Physical design flow Challenges at 28nm on Multi-million gate blocks
AGENDA

1. Introduction of 28nm technology ASIC
2. 28nm ASIC Physical design Challenges
   - Floorplanning
   - Congestion
   - Timing
   - Runtime
3. Results
4. Conclusion
5. Q&A
1. Introduction of 28nm technology ASIC

- 28nm has been in volume production for over 3 years
- There are those who believe 28nm is the last node of Moore’s Law
- Beyond 28nm the litho costs are expected to make it less cost effective.
- But most experts seem to agree that 28nm will have a long life

1. Introduction of 28nm technology ASIC

• So what does all this mean for the physical designer?
  – There will be more 28nm tape outs coming his way
  – Design will be getting more and more complex
  – All the issues seen in earlier nodes will continue to be there but more aggravated
    • Signal integrity, Leakage power impact
    • More placement rules, more routing rules, DFM and yield issues
    • Interconnect variations, Process variations
1. 28nm ASIC - A view

- 28nm TSMC HPM, 10Metal Layers + RDL
- Dimensions: >20mm each side
- Power: >100W
- 70+ blocks (400K to 2M gates)
- >200M gates; nearly 1Gb of RAM
- Typical clock freq 500/750MHz
2. 28nm ASIC Physical design Challenges - Floorplanning

- Need to get it right for smooth implementation

- Knowledge about the design helps
  - Know the design
  - Have flow diagrams

- Internal scripts to dump out groups of related macros helped designer come up with initial macro grouping

- Watch out for placement and routing blockages pushed down from top
- Cannot rotate macros by 90 degrees
- If you have clock channels going through the blocks, you will need to meet spacing requirements
  - Not only disrupts the macro placement
  - But also introduces placement pockets in the floorplan
- Trace macro fanouts/fanins to understand the connectivity(after first placement)
Warning: Avoid tracing through clock pins and test related pins
What if we do not have a flow diagram?

Evolve the floorplan!
Congestion not present in new netlist!
2. 28nm ASIC Physical design Challenges - Floorplanning

Congestion and timing problem
Solved using density screens. But use sparingly!
2. 28nm ASIC Physical design Challenges – Placement Congestion

Localized congestion

Congested Module placement

Module placement with instance padding

Try density screens?

No! Go with instance padding. Don’t go with cell padding!
2. 28nm ASIC Physical design Challenges - Timing

Other than going through path reports, inspecting visually can give insight to the problem.

The Display Timing Map feature
Timing was looking great until we detail routed the design.

Layer assignment differences
1. More buffering
2. Detouring

✓ These options helped improve correlation:
setTrialRouteMode -skipTracks "M10 1:5"
setTrialRouteMode -skipTracks "M9 1:5"
2. 28nm ASIC Physical design Challenges – Timing issues: Fanout and placement

- Very bad TNS!
  - ~40K Fanouts endpoints
  - ~500 startpoints.

- Bounding the startpoints at the centre worked!

![Diagram showing physical design challenges and solutions.]

--

28nm ASIC Physical design Challenges – Timing issues: Fanout and placement

- Very bad TNS!
  - ~40K Fanouts endpoints
  - ~500 startpoints.

- Bounding the startpoints at the centre worked!
## 2. 28nm ASIC Physical design Challenges – Run time

One of the worst runtime blocks that we had

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floorplan</td>
<td>3 hrs</td>
</tr>
<tr>
<td>Place and Optimization</td>
<td>43 hrs</td>
</tr>
<tr>
<td>CTS</td>
<td>20 hrs</td>
</tr>
<tr>
<td>Routing and Optimization</td>
<td>70 hrs</td>
</tr>
<tr>
<td>Metal and DFM fills + GDS generation</td>
<td>22 hrs</td>
</tr>
<tr>
<td>Extraction, timing, Vt swaps, Noise, DRC, LVS, Antenna, Signal EM</td>
<td>40 hrs</td>
</tr>
<tr>
<td>Total</td>
<td>198 hrs</td>
</tr>
</tbody>
</table>

~4M instances, 180+ macros

An iteration time of over 3 days!

8+ days!!!

Anything that can possibly help avoid an iteration is welcome!

Related concerns:
1. Diskspace
2. LSF and compute license
Historically, we had been having just FF hold corners enabled during optimization. This was found to be sufficient. The belief was that we didn’t need any other corner, and adding more would impact the runtimes.

But on 28nm, some blocks were found to have significant hold violations in the Slow corner. Found that adding a couple more hold corners did not impact total runtime too much. Runtime impact was typically <2hrs. Most importantly – Signoff hold fixing became minimal and less disruptive!
It was observed that nets with significant M9/M10 (thick metals) routing were more likely to end up with signal EM issues. Sometimes with 100s of violations.
- These EM violations were on the lower layers
- Easy fix is to simply apply double width NDR on these nets and reroute them
- Risks disruption of nets in the vicinity

Could this be avoided? Or could we identify these nets early?
Yes, We can!

Immediately after the first detailed routing, applying NDR on the nets with high wire cap driven by high drive cells helps to avoid a significant number of EM violations.

Using a condition like “if (16x & wcap > 0.140) || (12x & wcap > 0.220) then apply NDR”, we were able to bring down the EM violations to under 10 in most blocks.
3. Improved Results

1. We were able to successfully execute highly complex blocks in 28nm
2. Increased efficiency – a designer could handle up to 4 to 6 blocks
3. Once the recipe had been fine-tuned, most blocks could be closed from scratch in under 2 weeks!!!
4. Conclusion – Takeaways

- 28nm presents challenges – like every other node
- **Floor-planning**: When we have lots of macros to be placed, flow diagrams help; Watch out for clock channels and macro orientation restrictions
- **Congestion**: Solve using instance padding, density screens and blockages as needed; Or Even netlist updates
- **Timing**: Use bounding instances, density screens and blockages as needed; Or skipping tracks if it’s a post-detailed-route correlation issue
- **Runtime** – enable more modes/corners if needed; Plan to reduce the number of iterations; Go for avoidance whenever possible, eg. signal EM
- Cadence EDI flow enables us to execute these highly complex chips; all the hooks and knobs are there – we just need to figure out the right ones to use!
- **Automation** – Everyone’s efficiency improves, Engineer can handle 4 blocks.
Acknowledgement

1. Team mates working on block/chip level closure – those who actually spent their time debugging, analyzing and trying out multiple options until we could solve each problem
Thank You

Name : Nilesh Ranpura
Name : Vineeth Mathramkote
Email ID : marketing@einfochips.com