proAptiv: **Efficient Performance**
on a Fully-Synthesizable Core

28 August 2012

Ranganathan “Suds” Sudhakar
Chief Architect
Three new cores optimized for embedded markets
Aptiv Core Portfolio

Classic MIPS Products

- **24K/24KE Series**: 8-stage pipeline
- **34K Series**: Multi-threading, 9-stage pipeline
- **74K Series**: Out of Order Dual issue, MP version
- **1004K Series**: Multi-threading, 9-stage pipeline
- **1074K Series**: MP version

Aptiv™ Generation

- **proAptiv™ Family**: Single-Threaded Area Optimized
  - 1 to 6 core configs, Hi-speed FPU and L2 cache controller

- **interAptiv™ Family**: Multi-Threaded Power Optimized
  - 1 to 4 core configs, 2-level MT/FPU and L2 cache controller

- **microAptiv™ Family**: DSP-Accelerated Energy Optimized
  - MCU (cacheless) or MPU (caches/TLBs) with real-time/security

Classic MIPS Products

- **24K/24KE Series**: 4 stage pipeline
- **M4K/4KE Series**: 4 stage pipeline
- **M14K/c Series**: Code compression, 5 stage pipeline
- **1004K Series**: Multi-threading, 9-stage pipeline
- **74K Series**: Out of Order Dual issue, MP version
- **1074K Series**: MP version
What is a “Soft” Core?

- **Fully synthesizable “package”**
  - Design data
    - RTL
    - Configurator – MP/MT, FPU, Trace/Debug, cache/TLB/SPRAM/buffer sizes, bus widths
  - Physical design support
    - Reference floorplans, Synthesis + Place-and-Route scripts
    - DFT/Scan, Timing and Power Analysis scripts
  - Simulation models
    - Bus Functional Models and compliance checkers
    - Instruction accurate simulators, Cycle exact simulators
  - Verification collateral
    - Architectural Verification Test suites, core diagnostics
    - Sample testbench, build and run scripts
  - Documentation
    - ISA manuals, global configuration register tables, memory maps, boot procedures

- **Available separately**
  - FPGA development boards
  - EJTAG/debug probes
  - OS components, libraries, software toolchains (compiler, libraries, JITs, codecs)
What is a “Hardened” Core?

- **Tapeout-ready GDS, built on a generic ASIC flow, using:**
  - Configured soft core
  - Floorplan – placement of RAMs, bounding box
  - Physical IP for some process technology
    - Standard cell library
      - e.g. 28nm low-leakage 12-track mixed-Vt with booster flops
    - Compiled memories
      - e.g. 28nm high-speed LVT single + dual-port bit-writable memories
    - Fab conditions
      - Process corner *(usually worst-case slow-slow, high-temp, low voltage)*
      - Number of metal layers, DRC/LVS, power grid, IR drop, OCV/AOCV, PLL jitter

- **Not to be confused with a “hard core”**
  - Frequency and power improvements beyond simple hardening:
    - Custom std cells, flops, clk-gaters characterized for typical silicon
      - e.g. 1.x GHz worst-case SVT → 2.x GHz typical with LVT, overdrive, cooling
    - Multi-port register files and custom memories for cache arrays
      - Hierarchical floorplans, structured placement, mesh clocking
Hardened proAptiv Layout

Branch Prediction RAMs (1R1W)

MBIST Engines Included

Stdcells, Clock tree, Power grid

I-cache RAMs

TLB RAMs

D-cache RAMs
Soft Core Design Considerations

- **Life revolves around flops (and muxes)**
  - No CAMs – schedulers, TLBs, BTBs all built from flops
  - No ROMs – div/sqrt lookup tables all built from gates
  - No multiports – Register files, reorder buffers all built from flops
    * Read ports are large muxes \(\sim O(\text{num\_entries})\)
    * Write ports are small muxes \(\sim O(\text{num\_ports})\)
  - Exceptions are:
    * 1RW RAMs for use in cache/TLB arrays
    * 1R1W RAMs for use in branch prediction arrays
      * Used judiciously -- proAptiv is the first MIPS soft core to use these

- **Sophisticated techniques cannot be easily employed**
  * Banking, sum-addressing or one-hot-indexing
  * Dynamic circuits, especially negedge-triggered

- **More pipestages needed for a given frequency**
  * MIPS’s pure RISC ISA helps counteract this
Soft Core Timing and Verification Challenges

- **Timing paths not consistent**
  - Variations in floorplan, configuration, stdcell, memory IP
  - Variations in operating point – fab, process, Vt mix, overdrive
  - Variations in EDA tool margins, flows, vendors and versions

- **But good enough!**
  - Balance logic across pipestages
  - Ensure loop paths are minimal and reflected in the microarchitecture
  - Ensure floorplan reflects critical unit and pin placement

- **Specific considerations for high-frequency pipelines**
  - Any CAM-RAM structures take at least 2 clock cycles
  - Regfile read+bypass takes at least 2 clock cycles

- **Need to fix timing paths at all phases of the implementation**
  - Synthesis, Place, Route, Clocking (No ECOs or manual tuning allowed)

- **Verification challenges**
  - Dozens of configuration variables but still need high code+functional coverage
proAptiv Coherent Processing System (CPS)

- **Modern Superscalar Core**
  - Integrated L2$ Controller with ECC
  - Improved CM performance
  - Supports configurations up to 6 cores

- **Enhanced Coherence Manager**
  - Up to 2 IOCU Blocks

- **IO Coherence Unit (IOCU)**
  - Up to 256 system interrupts

- **Global Interrupt Controller**
  - Voltage domain/gating per core
  - Clock gating per core
  - Software programmable

- **Cluster Power Controller**
  - Up to 256 system interrupts

- **PDtrace™ – cluster level support**

---

### proAptiv Coherent Processing System (CPS)

- **proAptiv CPU 1**
- **proAptiv CPU n (Up to 6)**
- **Coherence Manager**
  - L2 Cache Controller
  - Main Memory Non-Coherent I/O
  - Coherent IO Devices
- **Global Interrupt Controller**
- **Configuration Registers**
- **IOCU**
- **PDtrace**
- **Cluster Power Controller**
- **Interv Port**
- **Req Port**
- **Optional**
proAptiv Design Goals

- **Fast**
  - Optimized for mobile computing and networking
  - Multi-issue dynamically-scheduled operation
  - Deep pipeline to achieve multi-gigahertz operation
  - Brand new high-frequency FPU matched to core

- **Efficient**
  - Elegant balanced microarchitecture, not brute force width and depth
  - Minimal area for cost and leakage; fine-grain clock gating
  - Reduces the need for costly heterogeneous schemes

- **Scalable**
  - New 6-core Coherence Manager and 256-bit L2 cache controller

- **Robust**
  - Age-based scheduling, careful tuning of predictors/prefetchers
  - Easy to add features and performance, vary microarch parameters

- **Feature set**
  - MIPS32 R3 / MIPS16e, DSP ASE v2, PDtrace v6, Enhanced VA
proAptiv Base Core Architecture

- **Superscalar OoO CPU – 16 stage**
  - Quad inst fetch
  - Triple bonded dispatch
  - Inst peak issue: quad integer; dual FP

- **Sophisticated branch prediction and L0/L1/L2 BTBs, RPS, JRC, way predicted instruction cache**

- **High performance, multi-level TLBs, way predicted data cache**

- **Instruction Bonding makes six issue pipes look like eight**

- **Fast integer divide, multiply and multiply-accumulate operations**

- **Dual Issue FPU**
  - Higher speed (1:1 with CPU)
  - Lower latency on most operations
  - Single-pass double precision
  - More parallelism and dedicated schedulers – more ops in flight
proAptiv Pipeline

16 stage integer load pipeline

- Instruction Fetch: 4 cycles
- Branch Prediction: 2 cycles
- Bonded Dispatch: 4 cycles
- Issue Regread: 4 cycles
- Agen Dcache: 2 cycles
- Regwrite Graduate: 2 cycles

Floating-Point Decode, Rename, Issue, Regread: 4 cycles
FADD FMUL: 4 cycles
Regwrite Graduate: 2 cycles

User Defined Instruction Unit: Variable cycles
Regwrite Graduate: 2 cycles
proAptiv Instruction Fetch

- 16-entry ITLB
- 32 or 64KB I-cache
- 4-way associative LRU
- 32-byte line
- Parity protected
- Fetches 16 bytes per cycle
  - Aligned fetch reduces power
  - Dynamic 8 byte bundle reduces power on branches or MIPS16e
- Next Fetch Way predictor reduces power by reading only one way
  - Sequential fetch way from SRAM
  - Target way from BTB
- Next Line Prefetcher
  - Variable number of lines on a miss
  - Direct bypass from refetch queue
- Extra pipestages inserted for MIPS16e fetching and unpacking
- Credit-based instruction buffer
proAptiv Branch Prediction

**Branch History Tables**
- Predicts branch direction
- Novel algorithms deliver class-leading prediction accuracy
- Uses sophisticated global history
- Can predict 2 (MIPS32) or 4 (MIPS16e) branches per cycle
- Multiple SRAM-based tables
  - Only 1R1W structures on chip
- Leverages delay slots to minimize storage capacity needed
  - In MIPS, unlike some ISAs, the delay slot cannot itself be a branch

**Branch Target Buffers**
- Provides fast target prediction
- Multiple buffers with various latencies and sizes, up to 512 entry 4-way

**Jump Register Cache**
- Predicts indirect jumps
- Multiple targets per jump PC

**Return Prediction Stack**
proAptiv Instruction Dispatch – Bonding

- **Combine adjacent instructions into single bonded op**
  - e.g. consecutive LW or SW instructions
  - e.g. branch with certain instructions in delay slot
    - Fused compare-branch is already part of MIPS integer ISA

- **Load/Store bonding makes one memory pipe look like two**
  - 1 DTLB, 1 tag array, single-ported data array saves area
  - Single DTLB and cache access saves energy, power
  - Occupies only 1 entry in various queues/buffers – more ILP
  - Carried forward as one operation on L1-miss – more MLP
  - Speeds memset, bcopy, strcmp, spill-fill, GPU communication

- **Design decisions**
  - Initially limit to two instructions, aligned addresses and ST
    - But designed to scale to four, misaligned accesses and MT
  - Therefore, needs a **bonding predictor** in the front-end
    - Trained by LSU (memtype must be cacheable or write-combining)
    - Indexed by PC and other control flow information

MemCopy Loop:

```
lw r1, 0x0(r20)
lw r2, 0x4(r20)
lw r3, 0x8(r20)
lw r4, 0xc(r20)
lw r5, 0x10(r20)
lw r6, 0x14(r20)
lw r7, 0x18(r20)
lw r8, 0x1c(r20)
sw r1, 0x0(r21)
sw r2, 0x4(r21)
sw r3, 0x8(r21)
sw r4, 0xc(r21)
sw r5, 0x10(r21)
sw r6, 0x14(r21)
sw r7, 0x18(r21)
sw r8, 0x1c(r21)
addiu r20, r20, 0x20
addiu r21, r21, 0x20
bnez r23, Loop
sub r23, r23, r22
```
Bonded stores have 3 source registers
- 1 address and 2 data GPRs
  - Compared to 2 sources for ordinary stores
  - Requires 1 more read port at execute than unbonded machine

Hence cracked into decoupled operations
- STA (Store Address) – 1 reg source
- STD (Store Data) – 2 reg sources

STA reads cache tags and detects L1 miss early
- Requires only 1 read port in load-store pipe

STD delivers data to LSU in memory aligned format
- Requires only 2 read ports
- Thus avoiding the need for any pipe to have 3 ports

Some stores are never cracked
- e.g. Misaligned stores, where data depends on address

Some stores are always cracked
- e.g. FPU stores, where the integer scheduler has no visibility or control over the FP register file and issue ports
proAptiv Instruction Issue – Segmented Scheduler

- **Two issue queues**
  - Neither single large unified queue (low-frequency)
  - Nor too many small distributed schedulers (high power)

- **1 ALU issue queue and 1 AGU issue queue**
  - Check dependencies and structural hazards
  - STA and STD share same scheduler entry, reducing area/power

- **Age-priority scheduling**
  - Requires age-vector per entry to pick oldest
  - Allows non-shifting schedulers with fewer comparators/muxes for low power
  - Minimal CAM logic – timing friendly

- **No reservation stations**
  - Read registers after scheduling – low power
Holy grail of OoO scheduler design:
- Large (40 – 64 entries) yet fast (able to follow single-cycle dependency chains)

Typical schedulers employ one of two wakeup techniques
- Encoded register-number wakeup (e.g. MIPS R10K)
  - (Wakeup → Pick → Mux) → (Wakeup → Pick → Mux) → …
  - Pick and Mux can sometimes be overlapped
- Decoded entry-number wakeup (e.g. MIPS 1074K)
  - (Wakeup → Pick) → (Wakeup → Pick) → …
  - Usually multi-hot vectors for dependency checking

proAptiv can utilize a third technique
- Transitive Wakeup
  - (Wakeup) → (Wakeup) → (Wakeup) → …
- Only works with decoded entry-numbers
  - Relies on multi-hot broadcasts
  - \{1\} → \{1, 2, 3\} → \{1, 2, 3, 4, 5, 7\} → \{1, 2, 3, 4, 5, 6, 7\}
- Requires strict age-priority scheduling and other constraints
  - Prevents premature pick of a younger op dependent on an older op
  - e.g. inst 6 before inst 4
proAptiv Integer Execution

- **One simple ALU pipe**
  - Handles arithmetic, logical ops and small shifts

- **One complex ALU pipe**
  - Handles a superset of the simple ALU ops – such as large Shifts
  - Handles DSP operations that involve reading or writing the 64b accumulators
    - Accumulators are renamed and treated as two 32b registers
      - Saves power and area compared to designs using 64b rename pool
      - DSP flags are renamed using separate 13b wide pool
        - Allows easy handling of sticky status bit fields
    - Interfaces with Multiply-Divide Unit which also uses the accumulators
      - Supports single-cycle bypass for integer multiply-accumulate
      - New designs for fast multiplication and very fast division

- **One branch/store-data pipe**
- **One load/store pipe**
- **Pipes share read and write ports to further bring down area/power**

- **Thanks to bonding, the 4 physical pipes can actually execute up to 6 MIPS32 integer instructions on a particular clock cycle**
proAptiv Memory Subsystem

- **Designed for large modern workloads**
  - Enhanced Virtual Addressing (EVA) allows efficient access > 3GB
    - Via programmable segments and new kernel load-store instructions

- **LSU**
  - Out-of-Order operation: loads/stores can (with some restrictions) overtake each other
    - Important for performance
    - And efficiency (maximizes utilization of single load-store pipe)
    - But requires:
      - Excellent memory disambiguation and “RAW” hazard avoidance
        - Overeager Load Predictor accessed before insertion into scheduler
          - LSU CAMs detect failure to forward from store buffer and trains predictor
            - Mark a specific load as overeager
          - Predictor forces marked loads to be uneager
            - Scheduler holds overeager loads until all older STA and STD have issued
  - Enforce MIPS’ weakly-ordered memory consistency model
    - Store merging, lightweight and heavyweight SYNCs, cache-ops
    - FP stores can graduate even before receiving store data from FPU

- **BIU**
  - Write-combining and bonding to support streaming writes
proAptiv Memory Management

- **MIPS dual-entry scheme in TLBs**
  - Two VAs differing by 1 address bit share CAM/index portion of entry
  - Separate PA for each of the two VAs

- **Instruction and Data TLBs**
  - Holds 16KB or 4KB pages or sub-pages from VTLB/FTLB
  - 16 entry Instruction TLB
  - 32 dual entry Data TLB
    - Fast adder-comparator logic

- **Variable page size TLB (VTLB)**
  - 64 dual entries, fully associative
  - Holds pages from 4KB – 256MB

- **Fixed page size TLB (FTLB)**
  - 512 dual entries, 4-way assoc
  - Holds either 16KB or 4KB pages
  - Optional at build and runtime
  - SRAM-based implementation
proAptiv Floating Point

- **Brand new high-speed design**
  - Can run 1:1 with proAptiv up to top achievable core frequency
  - Native double-precision datapath
  - FMAC-based pipeline with early and late bypass for FADD/FMUL
    - 4-cycle FADD, 4-cycle FMUL, 7-cycle FMAC
  - Low latency and high throughput for long ops like div/sqrt/recip/rsqrt
    - Functional iterative algorithms and lookup tables compared to bitwise SRT
    - Can run independent instructions under a long op, including other long ops

- **Coprocessor style FPU**
  - Has its own decoupled pipeline, regfile and load/store interface buffers
  - Non-stalling design using shelving buffers to reduce power, improve perf
    - Lower power than PRF-style renaming, given flop-based implementation

- **Formal verification**
  - Against a precise IEEE-compliant mathematical model
proAptiv L2 Cache Controller

- Accompanies both proAptiv and interAptiv cores
- 256KB to 8MB shared across 1 to 6 cores
- 8-way associative
- Selectable 32 or 64B line size
- 256-bit internal datapaths and buffers
- Up to 256-bit interface to system interconnect
- Optional wait states on tag, data or control RAMs
  - Accommodates slow memories, due to:
    - Large size or high-frequency operation
    - HD bitcells, pipelined RAMs, low-voltage operation
- Optional ECC on all RAMs
  - Adds one pipestage
- L2 storage non-inclusive to L1
- Critical-word first; can interleave responses to multiple cores
proAptiv Dual-Core Floorplan

Configuration:

**Per base core**
- FPU
- 32KB/32KB I/D L1$s
- TLB
  - I and D TLBs
  - 128 entry VTLB
  - 1024 entry FTLB
- PDtrace

**Cluster level**
- Dual core coherence
- 1MB L2$ with ECC
- PDtrace aggregator
- 64-Interrupt Controller
- HW IO coherence
- Cluster power controller
- Probe interface block
proAptiv Quad-Core Floorplan

proAptiv Core 1

proAptiv Core 2

proAptiv Core 3

proAptiv Core 4

1MB RAM

CM2

1MB RAM

System I/O
proAptiv Summary

- **Fast**
  - 4.5 EEMBC CoreMark/MHz
    - Highest single-threaded score published for any licensable CPU*
      - 75% over prior MIPS 1074K core
    - Operating frequency > 1GHz worst-case, >> 2GHz typical at 40nm

- **Slim**
  - Highest CoreMark/mm² for any licensable CPU*
    - Dual core area ~ 1MB L2 cache

- **Cool**
  - Highest CoreMark/mW for any licensable CPU*
    - Sub half-watt power at 40nm

- **Efficient performance** on a **fully-synthesizable core**

* CoreMark/MHz derived from publicly available and published scores at [http://www.coremark.org](http://www.coremark.org)
Area and power efficiencies based on MIPS internal and competitive estimates
Thank You!

Questions?

At the core of the user experience®