Best Practices for Implementing ARM Cortex®-A12 Processor and Mali™-T6XX GPUs for Mid-Range Mobile SoCs.
Cortex-A12: ARM-Cadence collaboration

- Joint team working on ARM® Cortex®-A12 iRM flow

- iRM content:
  - Design constraints
  - Floorplan files
  - CPF files for low power flows.
  - Cadence flow scripts, e.g. RC & EDI, based on Foundation flow scripts.
  - Makefile to run the whole flow.
  - iRM User Guide document
Cortex-A12 Cadence iRM Flow

RTL Libraries CPF FP SDC

Synthesis & Physical Synthesis
RTL Compiler Physical v12.2

netlist SDC CPF PLACEMENT

P&R (gigaOpt, CCOPT, Nanoroute)
Encounter Digital Implementation v11.1

netlist SDC CPF LAYOUT

ATPG
Encounter Test v12.10

Parasitics Extraction
QRC v11.12

Logic Equivalence
Low power checks
Conformal
LEC + CLP v12.10

Static Timing Analysis
Encounter Timing System v11.1

signoff
Mali T678: ARM-Cadence collaboration

• Extend collaboration started on ARM® Mali™-T604 back in 2011

• Technology used:
  – TSMC 28nm HPM
  – Base C35 LVT, Base C35 SVT
  – PMK C35 HVT, PMK C35 SVT, PMK C35 LVT

• Goals
  – Develop the optimal methodology to achieve PPA targets
  – Deliver a set of customer ready deployable scripts
  – Deliver a methodology that can be used to implement the Mali-T678-MP4 being developed for the ARM Reference Platform
  – Demonstrate Cadence methodology for future Mali GPU’s.
GPU Core Challenges

• **GPU Processors**
  – Hugely data parallel workloads
  – 100s-1000s threads
  – 1 - Many cores
  – Long pipelines (>50 stages)
  – Copes with very high latency
  – High throughput

Multiple shader cores
  – Unique challenge to hierarchical design

Complex datapath structures
  – GPU = complex math

• **Power efficiency is key**
  – Optimizing for maximum performance
  – Selecting the appropriate shader core configuration
Mali-T678 iRM floorplan

- 9 power domains
- 8 switchable power domains
- Bottom up flow with 6 share core macros
Flexible Hierarchical Implementation Flow

- **ARM Mali GPU supports multiple numbers of shader cores**
  - Each shader core is 2.6M instances
  - 2-8 shader cores in a configuration

- **Flow must support separate shader core / top level**
  - Supporting parallel runs
  - Goal is to minimize iterations regardless of configuration

- **Developed hierarchical RTL synthesis strategy**
  - Complementary to proven hierarchical implementation flow
Multi-Bit Cell Inferencing (MBCI)

- **MBCI Flow**
  - Merge single-bit flops into multi-bit version of flops – dual or quad flop versions (*requires MB cells in library*)
  - QOR driven multi-bit mapping
  - Clean verification flow with LEC

- **Benefit**
  - Sharing clock enables reduces power

- **Impact on Mali-T678**
  - Did MBCI on the Shader
  - 88% conversion rate (swap ratio)
    - Using 2x only flops
  - No loss in performance
Reduce Power up to 10% while meeting Timing
Physical Aware Multi-Bit Cell Inferencing (PA-MBCI)

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>Logical MBCI</th>
<th>Physical MBCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>WNS</td>
<td>-0.121ns</td>
<td>-0.129ns</td>
<td>-0.123ns</td>
</tr>
<tr>
<td>TNS</td>
<td>-104.69ns</td>
<td>-140.84ns</td>
<td>-104.49ns</td>
</tr>
<tr>
<td>Utilization</td>
<td>77.04%</td>
<td>73.39%</td>
<td>74.59%</td>
</tr>
<tr>
<td>Seq Power</td>
<td>97.55</td>
<td>47.18</td>
<td>65.04</td>
</tr>
<tr>
<td>Total Power</td>
<td>450.67</td>
<td>277.60</td>
<td>345.77</td>
</tr>
</tbody>
</table>
Key Shader Core Challenges and Solutions

- **Clock SI**
  - Challenge: much higher SI at 28nm, too much congestion to fix traditionally
  - Solution: use CCOpt to shield nets higher in the clock tree
    - No impact to congestion

- **Data SI**
  - Challenge: predictable closure flow
  - Solution: tighten pre-SI timing to enable more pre-IPO transforms
    - Cuts down on updates post-route
    - Fewer jumps in closure

- **Hold Fixing**
  - Challenge: lots more hold fixes needed at 28nm
    - Shader core had regions of high density
  - Solution: used cell padding on registers to allow room for hold buffer insertion
Key technologies for implementing high-performance GHz + ARM processors and Mali GPU's
Key improvements to Encounter digital tools

**RC/RCP**
- Multi-objective low-power and physical aware global synthesis
- Better WNS and total power
- Tighter correlation to EDI System
- Multi-bit cell inference (MBCI)
- Improved TAT

**GigaOpt**
- Fine grained, multi-threaded logic transformations + TNS driven optimization
- Dynamic/leakage power minimization
- Route-driven/layer-aware opt
- Much improved WNS/TNS/TAT*

**CCOpt**
- CCOpt improves even fine-tuned ARM processor timing
- Useful skew based time borrowing across entire pipeline chain
- Improved MMMC/OCV/setup/hold
- Improved PPA than std. CTS solution

*WNS = worst negative slack
TNS = total negative slack
TAT = turn-around-time
CCOpt=Clock Concurrent Optimization

In-Design Signoff
(RCX, Timing, Power, DFM)
Next-Gen Physical Aware RTL Synthesis
Next-Gen Physical Aware RTL Synthesis
Faster Synthesis, Better QoR and Convergence

GigaHertz Performance
Up to 50+% TNS, 10% WNS, 10% total power savings

Congestion Optimization
Structure and map generic logic with physical placement knowledge at the RTL stage

Tight Correlation to EDIS
Generate predictive netlist for P&R
Faster full-flow TAT by reducing Synthesis to P&R iterations

Available in RTL Compiler 13.1 (Nov’13):
PAS/PAM/PAR, PA-MBCI, PA-ECO, PA-DFT

Early access software in use by 9 of the top 10 Semiconductor Design Companies
Unique Placement Technology

Reg array 1st stage muxes 2nd, 3rd … stage muxes 1st stage muxes

Traditional Placement

Desired Placement

15 © 2012 Cadence Design Systems, Inc. All rights reserved.
Unique Placement Technology

Encounter Placement Engine – streamlines data line and select lines

Traditional Placement
GigaOpt: Next generation optimization technology pervasive through the EDI System flow
GigaOpt technology

ROI-Driven Optimization
Power vs. Performance vs. Area

Score = \frac{gain}{pain}

Logic Transforms
Placement Transforms
Routing Transforms

Threaded "sub-network" transform engine

Threaded MMMC timing
Threaded current source model base+SI delay calc
Threaded extraction
Threaded routing
Design database
Route-driven optimization

- **Size a Gate**
- **Buffer a Net on a high metal layer**
- **Gate composition e.g. NAND/NOR → AOI**
- **Gate decomposition e.g. AOI → NAND/NOR**
- **Bubble push**

**GigaOpt**

**13.1 default**

- M9/M10
- M7/M8
- M5/M6

**Buffer a Net**

- **Route-driven optimization**
  
  - **Buffer a net**
  - **Gate composition** e.g. NAND/NOR → AOI
  - **Gate decomposition** e.g. AOI → NAND/NOR
  - **Bubble push**

**Graph:**
- Capacitance/um
- Resistance/um

(data normalized to 1.0 at 90nm)
Route-driven optimization results

<table>
<thead>
<tr>
<th>Design</th>
<th>11.1 Freq</th>
<th>13.1 Freq</th>
<th>11.1 TNS (ns)</th>
<th>13.1 TNS (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design A</td>
<td>360 MHz</td>
<td>723 MHz</td>
<td>-3,400</td>
<td>0</td>
</tr>
<tr>
<td>Design B</td>
<td>347 MHz</td>
<td>725 MHz</td>
<td>-5,153</td>
<td>0</td>
</tr>
<tr>
<td>Design C</td>
<td>420 MHz</td>
<td>645 MHz</td>
<td>-2,080</td>
<td>-27</td>
</tr>
<tr>
<td>Design D</td>
<td>435 MHz</td>
<td>631 MHz</td>
<td>-5,694</td>
<td>-77</td>
</tr>
<tr>
<td>Design E</td>
<td>435 MHz</td>
<td>684 MHz</td>
<td>-13,624</td>
<td>-32</td>
</tr>
</tbody>
</table>

57% Average Improvement
CCOpt: Native integration in EDI System
Clock Concurrent Optimization Technology

Traditional CTS and Post-CTS Opt

Clock Concurrent Optimization

CCOpt is natively integrated in the EDI System, delivering Excellent PPA and TAT improvements.
Native CCOpt integration

Native CCOpt is a limited-access feature in release 13.1
Native CCOpt results (28nm node)

<table>
<thead>
<tr>
<th>Tool</th>
<th>WNS</th>
<th>TNS</th>
<th>Runtime</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-power mobile 32-bit CPU</td>
<td>Scripted</td>
<td>-57ps</td>
<td>-12ns</td>
<td>34hr 15min</td>
</tr>
<tr>
<td></td>
<td>Native</td>
<td>-15ps</td>
<td>-0ns</td>
<td>3hr 56min</td>
</tr>
<tr>
<td>32-bit CPU for tablets and smartphones</td>
<td>Scripted</td>
<td>-85ps</td>
<td>-297ns</td>
<td>101hr 17min</td>
</tr>
<tr>
<td></td>
<td>Native</td>
<td>-12ps</td>
<td>-0ns</td>
<td>25hr 35min</td>
</tr>
<tr>
<td>Complex 64-bit CPU for datacenters</td>
<td>Scripted</td>
<td>-99ps</td>
<td>-26ns</td>
<td>34hr 36min</td>
</tr>
<tr>
<td></td>
<td>Native</td>
<td>-76ps</td>
<td>-3ns</td>
<td>23hr 25min</td>
</tr>
<tr>
<td>Quad-core 32-bit CPU hard-macro</td>
<td>Scripted</td>
<td>-77ps</td>
<td>-553ns</td>
<td>62hr 45min</td>
</tr>
<tr>
<td></td>
<td>Native</td>
<td>-15ps</td>
<td>-1ns</td>
<td>21hr 57min</td>
</tr>
</tbody>
</table>
Design signoff with Tempus
Introducing Tempus Timing Signoff Solution

New technology accelerates timing analysis and closure by weeks

Performance
Massively parallelized
Scalable to 100s of CPUs
Optimized data structures

Accuracy
10X faster path-based analysis
Advanced process modeling

Closure
10X reduction in closure time
Unlimited MMMC capacity
Path or graph-based optimization

Certified by TSMC for 20nm designs

Endorsed by Texas Instruments for advanced giga-scale, giga-performance ICs
Summary

Cortex A12

- ARM and Cadence are working with lead partners on the implementation of the Cortex-A12
  - Cadence Cortex-A12 iRM is available today from ARM.
- Next version of the Cortex-A12 iRM will use upcoming Cadence releases for EDI, RC/RCP, Tempus.

Mali T600 Series GPU

- ARM’s Mali GPU is a significant advancement in mobile graphics
  - Advanced scalable architecture
  - Optimal power/performance balancing
- Cadence’s implementation technology is perfectly suited to efficient Mali GPU implementation
  - Multi-objective global synthesis
  - Physically aware convergent flow throughout
  - Advanced clock tree optimization for power and performance
  - Power efficient multi-bit register merging
  - Efficient hierarchical flows
- The Cadence flow for the Mali-T678 is available from ARM