Advanced Implementation of ARM® Cortex®-A57 / Cortex-A53 CPUs and ARM Mali™ GPUs in TSMC 16nm FinFET Process

ARM Tech Symposia
Nov 2014
Today's Markets and Design Focus

Consumer
- >90nm
- 65nm
- 40nm
- 28nm

Mobile/Tablet
- 28nm
- 20nm
- 16nm

Enterprise
- 16nm
- 20nm
- 28nm
Implementation Targets

- High Performance Mobile Cluster (big)
  - Cortex-A57 MP2/4, Fmax
- High-Performance Server/Networking/Enterprise
  - Cortex-A57 MP4 + ECC, Fmax
- Low-Power Cluster (LITTLE)
  - Cortex-A53 MP4, Low-Power
- Low-Power Server
  - Cortex-A53 MP4+ECC, Fmax
Being Late to Market Costs $$$

- Development cycle for mobile SoCs are at least 6-8 months

- Typical life-cycle of mobile SoC is 12-18 months
  - Maximum value is in the initial stages

- Any delay in launching the SoC seriously impacts profitability
  - Loss of revenue due to product delay
  - Opportunity cost due to impact to subsequent products
The New Paradigm of Cortex-A Implementation Success

Predictable Results in a Predictable Time-Line
Comprehensive Platform for TSMC 16FFLL+ FinFET Process

**Products**
- **Standard Cell Libraries**
  - High Performance
  - High Density
  - Ultra High Density

- **Memory Compiler**
  - Single, Dual, 2-port and ROM
  - High performance and area/power optimized version
  - Multiple periphery options

- **POPTM IP**
  - Cortex®-A57/A53
  - Mali™-T760

- **GPIO**
  - 3.3V and 1.8V versions
  - High density, fully programmable

**Market & Solutions**
- **High-end Mobile**
  - Optimized performance with stringent power/area budgets for world-class smartphones
  - State of the art power management features

- **Server and Networking**
  - High performance IP for compute and networking applications
  - Advanced feature set enables optimized system integration

- **Implementation Solutions**
  - POP IP for best in class performance and TTM
  - Family of Architect products for fast and reliable chip design

**Availability**
- **Aggressive Schedule**
  - Production quality EAC (PDK 0.9) release available now
  - EAC release (PDK 1.0) under development, release planned for beginning 2015

- **Extensive Si Validation**
  - Multiple functional testchips with Cortex-A57 and A53 successfully taped out and verified
  - IP testchips with early production IP in fab
  - 1.0 PDK Testchips to be taped-out in Q1 2015
Platform IP & POP IP™ for Total Chip Success

- Majority of today's ARMv8-A cores are initially implemented with ARM Artisan® TSMC 16FF+ platform

- Artisan 16/14nm platforms provide excellent integration with ARM POP IP

- Combining ARM POP IP with ARM memory compilers and standard cells provide major advantages for design teams
  - Streamlined design flow based on consistent set of deliverables with identical look and feel, i.e. EDA views, PVT corners
  - Utilization of ARM Artisan Sign Off Architect and Power Grid Architect Technology for entire chip design
ARM Artisan Optimized Implementation Solutions
Easing transition to 16nm FinFET process technology

ARM Artisan Power Grid Architect
60% Utilization

ARM Artisan Signoff Architect
80% +10% Utilization Area Reduction

Improving entitlement for area scalability
Challenges and Solution for FinFET Power Grid Design

- Double pattern M1, M2 and M3
- Half track libraries
  - Different rails for VDD and VSS
  - Sometimes via1 needs to be inserted post-route
- Complex DRCs and Pitch misalignment
  - Row end, top/bottom, and corner cap boundary cells
  - Default usage of P&R commands will not create correct grids

ARM Artisan Power Grid Architect

- Insert power grid and power gates
- Inserts layout finish cells and detects violations

Faster floorplan generation reduces development time!
Optimized power rails result in higher utilization for smaller area!
Challenges and Solution for FinFET Design Sign-off

- Use of SB-OCV (also called Advanced OCV - AOCV)
  - Table of derate values for each cell indexed by path depth
  - Mature EDA support
  - ARM studies show that method is not comprehensive! Does not account for load and slew and can lead to a false sense of security about timing closure

ARM Artisan Signoff Architect

- Includes derate values for alternate slews and loads
- Includes methodology to identify instances with load/slew outside of the range of the initial table
- Reassigns more accurate derate values for these outlying instances

Increasing sign-off accuracy and reducing design risk!

![Diagram showing Delay Variation vs. Load/Slew](chart)

- High $\sigma/\mu$ means large variability
- Low $\sigma/\mu$ means small variability
- The arrow indicates that a single $\sigma/\mu$ point in the load/slew table is chosen for SB-OCV tables
- The variation cell delay for instances with loads or slews beyond that point is under-estimated by SB-OCV and is covered with ARM Artisan Signoff Architect
TSMC 16FFLL+ Process Offers a Wide Choice for Implementation

- 3 bit-cells
- 4 Vt
- 3 Channel Lengths (c16, c18, c20)
- 3 Logic Architectures (7.5T, 9T, 10.5T)
- Metal Stacks
- Implementation Tuning
Using Shmoo Analysis to Select the Right Vt/Channel Length

- Synthesis shmoo is generated to understand performance & leakage trends with different Vt/Channel length

- Helps us to make the right choices for implementation before the implementation

- Large gap between LVT and uLVT
  - Suboptimal Forward fill if starting with LVT
  - Suboptimal Backward fill if starting with uLVT
Selecting the CPU Configuration & Implementation Choices

<table>
<thead>
<tr>
<th></th>
<th>ARM Cortex-A57</th>
<th>ARM Cortex-A53</th>
<th>Mali-T860</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>High-performance (big)</td>
<td>Low-power (LITTLE) for mobile</td>
<td>Low Power Server</td>
</tr>
<tr>
<td>L1</td>
<td>MP4</td>
<td>MP4</td>
<td>MP4</td>
</tr>
<tr>
<td>L1</td>
<td>48K/32K</td>
<td>32K/32K</td>
<td>64K/64K</td>
</tr>
<tr>
<td>L2</td>
<td>2MB</td>
<td>1MB</td>
<td>1MB</td>
</tr>
<tr>
<td>Logic Architecture</td>
<td>SC10.5/SC9</td>
<td>SC9</td>
<td>SC10.5/SC9</td>
</tr>
<tr>
<td>Vt Allowed</td>
<td>SVt, LVt, uLVt</td>
<td>SVt, LVt</td>
<td>SVt, LVt, uLVt</td>
</tr>
<tr>
<td>Channel Length</td>
<td>c16,c18,c20</td>
<td>c16,c18,c20</td>
<td>c16,c18,c20</td>
</tr>
<tr>
<td>Metal Stack</td>
<td>9LM*/11LM</td>
<td>9LM*/11LM</td>
<td>11LM</td>
</tr>
</tbody>
</table>

* Customer Choice
POP IP Implementation Strategy

**Synthesis**

- Physically aware synthesis
- Synthesis based on floorplan and QRC techfile for accurate extraction
- Floorplan tuned over multiple iterations
  - Slack driven placement.

**Flat Derates**

- 5% on launch paths
- 7% on data paths
- 10% on capture paths
- These derates give a good correlation to the AOCV results for LVT based implementation
- Flat derates need to change for any other VT as AOCV is VT dependent.
Iterative Floor-Plan Analysis to Improve Performance

- POP IP development requires analysis of results on exhaustive floor-plan trials
- More than 20 iterations of the floor-plan with different placements & aspect ratios
- Start with data flow analysis
  - How data flows inside the RTL
Clock Tree Synthesis

Primary goals for building a clock tree

- Minimize the latency through the clock tree
  - Reduce OCV which is generally applied as a % of the delay and directly effects many setup and hold paths

- Minimize the power in the clock tree, dominated by dynamic power

- Reduce the skew but allow for applying useful skew as necessary
Variation of Dynamic and Total Power with Vt/Channel Length

In 16FFLL+, Vt/Channel Length Optimization is a trade-off of Performance v/s Total Power, not just leakage power!

### Normalized PPA across Vt/Channel Length

<table>
<thead>
<tr>
<th>Vt Splits</th>
<th>UL16</th>
<th>UL18</th>
<th>UL20</th>
<th>L16</th>
<th>L18</th>
<th>L20</th>
<th>S16</th>
<th>S18</th>
<th>S20</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssgn0.72v0.0c</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
</tr>
<tr>
<td>tt_0.80v85c</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
</tr>
<tr>
<td>tt_0.80v85c</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
</tr>
<tr>
<td>tt_0.80v85c</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
<td>by area</td>
</tr>
</tbody>
</table>

#### PPA Metrics

- **Performance (WC_NOM)**
- **Performance (TT_NOM)**
- **Leakage Power (TT_NOM)**
- **Dynamic Power (TT_NOM)**
- **Total Power (TT_NOM)**
- **Rectilinear Area**

#### Table Entries

- **9T ULVt-based 11LM**: 2.13x 1.70x 28.33x 1.47x 2.95x 1% 1% 59% 6% 7% 12% 1% 2% 11%
- **9T LVt-based 11LM**: 1.49x 1.30x 2.73x 1.30x 1.69x 1 0% 0% 0% 1% 3% 60% 5% 5% 26%
- **9T SVt-based 11LM**: 1.00x 1.00x 1.00x 1.00x 1.00x 1 0% 0% 0% 0% 0% 2% 9% 89%
New L2 Cache Instance Co-Development with CPU Micro-Architecture

Lower L2 Cache Power

30% reduction in L2 standby

Customized cache instances linked directly into ARM CPU micro-architecture
POP IP FCI Support ARMv8-A CPU ‘Light Sleep Mode’

ARMv8-A CPU support new power mode called ‘Light Sleep Mode’

‘Light Sleep Mode’ allows CPU to put L2D memory into sleep mode **speculatively** based on pipeline analysis.

1. CPU, based on pipeline analysis, determines L2 access will not be required for next few cycles.
2. LIGHT SLEEP signal asserted.
3. Memory periphery is powered down.
4. Note that CPU can continue executing from L1 while L2D is power-down.
5. CPU, based on pipeline analysis, determines need to access L2.
6. LIGHT SLEEP signal de-asserted (wake-up).
7. Memory is powered up in ONE cycle.

- **SoC-level leakage savings depends on software being run**
- **Example**
  - Dhrystone vector (no L2 access) ➔ Max leakage savings
  - Memcopy_L2/ max_power vector ➔ Lesser leakage savings

Copyright © 2014 ARM
Power Gating Topology

The power gating topology is determined based on the wake-up time requirements.

Typically we use the Hammer-Trickle combination as shown in the figure here for enabling the power gating in our design.

We can have a single chain or multiple chains to reduce the wake-up time.
Determining the Number of Power Switches

- Based on the below equation, we determine the number of trickle switches:
  - $$P_{\text{total}} = P_{\text{leakage}} + P_{\text{dynamic}}$$
  - $$I_{\text{total}} = \frac{P_{\text{total}}}{\text{Voltage}}$$
  - Total No of Trickle Switches = $$\frac{I_{\text{total}}}{I_{\text{D-Sat}}}$$
    - $I_{\text{D-Sat}}$: This is the saturation current for one switch cell

- The Headbuf/Head cells are selected based on following parameters:
  - Total count of trickle should be small enough to meet the wake-up requirement
  - The total area increase because of Hammer and Trickle switches should be within budget
  - The $V_t$ and CL of the gates should be selected such that the total leakage in ON state should be minimal.
ARM POP IP Enables New Era of Partnership

SoC Designer

Implementation Support

ARM POP IP

RTL Optimization

EDA Scripts

Process Optimization

Foundry

EDA Flow

ARM Cortex-A CPU

ARM POP IP Implementation Support

ARM POP IP RTL Optimization

ARM POP IP Process Optimization

ARM POP IP Foundry
Complexity of 64-bit CPU implementation
- 64-bit CPUs need more engineering & compute resources
- ARM Cortex-A57 run-time in several days

16FinFET is exponentially more challenging to implement
- 16FF has ~2X more design rules than 28nm

“Winner takes all” – time-to-market is very important

Flexible solution
Address a wide variety of market segments

Reduce Risk
Leverage ARM expertise

Reduce Development Cost
Phy. IP + Scripts + Implementation Knowledge
The New Paradigm of Cortex-A Implementation Success

Predictable Results in a Predictable Time-Line
Thank You

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Any other marks featured may be trademarks of their respective owners.