SILICON ACCESS™ NETWORKS

The iFlow Design Factory Infrastructure for a 17M-gate, 0.13μm, 333MHz design

Gilles-Eric Descamps (*)
ged@azulsystems.com

Satish Bagalkotkar
sbagalkotkar@yahoo.com

(*) now with Azul Systems

Agenda

- Common problems and their solutions
- Design Factory
  - Concept
  - Approach
  - Characteristics
- Elements of design factory
  - IT/CAD, Logic, IP, Physical Design
- Conclusion

Common problems

Common Problems

- Quality is traded-off to achieve dates
  - Constant pressure leads to cutting corners
- Teams are too inter-dependent
  - Source of inefficient use of resources
- Constant rework
  - Cost of change is not accounted
- Timing closure is unpredictable
  - Can impact schedule & performance
- Unqualified data causes snow-ball effect

Avoid Quality trade-offs

- “Measurable” quality requirement
  - 95% of paths using post-route spef should meet timing in Primetime
  - If you cannot measure, you will not be able to qualify
- Well defined qualification procedures using sign-off tools/scripts
- Qualification results reviewed by third party

Reduce team inter-dependency

- Hand-off complete only after qualification is met
  - Project leads handoff after signing on paper
- Margin based timing closure
  - Decouple FE & BE team
  - Let logic designers code RTL rather then chase false timing violation in BE
- Revision controlled handoffs
  - Customer can lock in with one version
- Well defined timeline for hand-off
  - No under the table transactions
  - Accumulate changes before hand-offs
  - Each hand-off has certain goals and deliverables

Avoid constant rework

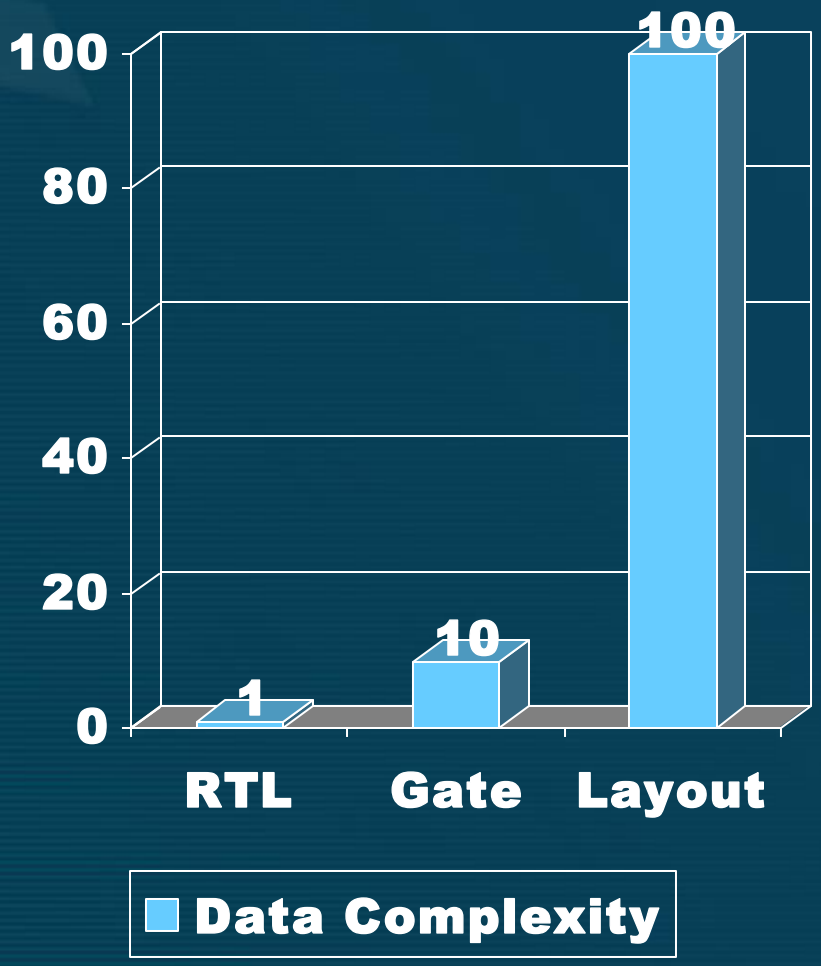
- Automated flow
  - Use of only Official hand-off versions
  - Eliminate user errors
  - Central queuing system for tools & machines
  - Allocated disk space
- Maturity
  - Always work on locked-in version
- Cost of change
  - Analyze how a change effects everything before making it

Timing Closure is unpredictable

- Rely on statistics
  - Test cases are run through the flow from RTL-to-tapeout quality
  - Metrics(runtime, timing, area etc.) are gathered as block goes through the flow
  - These metrics are then used to define margins at various phases(synthesis, placement, post-route) of design
- Margin based timing closure is single pass
  - As the block resembles the sample test case the results will be similar

Avoid snow ball effect

- “Pay me now or pay me later”
- Small problem can become overwhelming if gone undetected
- Catching the problem at source has huge payoffs



The “iFlow Design Factory”

What is a “Factory” approach ?

- Car manufacturer analogy:
- Various assembly lines produces parts.
- Automated forklift to transfer parts between assembly lines.
- Quality control procedures.

How does a Car “Factory” work

- Each assembly line is designed to do certain task
- Assembly lines are tuned to deliver desired quality output
- Inputs and Outputs from assembly line are sampled and qualified
- Forklift transfer parts between assembly lines
- Each worker follows a defined procedure
- Timeline on each assembly line is known
- Checkpoints are set for early problem detection

How can a Car Factory relate to Design Factory?

- The assembly lines are the design teams
  - logic, custom, physical, verification …
- Defined procedures
  - Standardized guidelines and methodology
  - automated flows tuned to deliver defined quality
- Tuned assembly lines
  - Tuning design process by sampling and regressions
- Automated forklift
  - Implemented through releases procedures
- Inputs & Outputs of assembly line are qualified
  - Before design, stdcell, memories, tech file, rtl etc are qualified
  - During design, by profiling based on metrics
  - At release time, by explicit checking for quality requirements

Why did we need a “Factory” ?

- Fast-growing startup (500%)
  - Multi-site Design Teams
  - Minimal employee ramp-up time
- Meet time-to-market with limited resources
- Design Complexity
  - 4 large SOCs (iPP, iAC, iAP, iCL)
  - 735 million transistors
  - 7.13 million standard cells
  - 0.13G TSMC process

What do you gain from a “Factory” approach ?

- Minimize risks
- Structured progress
- Organization-wide visibility
- Schedule / resources predictability
- Efficient usage of resources
- Eliminate non-creative actions
- Reduce dependencies

What are the characteristics ?

- Simplify
  - Single Methodology
  - Environment/tools
  - Directory organization
- Capture
  - Exchange/release
- Qualify

Simplify

- Single Methodology for all projects
- Standard environment/tools
- Company-wide directory organization
- Abstracting resources
- Single data source

Single Methodology

- Guidelines :
  - RTL coding
  - Project scheduling
  - Spare gates
  - Scan insertion
  - Timing closure
  - Circuit layout
  - DFT / DFM
- Standard implementation
- All chips follow the same design process

Environment / Tools

- Environment
  - normalized
  - one place for each thing (RTL, Circuit, BE, …)
  - one CAD team
- Tools
  - Gather statistics
  - Global License Broker
    - Platform LSF, or rtda FT/NC

Directory organization

- EDA tools go in:
  /tools/{vendor}/{tool}/{version}/…
- Technology files go in:
  /technology/{foundry}_{technology}_{process}/…
- Design Kits (stdcell lib…) go in:
  /products/{foundry}_{technology}_{process}/{vendor}/{product}/…
- Scripts go in:
  /scripts/current/bin
- Circuit / Custom deliverables go in:
  /delivered/{chip}/current/
- Logic (netlist) and Vault releases go in:
  /home/{chip}/sharedcache/{release}/…

Capture

- Consistent set
  - Expects well-known files
  - Standard sign-off tools to check results
  - Ensures ability to recreate
  - Data formats required by all customers
- Time-safe:
  - Version controlled
  - Data is migrated to a secure central place

Qualified Data Exchange

- Exchange / Release
- Automation & Knowledge Capture
- Metrics
- Thresholds & Color-Coding
- Dig-around Overview

Exchange / Release

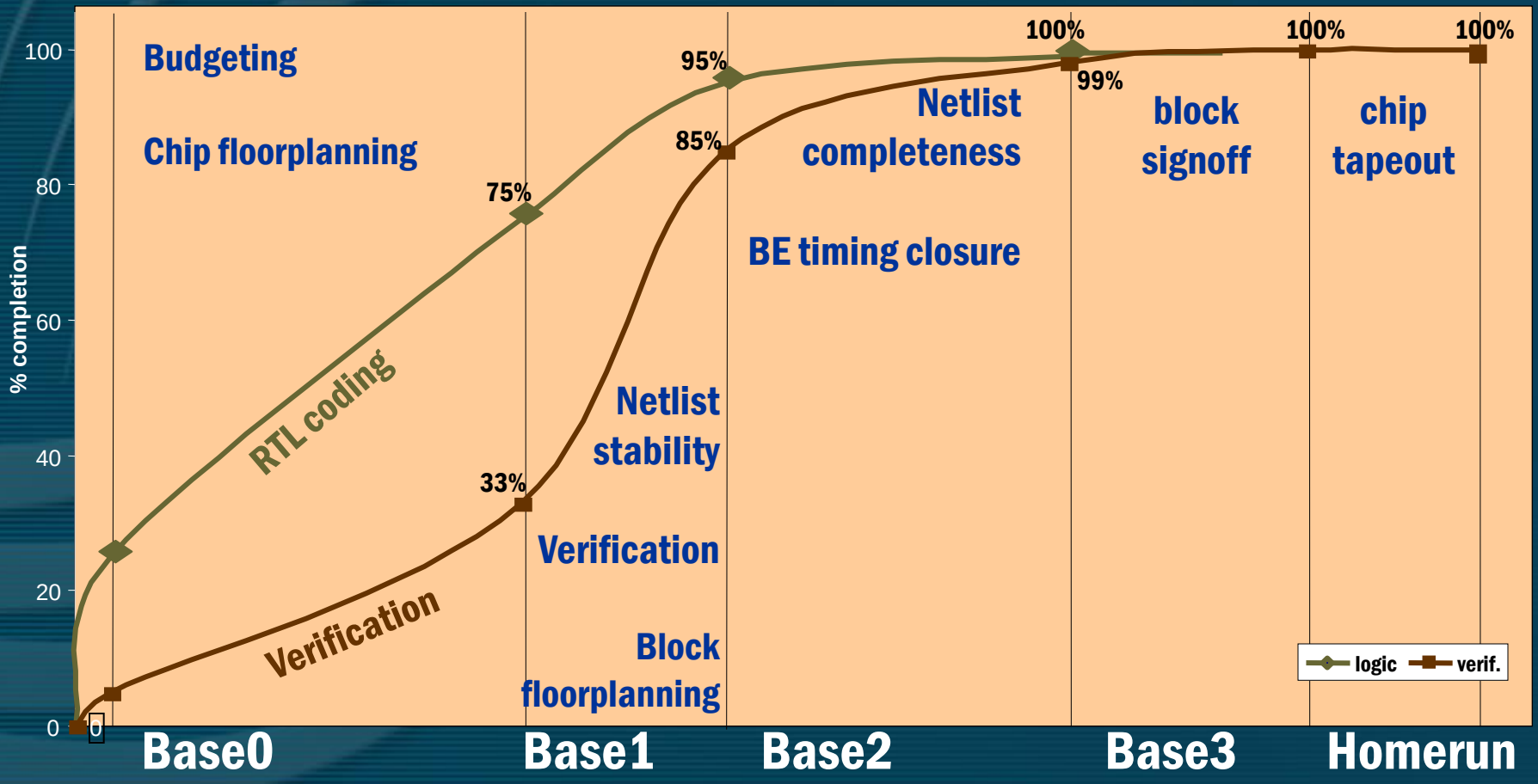
- No more email attachments
- publishing procedure
  - removes ownership of files when published
  - Captures time and date of files
- track 4W
  - Who, When, What, Why
- documented -> mirrored

Front-Loaded Methodology

Our Approach

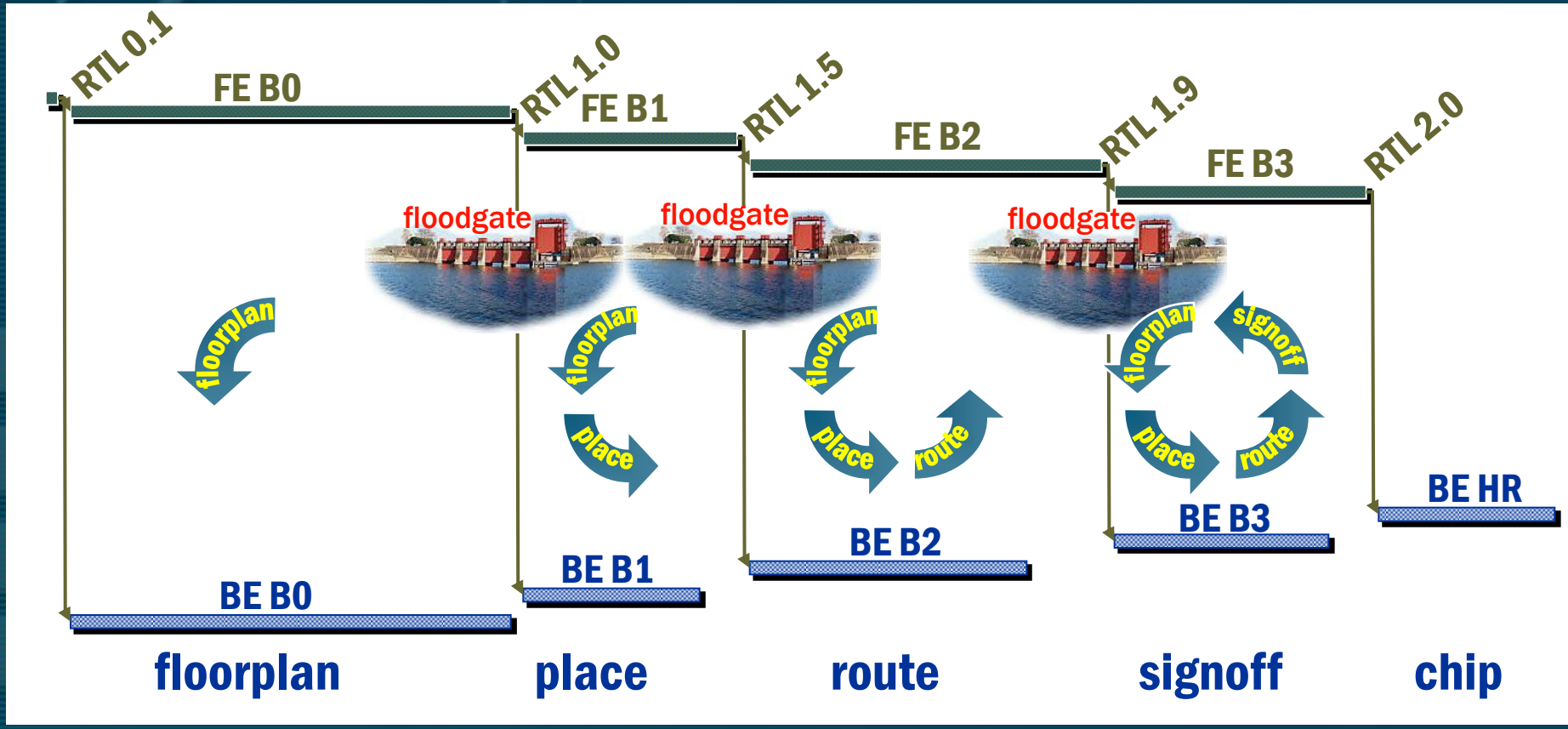
- Front-loaded methodology
- Decoupling of different aspects of design
  - RTL, verification, circuit, back-end, etc
  - Each team has its own specific goals
- Break down of the design timeline
  - Five phases with entry and exit criteria
  - Defined maturity for each phase
  - Structured progress for monotonic convergence
- Early detection of problems requiring trade-offs

Five phases for maturity



Monotonic Convergence

- Controlled releases from FE to BE based on predefined quality metrics



Elements of the Design Factory

IT
CAD
Logic
Hard Cores
Physical

IT / Abstracting resources

- Disk Storage
  - NAS, virtualization of disks
- Compute Machines
  - Central queueing system
- Tool Licenses
  - Management & Reporting
- Human resources
  - helpdesk

- Abstract Hardware
  - from: **`rlogin goliath`** to: **`run -r "sun7 RAM/20000 Apollo" myjob`**
  - Describe needs
- Abstract support
  - from: **`mail david`** to: **`open helpdesk ticket`**

CAD

- Centralized CAD team
- Proves solution before deploying
- Aggregate expert knowledge and define a single approach
- Automation focused
- Shared sense of urgency
- Helpdesk to prioritize requests

Logic Design

- Synchronous inter-block interfaces
- Margins inclusion into synthesis target period
  - Allows early detection of critical paths
    → appropriate micro-architecture trade-offs
  - Guarantees continuous timing closure
    → Ensures speedy convergence FE-BE
    → Margins are based on metrics from test cases
- Nightly build of design for QA
  - Compile - lint - mini regression - metrics gathering
    → Provides constant monitoring of design
    (no-cost status updates)

RTL nightly builds

- a software engineering practice
- RTL = software QA
  - lint
- Synthesis build = metrics
- Verification
  - daily fresh base
- Project Status for all / Visibility

Daily liveness status

DV apr09 nightbuild

To get the nightbuild RTL :

```
savc create -r ipp_s230_rtl_2006_NBapr09 -s build ipp/rtl ipp/setup
```

To get the nightbuild DV :

```
savc create -r ipp_s230_dv_2006_NBapr09 -s build reuse trc_system ipp/scripts ipp/verify
```

To get the nightbuild SYN :

```
savc create -r ipp_s230_syn_2000_NBapr09 -s build ipp/syn ipp/sta
```

Verification Sanity

Main Contacts: Parviz Yousefpour, Yuqing Niu, prime contacts

DV block name	rtl owner	runcsh	jobid	rundir	runlog	test name	test owner	test result
disp	gzhang	runcsh	2428229	rundir	runlog	ipp_disp_multi_ch_tb.e	yjpatil	ERR: FLEXLM Error: Unable to find Error-[UM] Undefined macro
						ipp_disp_sanity.e	yjpatil	ERR: FLEXLM Error: Unable to find Error-[UM] Undefined macro
ebas	sfrazer	runcsh	2428239	rundir	runlog	sanity.e	sfrazer	sanity_1.elog:SIMULATION PASSED
hash	qli	runcsh	2428259	rundir	runlog	crc_no_modulo.v	missing	crc_no_modulo.vlog:crc_no_modulo SIMULATION PASSED
						crc_with_modulo.v	missing	crc_with_modulo.vlog:crc_with_modulo SIMULATION PASSED
						rand_crc_no_modulo.v	missing	rand_crc_no_modulo.vlog:rand_crc_no_modulo SIMULATION PASSED
						reset_check.v	missing	reset_check.vlog:reset_check SIMULATION PASSED
iba	jjoseph	runcsh	2428265	rundir	runlog	ipp_blk_iba_ff_rw_lsram0_tb.e	missing	ipp_blk_iba_ff_rw_lsram0_tb_1.elog:25444:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_blk_iba_ff_rw_lsram1_tb.e	missing	ipp_blk_iba_ff_rw_lsram1_tb_1.elog:25444:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_blk_iba_ff_rw_qdr_36mb_pin3_tb.e	missing	ipp_blk_iba_ff_rw_qdr_36mb_pin3_tb_1.elog:42084:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_dack_gen_tb.e	missing	ipp_iba_dack_gen_tb_1.elog:12565:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_ocreq_wr_ctrl_tb.e	missing	ipp_iba_ocreq_wr_ctrl_tb_1.elog:21583:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_ocres_dp_hash_bk_tb.e	missing	ipp_iba_ocres_dp_hash_bk_tb_1.elog:12565:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_ocres_dp_oeg_bk_tb.e	missing	ipp_iba_ocres_dp_oeg_bk_tb_1.elog:12565:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_ocres_dp_onrs_tb.e	missing	ipp_iba_ocres_dp_onrs_tb_1.elog:13864:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
ibas	hding	runcsh	2428274	rundir	runlog	ipp_iba_ocres_dp_hash_tb.e	missing	ipp_iba_ocres_dp_hash_tb_1.elog:12525:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
						ipp_iba_oe_ocres_dp_hash_tb.e	missing	ipp_iba_oe_ocres_dp_hash_tb_1.elog:15184:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
icpss	nshah	runcsh	2428282	rundir	runlog	ipp_icpss_sanity.e	sshenoy	ipp_icpss_sanity_1.elog:67630:*MSG*trc_top_sim: SIMULATION PASSED with 0 WARNINGS
local_cam	ksanghan	runcsh	2428298	rundir	runlog	lcam_144_wr.v	aelnakhal	lcam_144_wr.vlog:= SIMULATION FAILED!!! At Cycle = 0000883b
						lcam_72_wr.v	aelnakhal	lcam_72_wr.vlog:= SIMULATION FAILED!!! At Cycle = 00008c5c
						lcam_lkup_144.v	aelnakhal	lcam_lkup_144.vlog:= SIMULATION FAILED!!! At Cycle = 00009824
						lcam_lkup_72.v	aelnakhal	lcam_lkup_72.vlog:= SIMULATION FAILED!!! At Cycle = 00009c45

Single-click access

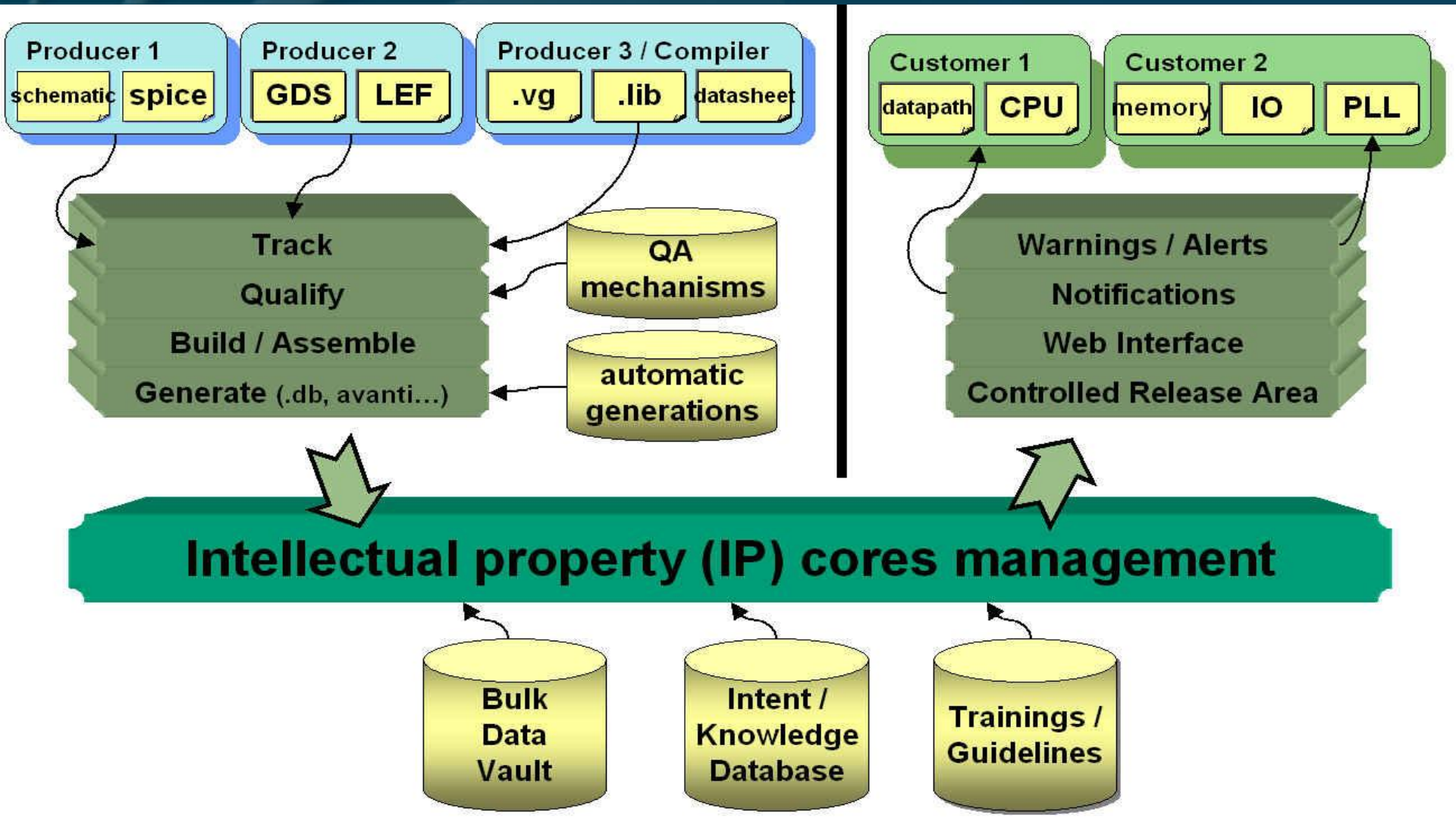
Hard Cores

- Custom design circuit
- Vendor IP re-verified & integrated

- 150 deliverables for 8 projects
  * 28 views per deliverable

  = ~8000 uploads / ~1500 releases
  ~365 000 managed files

IP Core Catalog



Single Core Overview

file: lef/l3_2_ra1st_4096x240cm8_0.lef **LEF - Library Exchange Format (LEF file)** What content should go into this file ? [2001-05-30 11:44:09] d:1113,1:104	latest final: **ul4** [chg2final] [chg2phldr] by awong on 2002-02-08 22:08:39 Browse CVS Vault upload new	- QA script [2001-08-16 17:47:12] - QA description [2001-06-13 13:44:24] - QA report: [2002-02-08 22:09:58] - QA status: passed by awong redo QA
file: cdump/l3_2_ra1st_4096x240cm8_0.cdump **Silicon Perspective First Encounter (Cdump file)** - no content description available d:1113,1:114	latest final: **ul7** by awong on 2002-02-08 22:08:49 Browse CVS Vault **AUTOGENERATED**	- GEN script [2001-04-23 15:57:51] - GEN report: [2002-02-08 22:10:03] redo autogenerate

- Document, Track, Qualify
- autogenerated views
- Queue jobs (QA, gen)

Common Myths in Physical Design

- *Good tools can build complex chips*

  Bad flow = Bad methodology + Bad tools
  Medium flow = Good methodology + Bad tools
  Good flow = Good methodology + Good tools

- *Doing "early & often" helps schedule*

  Immature data results in illusion of progress

- *Common solution are too costly*

  It is easier to steer one ship then 100 boats

- *Each block needs its own flow*

  Single flow can be designed with options

Correct-by-construction approach

- Top-level Clock H-trees
  - Using tunable buffers to reach a 5ps skew
- Noise avoidance and correction
- Custom-designed power grid
- Length based buffer insertion at top-level
- Margin based timing closure
- Area growth statistics

Physical Design Flow

- One-Button Gates-to-GDSII block flow
- Best of Breed 15+ tools, 60+ steps
- Single color-coded GUI
- 4 exit points – floorplan, place, route, signoff
- Data & results captured, qualified and published on a web page

Ensure flow stability

- New tools are integrated in parallel to existing tools as conditional elements
- Each tool version is tested before deployed
- Silicon proven blocks are used for flow regression
- Nightly and weekly regression ensures flow is fully operational
- Detail metrics are gathered and archived for reference
- Results of each regression run ensures quality
  - Flow is tuned to deliver tapeout quality results on all regression test cases

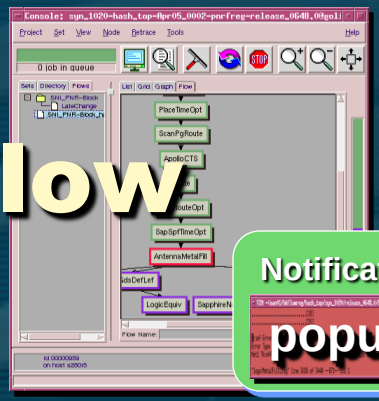
Block Flow

nightly regressions

testcase 1

testcase 2

testcase 3

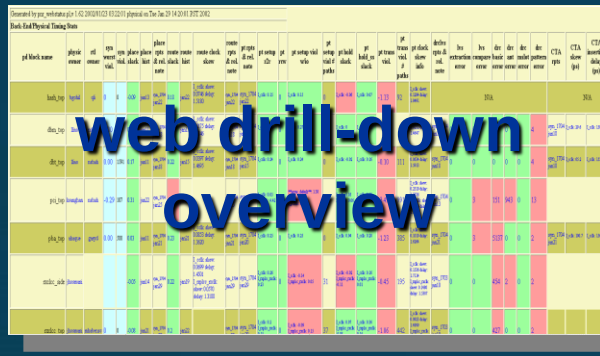


user interface

Notification

popup

email



web drill-down overview

capsule synopsys

capsule cadence

capsule magma

.vg .LEF .DEF .spf .spef .sdc

Milkyway

Resource management

Netlist Floorplan Timing DFT

Database GDS, .vg Reports Profiles

Project Knowledge

global.params

Design Dependent Knowledge

user.params

Design Process Knowledge

templates

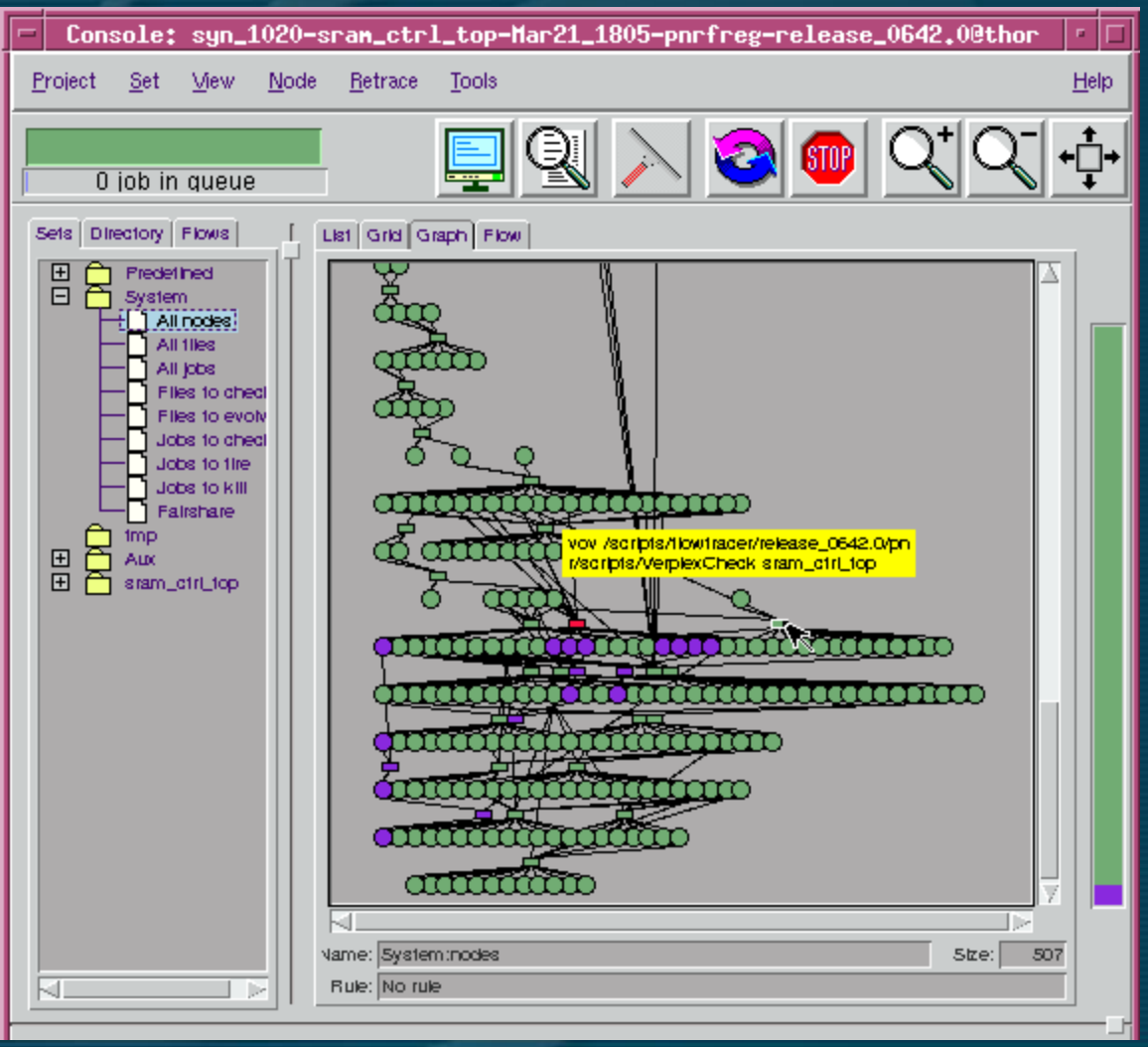
Distributed File watchdog

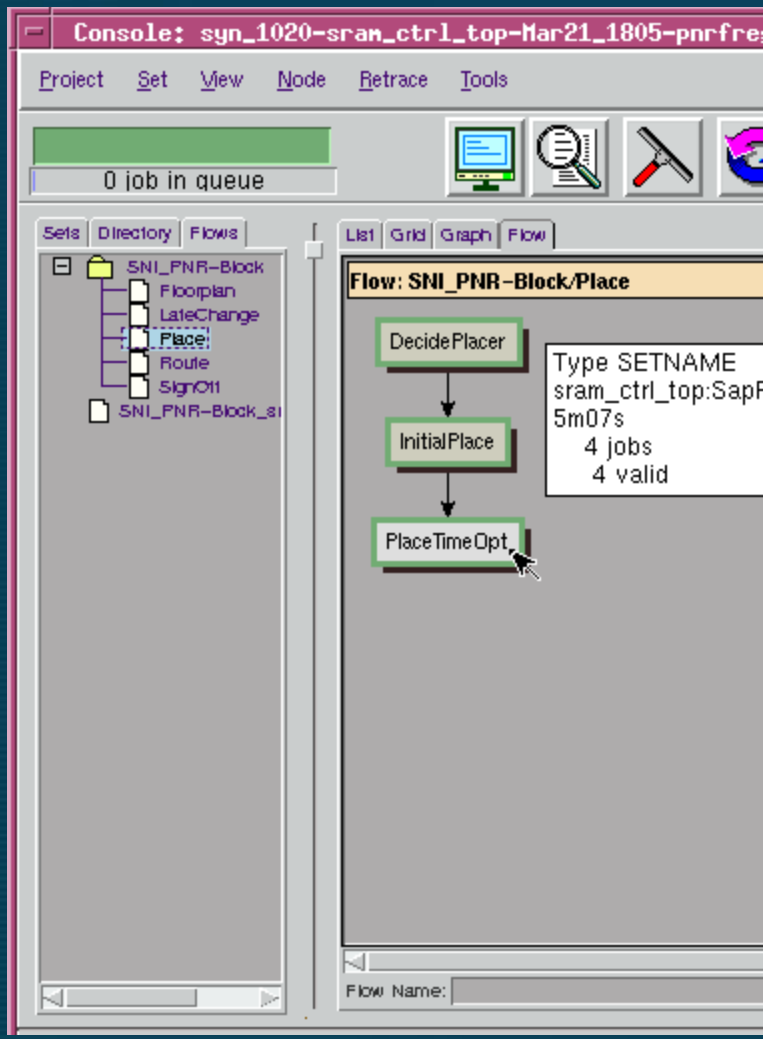
licenses

Compute Server Farm

September 2003

physical block flow graph



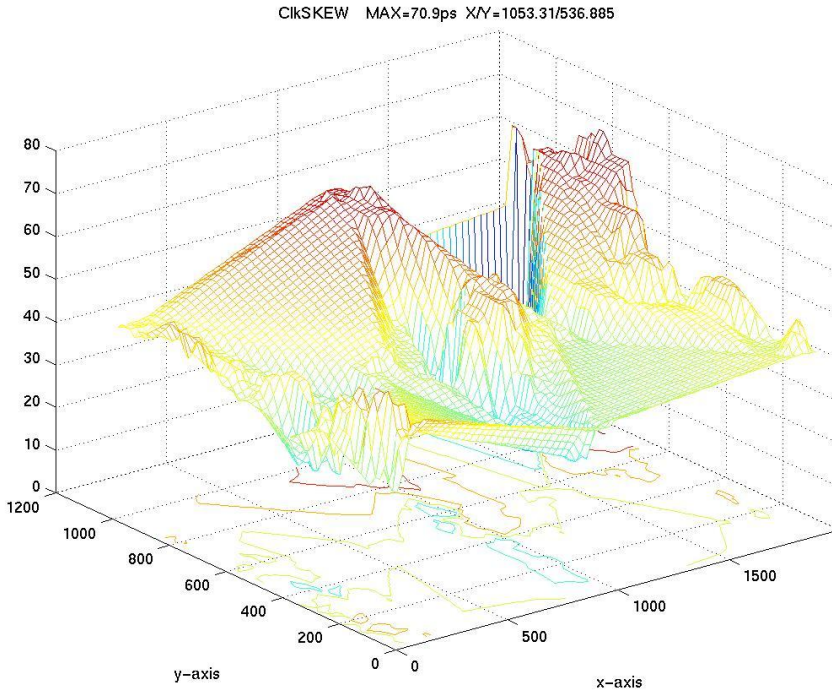


- r t d a Flowtracer

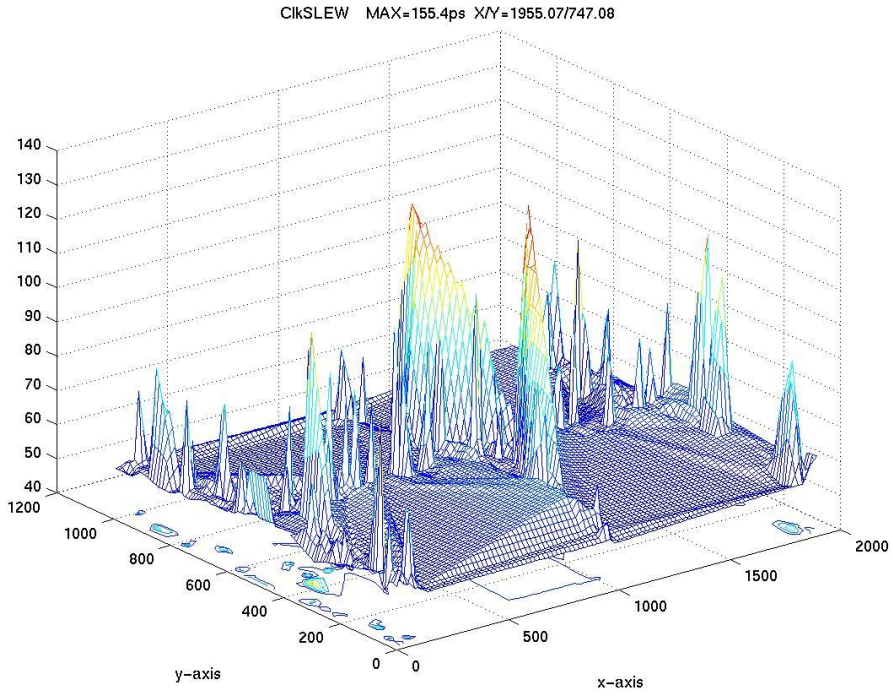
Physical Block Overview

syn viol.	place slack	place hist	place rpts & rel. note	route slack	route hist	route clock skew	route rpts & rel. note	pt rpts & rel. note	pt setup r2r	pt rcw	pt setup viol w/io	pt setup viol # paths	pt hold slack	pt hold_ss slack	pt trans viol.	pt trans viol. # paths	pt clock skew info	drclvs rpts & rel. note	lvs
	-0.09	jan13	syn_1704 feb20	0.22	feb19	I_cclk: skew: 0.0746 delay:	syn_1704 feb20	syn_1704 feb19	I_cclk:	0	I_cclk:	0				61		syn_1704 feb20	0
	0.12	jan29	syn_1704 feb15	0.24	feb06	I_cclk: skew	syn_1704 feb15	syn_1704 jan29	I_cclk:	0	I_cclk:	87				0		syn_1704 feb15	0
	0.06	feb05	syn_1705 feb15	-0.01	feb08	I_cclk: skew	syn_1705 feb15	syn_1705 feb07	I_cclk:	0	I_cclk:	1				4		syn_1705 feb15	0
	-0.01	feb20	syn_1705 mar23	-1.04	mar22	I_cclk: skew I_pciclk: sk	syn_1705 mar23	syn_1705 mar23	I_cclk: I_pciclk:	203	**async_default**: I_cclk: **default**: I_pciclk:	8				5198		syn_1705 mar23	0 (warn
	0.05	feb09	syn_1705 feb21	-0.04	feb21	I_cclk: skew	syn_1705 feb21	syn_1705 feb18	I_cclk:	0	I_cclk:	14				123		syn_1705 feb21	0 (warn
	-0.04	jan29	syn_1704 mar18	-0.08	feb13	I_cclk: sk I_rxplcc_	syn_1704 mar18	syn_1704 mar15	I_cclk: I_rxplcc_rxclk:	0	I_cclk: I_rxplcc_rxclk:	0				84		syn_1704 mar18	0 (warn

Single-click access



•Clock Skew



•Clock Slew Rate

Conclusion

- Margin based timing closure ensures predictability
- Early emphasis on quality significantly improved overall schedule
- Qualified deliveries at milestones
- Capture & Measure provide company-wide visibility into the project