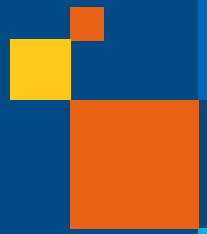
Flex Beijing FPGA Sharing

Quartus Scripting & Workflow

Yu, Chunxi

2022-09-19



intel®

Agenda

- Quartus Scripting
  - Environment Setup
  - FPGA Build Example Flow
  - Qsys Simulation Example Flow
  - References
- Repo Management
  - Basic Workflow
  - GitIgnore for IP/Qsys

* Quartus Prime Pro Linux for example

Quartus Scripting

- Why?
  - Automate flow
  - Easy to Integrate into other tools' flows
  - Tracked Modifications
  - Improve Performance (as said in User Guide)

1. Command Line Scripting

FPGA design software that easily integrates into your design flow saves time and improves productivity. The Intel® Quartus® Prime software provides you with a command-line executable for each step of the FPGA design flow to make the design process customizable and flexible.

The command-line executables are completely interchangeable with the Intel Quartus Prime GUI, allowing you to use the exact combination of tools that best suits your needs.

2. Tcl Scripting based on tcl language (https://www.tcl.tk/man/tcl)

You can use Tcl scripts, as an alternative to the GUI, to control the function and operation of Intel Quartus Prime software.

For example, you can use Tcl scripts to perform the following tasks:

- Manage Intel Quartus Prime projects
- Specify assignments and constraints
- Compile your design
- Perform timing analysis
- Generate and view reports about your project

Quartus Scripting – Environment Setup

- Configure License (could also be set in Quartus GUI in Windows)
  - `setenv LM_LICENSE_FILE 1800@altera02p.elic.intel.com:1800@altera05p.elic.intel.com`
- Add below paths into $PATH
  - `setenv PATH $QUARTUS_INSTALL_DIR/qsys/bin:$PATH` //for Qsys scripting
  - `setenv PATH $QUATUS_INSTALL_DIR/quartus/bin:$PATH` //for Quartus scripting and Quartus GUI

```
scc809028> ls /p/hdk/rtl/cad/x86-64_linux26/altera/quartus_prime_pro/21.4/qsys/bin/
ip-cache-check  ip-make-simscript           nios2-bsp-query-settings     qsys-edit
ip-catalog      ip-setup-simulation         nios2-bsp-update-settings    qsys-generate
ip-datasheet    nios2-app-generate-makefile nios2-lib-generate-makefile  qsys-script
ip-deploy       nios2-app-update-makefile   nios2-lib-update-makefile    qsys-validate
ip-generate     nios2-bsp-create-settings   pd-diff                      sim-script-gen
ip-lint         nios2-bsp-editor            pd-system-diff-tool
ip-make-ipx     nios2-bsp-generate-files    qsys-archive
scc809028> ls /p/hdk/rtl/cad/x86-64_linux26/altera/quartus_prime_pro/21.4/quartus/bin
clearbox                qcrypt              quartus_dse          quartus_rv
constra                 qemit               quartus_dsew         quartus_sh
dmf_ver                 qeslc               quartus_easm         quartus_sign
encrypt_1735            qfid                quartus_eda          quartus_sta
jtagconfig              qmegawiz            quartus_encrypt      quartus_staw
jtagd                   qnsm                quartus_fid          quartus_stp
jtagquery               qnui                quartus_fif          quartus_stp_tcl
juart-terminal          qpfgw               quartus_fit          quartus_stpw
mega_symc               qpgmt               quartus_ftk          quartus_syn
mega_symcng             qppl                quartus_hps          quartus_template
mif2hex                 qpro                quartus_ipgenerate   quartus_tlg
mw-regenerate           qpro_sh             quartus_jbcc         quartus_worker
nios2-flash-programmer  qred                quartus_jli          qwed
nios2-gdb-server        qreg                quartus_map          rdb_convert
nios2-terminal          qsme                quartus_npp          remote_debug_tester_app
openocd                 quartus             quartus_pdp          tclsh
openocd-cfg-gen         quartus-ip-catalog  quartus_pfg          uniphy_mcc
pgm_py                  quartus_asm         quartus_pgm          wish
pll_cmd                 quartus_bpps        quartus_pgmw         writeback
qatc                    quartus_cdb         quartus_pow          xcvr_diffmifgen
qbnl                    quartus_cmd         quartus_ptc
qcmd                    quartus_cpf         quartus_py
scc809028>
```

```
# Quartus in EC

set quartus_version = 21.4

modpath /p/hdk/rtl/cad/x86-
64_linux26/altera/quartus_prime_pro/${quartus_version}/quartus/bin
modpath /p/hdk/rtl/cad/x86-
64_linux26/altera/quartus_prime_pro/${quartus_version}/qsys/bin

setenv LM_LICENSE_FILE
1800@altera02p.elic.intel.com:1800@altera05p.elic.intel.com
```

Quartus Scripting – FPGA Build Example Flow

- `quartus_sh -t setup_proj.tcl`

```
# --- setup_proj.tcl ---
# new project
project_new filtref -overwrite
# Assign family, device, and top-level file
set_global_assignment -name FAMILY "Arria 10"
set_global_assignment -name DEVICE <Device>
set_global_assignment -name VERILOG_FILE filtref.v
# Assign pins
set_location_assignment -to clk Pin_28
set_location_assignment -to clkx2 Pin_29
set_location_assignment -to d[0] Pin_139
set_location_assignment -to d[1] Pin_140
# close project
project_close
```

Working Directory:

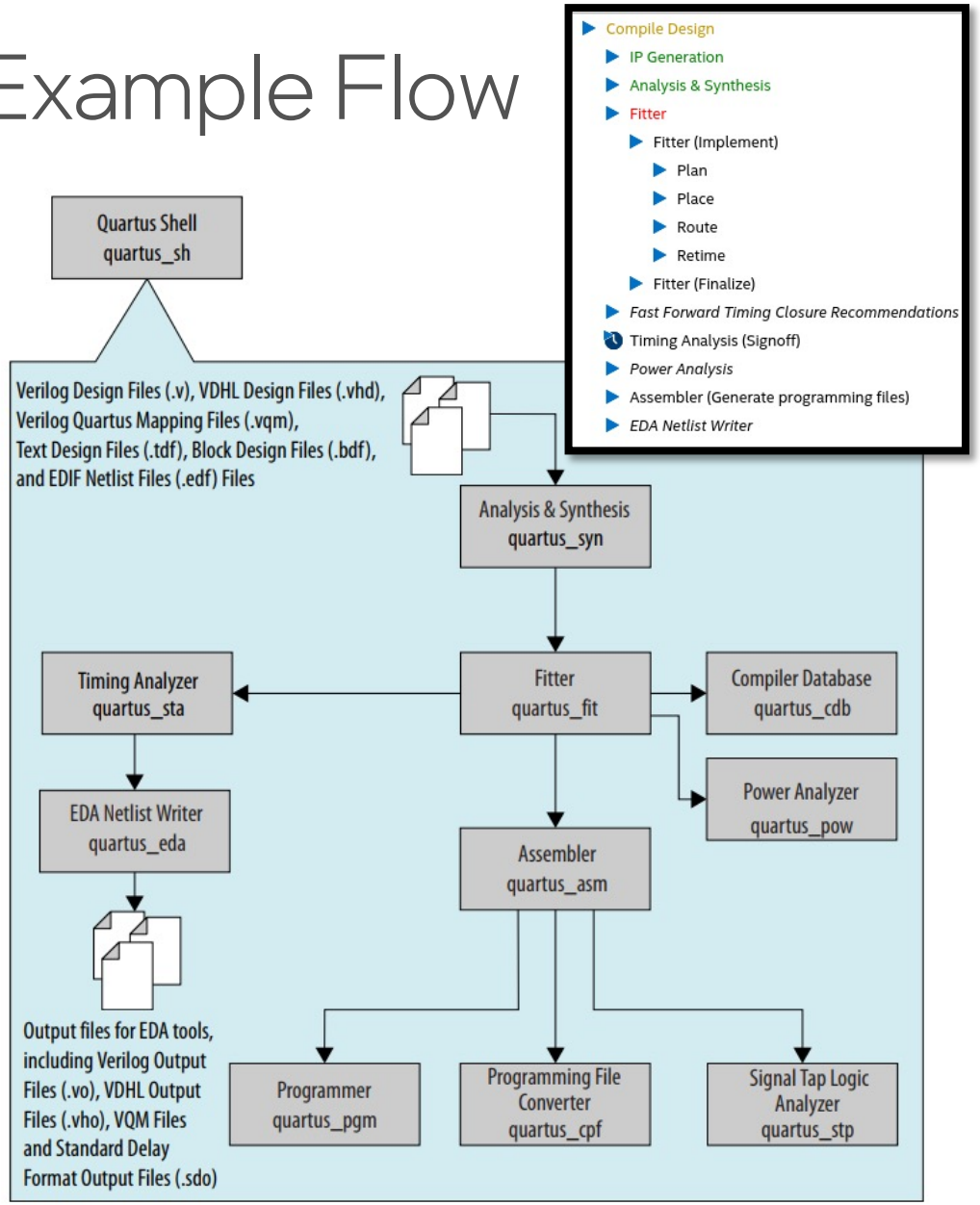
- <prj>.qpf
- <rev>.qsf: by default, <rev> is the same as <prj>
- ……

- `quartus_sh --flow compile filtref`

Note: difference between "quartus_sh" and "quartus_syn; quartus_fit; quartus_asm; quartus_sta"

`quartus_sh --flow`: flow can be "compile/implement/finalize/recompile"

`quartus_sh --flow`: flow are split into tasks, check tasks by `quartus_sh –flow <flow> <prj> -list_tasks`, control by "-start/end <task>"



Quartus Scripting – FPGA Build Example Flow

Shell (fpga_build.csh)

```
38 cd $work_dir
39 quartus_sh -t $MODEL_ROOT/synth/quartus/som_dpd.tcl
40 quartus_ipgenerate som_dpd -c $REV --generate_project_ip_files
   --synthesis=verilog --simulation=verilog --simulator=vcsmx
41 quartus_sh --flow compile som_dpd -c $REV
42 cd -
```

Tcl (som_dpd.tcl)

```
1  project_new som_dpd -overwrite
2  # Assign family, device, and top-level file
3
4  create_revision som_dpd_AGFB014R24A2E2V -based_on som_dpd -set_current
5  set_global_assignment -name FAMILY Agilex
6  set_global_assignment -name DEVICE AGFB014R24A2E2V
7  set_global_assignment -name TOP_LEVEL_ENTITY toplevel
8  source "$env(MODEL_ROOT)/synth/quartus/som_dpd_AGFB014R24A2E2V.settings.tcl"
9  source "$env(MODEL_ROOT)/synth/quartus/som_dpd.source.tcl"
10 source "$env(MODEL_ROOT)/synth/quartus/som_dpd_AGFB014R24A2E2V.qips.tcl"
11 source "$env(MODEL_ROOT)/synth/quartus/som_dpd_AGFB014R24A2E2V.pin.tcl"
12 source "$env(MODEL_ROOT)/synth/quartus./som_dpd.incdir.tcl"
13 source "$env(MODEL_ROOT)/synth/quartus./som_dpd.libdir.tcl"
14 source "$env(MODEL_ROOT)/synth/quartus./som_dpd.macros.tcl"
15 set_global_assignment -name SDC_FILE "$env(MODEL_ROOT)/synth/quartus/som_dpd.sdc"
16 export_assignments
17
18 create_revision msgdma_sa_sim_tb -based_on som_dpd -set_current
19 set_global_assignment -name FAMILY Agilex
20 set_global_assignment -name DEVICE AGFB014R24A2E2V
21 set_global_assignment -name TOP_LEVEL_ENTITY msgdma_sa_sim_tb
22 source "$env(MODEL_ROOT)/synth/quartus/som_dpd_AGFB014R24A2E2V.settings.tcl"
23 source "$env(MODEL_ROOT)/synth/quartus/som_dpd_AGFB014R24A2E2V.qips.tcl"
24 export_assignments
25
26 delete_revision som_dpd
27
28 project_close
```

```
<prj>.qpf
<rev1>.qsf
<rev2>.qsf
……
```

- 在 tcl 里加环境变量用 `$env(MODEL_ROOT)` 即可
- 添加 Source File
  - `set_global_assignment -name SYSTEMVERILOG_FILE source.sv`
- 添加 Macro
  - `set_global_assignment -name VERILOG_MACRO "use_input_a=1"`
- 添加 Include Path
  - `set_global_assignment -name SEARCH_PATH "./../../includes"`
  - `set_global_assignment -name VERILOG_INCLUDE_FILE "./../../includes/top_defines.vh"`
- 添加 Quartus IP File 或者 QSYS File
  - `set_global_assignment -name QSYS_FILE "./axi_crossbar_qsys/axi_crossbar.qsys"`
  - `set_global_assignment -name IP_FILE "./axi_crossbar_qsys/ip/axi_bridge_0.ip"`
- 添加 tcl File
  - `set_global_assignment -name SOURCE_TCL_SCRIPT_FILE "./hahaha.tcl"`
- 添加顶层使用的 Parameter，以 array parameter 为例 `{4'd3,4'd4,4'd3}`
  - `set_parameter -name NUM_OF_RX_FIFO "\{4'd3,4'd4,4'd3\}"`
- 添加其他的 Settings
  - `set_global_assignment -name ENABLE_INTERMEDIATE_SNAPSHOTS On`
- 添加 Pin Assignments
  - `set_location_assignment -to port_1 PIN_BR46` # location
  - `set_instance_assignment -to port_1 -name IO_STANDARD "1.8 V"` # io standard
  - `set_instance_assignment -name CURRENT_STRENGTH_NEW DEFAULT -to port_1` # drive strength

Note: difference between “source tcl” and “set_global_assignment –name SOURCE_TCL_SCRIPT_FILE”

Quartus Scripting – Qsys Simulation Example Flow

project name:
som_dpd

revision name:
msgdma_sa_sim_tb

qsys top:
msgdma_sa_sim_tb
(a Qsys, including msgdma, avmm bfm, etc.)

testbench:
msgdma_sa_sim_tb_wrapper
(provides clk, rst, transactions)

1. Create a directory to place the simulation files

```
23 cd $target_dir
```

2. Create Quartus project based on revision

```
25 quartus_sh -t $MODEL_ROOT/synth/quartus/som_dpd.tcl
```

3. Generate Qsys/IP files (tool: vcsmx)

```
26 quartus_ipgenerate som_dpd -c msgdma_sa_sim_tb --generate_project_ip_files
   --synthesis=verilog --simulation=verilog --simulator=vcsmx
```

```
scc809028> tree source/subsystems/AGFB014R24A2E2V/msgdma_sa_sim_tb/sim/
source/subsystems/AGFB014R24A2E2V/msgdma_sa_sim_tb/sim/
├── common
│   └── vcsmx_files.tcl
├── msgdma_sa_sim_tb.v
└── synopsys
    └── vcsmx
        ├── synopsys_sim.setup
        └── vcsmx_setup.sh

3 directories, 4 files
```

<tool_name>_setup.sh is used to do simulation compile of the Qsys and its dependencies. Also, it contains a script template if you have user source file to add to simulation.

Quartus Scripting – Qsys Simulation Example Flow

project name:
som_dpd

revision name:
msgdma_sa_sim_tb

qsys top:
msgdma_sa_sim_tb
(a Qsys, including msgdma, avmm bfm, etc.)

testbench:
msgdma_sa_sim_tb_wrapper
(provides clk, rst, transactions)

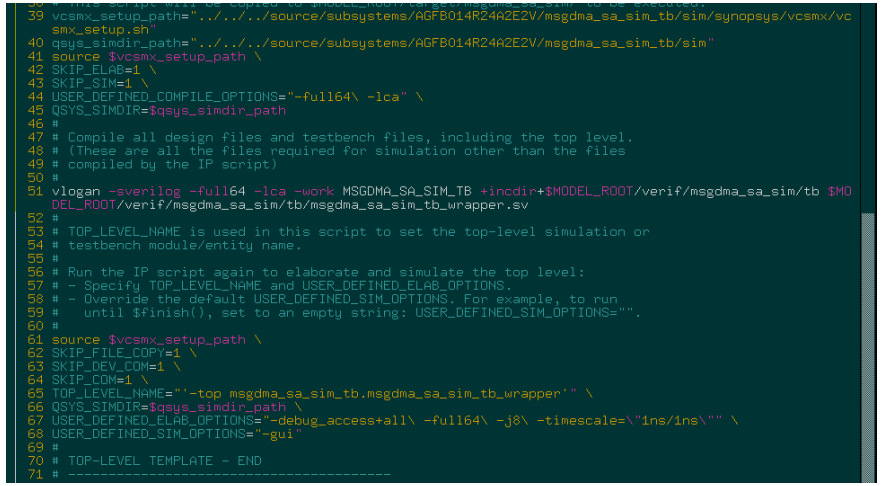
4. Copy necessary generated simulation files into the directory (user script template will tell you to do so)

```
30   cp $qsys_simdir/synopsys/vcsmx/* $target_dir/
```

5. Run user script to compile/elaborate/simulate

```
36 set msgdma_sim_script = $MODEL_ROOT/verif/msgdma_sa_sim/vcs/msgdma_sa_sim_vcsmx.sh
```

```
44 sh $msgdma_sim_script
```



User Script can be written according to the template in <tool_name>_setup.sh

1. Correct paths for setup.sh if you have a different path than Quartus thought.
2. Add compile step for all your user source files.
3. Modify compile/elaborate options as you want for your simulation. (Remember to escape spaces)

Quartus Scripting – Resources

- Quartus User Guide
  - Platform Designer: Command-line and Tcl scripting related to IP/Qsys
    - Intel® Quartus® Prime Software User Guides: Platform Designer - Command-Line Utilities
  - Scripting: Command-line and Tcl scripting (except Platform Designer)
    - Intel® Quartus® Prime Software User Guides: Scripting
- Command-line Help (other help methods refer to User Guide)
  - `quartus_xxx --help` for all arguments quartus_xxx can take
    - e.g., `quartus_ipgenerate --help`
  - `quartus_xxx --help=argument` for all values argument can take
    - e.g., `quartus_ipgenerate --help=ip_file`

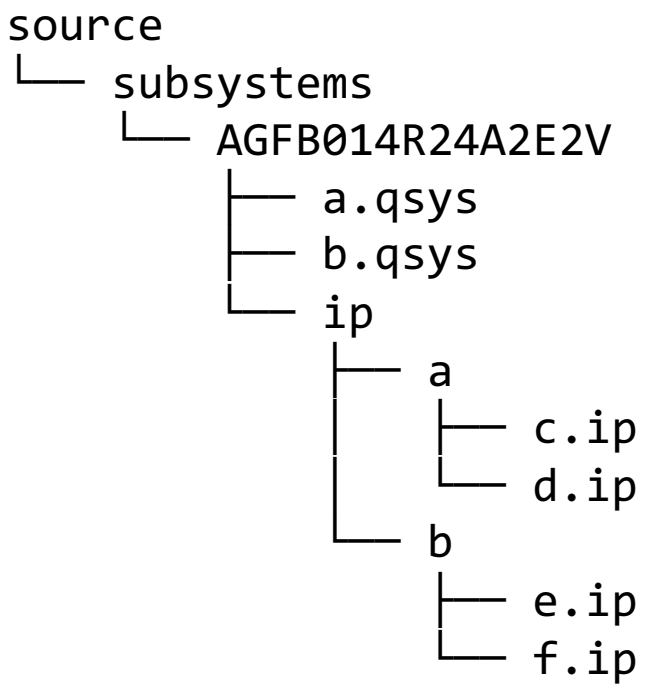
- Scripts feasible (but not written by me)
  - from exported tcl of Qsys to Ace hdl filelist (qsys-script, qsys-generate)
  - from Ace dumped filelist to Quartus tcl (not related to Quartus scripting)

Repo Management – Workflow

- Basic Workflow: Quartus flow using one-line command
  - Create a working directory (not tracked)
  - Create a Quartus project with appropriate revision and configuration
  - Run Quartus FPGA build flow (from ipgenerate to assembler)
  - Check results using scripts and send an email notification
  - Release image and related information to release area
- Notes
  - Be careful when switching branches.
  - You'll need to modify scripts inside ./synth/quartus for it to take effect next time.

Repo Management – Git Ignore

- To keep only .ip & .qsys inside source/subsystems path




```
# Quartus IP Gitignore
source/subsystems/**
!source/subsystems/**/
!*.ip
*.BAK.ip
!*.qsys
```

Explanation:

*: file and directory

/: directory

**: file and directory recursively

**/: directory recursively

files and subdirectories can be reserved only if parent directory is reserved.

Don’t even modify a “*” or “/”!!!


intel