

---

# Aurora 2001

Hierarchical design of a large graphics chip



Aurora/2001



# Introduction

---

- Founded 1997, ex-SGI: Paul Rodman, Margie Levine
- Physical Design services..."flow" ASP model
- 1999: CEO David Gregory
- Expert staff of physical design + software engineers
- Special expertise in hierarchical Avant! flows



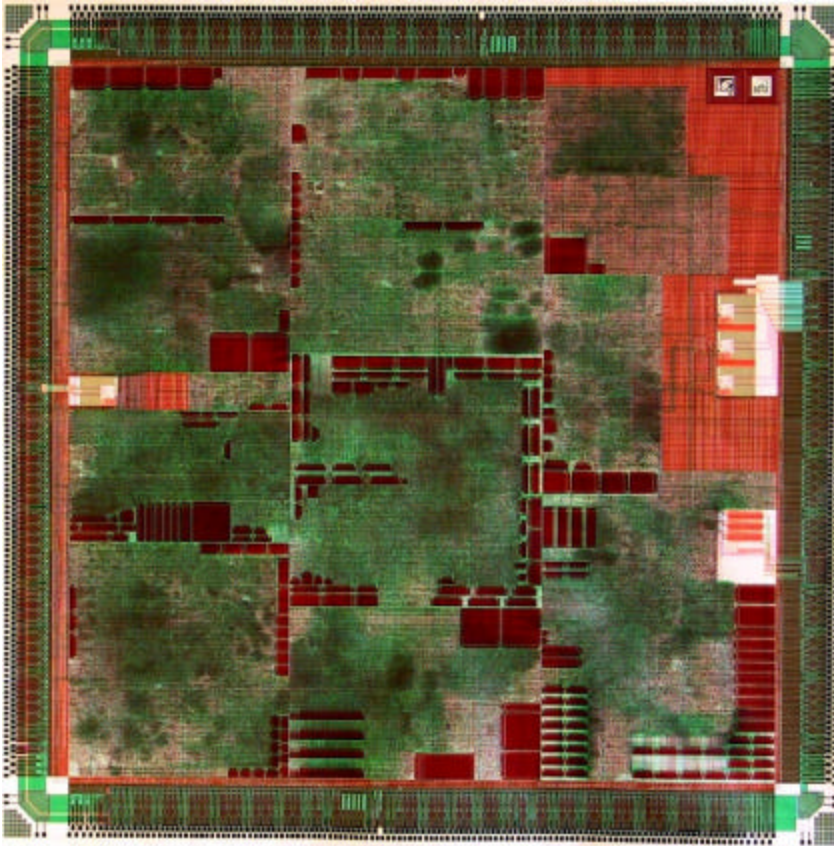
# Rampage design @ 3dfx

---

- New architecture in design, a very complex chip.
- Our experiments had showed Avanti! Tools QOR and speed much better than Cadence
- But problem: existing flat flow very slow for for 300k - 500k instance chip.
- Task: ReShape to develop hierarchical flow using Avanti toolset.



# “Rampage”



- TSMC/UMC .18u, 6LM
- pad limited, 35u pitch
- die area: 11.44 x 11.47 mm
- core: 10.6 x 10.6 mm
- ~1.5 Million placeable objects
- 30 Million xtors
- 200+ rams
- 18 clock domains
- 14 core blocks
- 8 pad ring blocks
- >10,000 repeaters (BUFX20)

# Hierarchy, pro

---

- Block level parallelism, for *both* computers and humans
- smaller netlists:
  - faster runtime
  - fewer tool problems
  - QOR better
- Partially complete design can start PD work
- more deterministic replay from new netlist releases
- late netlist changes possible effecting only subset of design

# Hierarchy, con

---

- Horizon effect
  - QOR issues ( e.g. congestion due to bad pin locations)
  - timing issues ( e.g. max transition on global wires)
  - DRC issues on block boundaries
  - Global structures, e.g. clock/power/repeater distribution more complex
- Data management issues
  - File explosion: 10,000 files for one netlist build
  - Experiments easy, therefore even more files around
  - Sharing data not always easy
  - Chip integration: which is the block we tapeout?

# Hierarchy, con (part II)

---

- Channels
  - cost area
  - bad for coupling
  - wasteful of metal: unused routing resources inside block
  - bad for timing: signals without feedthrus must go around blocks
  - complex flow: feedthrus make for complex verification problems
- Top level floorplanning required
  - extra work
  - painful to fix if block sizes change

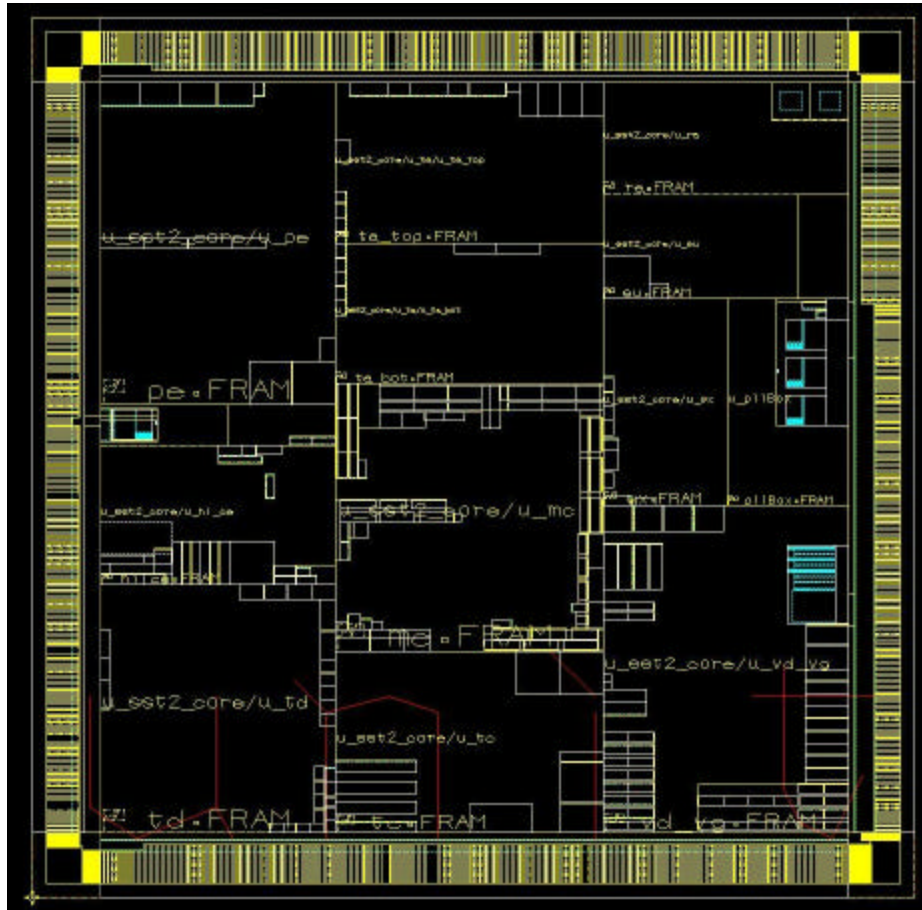
# Partitioning

---

- Relative sizes should be in similar range or floorplanning becomes difficult without wasting space, we grouped smaller netlists in two cases for this reason.
- “Reasonable” turn-around times imply blocks < 200k instances in size (We cut “ta” block into two blocks and wish we had cut “pe” block)
- pins/mm<sup>2</sup> ratio should be “reasonable”: mc block quite high
- global timing issues need to be addressed. RTL design paradigm can help, e.g. :
  - register--> buffer --> [chip crossing repeaters+wire] --> input gating --> setup + skew
  - no snake paths if possible



# Rampage floorplan



Name	Kinsts	Term	%util
pe	254	2546	72
mc	123	6100	59
hice	97	4239	72
ta1	132	3707	63
ta2	129	1601	65
tc	123	2191	73
td	235	2435	79
vdvg	179	1714	73
vp	15	906	64
ra	66	425	51
su	68	639	55
wx	87	1118	62
pll1	.8	352	16
pll2	.8	1108	26
east	2.8	286	41
west	10.0	885	46
north	5.3	341	56
south	5.3	335	56

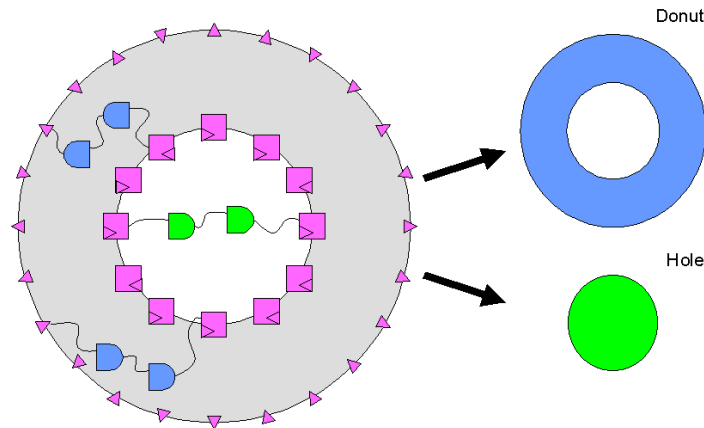
# Concurrent Logical/Physical Design

---

- The way humans work on complex problems
- Extraction
  - Placement-based RCs ok and FAST.
  - Do a full route and correlate with StarRC-XT from time to time
- RTL engineers converge on *real timing*, congestion problems are detected ASAP.
- Wireload models
  - generated per block
  - only used for synthesis run, crude A vs. B netlist comparisons
  - 85% percentile by default, increased if ECO size > ~10% of cells
  - useful rule of thumb: if 10% cells change ->  $10/4 = 2.5\%$  area increase

# Global Timing

- “Donut flow” used for top level timing and convergence ECO
  - split block level netlist into “donut” and “hole” (core)
  - split block level parasitics “ “ “ “
  - top level netlist + abutment dspf (0 ohm jumpers) connects donut R and C.

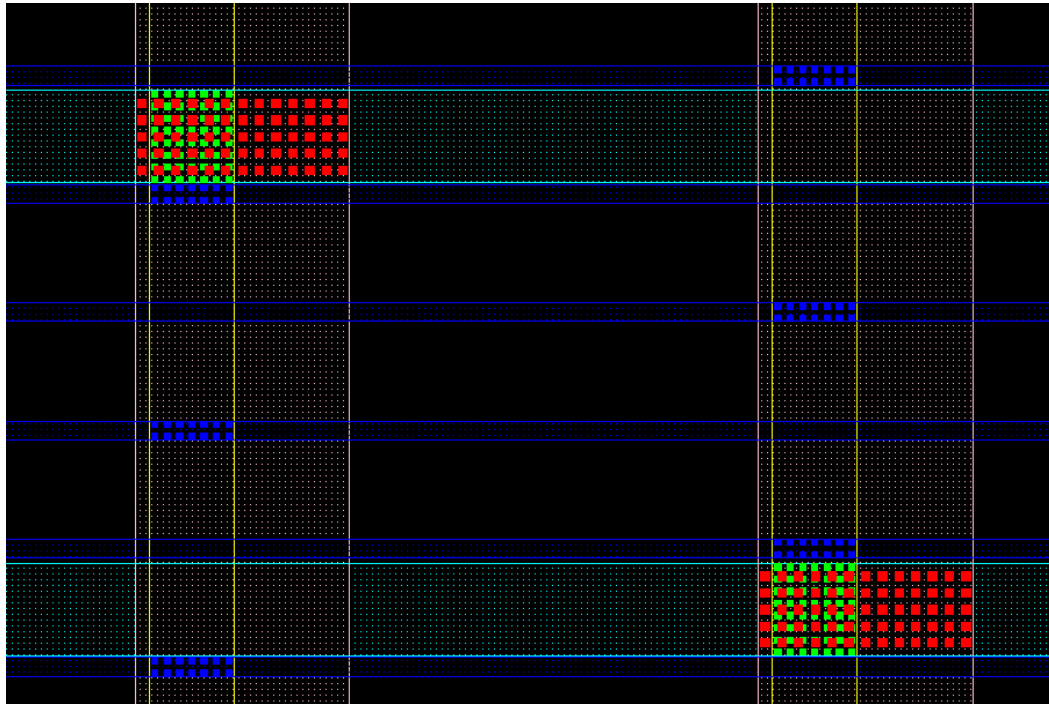


# Global Timing, 2

---

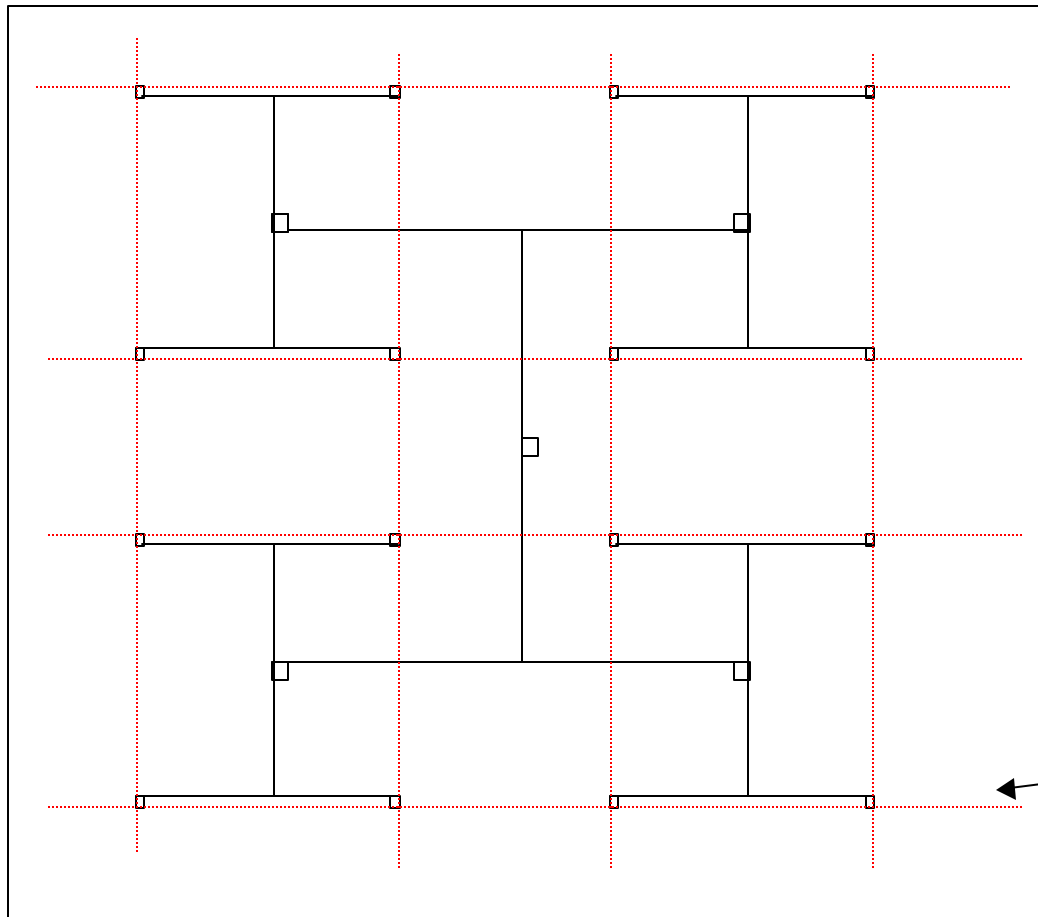
- Top level timing run now ~4X smaller
- Ratio somewhat indicative of partition quality
- Top level timing + ECOs in 2.5hrs
- No block level constraints required except “easy” ones
- Split ECO into ECO-per-block

# Power Distribution



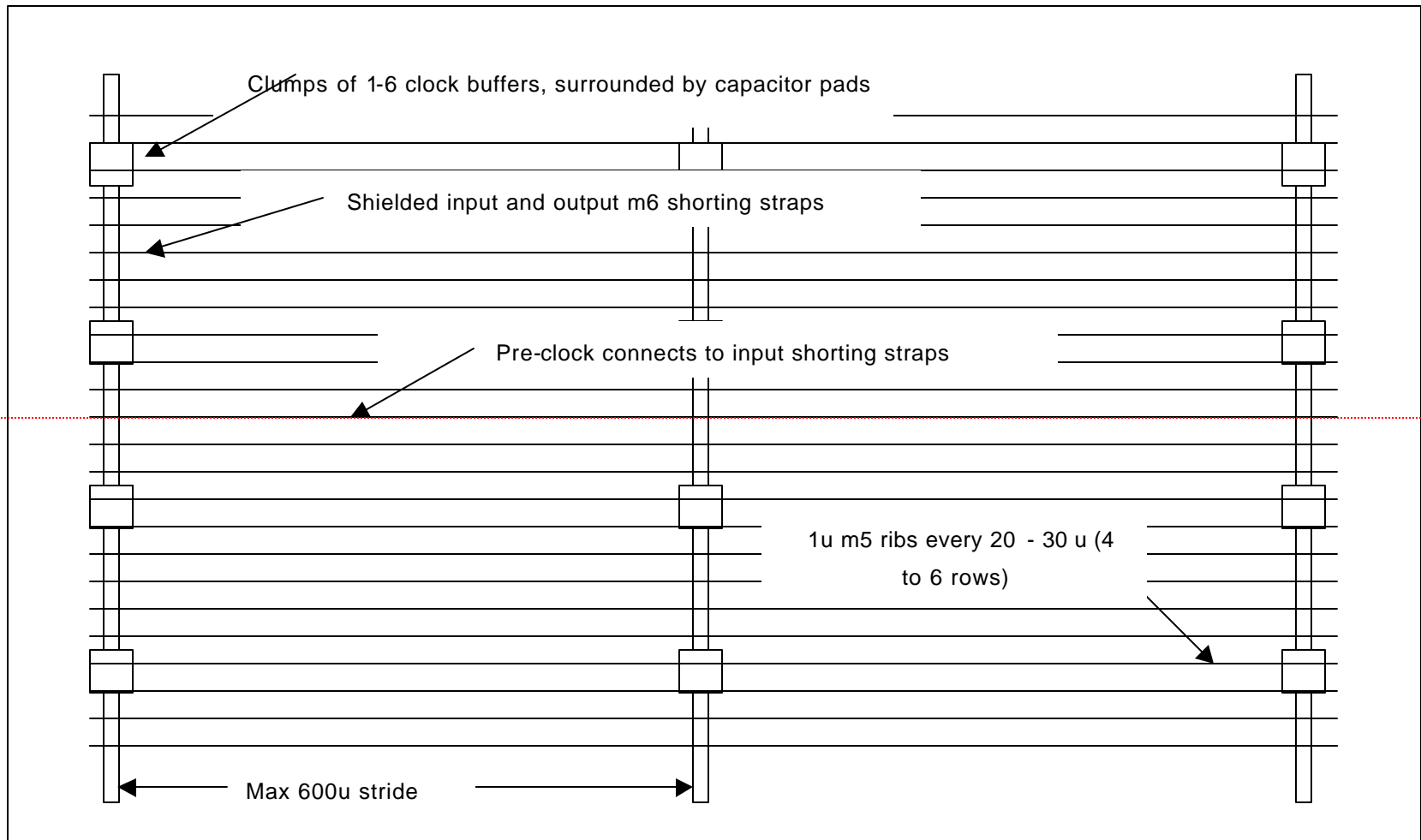
- “Fine grain” mesh
  - Vertical M6
    - width 9u
    - stride 53u
  - Vertical M4
    - width 3.5u
    - stride 53u
  - Horizontal M5
    - width 4.4u
    - stride 40u
- vss/vdd meshes offset 1/2 of stride

# Pre-clock 2 Level H-tree



- All routes 5-6u M6/5, shielded with 1u grounds
- ~10 buffers per node
- output mesh must hit every sub-block

# Block level clock distribution



# Pad Ring Assembly

---

- Fully re-playable and hierarchical (l.e. north, south, east, west blocks)
- “Edge logic”
  - allow some std cells to be associated with each pad cell
    - use verilog hierarchy to group pad cell with it's associated std cells
    - but, allow overrides from engineers for shared control logic.
    - Apollo places remaining cells
  - fixed, deterministic placement important (l.e. pre-place these cells)
- AGP/DDR, balanced and shielded bus routing.
  - Pre-routes for bus trunks, main stub
  - Avanti special route rules for stubs off pre-routes
- Bond pad stagger, end spread, power cuts, corner blocks and fill cells
- Clock distribution: T-bar





# “PE” block

---

- 250,000 Instance block, only 72% utilization, with nasty hot spot
- Over 1200 nets w/ fanout >16.
- ‘cam memory’ mux selects. Synopsys chose AOI instead of 2:1 mux
- Infer\_mux and simple buffer tree optimization solved problem

# Late breaking changes

---

- Complete re-synthesis of 3 blocks resulting in approximately 30k new gates.
- New PnR done in parallel with closure on other blocks
- Virtually impossible with flat design
- Retarget from umc to tsmc

# Hints and Kinks

---

- NullDisplay
  - prevents core dumps due to X problems with huge log output
  - open permissions on /tmp/.X11-unix
  - may need to kick NullXServer
- Aserver
  - forked “make” waits forever if Aserver needs spawning by your job
  - Many versions get started depending on apollo versions
  - solution: crontab to spawn Aserver on all machines
- Fixed X,Y floorplan
  - solution: create dummy pins
  - find min pin width on each side
  - add negative core-to-boundary offsets
  - delete dummy pins

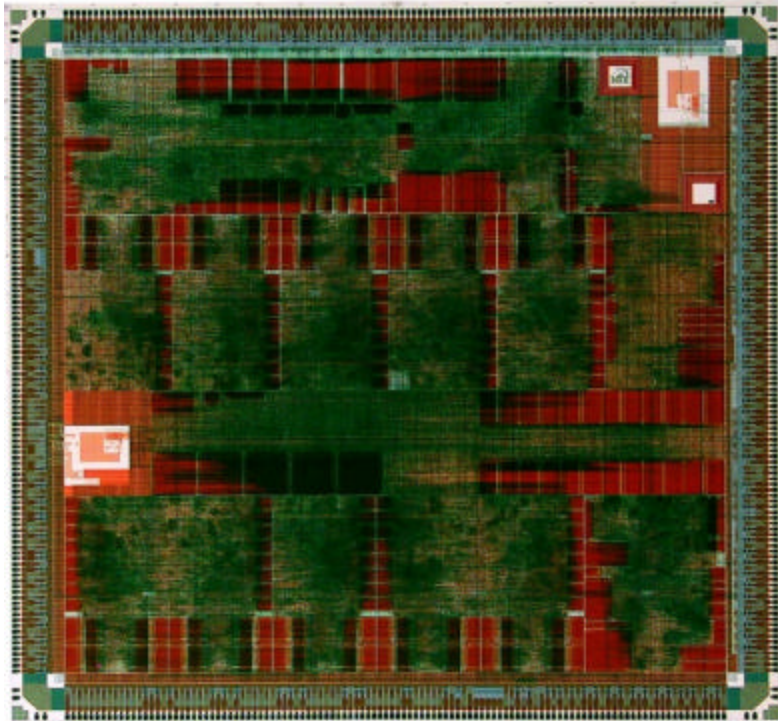
# Hints and Kinks, II

---

- Proactive Log grepping for problems
  - Fail to zqInitialRoute net \S+ , skip it
  - Cell Master \S+ does not exist
  - etc, many more
- Utilization > 100%
  - expand diearea and then refloorplan later
- Feedthroughs and multi-pin nets
  - hierarchical lvs issues solved with pulling up feedthroughs to the top level
  - extraction wants to short pins

# Another chip: “Sage”

---



Name	Kinsts	Term	%util
cbe	37	1504	73
cfe	58	875	90
pif	23	1639	50
ppe	91	948	77
sif	17	1639	58
vpc	36	4828	64
vpu0	43	548	86
.			
.			
vpu9	43	548	86
pll1	1	168	31
pll2	.3	335	31

# Futures

---

- Physical Synthesis
  - too slow until MP synthesis integrated into placement
- Always need better placers!
  - ...It's all downhill from placement
- Timing driven router spacing
  - needed to keep deterministic closure on post route timing
- Coupling detection/prevention
- Point tools
  - flip / C4
  - partial clock meshes
  - power tapering

# Acknowledgements

---

- 3dfx team
- ReShape team
- Avanti AE support

