Moore noted that a major shift in scaling of VLSI feature sizes occurs about every 18 months. Over the last few years we have gone from 65nm in 2007, to 45 nm in 2010 and soon to 32 nm in 2013. Why didn’t we just skip the 45nm technology node and come out with 32 nm instead? Well the answer is that it is very difficult and costly to move from one node to another in terms of research, development and manufacturing equipment. Simply put, it is very costly. Thus, while researchers are currently working on 22nm and 14 nm nodes, the yields are still very low. That is, if one manufactured one million transistors, few if any would work as desired. It usually takes several years of tuning a technology to learn how to increase the yield to very high levels; this is the process known as “yield learning”. I leave it to the reader to calculate the minimum required probability of manufacturing a good transistor so that if a chip has one-billion transistors, the probability of all of them are good is 0.5. Well the answer looks something like 0.999...9. As a result, it is estimated that from 30-70% of complex (multi-million transistors) chips manufactured today using the latest technology nodes are defective. Yet they all look the same. Manufacturing test is the process of separating out the good chips from the bad ones. Again, most aspects of test development and testing itself have been automated. The main topics addressed in testing are: design for test, built in self tests, test generation, fault simulation, classes of faults, fault diagnosis, etc.
Verification

It is not humanly possible to design a multi-million gate chip without making several or many mistakes. Also, many aspects of the circuit design process are based on imprecise models, approximations and assumptions. Thus the final product usually has many problems. Thus verification is a primary tool associated with design. Verification must address many aspects of a design, such as its temporal attributes, its electrical characteristics, naturally the logic itself, and finally the physical characteristics. Properties must be checked that were not intended to be part of the design, such as simultaneous switching noise, and ringing due to inductance. Verification is one of the last frontiers in CAD that has not been successfully automated.
Test (E658): VLSI Design Flow

- Design specification
- Behavioral description
- RTL description (HDL)
- Functional verification
- Logic synthesis
- Gate-level netlist
- Logic verification
- Floor planning, place and route
- Physical layout
  - Layout verification and implementation
    - FAB
      - Testing
        - Sort/bin
          - Fail
            - Trash
          - Pass
          - Slow
            - Less capability

CAD I and III: VLSI Design Flow

- Design specification
- Behavioral description
- RTL description (HDL)
- Functional verification
- Logic synthesis + DFT + BIST
- Gate-level netlist
- Logic verification
- Floor planning, place and route

Physical layout

- Layout verification and implementation
- FAB

Testing

- Sort/bin
  - Fail
  - Slow
  - Trash
  - Pass
  - Less capability

8/16/10

lasted edited by Mel Breuer
Schedule VLSI students only taking CAD I

<table>
<thead>
<tr>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE CAD I (4)</td>
<td>EE 577a (3)</td>
</tr>
<tr>
<td>EE 477 (4)</td>
<td>CSCI 455 (3)</td>
</tr>
<tr>
<td>Remedial class and/or English</td>
<td>EE 552 Asynchronous circuits (3) or EE560 or …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 577b (2)</td>
<td>Analog circuits and/or fabrication (3)</td>
</tr>
<tr>
<td>EE 658 Testing (3)</td>
<td>Computer architecture and/or networks (3)</td>
</tr>
<tr>
<td>Application:</td>
<td></td>
</tr>
<tr>
<td>Communication, signal processing, biomed</td>
<td>EE590 Directed research (0, 1, 2, 3, 4)</td>
</tr>
</tbody>
</table>

28 units shown
Schedule VLSI students only taking CAD I, II, III

<table>
<thead>
<tr>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE CAD I (4)</td>
<td>EE 577a (3)</td>
</tr>
<tr>
<td>EE 477 (4)</td>
<td>CSCI 455 (3)</td>
</tr>
<tr>
<td>Remedial class and/or English</td>
<td>EE CAD II (3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fall</th>
<th>Spring</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE 577b (2)</td>
<td>Analog circuits, fabrication, computer architecture, networks or asynchronous circuits (3)</td>
</tr>
<tr>
<td>EE 658 Testing (3)</td>
<td>Application: Communication, signal processing, biomed</td>
</tr>
<tr>
<td>EE CAD III (3)</td>
<td>EE590 Directed research (0, 1, 2, 3, 4)</td>
</tr>
</tbody>
</table>

28 units shown
And you must keep on learning

What, How, Why, When