ASIC Design Methodology
ASIC Design Methodology

- ASIC design process is very, very complex
- No single step is all that complicated, but...
- An enormous number of steps are involved
- Many of the steps are repetitious (use a computer!)
- Scripts make the computer do the grunt work for us
A piece of the design flow...
Scripting provides many advantages

- One script works with few mods for many chips (Reusability)
- Self-documenting - better than separate documentation)
- It’s completely repeatable
- Automatic, requiring no human intervention (productivity again)
HDLs use shell scripting to build ASICs

- PERL, Bash, TCL used extensively
- Top level script sets up environment, calls tools, checks output
- Lower level scripts run individual tools
A simple Bash shell program "doit.sh"

"doit.sh" compiles Verilog file with vlog, then starts the simulator

```bash
#!/bin/bash

if [ ! -d "work" ] ; then
    echo "work does not exist, making it"
    vlib work
fi

if [ -s "adder.sv" ] ; then
    vlog -novopt adder.sv
fi

echo "***** Simulating adder at rtl level"
vsim adder -do adder.do
```
A simple TCL program ”adder.do”

”adder.do” directs the operation of the simulator

```
view signals
add wave -r /*
force data 16#002
force rd_fifo 0
force first_select 1
force -freeze /clk 1 -repeat 100
force -freeze /clk 0 50 -repeat 100
force reset_n 0
run 90
force reset_n 1
run 10
run 100
```
The design process transforms code to silicon in steps

- At each step, the design is checked against a known correct reference
- The reference is known as the "golden reference"
- Can be as simple as a bit vector set or a set of rules to check
- Each check against the reference is called regression
Testbenches are used to facilitate checking at each step

- The testbench is a top level module that encompasses the entire design
- It helps feed data and stimulus to the design
- It is not synthesible