Design Closure: Power Constraints, best practices for an accurate Report Power estimation

Feb 2021
Design Closure Sessions

- **Session 1**
  Methodology, tips, and tricks for achieving better Quality-of-Results

- **Session 2**
  Using Timing Closure Assistance tools to address tough timing issues

- **Session 3**
  Power Constraints, best practices for an accurate Report Power estimation
Agenda

- Power impact and Time to Market
- Design Closure an efficient approach
- Understanding design power
- Design Power Constraints
- Vivado Commands
Power impact and Time to Market
Why is Power Closure so important?

- Board Design is Fixed
- Power and Thermal Issues take a long time to correct
- Design Changes (Typically Weeks)
  - Re-Run P&R
  - HDL Changes
  - Reducing design specifications
- Hardware changes (Typically Months)
  - Board Re-spin
  - Power Delivery Changes
  - Thermal Solution Changes
Converge and ensure all original assumption/design constraints are met

Designed within Budget?

Yes

SUCCESS

No

Modify Design

If power cannot be reduced, Thermal and Board Design needs to be redone (very time consuming)

DISSECTION

• What went well?
• Lessons learned?
Design Closure an efficient approach
Design closure – combining Timing and Power

- More efficient, build a complete picture of timing AND power

- Place & Route
- Timing Closure
  - Timing Clean?
    - Yes
    - No
      - Design within Budget?
        - Yes
        - No
          - Timing Still Clean?
            - Yes
            - No
              - Generate .bit / .pdi and HW test

- Power Closure
  - No

- Timing Clean?
  - Yes
  - No
    - Analyse results for lowest power
    - Generate .bit / .pdi and HW test
    - Reduce Time To Market
Combining Timing and Power gives much more information

- Allows users to take the best run from a Timing AND power perspective, not just the best timing run

**Power / Timing Slack for Different P&R Runs**

Best Run: Power Budget Exceeded

Lowest Power: Still Timing Clean
Understanding design power
Design Power – Sum of two Parts

- Dynamic power directly related to the user design
  - Resource count, Fmax, toggle rates

- Static Power Junction temperature related
  - Rises as Junction temperature rises
  - Xilinx provides Typical and Maximum process numbers
  - Maximum process should be used for worst case power

- Static power often overlooked
  - Causes Power Analysis inaccuracies

- Static power reduction via:
  - Thermal solution improvements
  - Low Voltage devices
Example of Total Design power over temperature

- Total Power Range 72 – 90W
- Dynamic power 70.3W
- Static power range 2.2 – 20.5W
- 18.2W range
Device Selection – impact on Static power

- Best Practice is to design for low voltage devices
  - Easier to move up if timing is challenged
  - Harder to move from High or Mid voltage to Low if power is too high

- Versal has 3 Voltages
  - Low, Mid and High
  - Also, Low and Standard Static Screen

- UltraScale Plus
  - Supports for Vlow (0.72v)
Design Power Constraints
Minimum recommended power constraints

- Ensure a power budget is defined
- Maximum process is set
- Without thermal information the MAX Junction Temp (Tj) should be used

```bash
set_operating_conditions -design_power_budget <Power in Watts>
set_operating_conditions -process maximum
set_operating_conditions -junction_temp <Max Tj based on Temp Grade>
```
Best accuracy Power Constraints

- Minimum constraints use the Max junction temp
  - Worst case Power analysis

- Power Estimate can be refined further
  - Defined maximum ambient the application will support
  - Define Theta Ja - Thermal solution efficiency (C/W)
    - For example: 1.5 C/W for every W dissipated Tj increases by 1.5C

```
set_operating_conditions -ambient_temp <Max Supported by Application>
set_operating_conditions -thetaja <Increase in Tj for every W dissipated C/W>
```
Refining Power Estimation using $\theta_{ja}$

1. No Constraints – 55W

2. Worst Case – 77W
   Max Process & $T_j$ Max

3. Refined – 63W
   $T_a$ Max - 35°C & $\theta_{ja}$ – 0.6 C/W

Ensures accurate modelling of Static and Total Power

© Copyright 2021 Xilinx
## Improving the Confidence Level

- **Try to achieve High Confidence Level of accuracy - review power reports**
  - Review switching activity constraints: specify missing and correct invalid constraints
  - Report the power after Implementation for accurate signal power
  - Use most recent Vivado version for most up-to-date power characterization data

### Confidence Level Details

<table>
<thead>
<tr>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design State: Low</td>
<td>Design is synthesized</td>
<td>Design is routed</td>
</tr>
<tr>
<td>Clock Activity: High</td>
<td>User specified more than 95% of clocks</td>
<td>User specified more than 95% of clocks</td>
</tr>
<tr>
<td>I/O Activity: Low</td>
<td>More than 75% of inputs are missing user specification</td>
<td>User specified more than 95% of inputs</td>
</tr>
<tr>
<td>Internal Activity: Medium</td>
<td>User specified less than 25% of internal nodes</td>
<td>User specified more than 95% of inputs</td>
</tr>
<tr>
<td>Characterization Data: High</td>
<td>Device models are Production</td>
<td>Device models are Production</td>
</tr>
</tbody>
</table>

- **Low:** Unrouted design, No power constraints
- **Medium:** Unrouted design, Some power constraints
- **High:** Routed design, Good power constraints
Power Constraints Advisor

- Available from the Launch Power Constraints Advisor in Power Summary
- Simple GUI that shows confidence level of Sets / Resets and Enables
  - Allows sorting and filtering
  - Start with Low Confidence High Fanout nets
- Negative power margin clearly shown
Power Rail Constraints

- New Feature in Vivado 2020.2
- Should be used in addition to Design Power Budget
- Allows regulator power to be validated

Create a new power rail:
  - `create_power_rail <power rail name> -power_sources {supply1, supply2, ..}`

Add power sources to an existing power rail:
  - `add_to_power_rail <power rail name> -power_sources {supply1, supply2, ..}`

Define current budget:
  - `set_operating_conditions -supply_current_budget {<supply rail name> <current budget in Amp>} -voltage {<supply rail name> <voltage>}`
Power Rail Constraints – Example

- Direct: Single regulator to 1 or more rails
  
  ```
  create_power_rail board_85V -power_sources {VCCINT VCCBRAM VCCINT_IO}
  set_operating_conditions -supply_current_budget {board_85V 55} -voltage {board_85V 0.85}
  ```

- Indirect: Two or more regulator stages
  
  - Can define multiple regulation stages
    - Example: Board supply power and Rail current

  ```
  set_operating_conditions -supply_current_budget {VCCINT 50 VCCBRAM 1 VCCINT_IO 4}
  ```
Power Rail Constraints - Results

- Single regulator to 1 or more rails
  - Critical Warning Generated

- Margin Reported in Text report:

<table>
<thead>
<tr>
<th>Source</th>
<th>Voltage (V)</th>
<th>Total (A)</th>
<th>Budget (A)</th>
<th>Margin (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>board_8SV</td>
<td>0.850</td>
<td>55.409</td>
<td>55.000</td>
<td>-0.409 (VIOLATED)</td>
</tr>
<tr>
<td>VCCINT</td>
<td>0.850</td>
<td>47.652</td>
<td>47.652</td>
<td>-0.000</td>
</tr>
<tr>
<td>VCCBRAM</td>
<td>0.850</td>
<td>0.644</td>
<td>0.644</td>
<td>-0.000</td>
</tr>
<tr>
<td>VCCINT_IO</td>
<td>0.850</td>
<td>7.113</td>
<td>7.113</td>
<td>-0.000</td>
</tr>
<tr>
<td>VCCAUX</td>
<td>1.800</td>
<td>1.420</td>
<td>1.420</td>
<td>-0.000</td>
</tr>
</tbody>
</table>

- Two or more regulator stages

Critical Warning Generated

Margin Reported in Text report:
Vivado Commands
Vivado Power Optimization commands

- Vivado has some powerful options that can save up to 30% on dynamic power
  - Intelligent Clock Gating

- Power optimization available via the following commands:
  - `power_opt_design`
  - `opt_design`

- `power_opt_design` can be run before or after placement
  - Best results before placement
  - After placement it will preserve timing

- `opt_design`
  - Runs Block RAM power optimizations – Can impact timing
  - Disabled via `-directive NoBramPowerOpt`
### Potential Power Impact of Place & Route directive

<table>
<thead>
<tr>
<th>Synthesis</th>
<th>opt_design</th>
<th>Place</th>
<th>Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>AreaOptimized_high</td>
<td>ExploreArea</td>
<td>ExtraNetDelay_high</td>
<td>HigherDelayCost</td>
</tr>
<tr>
<td>AlternateRoutability</td>
<td>ExploreSequentialArea</td>
<td>ExtraPostPlacementOpt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ExploreWithRemap</td>
<td>WLDriivenBlockPlacement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSI_SpreadLogic_high</td>
<td>SSI_HighUtilSLRs</td>
</tr>
</tbody>
</table>

- Examples of what we have found beneficial for power
- Every design is different experiment to understand the impact to Timing and Power
- The earlier in the flow the better the results i.e Synthesis and opt_design
- -flatten_hierachy full generally improves power
Power Design Closure Conclusions

Utilize Timing Closure to get the best understanding of Design Power
Tcl Script for Timing and Power Closure - AR 76056

Largest Dependency on user input out of all Design Closure steps
User application Power Delivery, Ambient and Thermal solution information critical

Thermal Data should be used to refine Power Estimation
Ensuring a successful Power Delivery and Board design

Power impacts every design
If not correctly addressed can have the largest Time To Market Impact
Thank You
Additional Resources

- Xilinx Power Page [Xilinx.com/power](Xilinx.com/power)
- Vivado Power Analysis & Optimization User Guide - [UG 907](UG 907)
- UltraFast Design Methodology User Guide – [UG949](UG949)
- Vivado Design Suite Tcl Command Reference Guide – [UG835](UG835)
- Xilinx Power Estimator User Guide
  - Versal ACAP [UG1275](UG1275)
  - UltraScale Plus [UG440](UG440)
- Power & Thermal Checklist - [AR 76055](AR 76055)
- Tcl Script for Timing and Power Closure - [AR 76056](AR 76056)