EE681 Course
Computer-Aided Design of Integrated Systems II

Instructor: Peter A. Beerel
Coordinates: EEB 350, x0-4481
pabeerel@eiger.usc.edu

http://www-classes.usc.edu/engr/ee-s/681b/ee681.html
**What Is EE681 About?**

**Algorithms and tools** to aid the analysis and design of VLSI circuits and systems

Required functionality may be described behaviorally as an algorithm

- Greatest common divisor or MPEG encoder/decoder described in C

VLSI circuit may be described structurally as a hierarchical interconnection of modules

- Interconnection of datapaths, registers, FSMs, control logic, buses, etc.
Compute the greatest common divisor of numbers \( x \) and \( y \)

```
while ( x != y ) {
    if ( x > y )
        x = x - y ;
    else
        y = y - x ;
}
gcd = x ;
```
HDL Specification

Interface is specified in VHDL

entity GCD is
  port(
    xi, yi : in BIT_VECTOR(1 to SIZE);
gcd : out BIT_VECTOR(1 to SIZE);
x, y : buffer BIT_VECTOR(1 to SIZE);
rst : buffer BIT;
  );
end VHDL;
architecture GCD_BEHAVIOR of GCD is
begin
    process begin
        variable gtr, equ : BIT ;
        wait until CLOCK'event and CLOCK = ‘1’ ;
        if ( rst = ‘1’ ) then
            x <= xi ; y <= yi ; rst <= 0 ;
        else
            gtr := x > y ;
            if ( gtr = ‘1’ ) then
                x <= x - y ;
            else
                equ := x = y ;
                if ( equ = ‘1’ ) then
                    gcd <= x ; rst <= ‘1’ ;
                else
                    y <= y - x ;
                end if ;
            end if ;
        end if ;
    end process ;
end VHDL_1 ;
Logic-Level Circuit
Gate-Level Circuit

![Diagram of a gate-level circuit](image)
Transistor and Layout Levels
Why CAD?

CAD tools are an enabling technology

– Have raised the abstraction level of design entry. From “rectangle pushing” to “schematic editing” to “program writing”

CAD tools have

– Significantly reduced design turnaround time
– Helped to manage design complexity
– Allowed design space exploration
– Helped to improve design characteristics
**Aspects to VLSI CAD**

**Simulation**: Analyze a design at different levels of abstraction, e.g., device-level, logic-level, behavioral-level.

**Synthesis**: Translate design from higher (e.g., algorithmic) to lower (e.g., logic) level of abstraction.

**Verification**: Ensure that the design is functionally correct and meets timing requirements.

**Testing**: Once design has been fabricated on integrated circuit, generate tests that will check correctness of fabricated circuit.
Synthesis

Involves taking an architectural specification and producing a VLSI circuit layout that

– Satisfies area, timing, and power dissipation specs
– Is the smallest or fastest or most power efficient

Many, many alternatives in design synthesis

– Need to be able to make correct choices by systematically exploring design space
– Getting a design is easy, obtaining a feasible design or the “best” design is very difficult
Verify that the VLSI circuit layout correctly implements the specified functionality.

if ( a = ‘1’ )
    b := ‘0’ ;
else
    b := ‘1’ ;
Test Generation

Check that the fabricated circuit implements the logical functionality by generating and applying a set of test vectors

Defect (misaligned gate) impairs logic functionality
Aspects of EE681

- Boolean algebra
- Logic synthesis
- Combinatorial optimization
- Automata theory
- Computer Architecture
- Architectural synthesis
- Compiler optimization
- Verification and Test
- Graph theory
Typical Assignment:

- You will be given a CAD algorithm implemented in a C software package, with input-output and algorithm structure well-defined.
- Modify the core of the algorithm (usually selection heuristics) to improve the results on given set of benchmarks.
- Expect to write no more than ~500 lines of C code.