

Build Flow Automation



by Iain Waugh



Overview

- Aim: Type "`make`" to get a bitfile that you can program into an FPGA
- Prerequisites
- Method for ISE projects
- Method for Vivado projects

Sample Code

- All sample code is here:

https://github.com/iain-waugh/fpga_dev_board

- You are free to clone and use it.
- Note the "hyphen"

Prerequisites

- Aimed at Windows users
- Linux users can skip parts where we set up a shell
- Mac folk need a VM

Get A Windows Posix Shell

- Windows doesn't have a "make" command
- You need to install either Cygwin or MSys2
- I prefer MSys2:
<https://www.msys2.org/>
- Update it: `pacman -Syuu`
- Install dev tools: `pacman -S base-devel`

Environment Variables

- ISE users on Windows 10:
Use the free Linux VM download from xilinx.com
Developers -> ISE Design Suite -> Downloads -> 14.7 Window 10
- Vivado users need to add to their PATH:
`<path to vivado>\Vivado\2018.3\bin`

Check

- Open an MSys (or POSIX) shell
- Can you run "`make -v`"?
- Can you run "`vivado -help`"?
- If not, go back and fix it

Make: How Does It Work?

- You describe the project dependencies in a "recipe"
- Make checks timestamps on dependencies that exist
- If a file doesn't exist, Make looks for a rule to build it
- Make rebuilds files that are older than dependencies
- If a dependency is not already on the filesystem or does not have a recipe, you get an error

Makefile Syntax Basics

- Define variables for use later on

```
variable = value
```

```
variable = value</pre
```

- "?=" sets a variable if its value hasn't already been set, such as in a global environment variable
- Use `${variable}` to access a variable's value

Makefile Syntax Basics

- Create target recipes

```
target: dependency1 dependency2 ...
```

```
<tab> command-line command
```

```
<tab> another command-line command
```

```
next_target: dependency1 dependency2 ...
```

- The <tab> character is essential for command lines
- The first target in the file is special; it is used when you type "make" just by itself.

Makefile Syntax Basics

- Q: You want to type "`make synth`" to run the synthesis process, but "`synth`" isn't going to be a file on the filesystem. What do you do?
- A: Define "`synth`" as a phony target like this:

```
.PHONY: synth
```

```
synth: ${PROJECT}.ngc ${PROJECT}.prj
```

ISE Make Flow

- Main command flow is:
 - Synthesis (HDL to netlist) - `xst`
 - NGD (Xilinx Native Generic Database) - `ngdbuild`
 - MAP (Database to specific part primitives) - `map`
 - P&R (Place and Route) - `par`
 - BitGen (Bitstream Generation) - `bitgen`

Dependency Limitations

- Limitation: Make doesn't know about each HDL source file unless you specifically tell it
- Maintaining that is very tedious
- Most projects store HDL file lists in a separate "project" file
- TL;DR: If you change one HDL file, just run the whole flow from the start

ISE Makefile Demo

```
fpga_dev_board\syn\ax309_board\Makefile
```

Vivado Make Flow

- Main command sequence is:
 - Load design/cores: `read_vhdl xxx.vhd`
`read_checkpoint xxx.dcp`
 - Synthesis: `synth_design`
 - Optimisation: `opt_design -verbose`
 - Place: `place_design`
 - Route: `route_design`
 - BitGen (Bitstream Generation): `write_bitstream`

My Vivado Simplification

- I only ever look at the synthesised result and the final routed result
- I've designed my "makefile" with just 2 options:
 - `make synth`
 - `make route`

Biggest Vivado Make Challenge

- Q: How do you pass arguments from a shell command to the Vivado ".tcl" script?

- A: With this command line:

```
vivado -mode batch -source ${SYNTH_NAME}.tcl \  
-tclargs ${OUT_DIR} ${SYNTH_NAME} ${DEVICE} ${PROJECT}
```

- The vivado ".tcl" script file starts like this:

```
## Read TCL script command-line arguments  
set OUT_DIR      [ lindex $argv 0 ]  
set SYNTH_DCP    [ lindex $argv 1 ]  
set DEVICE       [ lindex $argv 2 ]  
set PROJECT      [ lindex $argv 3 ]
```

Vivado Makefile Demo

```
fpga_dev_board\syn\zedboard\Makefile
```

Possible MSys Issue

- If you run "`make`" and see errors caused by Vivado looking for Linux files
- It's because MSys can make Windows look too much like Linux and Vivado gets confused
- Solution: Change "`vivado`" call to "`vivado.bat`"

Questions?