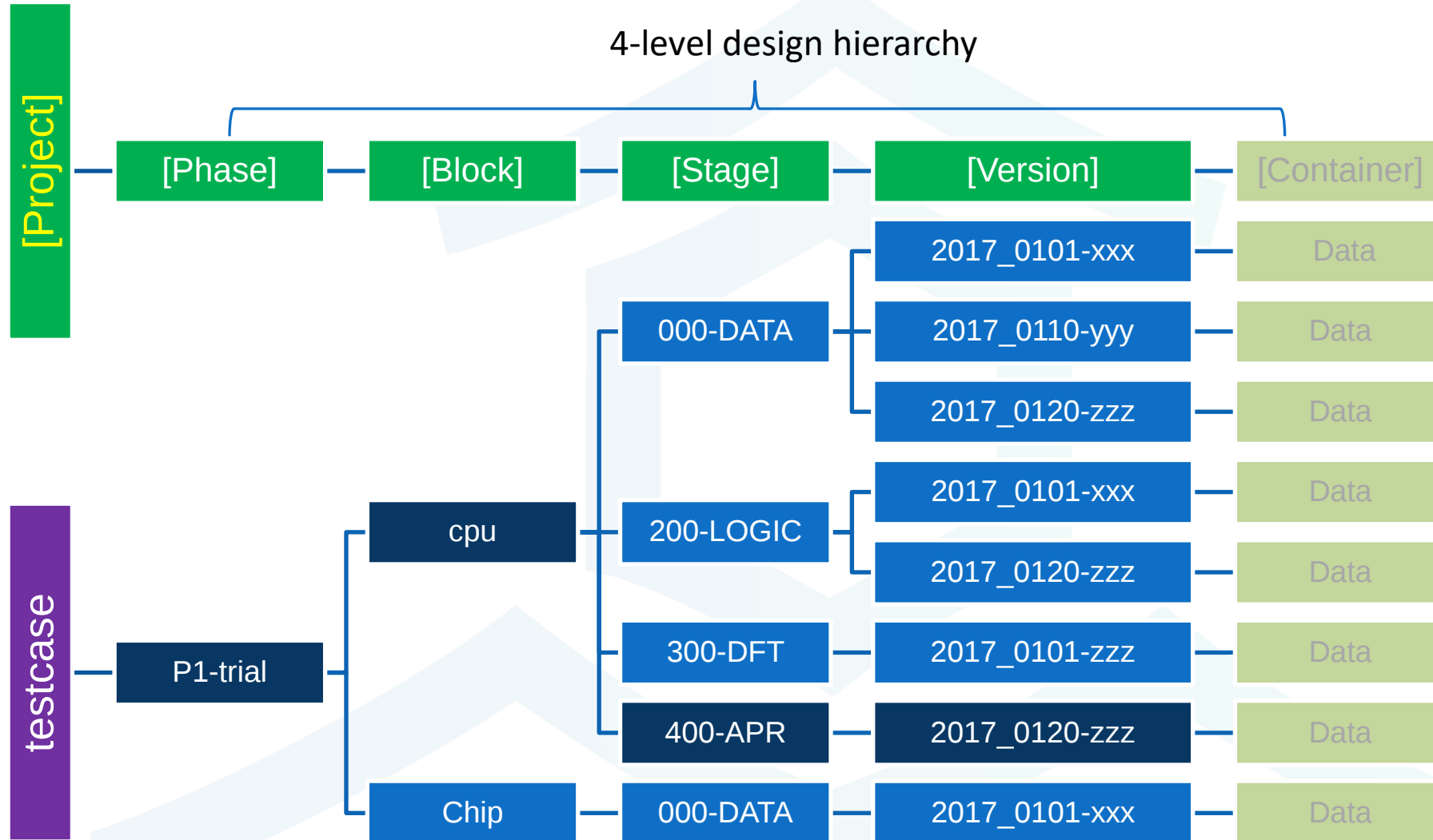


DVC Native Command Usage model

汇报人: Louis Liu

2025/5

Design Version Control Directory Hierarchy



<https://github.com/VirtualChip/dvc.git>

Design Version Hierarchy Browser

DESIGN::/ testcase/ P3-final / cpu/ 400-APR / 2017_0120-xxx

Phase

- P1-trial
- P2-stable
- P3-final
- P4-silicon

Block

- cpu
- chip

Stage

- 000-DATA
- 200-LOGIC
- 400-APR
- 500-SIGNOFF

Version

- 2017_0110-xxx
- 2017_0115-xxx
- 2017_0120-xxx
- 2017_0122-xxx
- 2017_0125-xxx
- 2017_0127-xxx

Container

- .
- place_opt
- route_opt

Filename	Size	Variable	Format	Description
block.v	100K	VFILE	VERILOG	Verilog Netlist File
blcok.sdc	20K	SDC_FILE	SDC	Timing Constraint File
block.def	3M	DEF_FILE	DEF	Physical Design File
block.spef.gz	120M	SPEF_FILE	SPEF	RC Extraction File
block.sdf	10M	SDF_FILE	SPEF	Standard Delay File
report_vio.rpt	20K	STA_RPT	RPT	Timing Report

Design Version Control Commands (2017_1025a)

Website: [<https://github.com/icdop/dvc/>]

Archive: [https://github.com/icdop/archive/2017_1025a.zip]

- Initialize Server
 - dvc_set_server ; dvc_get_server
 - dvc_init_server [start|stop]
- Create Project
 - dvc_create_project ; dvc_checkout_project
- Create Design (Phase/Block/Stage/Version)
 - dvc_create_design; dvc_checkout_design; dvc_checkin_design
- Data management with Container/Objects model
 - dvc_create_container; dvc_update_container; dvc_commit_container
 - dvc_copy_object, dvc_link_object, dvc_rename_object; dvc_remove_object;

SVN server global variables

- **dvc_set_server** <var_name> <value>
- **dvc_get_server --info**
- All server variables are stored in \$SVN_ROOT/.dop/env/

Variable Name	Default Value	Comments
SVN_ROOT	\$env(SVN_ROOT)	Must be set with unix environment variable
SVN_MODE	file	Available value are [svn file]
SVN_HOST	\$(shell hostname)	Default svn server hostname
SVN_PORT	3690	Default svn server port

How to start SVN server and create project repository

Assign svn server database path (default: \$HOME/svn_root)

```
setenv SVN_ROOT      /nfs/disks/svn_root
```

```
setenv SVN_MODE      svn
```

Initialize SVN server - (server pid/log are stored in \$SVN_ROOT/.dvc)

```
dvc_init_server
```

```
--root $SVN_ROOT          (Default: $HOME/svn_root)
```

```
--mode $SVN_MODE          (Default: file)
```

```
--host $SVN_HOST          (Default: localhost)
```

```
--port $SVN_PORT          (Default: 3690)
```

```
[start/stop/restart/status]
```

Create project repository - (db repo is stores in \$SVN_ROOT/testcase/)

```
dvc_create_project  testcase
```

DVC global design environment variables

- **dvc_set_env** <var_name> <value>
- **dvc_get_env** [--tcl | --csh | --sh | --csv | --info] [--all | <var_name> ...]
- Value of each variable is stored in \$PWD/.dop/env/\$var_name

Variable Name	Default Value	Comments
DESIGN_PROJT	—	Set by dvc_checkout_project
DESIGN_PHASE	:	Current working phase name
DESIGN_BLOCK	:	Current working block name
DESIGN_STAGE	:	Current working stage name
DESIGN_VERSN	:	Current working version name
DESIGN_CONTR	.	Current working container name

How to import data into design repository

Assign svn server database path (default: \$HOME/svn_root)

```
setenv SVN_ROOT      /nfs/disks/svn_root
```

Assign project name and create database root directory (default: '_')

```
dvc_checkout_project testcase [<PROJT_PATH>]
```

Create design version folder and checkin design files

```
dvc_create_design    P1-trial/chip/000-DATA/2017_0910-xxx
```

```
dvc_checkout_design
```

dvc_link_object	/nfs/disks/rundir	RUNDIR
-----------------	-------------------	--------

dvc_copy_object	/sfs/disks/rundir/design.v	design.v
-----------------	----------------------------	----------

dvc_copy_object	/sfs/disks/rundir/design.sdc	design.sdc
-----------------	------------------------------	------------

dvc_set_dqi	CELL_COUNT	11566
-------------	------------	-------

```
dvc_checkin_design
```

Handling APR design folder with multiple steps

Create design version folder and checkin design files

```
dvc_create_design      P1-trial/chip/400-APR/2017_0910-xxx
```

```
dvc_checkout_design
```

```
    dvc_link_object      /nfs/disks/apr/rundirRUNDIR
```

```
    dvc_create_container place
```

```
        dvc_copy_object  src/design.v              design.v
```

```
        dvc_copy_object  src/design.sdc             design.sdc
```

```
        dvc_set_dqi       CELL_COUNT                11566
```

```
dvc_checkin_container
```

```
dvc_create_container cts
```

```
    dvc_copy_object      cts/design.v              design.v
```

```
    dvc_copy_object      cts/design.sdc             design.sdc
```

```
    dvc_set_dqi          CELL_COUNT                12345
```

```
dvc_checkin_container
```

```
dvc_checkin_design
```

Handling STA design folder with multiple modes

Create design version folder and checkin design files

dvc_create_design *P1-trial/chip/520-sta/2017_0910-xxx*

dvc_checkout_design

dvc_link_object */nfs/disks/sta/rundirRUNDIR*

dvc_create_container *func/0p80V/ttt_85C_rctyp*

dvc_copy_object *src/design_func.sdc* *design.sdc*

dvc_copy_object *sta/000/design.rpt* *sta.rpt*

dvc_set_dqi *WNS* *50*

dvc_checkin_container

dvc_create_container *scan/0.72V/sss_125c_rcmax*

dvc_copy_object *src/design_scan.sdc* *design.sdc*

dvc_copy_object *sta/005/design.rpt* *sta.rpt*

dvc_set_dqi *WNS* *-100*

dvc_checkin_container

dvc_checkin_design

How to checkout data for design review

Checkout all designs(phase/block/stage/versions) in project_name
`dvc_checkout_project --recursive project_name [<projt_root>]`

Checkout all subfolders(blocks,stages,versions) under phase_name
`dvc_checkout_phase --recursive phase_name`

Checkout all subfolders(stages,versions) under block_name
`dvc_checkout_block --recursive block_name`

Checkout all subfolders(versions) under stage_name
`dvc_checkout_stage --recursive stage_name`

Checkout all subfolders(containers) under version_name
`dvc_checkout_version --recursive version_name`

Block1

000-DATA

2017_0110

410-FLOORPLAN

2017_0115

420-PLACE

2017_0120

430-CTS

2017_0122

440-ROUTE

2017_0125

2017_0126

2017_0127

README

- Post route hold fix with uncertainty 25ps

FILES

- block.v
- block.sdc
- block.def
- block.spf
- block.sdf
- sta.rpt

HISTORY

- 000-DATA/170910-ww38
- 200-LOGIC/170910-ww40
- 300-DFT/170910-ww41
- 400-APR/17910-ww43

2017_0127

01-LogicRule

L-01-ERC

02-Timing

T-01-SDC

03-Reliability

R-01-ESD

P-01-LVS

04-Physical

P-02-QEA

P-03-DRC

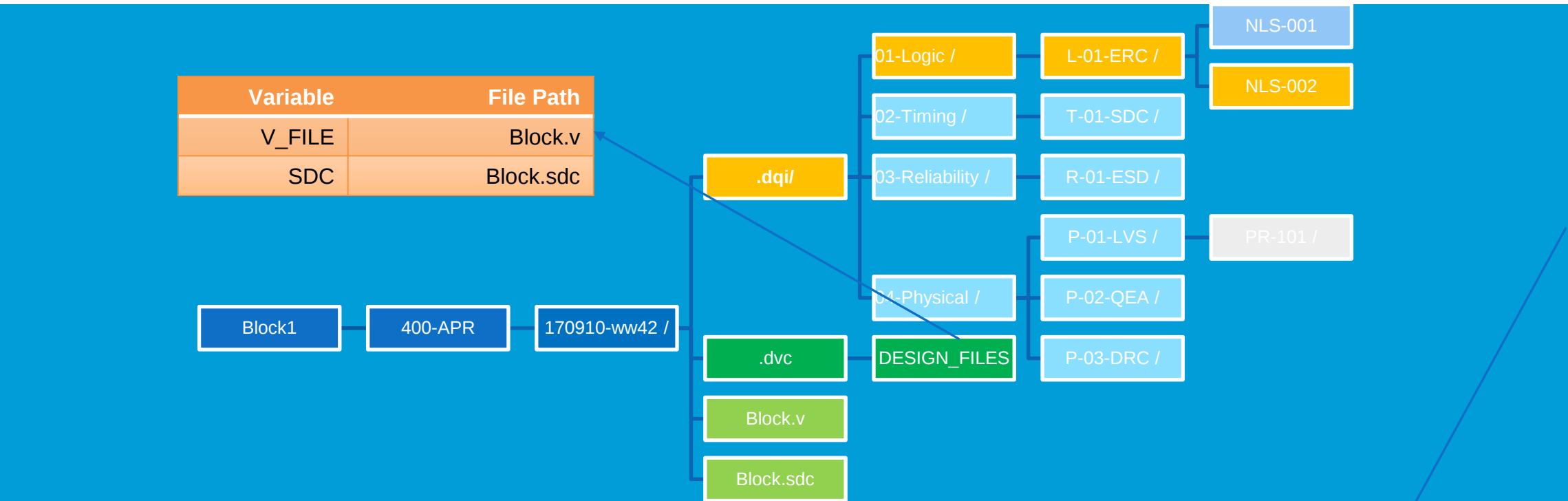
L-01-ERC

Type	Status	DQI	Title	Value	Unit
INFO	Manual	LR-001	Design Instance Count	100	K
WARNING	Auto	LR-002	Assign statement in netlist	20	
WARNING	Error	LR-003	Use Clock buffer in data path	50	
WARNING	Reviewed	LR-004	Floating inout pin	20	
ERROR	Error	LR-005	Floating input pin		
ERROR	Waived	LR-006	Unused I/O pad		

Quality Indicator
Detail Review

DQI Summary
by Category

DQI Value Update Tracing Log



NLS-002

Trace the DQI value change history and reviewer's comment

Value	Status	Time	User	Script	Comment
20	AUTO	2017_0127_1030	Mike	NLS.pl	/block1/305-postRoute/2017_0127/block1.v
5	ERROR	2017_0128_1405	York	Manual	Check assign statements, 5 suspicious
20	WAIVED	2017_0128_1525	Albert	Manual	Confirmed assign statements are all required

How to prepare a DQI Extraction plugin

- Create a directory with plugin name, prepare Makefile and required scripts
- Makefile Example:
design:
 # create design data container folder
import: DESIGN_FILES
 # copy the required files into design data container
check: REQUIRE_FILES
 # check if all required files exist
extract: check
 # extract DQI from report files and generate DQI.csv
save:
 # write to DQI.csv into DQI database
html:
 # generate HTML report using pre-defined template and annotate dqj value

Thank you

谢谢您

謝謝您

ありがとう
ございました