Clock Distribution and Balancing Methodology For Large and Complex ASIC Designs

Dr. Kaijian Shi
Principal Consultant
Synopsys, Inc. (Professional Services)
Agenda

- Part 1: Clock tree synthesis: Fundamentals
  - Classical clock tree synthesis methods
  - Advanced clock tree synthesis

- Part 2: Clock tree synthesis: Engineer perspective
  - Custom vs. automatic clock tree synthesis
  - Clock phase delay control
  - Clock skew control
  - Clock duty cycle distortion control
  - Clock gating efficiency
  - Clock signal integrity

- Summaries
Classical clock tree synthesis methods

- Step 1: Generate a clock tree
- Step 2: Tune the clock tree to meet:
  - Skew target
  - Slew target
  - Other required constraints
Clock tree generation based on structure and load balance (H-tree)

Structure balance

Structure and load balance
Clock tree generation based on structure and load balance (Fish-bone)
Clock tree tuning

- Extract clock tree from layout
- Build buffered RC network or Spice deck
- Calculate clock tree skews
- Tune clock tree branch delays
  - Adjust tapping points
  - Size buffers
  - Size wires
  - Snake route wires
  - Add dummy load
Advanced clock tree synthesis methods

- 0-skew clock tree synthesis
- Clock tree synthesis considering process variations
0-skew clock tree synthesis method

- Integrate 0-skew clock tuning into each level CTS
- Bottom up hierarchical process:
  - Cluster clock nodes and build a local tree by the load balance based CTS methods
  - Create a buffered RC network from the local clock tree
  - Minimize clock skew by wire sizing and snake routing
- Advantages
  - Eliminate port-CTS tuning process
  - Interconnect aware skew balance
  - Efficient (local clock tree skew optimization)
Advanced clock tree synthesis methods

- 0-skew clock tree synthesis
- Clock tree synthesis considering process variations
Clock tree synthesis considering process variation

- P-variations cause unpredictable delay variations in transistors and wires -> uncontrollable skew

- The delay variations in common part of clock tree between launch and capture flops do not cause skew

Principle of minimizing P-variation effect
Minimize non-common part of clock tree between launch and capture clock nodes
Apply to clock tree topology

- Group launch and capture clock nodes and cluster them in the bottom up fashion in clock tree topology generation.

More sophisticated methods:
- Create a weighted direction graph to model launch and capture clock node relationship in a complex clock structure.
- Use the graph as constraint in clock tree generation.
- Reduce complexity by considering only clock nodes in timing critical paths.
Apply to clock tree layout

- Place clock tree buffers to reduce the length of the non-common portion of the clock tree

Bad

Good
Clock tree synthesis: Engineer perspective

- Custom vs. automatic clock tree synthesis
- Clock structure in large and complex ASIC designs
- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Custom clock tree distribution and balancing

- Manually define top levels of clock tree to blocks
  - H-tree, wide/shield wires, differential buffers etc.
- Build local mesh or tree to distribute clock to leaf cells
- Extract clock tree and build SPICE deck.
- Analyze the extracted clock tree and tune it manually.
- Pros:
  - Low skew clock tree
- Cons:
  - Long and complex process
  - Large power dissipation in clock mesh
- Popular in high speed MPU design, but not suitable for ASIC and low-power designs
Automatic clock tree synthesis (CTS)

- **Pros:**
  - Flexible and relatively easy to use
  - Low skew, if clock structure is not complex
  - Clock phase delay and wire length are minimized
  - max_fanout, max_cap, max_slew targets are honored
  - Good correlation between pre- and post-route clock skew

- **Limitations:**
  - Optimization is focused on expanded clock buffer trees.
  - Quality drops when clock structure becomes complex and CTS constraints are not provided.

- **Need CTS constraints and guidance.**
Clock tree synthesis: Engineer perspective

- Custom vs. automatic clock tree synthesis
- Clock structure in large and complex ASIC designs
- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Clock structure in large and complex ASIC designs

- For power saving
  - Multi-level clock gating
  - Many gating domains
  - Multi-clock speed domains

- For user programmability and debug support
  - User programmable clock dividers
  - Various operation mode dependent clock distributions to support emulation and debug

- Results:
  - Complex clock structure with control logic
  - Challenges in clock distribution and balancing
Issues and practical solutions in automatic clock tree synthesis

- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Clock phase delay reduction

- Large phase delay => large clock tree and power dissipation
- Contributions to clock phase delay
  - Delays of the expanded clock buffer tree
  - Delays of “non-CTS” cells in the clock distribution. (gating, mux, divider)
- Large clock phase delay: causes and solutions
Cause 1: Bad placement of “non-CTS” cells

- Large load cause large slew at “non-CTS” cell outputs
- Cascaded “non-CTS” cells amplify slew
- To fix slew violation, CTS tools insert back-to-back buffer chain
  
  => buffer chain delay is added to clock tree insertion delay!
Cause1: Bad placement of “non-CTS” cells

- Large “non-CTS” cell delays due to large load
  - The cell delays are part of the clock phase delay
  - The cell delays cannot be reduced by CTS tools

- A practical solution
  - Find “non-CTS” output nets
  - Apply heavy net-weight to the nets in cell placement

  ➔ Minimize net length and load on “non-CTS” cells.
Cause 2: Bad placement of local cells at high levels of a clock tree

Though not require to be balanced in timing, the local flops are still balanced on loading => increase clock phase delay
Solution 1: place local flops in a small region

- Region the local flop in a small area.
- Effect is limited in a case of many local flops.
Solution 2: Isolate the local flops

- Insert and place an isolation buffer to minimize load on clock source => eliminate local flops’ load on clock tree
The longest clock path determines clock phase delay
Solution: CTS friendly floorplan

- Equalize steiner distance from clock source to leaf cells.
- Region cells connected to high level clock tree in a small area close to the clock source.
- Do not place a subchip in a position that can force clock paths detour routed around it.
An example of CTS friendly floorplan

- Challenges of rigid pin constraints and fixed size hard IP core.
- Equalize Steiner distance from clock source to leaf cells.
- Region clock management block in a small area close to PLL.
- No hard macros in std_cell placement area.
Issues and practical solutions in automatic clock tree synthesis

- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Localized clock skew can be used for good causes:
- Borrow time from non-critical paths to meet timing of critical paths
- Reduce IR drop and EM caused by simultaneous clock switching

In general case, skew degrades design speed and causes malfunction due to hold time violations.

» clock skew need to be minimized
Cause 1: Multiple clock dividing paths

- CTS tools usually balance through one dividing path
- High fan-in mux has large delay variations from inputs
Solution: balance dividing paths through a delay equalizer

Paths from Clock-in through the flop and the delay line are balanced.
Pros. Easy to implement

Cons. FSM flops need to be skewed earlier than bypass clock to avoid a cycle shift of the equalizer output clock
The delay equalizer implementation 2

- The flop in the equalizer is integrated into the divider as the last stage flop

  Pros.
  - Save one flop
  - Divided clocks and bypass clock are always in phase

  Cons. Need to redesign the clock divider logic
Constraints on dividing/bypass select signal

- Select during power-on => static signal => no constraint
- Run time programmable => need constraint

Signal switches only when clocks are in a same phase
Cause 2: Operation mode dependent multiple clock distributions

Four levels of Gating
1. Chip level clock gating
2. Functional domain level clock gating
3. Block level clock gating
4. Transaction level clock gating

Asynchronous module
Solution: A multi-mode clock balance strategy

- Extracted a common clock distribution topology
- Mode dependent clock path selection criteria
  - Minimize unbalanced leaf cells
  - Maximize balanced leaf cells in timing critical paths
- Defined constraints for local paths outside of the common clock distribution
- Balance the common clock distribution and local paths with defined constraints

=> Clock tree is balanced in various modes
Solution: A multi-mode clock balance strategy

Diagram showing the connections between various components such as DPLL1, DPLL2, Divider, and Functional path, with signals like Test_clk, Test_sel, Func_clk, block1_func_enable, block2_func_enable, and boundary_scan_enable.
Function mode inverting, test mode non-inverting clock paths

- Problem: inverter causes clock skew in test mode.
- Solution: Implement XNOR gate
Issues and practical solutions in automatic clock tree synthesis

- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Clock duty cycle distortion control

- Applicable to a design utilizing both clock edges
- Cause setup time violations in timing critical paths
- Main causes of clock duty cycle distortion
  - Rise and fall delay variations of library cells
    - The large load on the cells, the larger delay variations
    - The deeper a clock tree, the larger clock duty cycle distortion
  - PLL introduced distortion
Solution 1: insert inverter pairs in a clock tree

- Find a mid-point in a clock tree to invert rise/fall delay variation so as to cancel the non-inverted rise/fall delay variation.

Issues:
- A number of inverters need to be added after clock tree is expanded.
- Find optimal insertion points in the expanded clock tree is not easy.

Rise/fall delay delta = -100ps
Rise/fall delay delta = +100ps
Solution 2: Implement a clock phase chopper – case 1: longer high phase

- Restore 50/50 duty cycle by delaying rising edge of the high clock phase

- Pros.
  - Single insertion point and easy to implement
  - Implemented before clock tree expansion
  - Local delay tuning has little disturbance in layout
A clock phase chopper for 50-Td/50+Td duty cycle distortion

An OR gate chopper to delay falling edge of the clock
Issues and practical solutions in automatic clock tree synthesis

- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Clock gating efficiency

- Maximize gating effect to reduce idle power
- Minimize ungated part of the clock tree
  - Place clock gating cells close to clock sources
Issues and practical solutions in automatic clock tree synthesis

- Clock phase delay control
- Clock skew control
- Clock duty cycle distortion control
- Clock gating efficiency
- Clock signal integrity
Clock signal integrity

- Issues: Large net load on non-CTS cells in the clock paths can cause slew and IR drop violations.
- Solution: netweight and region based placement.

- Crosstalk induced violations
- Solution: Identify critical clock routes, reroute the nets with wide spacing or shielding
Summary

- Clock tree synthesis: Fundamentals
  - Classical clock tree synthesis methods
  - Advanced clock tree synthesis

- Clock tree synthesis: Engineer perspective
  - Custom vs. automatic clock tree synthesis
  - Clock phase delay control
  - Clock skew control
  - Clock duty cycle distortion control
  - Clock gating efficiency
  - Clock signal integrity