BASH Basics

Terminal

shell

Key
Board

Program
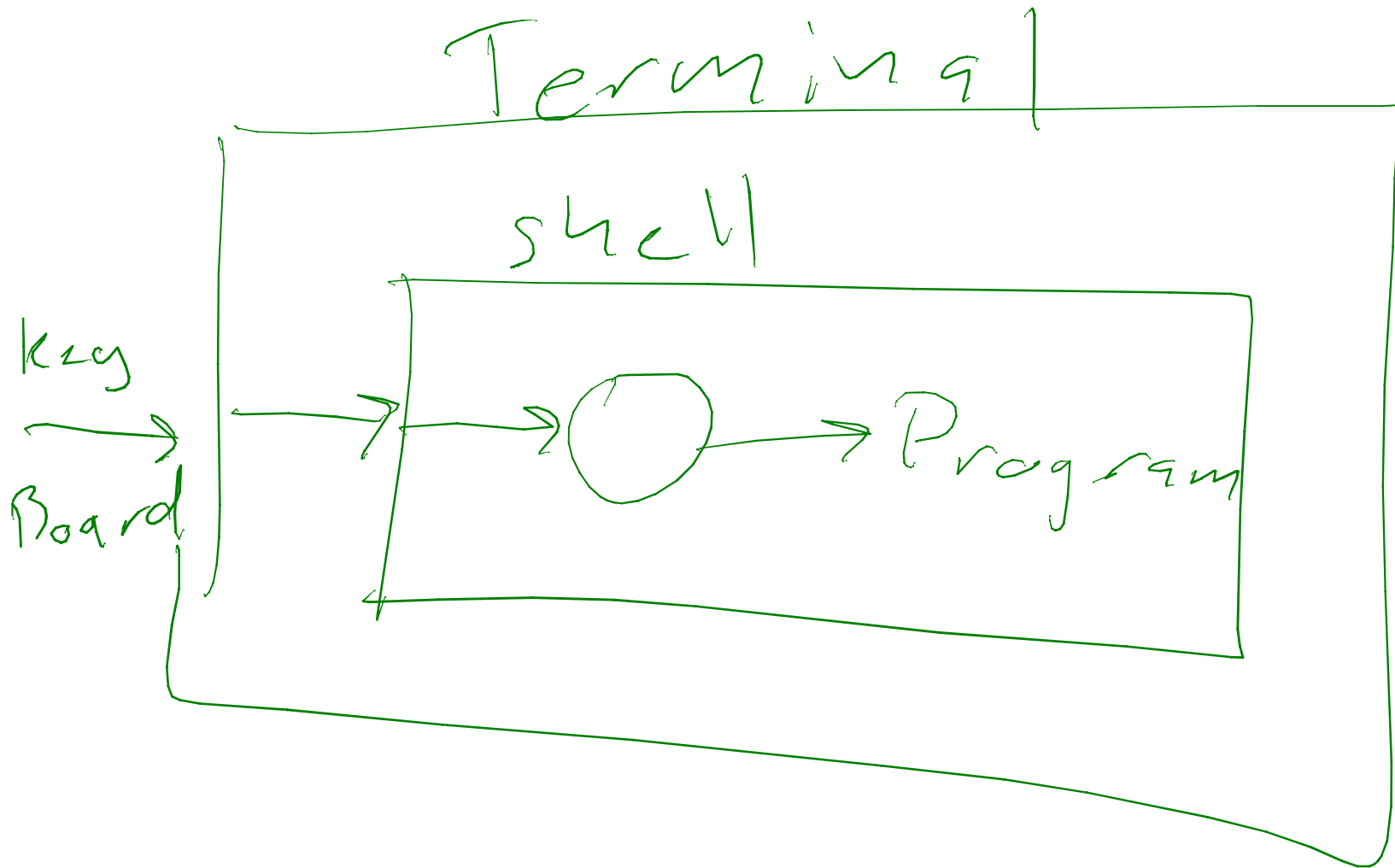
Bash Shell Programming Helps

- ▶ We use the Bash shell to orchestrate the chip building process
- ▶ Bash shell calls the other tools, does vector checking
- ▶ The shell script is a series of commands that the Bash interpreter executes

Bash Shell Programming Helps

▶ Bash shell scripts begin with a "she bang" followed by the path to the bash interpreter on line one.

▶ This tells the interpreter that this file is a shell script and also where the script resides. For a script file named "scriptfilename":

```
#!/bin/bash
echo "this is my first shell script"
```

▶ The shell script is not executable unless you make it so.

▶ The following adds execute permissions for the user.

```
chmod u+x <scriptfilename>
```

Bash Shell Programming Helps - Decisions with `if`

```
if [ condition_to_test ] ; then
    command1
elif
    command2
else
    command3
fi
```

-a bash.sh

► File based test conditions

Condition	True If
[-a file]	'file' exists
[-e file]	'file' exists (same as -a)
[-d directory]	'directory' exists and is a directory
[-f regularfile]	'regularfile' exists and is a regular file
[-s regularfile]	'regularfile' exists and has size >0
[-x excutablefile]	'excutablefile' exists and is executable by the shell

Bash Shell Programming Helps - Other test conditions

Condition	True If
[string1 == string2]	string1 is equal to string2
[string1 != string2]	string1 is not equal to string2
[numb1 -eq numb2]	numb1 is equal to numb2
[numb1 -ne numb2]	numb1 is not equal to numb2
[numb1 -gt numb2]	numb1 is greater than numb2

Bash Shell Programming Helps - Running VSIM

- ▶ vsim is always invoked on the module name, not the source file
  - ▶ `vsim modulename` **not** `vsim modulename.sv`
- ▶ Typical vsim invocation for use in bash scripts:
  - ▶ `vsim modulename -do dofilename -quiet -c -t 1ps`
- ▶ Minimizes chatter from the tool
- ▶ Runs without the gui
- ▶ Sets simulator to run with 1pS resolution
- ▶ For more vsim command line arguments, type: `vsim -help`

Bash Shell Programming Helps - File Redirection

- In BASH, there are three standard IO streams open:

stdin	file handle 0	standard input
stdout	file handle 1	standard output
stderr	file handle 1	standard error

- redirect stdout to file.txt, creating the file if necessary
  - `cmd > file.txt`
- forcefully redirect stdout to file.txt, even if **noclobber** is set
  - `cmd >| file.txt`
- redirect stdout to file.txt, appending to file.txt if it exists
  - `cmd >> file.txt`
- redirect stdout and stderr to file.txt
  - `cmd &> file.txt`
- pipe: send stdout of cmd1 to stdin of cmd2
  - `cmd1 | cmd2`

Bash Shell Programming Helps - Variables, translate

- Using a string variable to shorten names
  - `OSU_SYN_LIB="/nfs/guille/a1/cadlibs/osu_lib"`
- Dereferencing the variable using the dollar sign
  - `vlog $OSU_SYN_LIB/lib_ss_1.62_125c.v -work work`

- Translate lower case to uppercase in a list file using temporary file
  - `cat listfile | tr [a-z][A-Z] > $$tempfile`
  - `mv $$temfile listfile`

- `cat` sends the contents of `listfile` via a pipe into the translate utility `tr`. `tr` does the substitution and puts the results into a temporary file called `$$tempfile`. Then, `$$tempfile` is renamed to `listfile` with the altered contents.
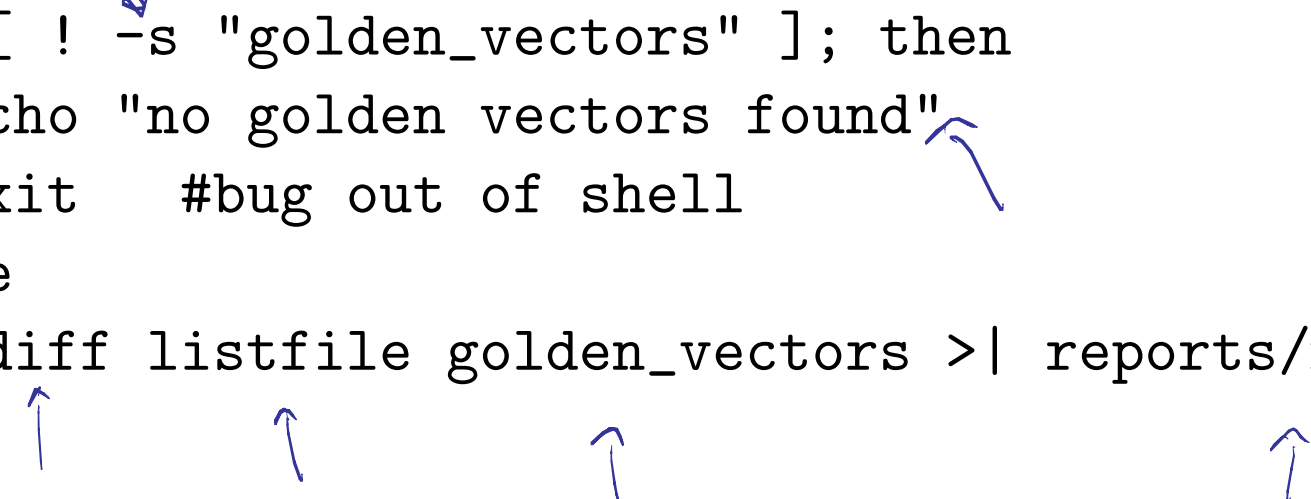
Bash Shell Programming Helps - Tee

- ▶ Sometimes you want to see the output of a compile on the screen and also also have it sent to a file. The utility `tee` does this for you. For example:
  - ▶ `vlog my_module.sv | tee compile_transcript`
- ▶ sends the output from `vlog` to the file named `compile_transcript` and also routes it to the screen.

Bash Shell Programming Helps - regression, diff, comments

- Check output vectors against a set of known good "golden" vectors.

```
if [ ! -s "golden_vectors" ]; then
   echo "no golden vectors found"
   exit    #bug out of shell
else
    diff listfile golden_vectors >| reports/miscompares
fi
```

- A check can then be made on the size of the miscompares file to see if there were any miscompares.

Bash Shell Programming Helps - good links

▶ For more information, see:

`http://tldp.org/LDP/Bash-Beginners-Guide/html/Bash-Beginners-Guide.html`

`http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html`

Makefiles

make

Block Diagram

counter

- >clk
- reset_n
- enable
- up
- down
- count_out

logic_blk

- >clk
- reset_n
- wr
- rd
- wr_ram
- rd_ram
- error_flag

clk
reset_n
write
read
data_in[7:0]

ram

- ram_rd
- ram_wr
- data_in
- data_out
- address

data_out [7:0]