Section III: Partitioning and Floorplanning

Overview

- Partitioning
- Floorplanning (top and block level)
- IO
- Clock distribution

Method

- Automatic vs. Manual?
  - Intense controversy
  - High performance designs force manual
  - But, not that hard
- Goal is two level netlist hierarchy:
  - N "manageable" PnR blocks,
  - pads, other edge logic flattened to top level

Block Area

- Examine total area under each branch of hierarchy
- Examine placeable objects under each branch of hierarchy
  - Sweet spot these days between 100k – 250k objects (I.e. 300 – 750k 'gates')
- Large numbers of rams can be problematic to place, so assume lower row utilization for these blocks
- Small blocks too difficult to fit in with large blocks: group with larger block and region later internally, if required

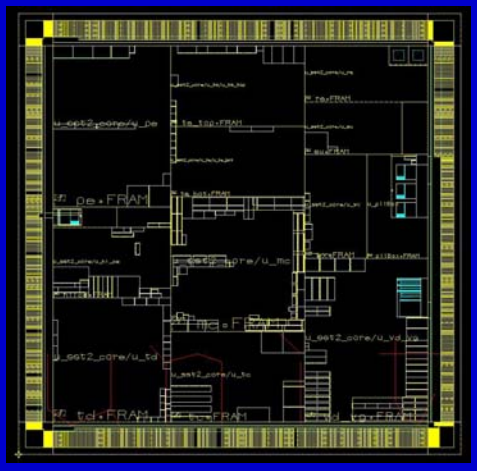
Block Area (#2)

- High Aspect Ratio blocks can be a problem for both congestion and/or timing
- Low utilization blocks may have better timing if ‘crunched down’ somewhat

Interconnection Density

- Rough rule of thumb: pin edge density should be less than 30% of the total potential pin bandwidth
- Abutted blocks must allow for feedthroughs as well (block in center feel more pressure)
- Channels, if present, probably will dominate the problem

Example Floorplan 1



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

Interconnection Density (#2)

- Typical boo-boo: RTL partitioning by function rather than connectivity
- Typical boo-boo #2: failure to replicate logic when appropriate e.g. Ram address muxing logic, io affinity control logic
- Both examples of failure to "Think Physical"

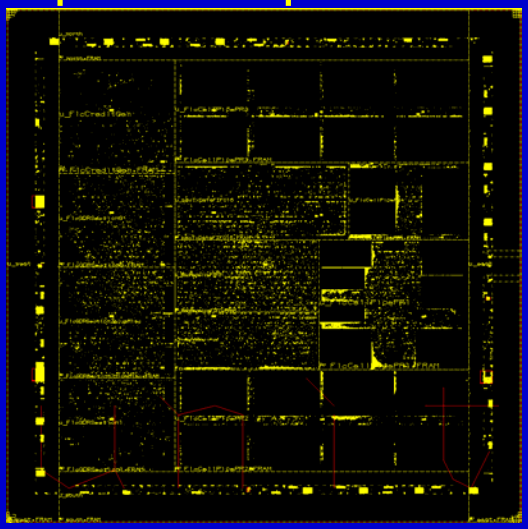
Interconnection Density (#3)

- Ram blocks: heads up! Pin layer choices involve tradeoff in floorplanning flexibility vs. wire bandwidth
- Low pin bandwidth: avoid m3 and below (assuming m1->m6 is HVHVHV)

Repeater Considerations

- Need floorplan ‘cracks’
- No IP bigger than repeater distance
- Package pinout and chip-crossing time may force PnR block locations
- Addition of diode for preemptive antenna fixing
- Must use length-based algorithm, not pure timing based

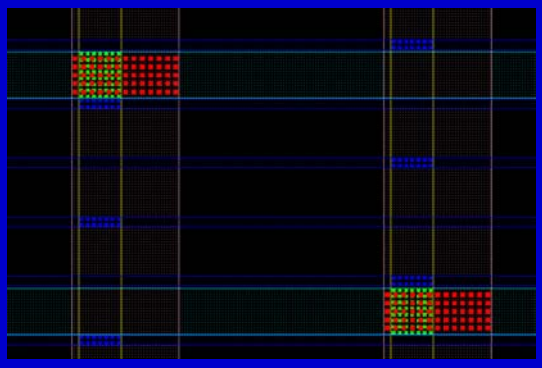
Example Floorplan 2



Low Utilization Blocks

- May need to be regioned down to reduce average distance
- Too much and coupling C is higher and timing will get worse
- High aspect ratios bad for timing, in general

Core Power Distribution Mesh



- "Fine grain"
  - Vertical M6
    - width 9u
    - stride 53u
  - Vertical M4
    - width 3.5u
    - stride 53u
  - Horizontal M5
    - width 4.4u
    - stride 40u

Special floorplanning issues

- Package pinout dictates analog IP
- PLL or other analog IP may require noisy macros (e.g. rams) placed farther away (e.g. 1200u)
- More inductive corner bond pads may force analog VDD/VSS to placed on edges (or double bonded)

Block Level Floorplanning

- Preplacing stdcells (“DIY data paths”)
  - Pro:
    - Reduces number of placeable objects in rest of block
    - Seeds placement of auto-placed cells
    - Increases area available since seeded stuff probably at high utilization
    - Deterministic results from run to run
    - Easier to change than full custom layout
    - Potential for faster ckts (T-gates, dynamic FFs, stacked latches, ??)

Block Level Floorplanning (#2)

- Preplacing stdcells
  - Con:
    - Sizes ups and buffering by tools a problem
    - Metal 2 or other interfering preroutes a problem
    - Dynamic power issues scary
    - Hand instantiation not portable to other libs
- Most common use: ‘edge logic’, register files

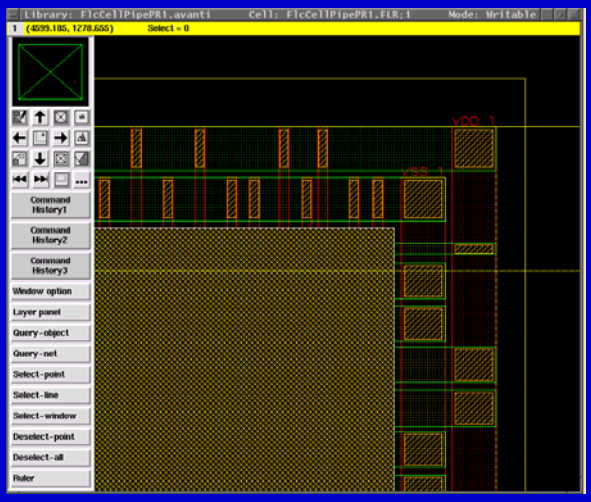
Typical Issues with Rams

- Number of layers blocked, 1,2,3…4: Place rams around block edges
- Connectivity to each other
- Connectivity to block pins
- Stdcell ‘canyons’ bad, but ok for repeaters (digression: why do all placers suck?)
- Two-side pin access problems, face pins inward

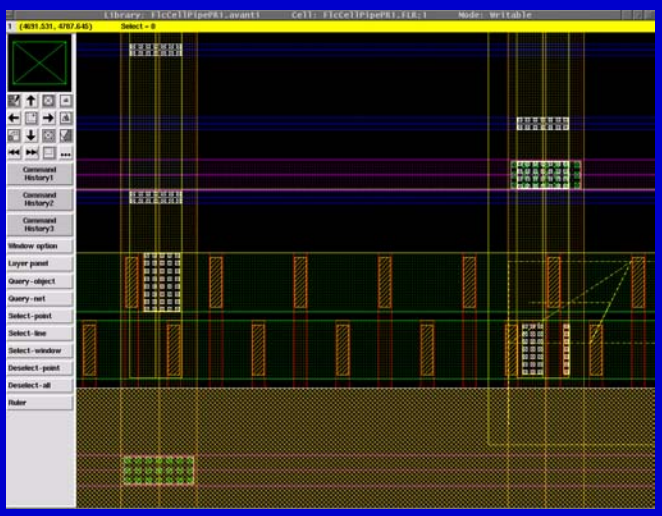
Typical Issues with Rams (#2)

- Corner hot spots
- Pin pitch issues
- Stdcell pin routing violations, keep them clear a few tracks
- Power hookup, rings or ‘internal pins’
- Bumps on top of memory array issues

Artisan Ram Rings



Artisan Ram Rings Connected



Grouping and Regioning

- Fixed regions (X0,Y0) (X1, Y1)
  - "Exclusive", rarely used, useful for regioning by clocks when clocks too expensive to distribute everywhere
  - "Non-Exclusive", most common region type
- Floating Regions
  - Takes MaxX MaxY, MaxHalfPerimeter
- Utilization "fluffers" shapes
  - Reduce congestion
  - Leave space for decoupling or more spares in very high row-utilization areas

Grouping and Regioning

- Global route "fluffers"
  - Determine headroom left in existing PnR
  - Reserve resources for later (I.e. model future impact of global feedthroughs, etc)
- Cases:
  - Grouping Based on clock domain
    - Reduce clock power, skew
  - Grouping for more deterministic placement
    - BIST logic
    - Merged hierarchy

Clock Distribution

- Distribute a clock with:
  - Minimum skew (performance and hold time issues)
  - Minimum cell area and metal use
  - (sometimes) minimal latency
  - (sometimes) particular latency
  - (sometimes) intermixed gating for power reduction
  - (sometimes) hold to particular duty cycle: e.g. 50:50 +- 1 percent

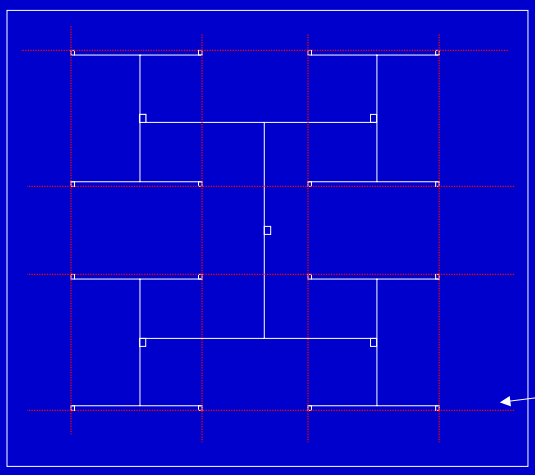
Clock Distribution (#2)

- Do all this in the face of:
  - Process variation from lot-to-lot
  - Process variation across the die
  - Radically different loading (ff density) around the die
  - Metal variation across the die
  - Power variation across the die (both static IR and dynamic)
  - Coupling (same and other layers)

ReShape Clocks Example

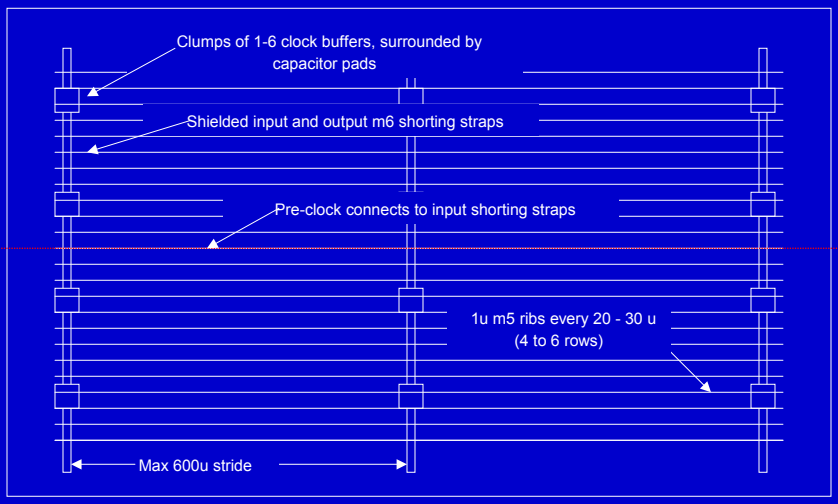
- Balanced, shielded H-tree for pre-clock distribution
- Mesh for Block level distribution

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 Mesh (.18u)



Problems with Meshes

- Burn more power at low frequencies
- Blocks more routing resources (solution, integrated power distribution with ribs can provide shielding for ‘free’)
- Difficult for ‘spare’ clock domains that will not tolerate regioning
- Post placement (and routing) tuning required
- No ‘beneficial skew’ (shudder) possible

Problems with Meshes (#2)

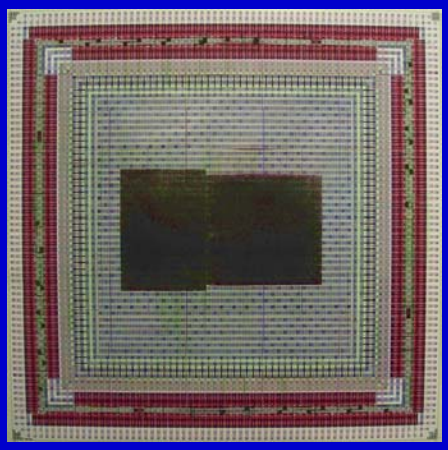
- Clock gating only easy at root
- Fighting tools to do analysis:
  - Clumped buffers a problem in Static Timing Analysis tools
  - Large shorted meshes a problem for STA tools
- Need Full extractions and Spice-Like simulation (e.g. Avant! Star-Sim) to determine skew

Benefits of Meshes (#3)

- Deterministic since shielded all the way down to rib distribution
- No ecoplacement required: all buffers preplaced before block placement
- Low latency since uses shorted drivers, therefore lower skew
- Ecoplacements of FFs later do not require rebalance of tree
- “Idealized” clocking environment for concurrent RTL design and timing convergence dance.

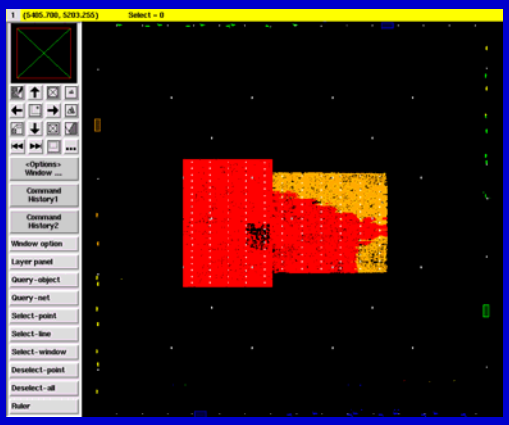
Mesh Example

- ~ 100k flops
- 6 blocks



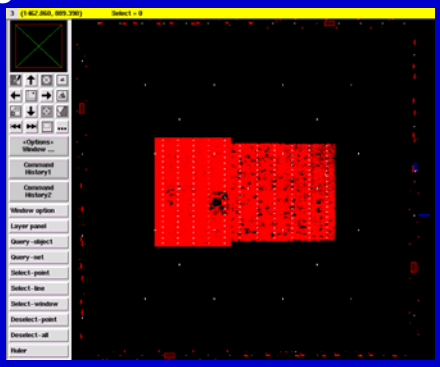
Clock Skew Thermal Map

- Pre-tuning



Clock Skew Thermal Map #2

- 50ps block/ 100ps global skew, post tuning



Other Difficult Nets

- Scan enable
- Reset Trees
  - Use synchronous fanout to each PnR block (chip quadrants if flat design)
- Massive muxing structures (e.g. CAMS, PLAs)
  - Use thermal maps to discover

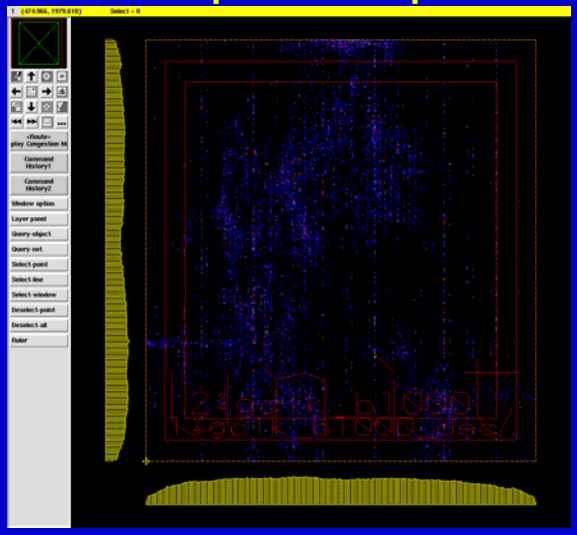
Other Difficult Nets (#2)

- Scan insertion
  - Beware if scan not in netlist: area, routeability hit to come
  - Block level insertion will create many loops, which may be tied together at the block level, confusing tools
  - Hierarchy ‘swizzles’ may occur if hierarchy manipulated in backend, or if test-insertion tools run incorrectly
  - To re-stitch post-placement or not

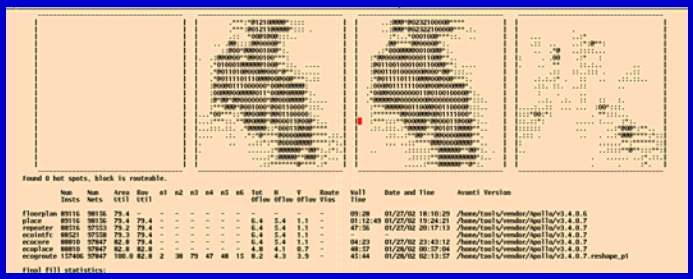
Congestion and Routeability

- Important for evaluation of floorplanning choices
- Global Routing:
  - GCELLS = Tiles
  - Basic global routing
  - Thermal Map and “Overcons”

Thermal Map Example



Thermal Map Example (#2)



Congestion and Routeability (#2)

- Detail Routing (Maze router)
  - Track assignment
  - SBOX routing of 6x6 GCELL SBOX, step and repeat with overlaps
  - Search and Repair. Welcome to “Vios”
    - Congestion vios
    - Pin accessibility vios (“chewing on rock”)
    - Maze router warts: large single SBOX routes
  - Eco re-route issues
  - Off grid pin issues
  - Non-preferred routing problems

Congestion and Routeability (#3)

- What does true congestion occur? Too much thermal map congestion for maze router to average over a ‘few’ SBOXes
- Scenic routes..more on this later…STAY AWAY

Congestion and Routeability (#4)

- Placer is using internal grouter
  - Old timing driven: single number for X, Y cap/len
  - Estimate congestion, used to be one number, now per GCELL
  - Average coupling per GCELL derived
  - Large effect on timing ECO, gate sizing, repeater insertion
  - Beware: if placement based thermal map does not look the same as post-groute thermal map!! (see mapoffsets in Apollo)

Notes on Clock Distribution

Clock Skew

- Most “high-profile” of clock network metrics
- Maximum difference in arrival times of clock signal to any 2 latches/FF’s fed by the network

Skew = max | $t_1 - t_2$ |

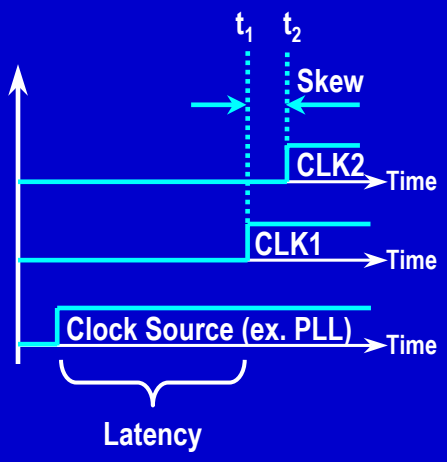


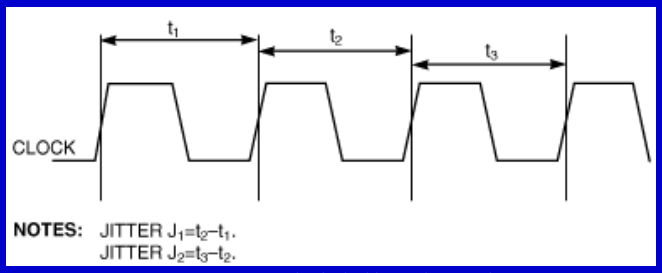
Fig. From Zarkesh-Ha

Clock Skew Causes

- Designed (unavoidable) variations – mismatch in buffer load sizes, interconnect lengths
- Process variation – process spread across die yielding different L_{eff}, T_{ox}, etc. values
- Temperature gradients – changes MOSFET performance across die
- IR voltage drop in power supply – changes MOSFET performance across die
- Note: Delay from clock generator to fan-out points (clock latency) is not important by itself
  - BUT: increased latency leads to larger skew for same amount of relative variation

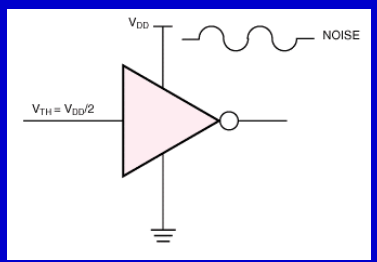
Clock Jitter

- Clock network delay uncertainty
  - From one clock cycle to the next, the period is not exactly the same each time
  - Maximum difference in phase of clock between any two periods is jitter
  - Must be considered in max path (setup) timing; typically O(50ps) for high-end designs



Clock Jitter Causes

- PLL oscillation frequency
- Various noise sources affecting clock generation and distribution
  - E.g., power supply noise dynamically alters drive strength of intermediate buffer stages
  - Jitter reduced by minimizing IR and L*(di/dt) noise

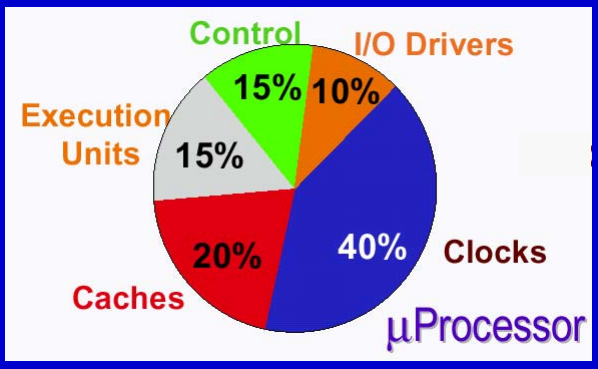


Courtesy Cypress Semi

Clock Power

- Power consumption in clocks due to:
  - Clock drivers
  - Long interconnections
  - Large clock loads – all clocked elements (latches, FF's) are driven
- Different components dominate
  - Depending on type of clock network used
  - Ex. Grid – huge pre-drivers & wire cap. drown out load cap.

Clock Power Is LARGE



$$P = \alpha C V_{dd}^2 f$$

Not only is the clock capacitance large, it switches every cycle!

Low-Power Clocking

- Gated clocks
  - Prevent switching in areas of chip not being used
  - Easier in static designs
- Edge-triggered flops in ARM rather than transparent latches in Alpha
  - Reduced load on clock for each latch/flop
  - Eliminated spurious power-consuming transitions during latch flow-through

Clock Area

- Clock networks consume silicon area (clock drivers, PLL, etc.) and routing area
- Routing area is most vital
- Top-level metals are used to reduce RC delays
  - These levels are precious resources (unscaled)
  - Power routing, clock routing, key global signals
- Reducing area also reduces wiring capacitance and power
- Typical #'s: Intel Itanium – 4% of M4/5 used in clock routing

Clock Slew Rates

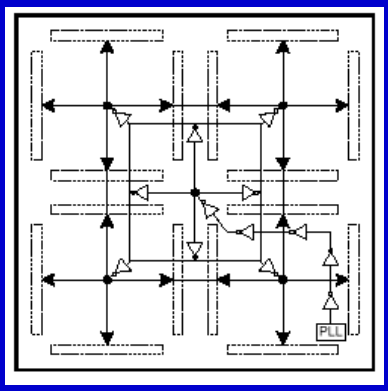
- To maintain signal integrity and latch performance, minimum slew rates are required
  - Too slow – clock is more susceptible to noise, latches are slowed down, setup times eat into timing budget [T_{setup} = 200 + 0.33 * T_{slew} (ps)], more short-circuit power for large clock drivers
  - Too fast – burns too much power, overdesigned network, enhanced ground bounce
- Rule-of-thumb: T_{rise} and T_{fall} of clock are each between 10-20% of clock period (10% - aggressive target)
  - 1 GHz clock; T_{rise} = T_{fall} = 100-200ps

Example: Alpha 21264

Grid + H-tree approach

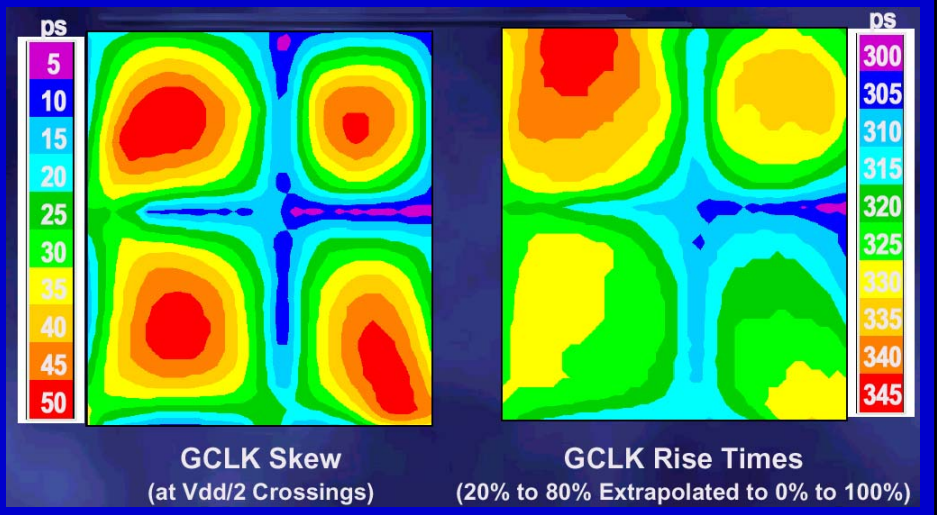
Power = 32% of total

Wire usage = 3% of metals 3 & 4



4 major clock quadrants, each with a large driver connected to local grid structures

Alpha 21264 Skew Map



Ref: Compaq, ASP-DAC00

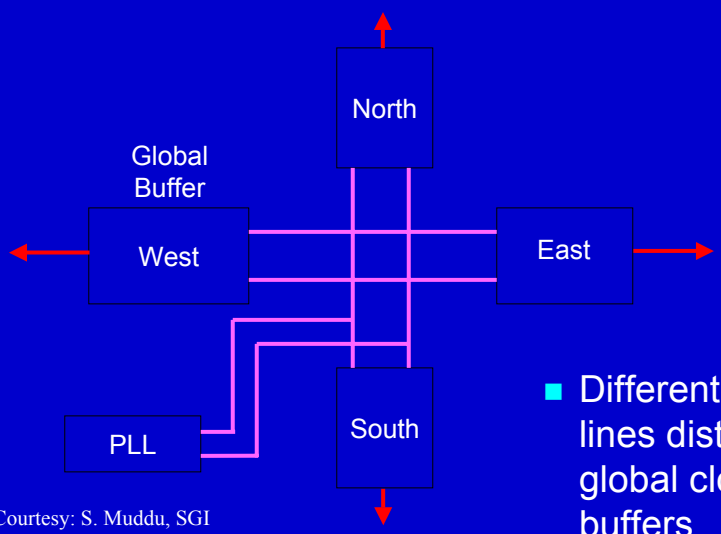
Clock Distribution Trends

- Timing
  - Clock period dropping fast, skew must follow
  - Slew rates must also scale with cycle time
  - Jitter – PLL's get better with CMOS scaling but other sources of noise increase
    - Power supply noise more important
    - Switching-dependent temperature gradients
- Materials
  - Cu reduces RC slew degradation, potential skew
  - Low-k decreases power, improves latency, skew, slews
- Power
  - Complexity, dynamic logic, pipelining → more clock sinks
  - Larger chips → bigger clock networks

Power vs. Skew

- Fundamental design decision
- Meeting skew requirements is easy with unlimited power budget
  - Wide wires reduce RC product but increase total C
  - Driver upsizing reduces latency (→ skew) but increases buffer cap
- SOC context: plastic package → power limit is 2-3 W

Global Clock Buffer Structure



- Differential clock lines distributed to global clock buffers

Courtesy: S. Muddu, SGI

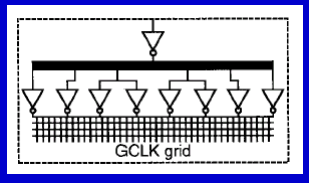
Hierarchy Management

- Mini-Block level Clock
  - Count clock nodes per Std. Block
    - total load (gate + wire)
  - Determine local clock tree levels/size
  - Estimate size of area clock buffer
  - Reserve space for clock buffers and clock wires/shields
  - Apply balanced clock routing
- Top-level Clock
  - Add clock grid topology for each Std. block
  - Estimate PLL to local buf. delays for all Std.blocks
  - Determine worst case delay
  - Add buffer-chains to align delays
  - Consider electromigration for high-activity, heavily-loaded wires
  - Add shielding inside, if necessary
  - Top-level balanced clock routing

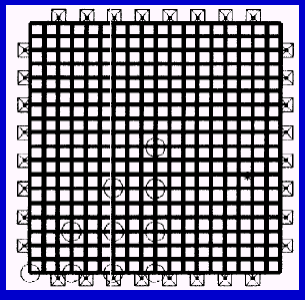
Courtesy: S. Muddu, SGI

Grid Networks

- Gridded clock distribution common on earlier DEC Alpha microprocessors
- Advantages:
  - Skew determined by grid density, not too sensitive to load position
  - Clock signals available everywhere
  - Tolerant to process variations
  - Usually yields extremely low skew values
- Disadvantages:
  - Huge amount of wiring and power
  - To minimize such penalties, need to make grid pitch coarser → lose the grid advantage



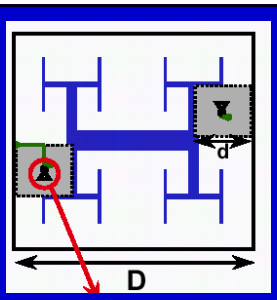
Pre-drivers



Global grid

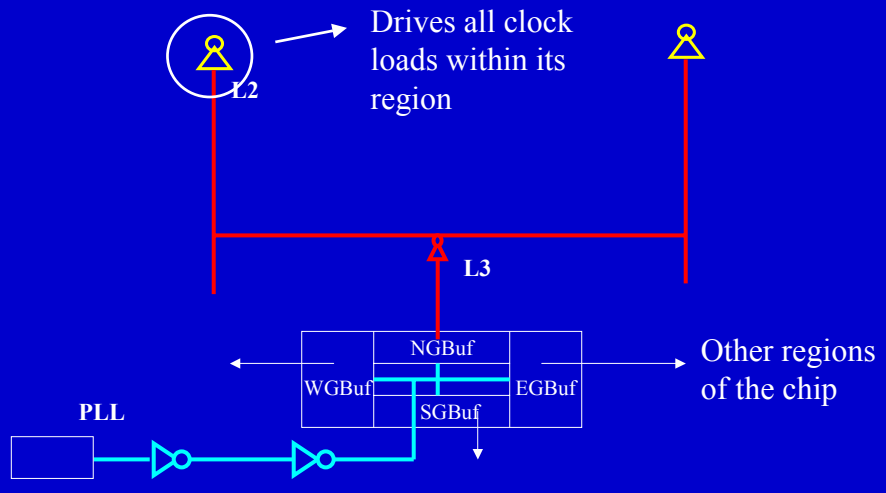
Tree Networks

- H-tree (Bakoglu)
  - One large central driver, recursive structure to match wirelengths
  - Halve wire width at branching points to reduce reflections
- Disadvantages
  - Slew degradation along long RC paths
  - Unrealistically large central driver
    - Clock drivers can create large temperature gradients (ex. Alpha 21064 ~30° C)
  - Non-uniform load distribution
  - Inherently non-scalable (wire R growth)
  - Partial solution: intermediate buffers at branching points



courtesy of P. Zarkesh-Ha

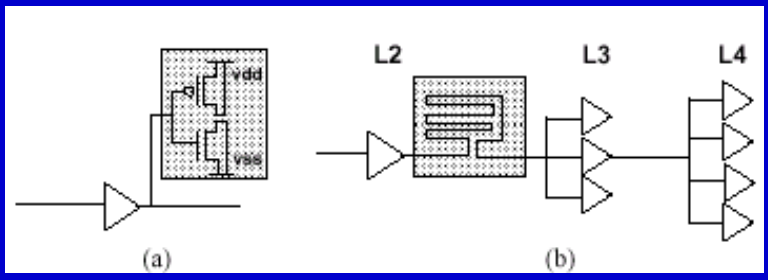
Buffered Clock Tree



Buffered H-tree

- Advantages
  - Ideally zero-skew
  - Can be low power (depending on skew requirements)
  - Low area (silicon and wiring)
  - CAD tool friendly (regular)
- Disadvantages
  - Sensitive to process variations
  - Local clocking loads inherently non-uniform

Tree Balancing



Con: Routing area often more valuable than Silicon

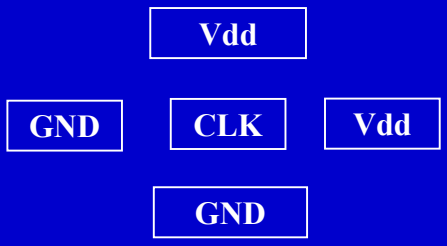
Some techniques:

a) Introduce dummy loads

b) Snaking of wirelength to match delays

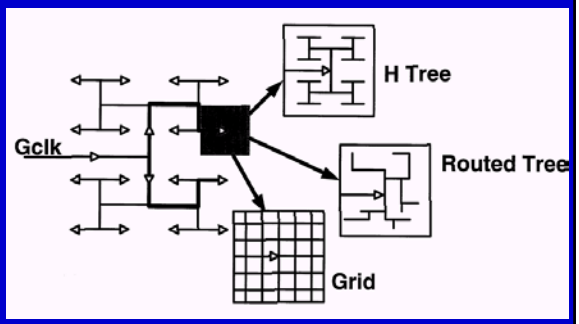
Clock Integrity

- Shield everywhere
  - Laterally and above/below
  - Provides current return paths, eliminates coupled noise effects (both C and L)

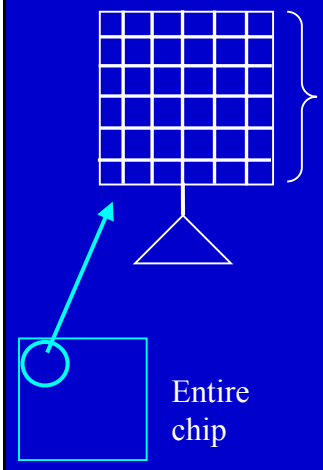


Network of Choice

- Globally – Tree
- Power requirements reduced relative to global grid
  - Smaller routing requirements, frees up global tracks
- Trees balanced easily at *global* level
  - Keeps global skew low (with minimal process variation)

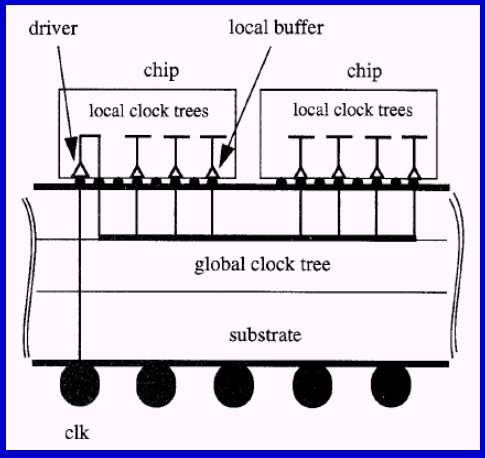


Network of Choice



- Locally – Grid
- Smaller grid distribution area allows for coarser grid pitch
  - Lower power in interconnect
  - Lower power in pre-drivers
  - Routing area reduced
- Local skew is kept very small
- Easy access to clock by simply connecting to grid

Skew Reduction Using Package

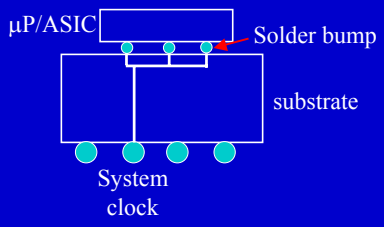


• Most clock network latency occurs at global level (largest distances spanned)

• Latency ∝ Skew

• With reverse scaling, routing low-RC signals at global level becomes more difficult & area-consuming

Skew Reduction Using Package



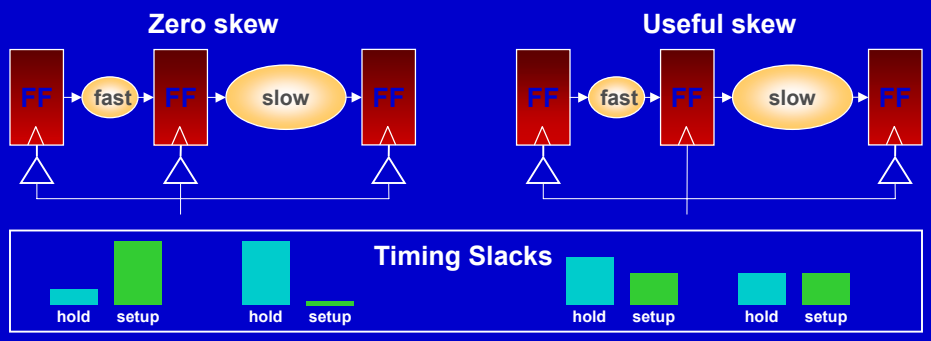
⇒ Incorporate *global* clock distribution into the package

⇒ Flip-chip packaging allows for high density, low parasitic access from substrate to IC

- RC of package-level wiring up to 4 orders of magnitude smaller than on-chip wiring
- Global skew reduced
- Lower capacitance → lower power
- Opens up global routing tracks
- Results not yet conclusive

Sylvester / Shepard, 2001

Useful Skew (= "cycle-stealing")



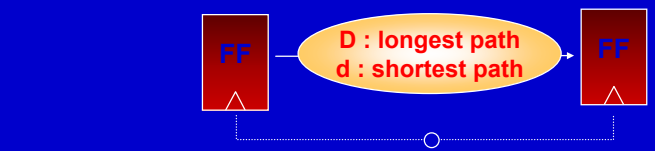
Zero skew
- **Global skew constraint**
- **All skew is bad**

Useful skew
- **Local skew constraints**
- **Shift slack to critical paths**

W. Dai, UC Santa Cruz

Skew = Local Constraint

- Timing is correct as long as the signal arrives in the permissible skew range

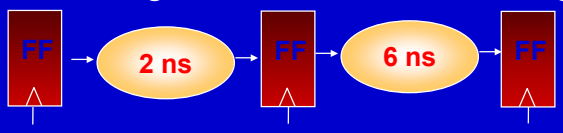


$-d + t_{hold} < Skew < T_{period} - D - t_{setup}$

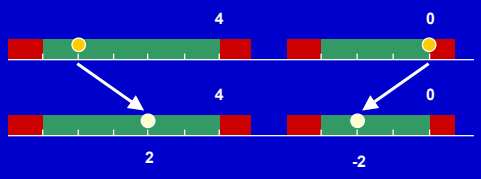
permissible range

Skew Scheduling for Design Robustness

- Design will be more robust if clock signal arrival time is in the middle of permissible skew range, rather than on the edge



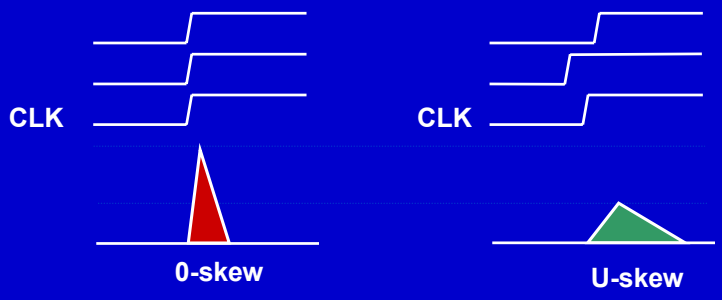
T = 6 ns



- "0 0 0": at verge of violation
- "2 0 2": more safety margin

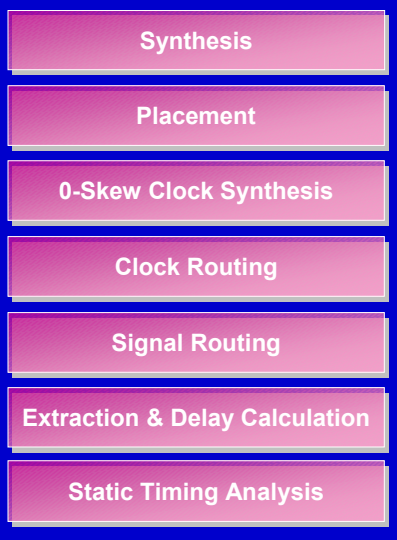
Potential Advantages

- Reduce peak current consumption by distributing the FF switch point in the range of permissible skew



- Can exploit extra margin to increase clock frequency or reduce sizing (= power)

Conventional Zero-Skew Flow



Useful-Skew Flow