A 600-MHz VLIW DSP

Sanjive Agarwala, Timothy Anderson, Anthony Hill, Michael D. Ales, Raguram Damodaran, Paul Wiley, Steven Mullinnix, Jerald Leach, Anthony Lell, Michael Gill, Arjun Rajagopal, Abhijeet Chachad, Manisha Agarwala, John Apostol, Manjeri Krishnan, Duc Bui, Quang An, N. S. Nagaraj, Tod Wolf, and Tony Thomas Elappuparackal

***Abstract*—A 600-MHz VLIW digital signal processor (DSP) delivers 4800 MIPS, 2400 (16 b) or 4800 (8 b) million multiply accumulates (MMACs) at 0.3 mW/MMAC (16 b). The chip has 64M transistors and dissipates 718 mW at 600 MHz and 1.2 V, and 200 mW at 300 MHZ and 0.9 V. It has an eight-way VLIW DSP core, a two-level memory system, and an I/O bandwidth of 2.4 GB/s. The chip integrates a c64X DSP core with Viterbi and turbo decoders. Architectural and circuit design approaches to achieve high performance and low power using a semi-custom standard cell methodology, while maintaining backward compatibility, are described. The chip is implemented in a 0.13-μm CMOS process with six layers of copper interconnect.**

***Index Terms*—Cache, digital signal processor (DSP), DMA, signal integrity, turbo decoder, Viterbi decoder, VLIW.**

I. Introduction

TYPICAL high-performance digital signal processing (DSP) applications are very demanding on compute, memory, and I/O bandwidth availability and having the right balance of available resources. A 600-MHz VLIW DSP [1] implements the C64x instruction set architecture (ISA) and delivers 4800 MIPS, 2400 16-b million multiply-accumulate (MMAC) and 4800 8-b MMACs at 0.3 mW/MMAC (16 bit) [1]. The DSP has a VLIW instruction set [2] coupled with 1 MB of on-chip memory and 1.6 GB of I/O bandwidth supported through a rich set of peripherals. The chip is targeted for high-performance DSP applications such as wireless infrastructure, imaging, and video. In particular, for the wireless infrastructure market, a synergistic solution has been devised using the C64x DSP integrated with two coprocessors, the Viterbi decoder and the Turbo decoder, which are required for forward error correction (FEC) algorithms. Cost is a function of the number of channels that a DSP can support for wireless base station systems. The FEC algorithms can consume from 70% to 90% of the DSP processing power [3]. Moving the FEC algorithms to programmable hardware coprocessors results in a greater number of supported channels from 2.7× voice channels and 14.0× data channels. The C64x DSP offers 4× the performance of the previous generation C62x DSP while maintaining backward compatibility. In this paper, the details of the architecture, implementation, and design methodology are presented.

II. Architecture

The processor has an eight-way VLIW DSP core with two-level memory system architecture. A block diagram of the processor is shown in Fig. 1 and the die photo in Fig. 2. The level-1 memories consist of 16 KB instruction and data caches which are connected to a level-2 memory controller. The L2 memory consists of 1 MB of SRAM, which can be partially configured as cache. The enhanced direct memory access (EDMA) controller serves as a concurrent crossbar between a number of peripherals, including timers, multichannel serial ports, PCI and Utopia II interfaces, an external memory interface (EMIF), and Viterbi and Turbo Code coprocessors. The chip has real-time emulation with instruction and data tracing capabilities.

A. CPU Architecture

The CPU core is an eight-way, exposed pipeline VLIW processor. To reduce program code size, the VLIW instruction words are stored in a compressed format in memory. This compressed format removes the instruction information for functional units on which the compiler has scheduled a no operation (NOP) instruction. The instruction format resembles a 32-bit RISC instruction set with marker bits within the 32-bit instructions to indicate which instructions should be issued in parallel. The instruction supply unit fetches and dispatches up to eight instructions in parallel.

One of the design requirements for the C64x architecture is to maintain binary compatibility with the previous C62x processors. The C62x programming model requires that, when a branch is taken, it will take precisely six cycles before the program flow changes to the branch target. Fig. 3(a) illustrates this phenomenon and how it relates to the processor pipeline. The pipeline stages have the following functions:

PG—Program Generate: Generates the next program address to be sent to the memory system

PS—Program Send: Contains some final address selection and allows for transport to the memory system

PW—Program Wait: The CPU is waiting for the memory system to access the memory.

PR—Program Receive: The CPU receives eight new instructions from the memory system.

DP—DisPatch: The CPU dispatches up to eight instructions to the appropriate functional units.

DC—DeCode: Any pending instructions are decoded

E1—Execution1: For instructions which take only a single cycle to execute, the register file will be read, the instruction executed, and the register file will be written.

Manuscript received April 8, 2002; revised June 10, 2002.

The authors are with Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: s-agarwala@ti.com).

Digital Object Identifier 10.1109/JSSC.2002.803954

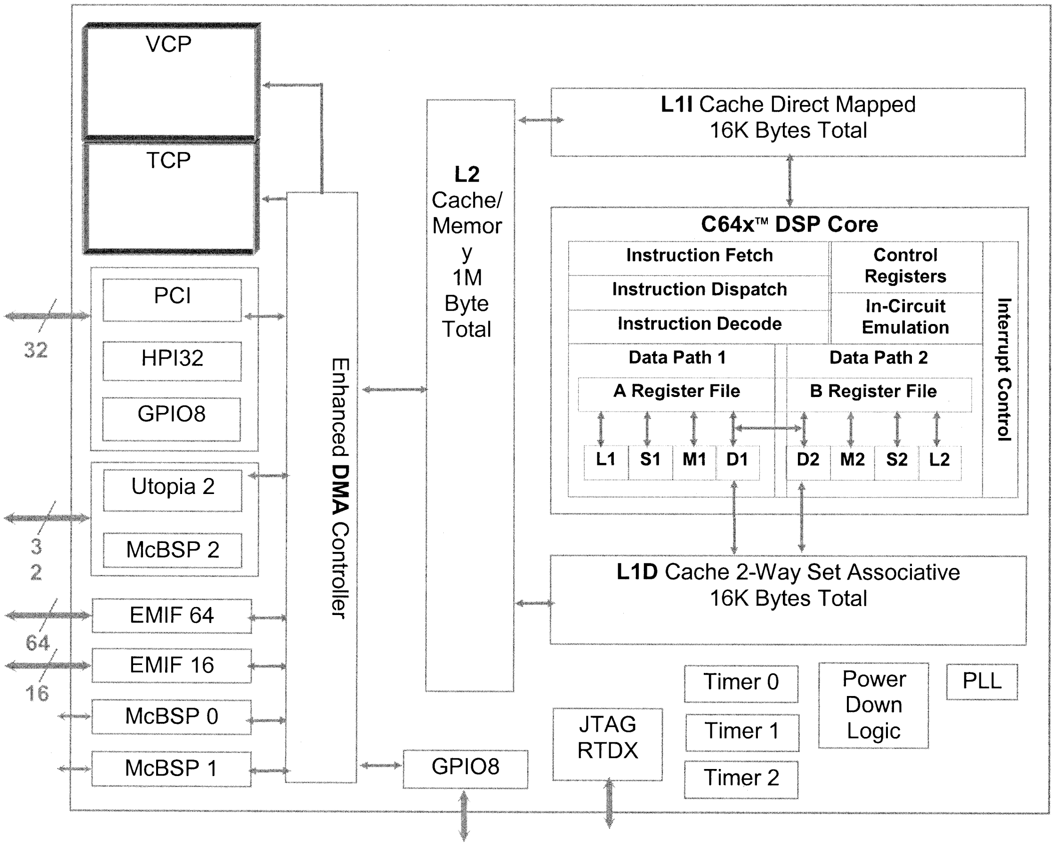


Fig. 1. Chip architecture.

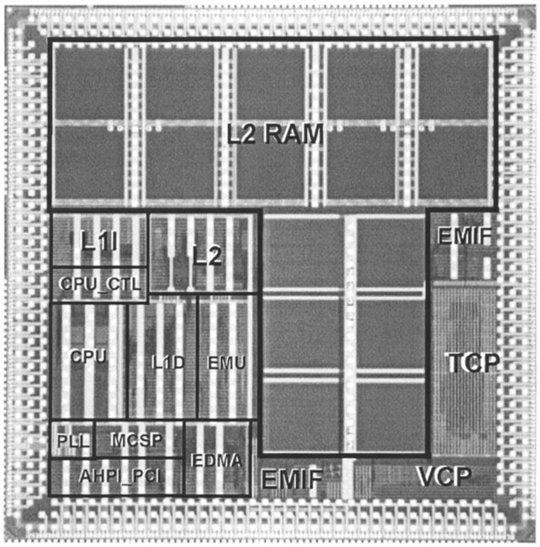


Fig. 2. Chip die photo.

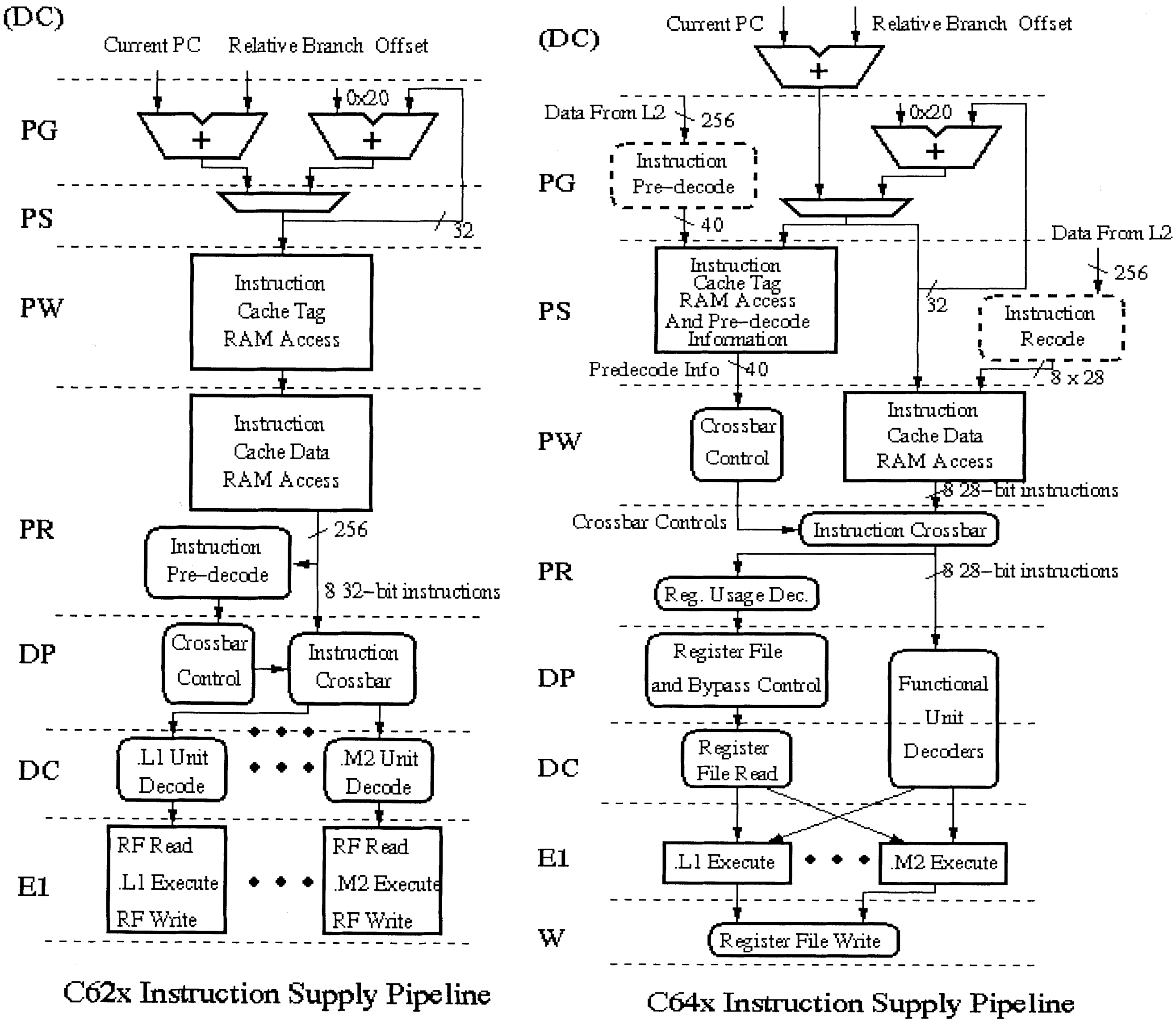
The E1 pipeline phase overlaps with the PG stage. When a branch instruction executes, it will take six cycles before the address generated in the PG stage will arrive at the E1 stage.

In order to achieve the 600-MHz frequency goal, register file bypass is used to help improve the critical E1 computational paths. With register file bypass, the register file (RF) is accessed in the pipeline phase before the execution phase and results are committed to the RF one cycle after the execution phase. Since the C64x has to maintain binary compatibility with the C62x processors, another pipeline stage cannot be added to accommodate the RF requirements. In addition, low power consumption is a primary design goal, so if the RF were to be bypassed in a given cycle the RF read(s) should be suppressed.

Fig. 3(b) shows how the C64x instruction supply pipeline is organized to meet these requirements. The "Crossbar Control" logic is moved back two pipeline phases into the PW stage to allow the RF read to occur in the DC phase. Up to eight instructions are dispatched in parallel at the beginning of the PR stage. The remainder of the PR phase is used to determine which registers each instruction will use. The register usage information is then used by the DP stage to calculate controls for the RF bypass logic and to also disable any RF reads that are not necessary due to RF bypass.

Instructions are recoded from the native 32-bit format into a 33-bit format before storage in the L1 Instruction Cache. The 33-bit format is divided into two parts: 5 bit are used by the dispatch logic in the PW and PR phases and 28 bit are used by the destination functional unit and the RF. These 5-bit codes contain the same information that the "Instruction Pre-decode" block produces in the PR stage of the C6211 pipeline, namely, instruction parallelism and functional unit destination. The 28-bit portion of the instruction is also encoded to optimize the "Register Usage Decoders" in the PR phase. Finally, the adder, which computes relative branch destination addresses, is placed in the DC phase, so that the next program address is available before the start of the PS phase.

The C64x ISA [2] contains many instruction additions over the previous generation to accelerate general DSP applications



(a) (b)

Fig. 3. C62x pipeline diagram.

as well as targeted instructions to address broadband communications, wireless communications, and video applications. Each multiplier unit is capable of performing two 16-bit or four 8-bit multiplies per cycle and can optionally add the results together, providing a total of 2400 16-bit MMACs or 4800 8-bit MMACs. SIMD instructions are provided to manipulate 16-bit and 8-bit data, including data movement, shifts, and arithmetic instructions. A Galois field multiplier is provided to accelerate Reed–Solomon codecs, bit-wise interleaving and gathering instructions for convolutional encoding, and 8-bit packed absolute difference instructions for motion estimation. Other specialized instructions include bit-reversal, bit counting, max/min selection, bit rotation, and endian-swapping instructions. The ISA provides a 2.0–4.7× cycle performance improvement over the previous generation ISA on Reed–Solomon decoding, a 7.1× improvement for MPEG motion compensation, and a 7.6× improvement in MPEG motion estimation.

B. Cache Architecture

The chip has a two-level memory system architecture [4] as shown in Fig. 4. The level-1 memories consist of a 16-KB direct mapped I-cache and a 16-KB two-way set associative D-cache. The level-2 memory consists of 1 MB of SRAM, which can be partially configured as a four-way set associative cache. The I-cache provides up to eight VLIW instructions to the CPU every cycle. The I-cache partially decodes the incoming instructions before storage into the data arrays to improve performance and reduce power in the CPU pipeline. The instructions are fetched from the I-cache in two separate CPU pipeline stages to allow optimization of the instruction RAM data bank accesses. The two-way set associative D-cache is capable of handling two 64-bit aligned load or store operations in parallel and one 64-bit nonaligned operation per cycle. Nonaligned accesses do not incur a stall penalty even when they cross a cache-line boundary. Store misses to consecutive addresses are combined

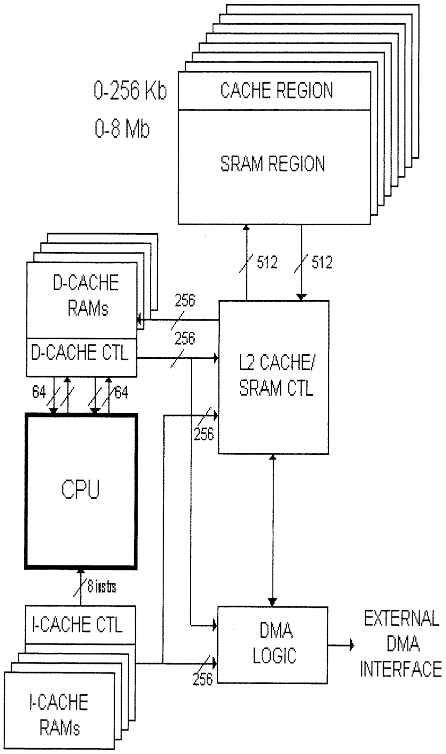


Fig. 4. Two-level memory system architecture.

before being dispatched to the L2 memory and may be buffered up to four 64-bit wide stores deep. The D-cache can pipeline up to four read misses to L2 memory in order to hide latency. The 1 MB of L2 memory may be mapped as all SRAM or as a mix of cache (up to 256 KB) and SRAM.

The level-2 memory interfaces to the DMA controller for cache-related data accesses and other DMA transfers. It also maintains data coherency between external memory and the level-1 caches. The level-2 memory can handle different memory sizes and varying latencies to access on-chip memories. L2 enables the CPU to initiate DMA transfers quickly. Additionally, the L2 also maintains configurable registers, which can be used for a number of operations such as flushing the caches in different modes.

Cache accesses are performed in three phases, with one for each of tag lookup, data access, and data alignment. Due to area, design-for-test (DFT), and electrical issues, single read/write port RAMs are preferred over dual read port RAMS. D-cache tags are duplicated to allow accesses from both datapath cores. Allowing level-2 memory access to D-cache tags for data coherence causes unnecessary stalls and power dissipation. This problem is limited by duplicating D-cache tags in level-2 memory. Data snoops to D-cache tags are performed only if the L2 snoop tag confirms the presence of a line in D-cache. The L2 snoop tag RAM is incorporated into the L2 DMA pipeline. This allows better pipelining of DMA accesses to L2 memory and keeps the DMA performance high.

Additional performance optimizations have been done to ensure that multiple DMA accesses to the same line do not incur the penalty of snooping data out of the D-cache. Coherency is maintained by invalidating lines from the D- and I-caches when written to by DMA accesses.

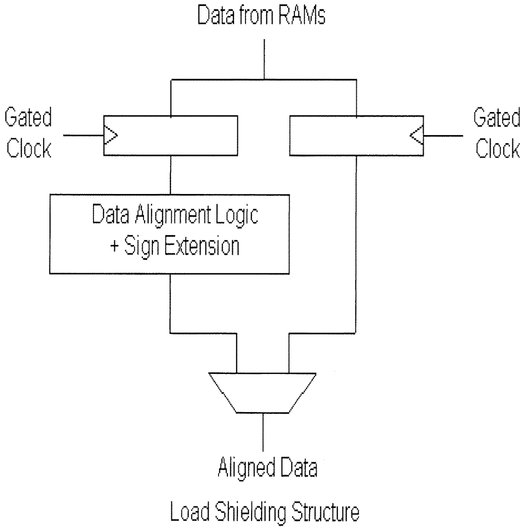


Fig. 5. Load-shielding structure.

Write-through RAMs are used to provide performance speed-up and reduce power. Typically, data is bypassed around the RAM for data written into the RAM that is needed at the output at the same time. Post-RAM muxing is reduced by using RAMs that internally pass write data to the output based on the write-through control. In addition, the RAM address, data, and control bits are kept quiet when the RAM is not used. The address bits are organized such that the lower bits have the highest activity. Data alignment logic used to rotate bytes is required for a number of types of accesses that occur less frequently but consume a lot more power. This logic (used in the data cache) is load-shielded as shown in Fig. 5. Bypassing the data alignment logic for word and double word accesses, which do not need alignment, also saves power.

Level-2 RAMs are typically large, requiring a lot of routing to reach them. These routes consume a considerable portion of the RAM's access power. Clock-gating the registers that drive them prevents unnecessary activity on these wires. They are only activated for valid accesses. Avoiding broadcast of addresses to all banks when only a few banks are accessed reduces power consumption.

C. *Enhanced DMA Architecture*

The 64-event, 4-channel EDMA controller [5] manages all on-chip data transfers (Fig. 6). This includes traffic between the 1 MB of L2 memory and the peripherals and between the different peripherals. The EDMA can transfer 64 bit of data every cycle at 300 MHz, for a maximum sustained bandwidth of 2.4 GB/s. Data transfers may be initiated through single-bit events (e.g., timer events and serial port events) through CPU-initiated transfer requests (e.g., L2 cache misses/evictions that result from normal load/store or program fetch operation), or through a mastering peripheral (e.g., PCI).

The data movement engine is called the transfer crossbar (TC). The TC is a time-multiplexed engine that operates on transfer requests that are queued into four different priority channels. Transfers within the same channel are processed in order. However, transfers in different channels can be

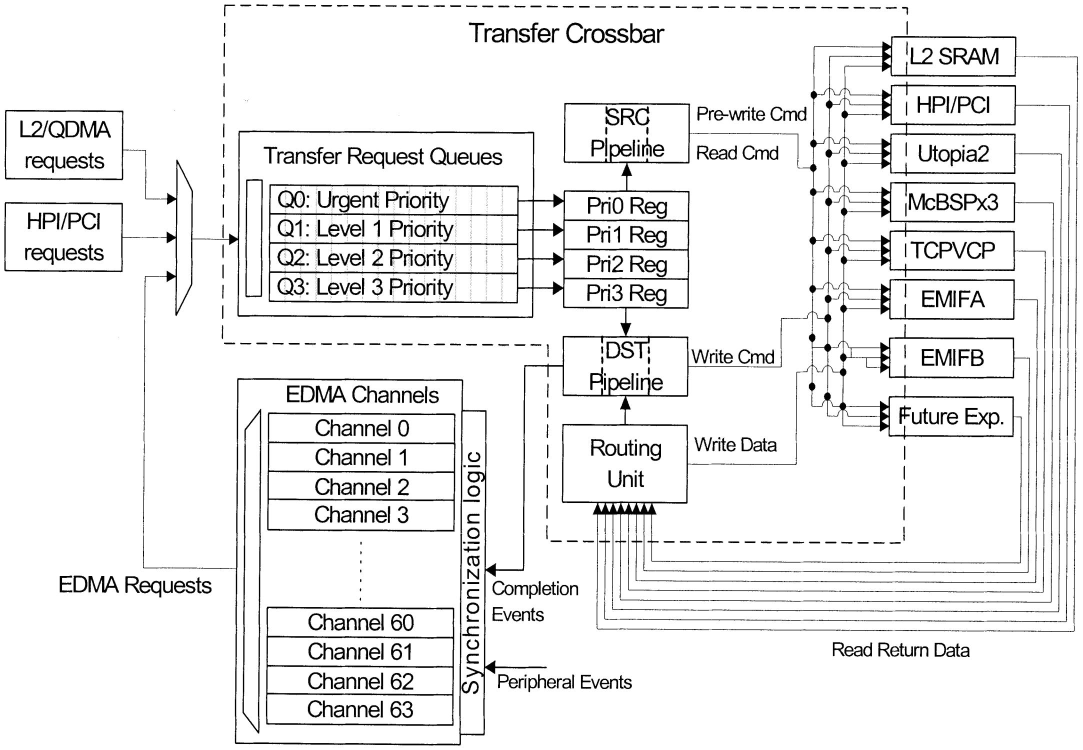


Fig. 6. EDMA architecture.

executed concurrently. The TC can interleave transfers on a cycle-by-cycle basis based on priority, resulting in efficient usage of the 64-bit bus. Head-of-line blocking is prevented by the TC due to its knowledge of the buffer status of each port in the system. The TC uses this knowledge to guarantee that every read request issued by the source pipeline is guaranteed to have a place to land in the destination port, which is serviced by the destination pipeline.

The EDMA channel module allows flexible transfer definitions suited to the needs of the wide range of peripherals and CPU transfer requests. The parameter RAM is capable of defining up to 85 unique parameter entries. Of these parameter entries, 64 can be active at any point in time and can be triggered by peripheral events or by CPU initiation. The remaining parameter sets can be dynamically reloaded onto one of the main 64 transfer sets upon conclusion of a transfer. Upon receipt of a peripheral event, the EDMA looks up the corresponding transfer definition in the transfer parameter table to obtain the source and destination addresses as well as word count and transfer modes for event-based transfer.

D. Datapath

The C64x has two identical datapaths with four functional units (M, L, S, and D) and one 32×32 bit register file per datapath. Each register file has six write ports and ten read ports, which enable parallel reads/writes of source/result operands from all functional units (Fig. 7). A cross-path bus enables each datapath to use the operands stored in the other datapath's register file. The datapaths support approximately 130 instructions and are capable of executing eight instructions per cycle. The functional units can execute several operations in parallel such as six 32-bit additions/subtractions, four 16-bit multiplies, eight 8-bit multiplies, two 8-bit Galois field multiplies, two dot products, Boolean operations, shifts, and compares. The addressing units (D1, D2) calculate memory locations for load/store operations.

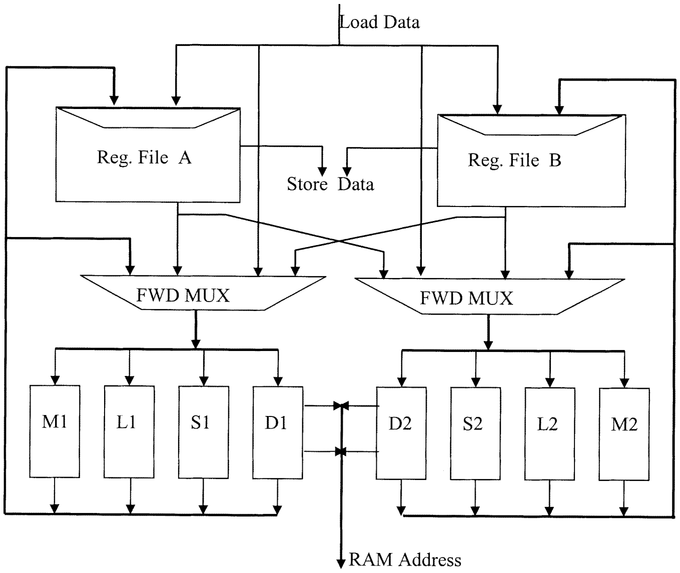


Fig. 7. C64x datapath.

To improve performance in the datapath for critical speed paths, clocks are skewed to borrow time from noncritical paths and latches are used to take advantage of time borrowing. A custom library of standard gates used in the design includes adder cells, flops/latches with integrated muxes, hot muxes with unbuffered transmission gate inputs, and register file cells. Most datapath cells are placed in rows with optimal routing already in position and are connected by abutment. Detailed floor planning is combined with special routing rules to further reduce wire length and improve performance. Critical buses and signals are routed with double pitch and wide metal to reduce RC delays and minimize capacitance. Critical signals are hand placed and routed. The register file is divided into two sections (odd/even registers) to enable 40-/64-bit reads and writes to registers without requiring extra read/write ports. A 32-bit result can be written to register0 while simultaneously writing an 8- or 32-bit result to register1. The data is multiplexed directly from the functional unit and the register file is bypassed whenever a register is written then read during the same cycle. This data forwarding yields a 10% reduction in cycle time.

The clock generation and distribution focused on minimizing power, as low power consumption is a critical design goal. Gated clocks are used extensively to reduce clock power. The flops and latches are placed in rows and the clocks routed by abutment to minimize clock routing. A header containing clock buffers and gated clock cells is placed on the end of each row of flops and latches. The clock trunks are prerouted to minimize interconnect. Latches are placed on the boundary of modules/submodules and gated off to prevent unnecessary data switching to submodules. As a result, signals inside the adders do not toggle when doing unrelated operations such as logical or shift instructions. These latches are integrated with the source multiplexers to reduce delay in critical paths. Application code is analyzed to determine instructions usage and this data is used to optimize the sub module partitions. Thus, highly used subfunctions are granted their own data latches while less frequently used sub functions shared data latches. Output busses driving large capacitances are gated to prevent unnecessary toggling. Flops are placed before the result busses to isolate these busses from decoding glitches. Gates driving signals with large slews are upsized to prevent crowbar current. Power analysis tools are used early in the design to identify areas of high power and detect extraneous switching on unselected mux inputs and within disabled logic blocks. Balancing logic delays to the inputs of a gate also reduces unnecessary glitching.

III. Coprocessors

A. Viterbi Decoder Coprocessor

A Viterbi decoder is typically used to decode the convolutional codes used in 2G and 3G wireless applications [3]. This algorithm comprises two steps: 1) computing state or path metrics forward through the code's trellis and 2) using the stored results from step 1, traversing backward through this data to construct the most likely codeword transmitted (known as traceback). State metric calculation is much more computationally intensive than traceback and consists mainly of add, compare and select (ACS) operations. An ACS operation determines the next value of each state metric in the trellis and does this by selecting the larger of two candidate metrics, one from each branch metric (BM) entering the state. The candidate metrics come from adding the respective branch metric to the respective previous state metric. The branch metrics are derived from the received data to be decoded. In addition, the ACS operation stores the branch that was chosen for use in the traceback process.

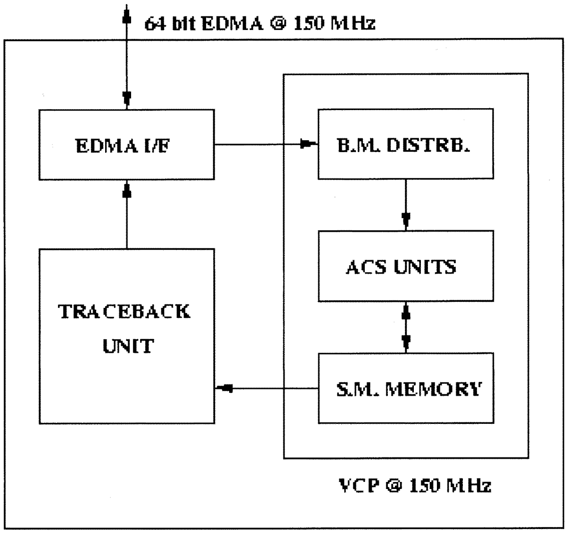


Fig. 8. VCP architecture.

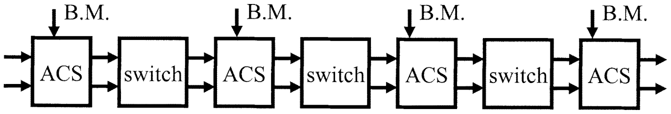


Fig. 9. ACS datapath.

The top-level architecture of the Viterbi coprocessor (VCP) is shown in Fig. 8 and consists of three major units: a state metric unit, a traceback unit, and a DSP interface unit. When operating at 150 MHz (300 MHz for its state metric memory), the state metric unit can perform 600×10^6 ACS butterfly operations per second and the VCP can decode at a rate of 4.7 Mb/s. This is equivalent to well over 350 voice channels for 3G systems.

The cascade structure in Fig. 9 is used to reduce the metric memory bandwidth to a reasonable level. This structure actually operates on a radix 16 subtrellis (16 states over four stages) and thus skips memory I/O for three of the four trellis stages resulting in a 75% reduction in memory bandwidth. This datapath also incorporates a unique register exchange over four trellis path bits (called the retraceback). The 4-bit segments will need no further traceback later; they can be used as integral items during the traceback process, increasing its speed. This cascade can also be operated at reduced lengths. The corresponding incorporated register exchange likewise then produces pretraceback results of the same length in bits.

The traceback unit operates in the traditional manner for backward traversing. This involves the repeated cycle of reading traceback memory to obtain the required word segment reflecting prior path decisions, shifting this word segment into the state index register to form the next state index needed for traceback and using this data to form the next memory address.

Flexibility is a key goal in the design of the VCP. The VCP can operate on single shift register convolutional codes with constraint lengths of $K = 5$–9 and code rates of 1/2, 1/3, and 1/4. The defining polynomials for the desired code are taken as input versus hardwiring only a select few. The VCP allows any puncturing pattern and has parameterized methods for par-

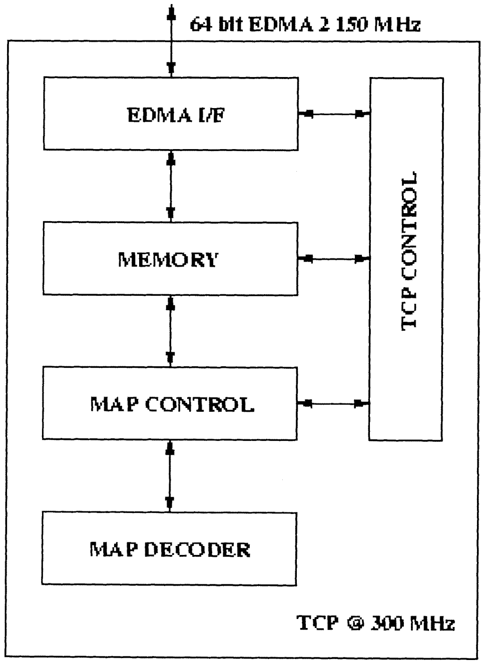


Fig. 10. TCP architecture.

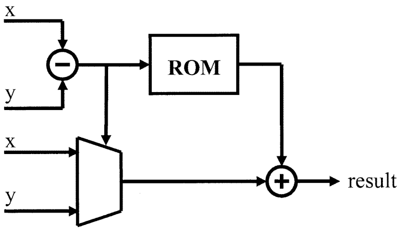


Fig. 11. Max* circuit for TCP.

titioning frames for traceback so that frame size essentially does not matter. The convergence distance can also be specified for partitioned frames. Thus, the VCP implementation can decode virtually any desired convolutional code found in the 2G, 2.5G, and 3G wireless standards.

B. Turbo Decoder Coprocessor

A turbo decoder is typically used to decode wireless data channels. The turbo decoder coprocessor (TCP) is an iterative decoder that uses the maximum *a posteriori* (MAP) algorithm [3]. Each iteration of the decoder executes the MAP decoder twice producing two sets of intermediate values called extrinsics. The first MAP decoder uses the noninterleaved data and the second MAP decoder uses the interleaved data. In each iteration, each MAP decoder feeds the other MAP decoder a new set of *a priori* estimates of the information bits. In this way, the MAP decoder pair can converge to a solution.

The data received from the channel is scaled by the signal noise variance prior to use by the MAP decoders. This scaling is performed by the DSP and sent to the TCP via the DMA at 150 MHz. The basic TCP architecture is shown in Fig. 10. The basic operating block within the TCP is the $\max^*$ circuit which uses a 2-bit lookup table and is shown in Fig. 11. When operating at 300 Mhz, the MAP decoder can process 9×10^9 $\max^*$ operations per second and the TCP can decode 28 384-Kb/s data channels at a rate of 11.3 Mb/s for eight iterations.

Flexible control allows the DSP to be configured to work in several modes. In the conceptually simplest mode, the DSP loads an entire block to the TCP, it performs a single MAP decode on the data, and the results are sent back to the DSP. In this case, the DSP will interleave the data between MAP decodes and is therefore involved in every iteration of the turbo decode. The data transfers are controlled by the EDMA. This particular operational mode allows the TCP to operate on a large variety of codes other than those specified in 3G, provided they use the same recursive systematic convolutional codes.

The TCP can also be set up to perform several iterations without DSP intervention. This greatly decreases the required bus bandwidth since the extrinsic results are not being passed back and forth. In this mode, the TCP uses a look-up table to perform interleaving and can therefore perform as many iterations as required to converge. The TCP controller is in charge of writing the correct systematic, parity, and *a priori* data to the MAP decoder. After a successful decode the DSP retrieves the corrected data via the EDMA.

Most of the time only 3–4 iterations are required for convergence, even though a maximum of 8–12 iterations are required to obtain the best turbo decoder performance. Using a stopping criterion that is a function of the MAP decoder outputs to decide when convergence has occurred often minimizes power consumption. Therefore, a stopping criterion can have a significant impact on the average MIP's requirements and hence the power level.

IV. Design Methodology

A. Design Flow and Library

The design methodology (Fig. 12) used in creating the processor has a number of unique features but builds upon basic, well-proven techniques. The circuit design style generally follows a standard-cell static-CMOS logic synthesis environment. This is extended with support for pass-transistor and hot-encoded muxes for timing and power optimizations. Clocks are created via clock-tree synthesis and optimized for power, but some extensions to handle a wide range of process variation and improvements for design robustness have been added. This is all wrapped around a static timing signoff flow.

The vast majority of logic created outside of the pseudo-custom tiled datapath is built using a traditional static-CMOS style. For speed and power reasons, large sets of cells are used which exposes unshielded pass-transistors to the synthesis environment, such as in the 2:1 inverting mux shown in Fig. 13. This requires some enhancement to off-the-shelf tools to handle both input capacitance characterization and timing characterization of propagation delays that depend on the drive strength of external gates. First, characterization for input capacitance is done for cases when the pass-gate is “on” and when the pass-gate is “off.” Input transition times covering the full operating range for the gate are applied, input capacitance is measured, and the minimum and maximum are extracted. Second, the delay from “select” signals (e.g., input ‘sa’ in Fig. 13) depends upon the driver on the related pass-gate inputs, which must discharge the

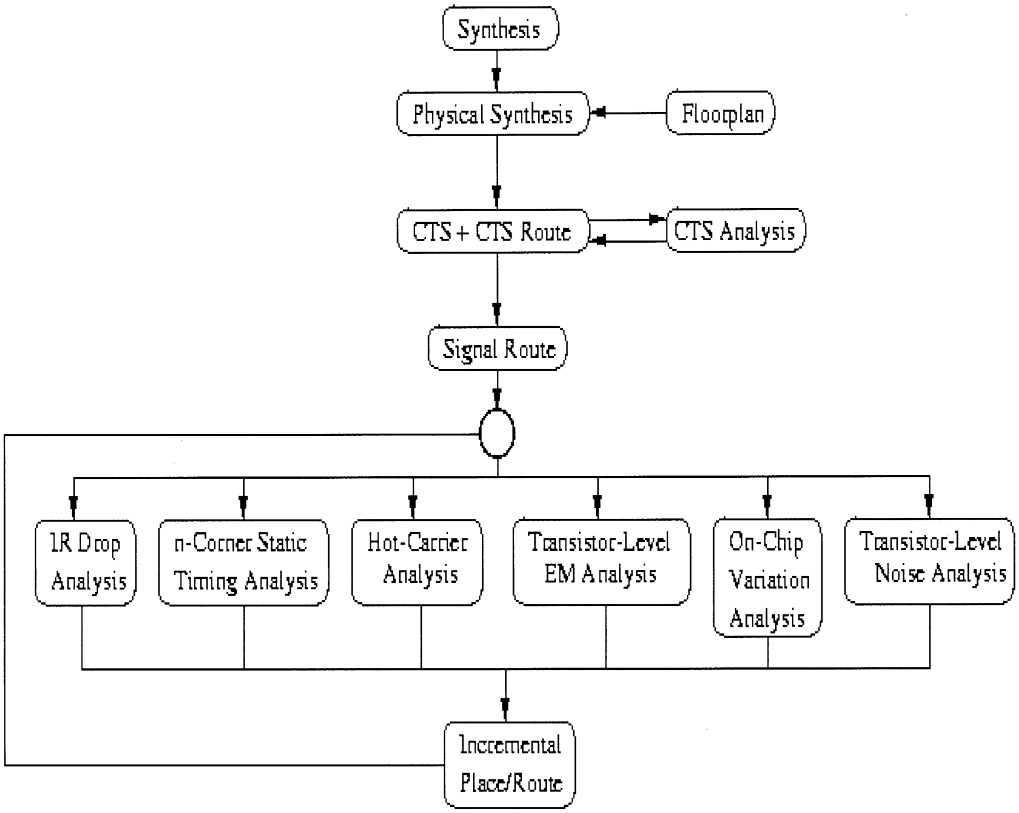


Fig. 12. Design and analysis flow.

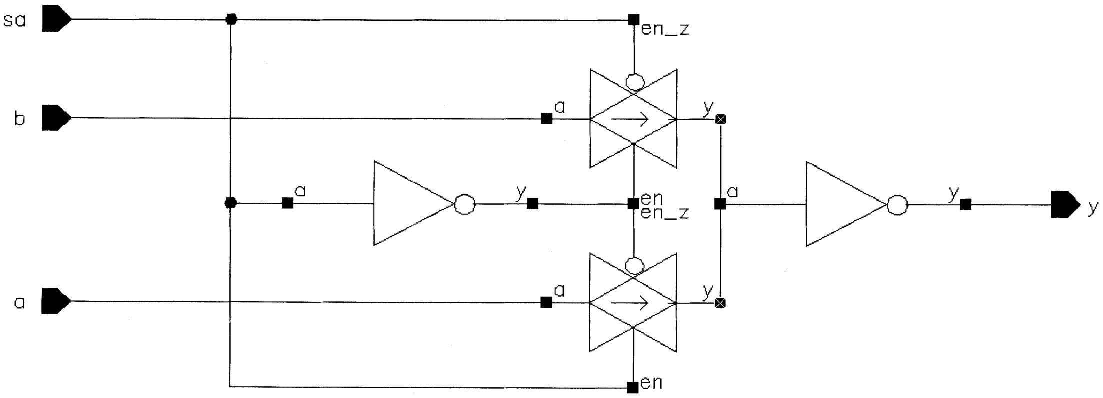


Fig. 13. A 2:1 inverting mux.

internal node of the pass-gate structure. Timing characterization for the "select" signal switching must be performed with the appropriate drivers attached to these related inputs. For each input driving a transistor source/drain terminal, a target max-transition time is selected. Based on the measured max input capacitance and this target time, a driving cell is selected. This driving cell is then attached to the input node and timing characterization for the related "select" signals is performed. This maximum transition time is annotated for synthesis as a design rule.

A second family of gates that is novel in a standard-cell design is the hot-encoded mux. An example hot-encoded mux is shown in Fig. 14. The assumption is that one and only one of the "select" inputs is on at any given time (i.e., the selects are 1-hot). These cells are used extensively for timing and power optimization throughout the design. The synthesis environment does not automatically infer these gates, they are specified in RTL, but the tool is capable of resizing these gates automatically based on usage conditions.

B. Clock Distribution

The on-chip phase-locked loop (PLL) generates 1x, 1/2x, and 1/4x clocks that are synchronously distributed to all flip-flops. The asynchronous peripheral logic interfaces with the internal clock through a rotating token synchronization scheme. The clock is physically implemented in three tiers with less than 50 ps of intrablock skew, 100 ps of interblock skew within a clock domain, and 150 ps overall. Clock gating is performed at the leaf level using clockgates, which cluster the leaf level loads around the clockgate to reduce skew and power. Efficient power management is achieved with almost 75% of the registers clock gated. A low-jitter point-to-point balanced tree structure is implemented with 12 levels of tunable clock buffers. Inductive re-

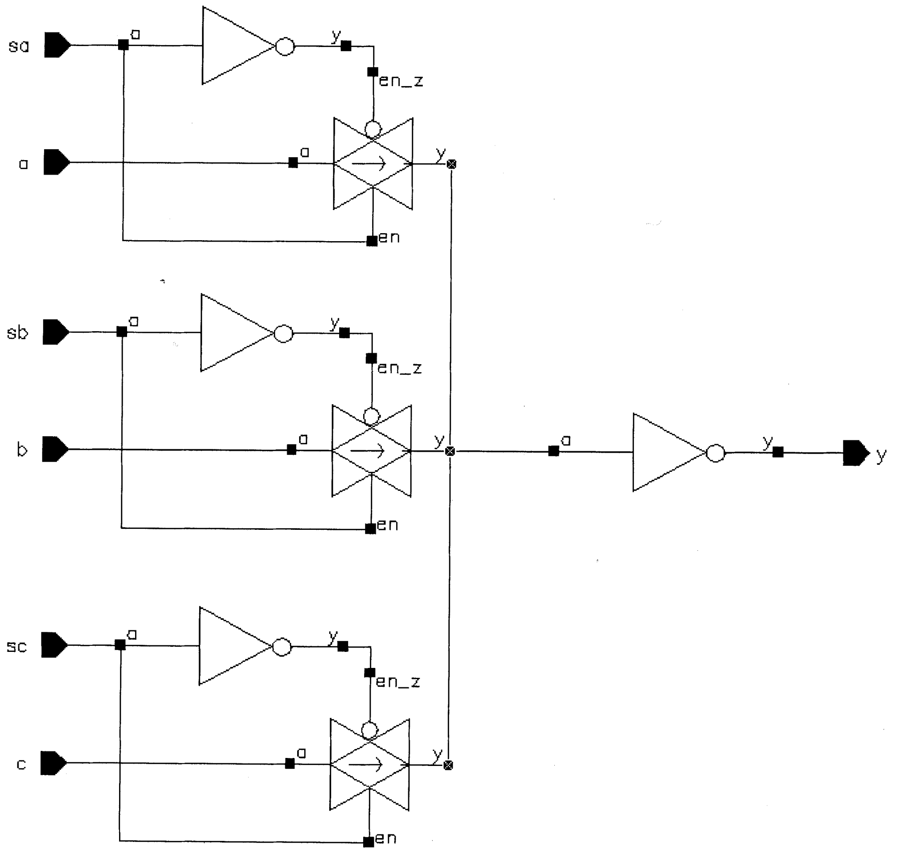


Fig. 14. Hot encoded mux.

turn paths are provided through clock shields that are tied to the power grid every 1200 μm. Restricting long routes and using a fine power grid minimizes inductance. Metal spacing guidelines keep parasitic capacitance to a minimum, avoiding noise problems. The gate delay and RC delay is balanced to enable the clock skew to track across process corners. The clock tree is analyzed and balanced using eight different combinations of interconnect and process corners. The combination of different corners are used to identify mismatch in interconnect, mismatch in gate delay and structure of the clock tree, check for voltage induced changes, and measure clock power dissipation.

C. Timing Analysis

One reality of deep-submicrometer design is that significant on-die process variation occurs. This variation impacts maximum clock frequency and can contribute to significant hold-time problems and yield loss. Additional top-level design margin is used to account for setup variation, but, for hold-time variation, on-chip variation analysis must be considered. In this analysis, the clock-tree delay to the driving register is derated by some percentage and the clock tree delay to the receiving register is uprated by another percentage. This simplistic model does not adequately account for the averaging effect, which occurs when the common point in the clock tree is a long way removed from the registers being analyzed. However, it is very simple to apply and did uncover potential problems on the processor.

TABLE I
POWER CONSUMPTION WITH TYPICAL ACTIVITY

CPU and L1I cache	L2 cache, DMA and Peripherals	IO	Total
0.310W	0.408W	0.384W	1.1W

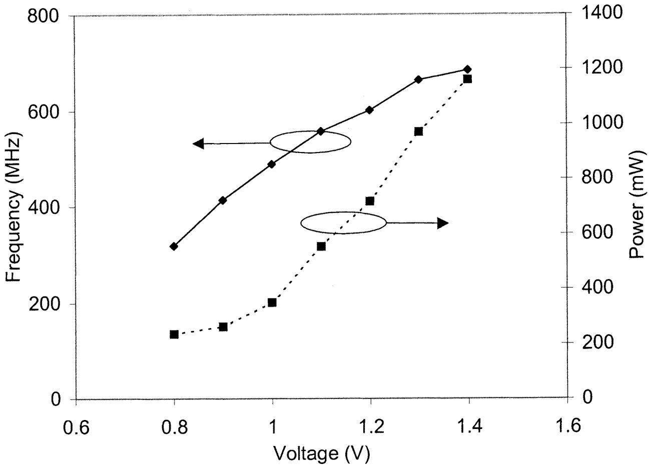


Fig. 15. Measured chip power and frequency.

Another reality in deep-submicrometer design is that die-to-die process variation is becoming more significant. For example, the interconnect parasitics could lie in a "max-capacitance" (maxc) corner or could be in a "max-resistance" (maxr) corner depending on the thickness of metal deposited or the width of the metal trenches. Similar to the analysis commonly done for transistor variation (e.g., weak, nominal, or strong), we also consider the variation of the interconnect. This

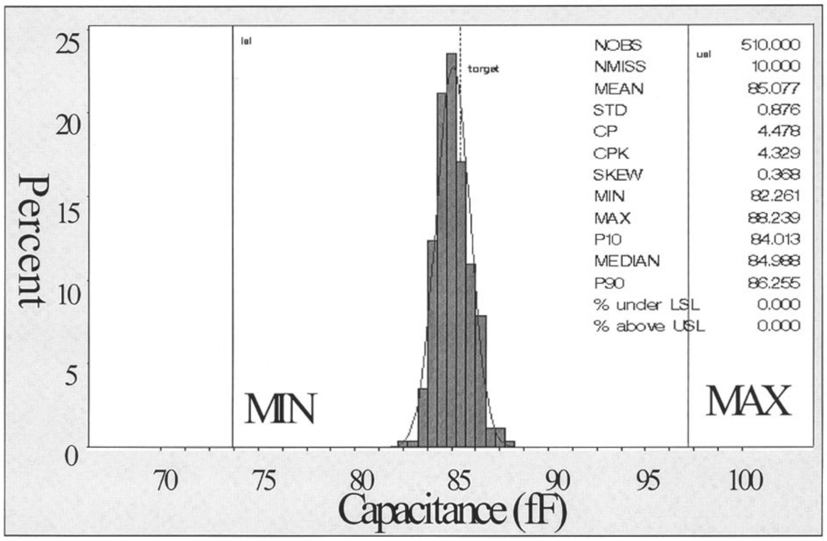


Fig. 16. Interconnect capacitance distribution for the MET1 process monitor structure.

requires that we analyze the design at a number of different process corners (e.g., strong_minc, strong_minr, weak_maxc, and weak_maxr). Each of these corners tends to show some unique design marginality issues. For example, long nets tend to violate timing as resistance becomes more dominant and localized high-fanout nets tend to limit performance in capacitance-dominant corners.

D. Power Analysis

Low power consumption is a critical design goal for the chip. A transistor-level simulation engine is used to accurately measure presilicon power dissipation and monitor the power budget of each functional unit during the entire design. The C64x power consumption is modeled as a combination of two levels: high DSP activity and low DSP activity. High DSP activity occurs during execution of the finite impulse response (FIR), the fast Fourier transform (FFT), or other optimized algorithms. Low DSP activity occurs when setting up registers or other less optimized code segments. A typical real-world application has a mix of 50% high and 50% low activity. Table I shows the power consumption measured on the C6416 silicon at 25 C, 600 MHz. The core and I/O supply voltages are 1.2 and 3.3 V, respectively. The typical I/O load is 20 pF. The DSP core dissipates 718 mW at 600 MHz, 1.2 V and 200 mW at 300 MHz, 0.9 V as shown in Fig. 15.

E. Signal Integrity

The signal integrity and reliability methodology covers a detailed analysis of crosstalk delay, noise, electromigration (EM) [5] on signal and power lines, gate oxide integrity (GOI) [6], negative bias temperature instability (NBTI) [6], static/dynamic IR-drop, and inductance effects. The typical analysis flow for each block after detail routing is shown in Fig. 12.

Special structures are implemented to measure and monitor interconnect parasitics, build interconnect models that accurately represent a copper metal system, and study wafer and lot level variation data and validation of parasitic extraction tools. The distribution of capacitance at 510 sites measured on 10 wafers (number of observations (NOBS) $=$ 510) on a process monitor structure for MET1 is shown in Fig. 16. Statistical metrics of the distribution are also shown in Fig. 16. The distribution of parasitic capacitance for the wafers shows the capacitance is tightly centered on the target specified for the process.

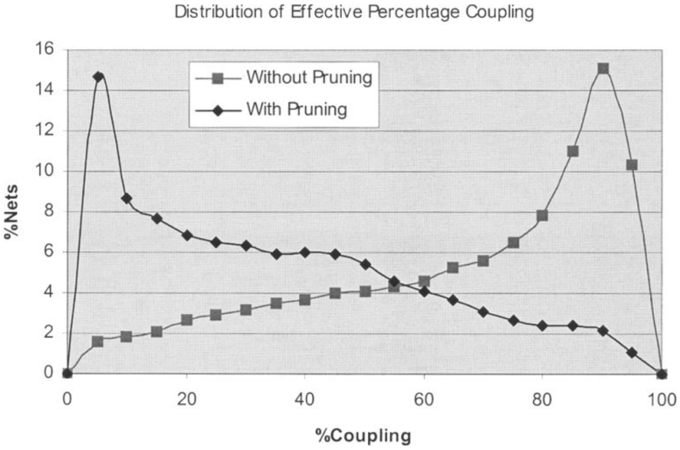


Fig. 17. Distribution of parasitic coupling before and after temporal pruning using timing windows.

Parasitic extraction includes effects of scattering that increases resistance for narrow lines in copper technology. The presence of fill metal used to obtain uniform chemical mechanical polishing (CMP) and silicon size adjustments of dense and narrow lines are both modeled in transistor-level parasitic extraction for datapath. Parasitic extraction at the datapath and chip level is done at multiple process corners to ensure robustness of timing/signal integrity and to enhance yield across multiple fabrication plants. The distribution of parasitic coupling before and after temporal pruning using timing windows is shown in Fig. 17. The parasitic coupling from initial parasitic extraction (without pruning) has a significant number of nets with large coupling percentages. Using timing windows for nets generated from a static timing analysis tool

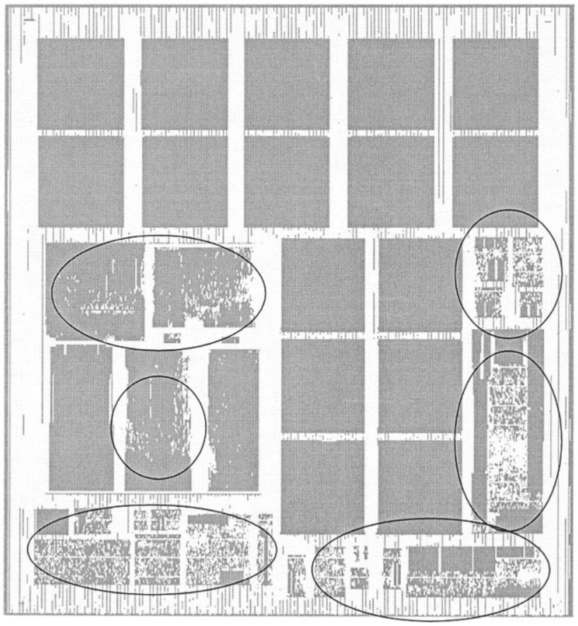


Fig. 18. Full chip analysis of parasitic coupling noise using "see through" coupled extraction to capture coupling across module boundaries.

(temporal pruning) helps reduce pessimism in timing and noise analysis. "See-through" coupled extraction enables coupled RC extraction for all nets that cross hierarchical boundaries and lumped RC extraction for nets internal to hierarchies. "See-through" coupled extraction was used to extract parasitic coupling to nets across module boundaries without a flat full-chip extraction. Traditional black box approaches used for analyzing parasitic coupling noise do not capture coupling across module boundaries. Parasitic coupling for the nets shown in circled regions in Fig. 18 are extracted using the "see-through" coupled extraction.

Inductance extraction is done using the PEEC [7] method on selected structures to study the impact of inductance on clock skew and noise and develop guidelines for shielding. Special methods are developed to analyze parasitic coupling effect on delay. Coupling parasitics are compensated based on early arrival time overlaps with aggressors for hold time verification and late arrival time overlaps with aggressors for setup time. This helps minimize pessimism in crosstalk delay analysis. Though parasitics are extracted hierarchically, coupling compensated parasitics are merged for final timing analysis. This helps to overcome potential timing violations across module hierarchies.

Noise analysis is done at the transistor level as the circuits contain pass gate configurations. Flip-flops, latches, and memory inputs are characterized to determine VL and VH noise margins and are checked for violations. Noise fixing is done for both peak violations and propagated noise to ensure robustness. Each module in the design is analyzed separately for noise and "see-through" coupled extraction is used to verify noise at full-chip level and detect noise events across module hierarchies. Separate checks are made for coupling on clock signals to detect undesired noise events that could affect clock skew.

Electromigration analysis is done on signal lines in two conditions to ensure complete coverage. The first condition covers minimum-metal-thickness metal and the second includes capacitance-dominated maximum thickness. A piecewise linear model is used for metal current density limits and via/contacts are verified separately. Power/ground supply networks are verified for average current densities for minimum metal thickness condition and IR drop. GOI is verified for the complete chip with gate oxide area approximately 0.025 cm^2 in size using a statistical model. Overshoots caused due to parasitic coupling are analyzed separately to calculate effective fail rates as the GOI phenomenon is exponentially accelerated by voltage. The NBTI effect is analyzed on timing and end-of-life spice models are used to ensure timing margin.

V. Results

A VLIW DSP that implements the C64x Architecture runs at 600 MHz with the core and I/O running at 1.2 and 3.3 V, respectively. The core power dissipation is 718 mW and full-chip power dissipation (including I/O) is 1.1 W. The DSP delivers 4800 MIPS, 2400 16-bit MMACs, and 4800 8-bit MMACs at 0.3 mW/MMAC (16 bit). The chip contains over 64 M transistors and is implemented in 0.13-μm CMOS technology with six-layer copper metallization in a flip-chip package. The design has been able to achieve high performance and low power using a semicustom standard cell methodology.

References

[1] S. Agarwala, P. Koeppen, T. Anderson, A. Hill, M. Ales, R. Damodaran, L. Nardini, P. Wiley, S. Mullinnix, J. Leach, A. Lell, M. Gill, J. Golston, D. Hoyle, A. Rajagopal, A. Chachad, M. Agarwala, R. Castille, N. Common, J. Apostol, H. Mahmood, M. Krishnan, D. Bui, Q. An, P. Groves, N. NS, L. Nguyen, and R. Simar, "A 600 MHz VLIW DSP," in *Proc. ISSCC Dig. Tech. Papers*, Feb. 2002, pp. 56–57.

[2] *TMS320C6000 CPU and Instruction Set Reference Guide*, Texas Instruments, 2000.

[3] A. Gatherer and E. Auslander, *The Application of Programmable DSP's in Mobile Communications*. New York: Wiley, 2001, ch. 3.

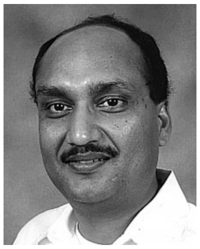
[4] S. Agarwala, C. Fuoco, T. Anderson, D. Comisky, and C. Mobley, "A multi-level memory system architecture for high performance DSP applications," in *Proc. ICCD*, Austin, TX, Oct. 2000.

[5] D. Comisky, S. Agarwala, and C. Fuoco, "A scalable high-performance DMA architecture for DSP applications," in *Proc. ICCD*, Austin, TX, Oct. 2000.

[6] T. Rost and B. Hunter, "Building in reliability," presented at the Proc. CICC Tutorial, San Diego, CA, 1999.

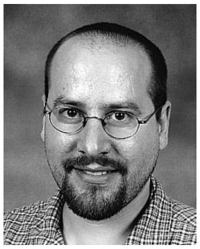
[7] G. La Rosa, S. Rauch, and F. Guarin, "New phenomena in device reliability physics of advanced CMOS submicron technologies," in *Proc. IRPS*, San Jose, CA, 2000.

[8] A. E. Ruehli, "Equivalent circuit models for three-dimensional conductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-22, pp. 216–221, Mar. 1974.



Sanjive Agarwala received the B.S.E.E. degree from Panjab University, India, in 1984 and the M.S.C.S. degree from Southern Methodist University, Dallas, TX, in 1990.

He joined the Texas Instruments (TI) R&D Lab, Dallas, in 1986, working on the development of circuit synthesis and timing analysis tools. Subsequently, he led the memory hierarchy development on two generations of x86 microprocessors at TI. In 1997, he led the design and development of the C62x DSP followed by the development of the high-performance C64x DSP at TI. He is currently a TI Fellow and Design Manager of the C64x design group at TI. He holds six U.S. patents.



Timothy Anderson received the B.S.E.E. degree and the B.A. degree in physics (*magna cum laude*) from Rice University, Houston, TX, in 1994.

He joined Texas Instruments, Dallas, TX in 1994, where he worked on the exception and interrupt logic of an X86 processor. In 1998, he was co-architect of the TMS320C6211 DSP, the first example of a two-level cache system on a mass-market DSP, and was responsible for implementing the super-scalar L1-Data Cache on that device. In 2000, he was co-architect of the TMS320C6416 processor, where he also defined and led the implemention of the instruction supply and other control logic of the CPU. He is a Senior Member of the Technical Staff at the DSP Design Center at Texas Instruments and holds five U.S. patents.

Anthony Hill received the B.S. degree from Oklahoma State University in 1992 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana Champaign in 1993 and 1996, respectively.

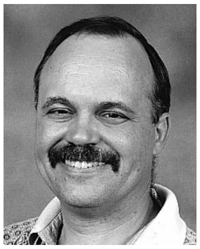
In 1992, he received a Graduate Research Fellowship from the National Sience Foundation. He joined Texas Instruments, Dallas, in 1996 working on low-power processor design. He is currently a Senior Member of Technical Staff for Texas Instruments, working on c64x DSP designs. His research interests are noise and IR-drop analysis in timing signoff and high-speed ASIC design methodologies.

Michael D. Ales received the B.S.E.E. degree from the University of Illinois at Urbana/Champaign in 1981.

He has worked in the DSP Division, Texas Instruments, Dallas, since 1981 where he is active in all areas of design, design for test, manufacturing test and design flows. He is currently a Distinguished Member of Technical Staff.

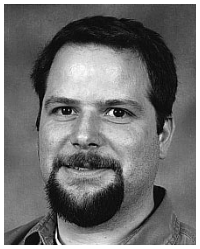
Raguram Damodaran received the B.Engg. degree from Coimbatore Institute of Technology, India, in 1991 and the M.S.E.E. degree from Texas A&M University, College Station, in 1993.

He is a Member of the Group Technical Staff at Texas Instruments (TI), Dallas. He was with the microprocessor design team from 1993 to 1996, where he was involved with functional verification, static timing analysis, and clock implementation. He has been with the DSP Department since 1997, where he was involved with the design of ALU, DMA controller and clock network implementation for the C62X architecture. He currently leads the memory system, clock, and low-power methodology teams for the C64x architecture.



Paul Wiley received the B.S.E.E. degree and the B.A. degree in mathematics from the University of Texas at Austin in 1979, the M.S.E.E. degree and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1982 and 1986, respectively.

He worked on developing and applying CAD tools at MCC and Motorola, with a concentration on timing analysis tools. He joined Texas Instruments, Dallas, in June 1996 to support the design of high-performance circuits. Currently, he is a member of the Dallas DSP Design Team, responsible for the Integration and Methodology team.



Steven Mullinnix received the B.S. degree in computer engineering from Iowa State University, Ames, in 1993 and the M.S.E.E. degree from the University of Texas at Dallas in 1997.

He joined Texas Instruments Defense Systems Group, Dallas, in 1993 and worked for a year as a Test Engineer followed by a year as a Software Engineer. He has since worked in the Semiconductor Group on IC functional design verification of various microprocessors and DSPs. He is currently a Member of Technical Staff working on verification of the TI C64x DSP.

Jerald Leach received the B.S.E.E. degree from Texas A&M University in 1976.

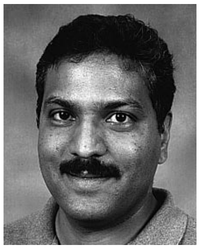
He joined Texas Instruments, Dallas, in 1976 doing calculator design. He was the Lead Designer for C3X and C4X DSPs and Lead Datapath Designer for C62xx, C67xx, C64xx DSPs. He is currently a Distinguished Member of the Technical Staff at Texas Instruments.

Anthony Lell received the B.S.E.E. degree from Texas A&M University, College Station, in 1994.

He worked as a Design Engineer in a variety of areas on the Texas Instruments' C6x DSPs for the past eight years. He is currently Member of Group Technical Staff.

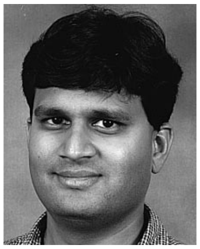
Michael Gill received the B.S. degrees in physics and computer engineering from Iowa State University, Ames, in 1982 and the M.S. degree in electrical engineering from the Southern Methodist University, Dallas, TX, in 1986.

From 1982 to present, he has worked for Texas Instruments, Dallas, as a CMOS Design Engineer, dealing with floating-point architecture and CAD methodology. His current area of interest is deep submicrometer physical design. He is currently a Distinguished Member of Technical Staff.



Arjun Rajagopal received the B.Tech. (Hons.) degree in electrical engineering. from Birla Institute of Technology and Science, Pilani, India, in 1991 and the M.S.E.E. degree from Texas A&M University, College Station, in 1993.

He joined Texas Instruments, Dallas, in 1993, working on functional design verification of x86 microprocessors. He is currently in the Dallas DSP design team and has worked on clock distribution, interconnect modeling, parasitic extraction, and power grid analysis for two generations of DSPs, the C62x and C64x.

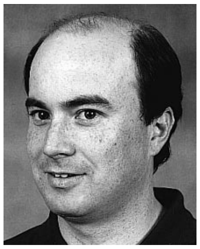


Abhijeet Chachad received the B.S. degree in electronics from VJTI, University of Bombay, India, in 1994 and the M.S.E.E. degree from the University of Texas at Dallas in 1997.

He joined Texas Instruments, Dallas, in 1997 and has been a part of the DSP Design team. He initially worked on the EDMA controller and peripherals for the c62x DSP. Since 2000, he has been working on development of the memory subsystem, specifically, the level-2 memory controller.

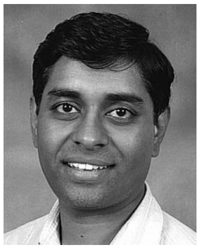
Manisha Agarwala received the M.S.E.E. degree from the University of Texas at Dallas in 1992.

She joined Texas Instruments, Dallas, upon graduation and is currently working on design and development of real-time emulation technology for DSPs. She has been awarded three patents.



John Apostol received the B.S. degree in electrical engineering from Iowa State University, Ames and the M.S. degree in physics from the University of Illinois at Urban-Champaign.

He joined Texas Instruments, Dallas, in 1992 and has worked on projects ranging from defense systems, X86 class processor, and, recently, high-speed DSPs including the TMS320C6416. His current responsibilities include design flows/methodology, standard cell modeling, and chip integration. He is a Member of the Group Technical Staff.

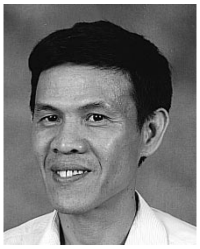


Manjeri Krishnan received the M.S.E.E. degree from the University of Texas at Dallas in 1994.

He joined Texas Instruments (TI), Dallas, upon graduation. At TI, he has worked on several projects involving high-performance microprocessor design, in such areas as functional verification, microarchitecture, testability, and static timing analysis. He is currently working on TI's C64x high-performance family of DSPs.

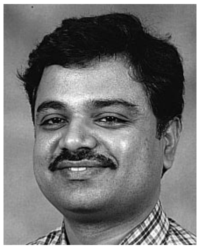
Duc Bui received the B.S.E.E. degree from the University of Illinois at Urbana-Champaign in 1992 and the M.S.E.E. degree from the University of Texas at Dallas in 1997.

He worked at IBM for three years as a Floating Point Unit designer for a PowerPC microprocessor before joining Texas Instruments (TI), Dallas, in 1995. At TI, his responsibilities have included designing CPU datapath and control logic and power analysis for various DSPs. He is the holder of two U.S. patents.

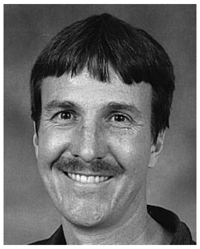


Quang An received the M.S.E.E degree from National Taiwan University, Taiwan.

He joined Texas Instruments, Dallas, in 1986. He has designed microsequencer, fifos, cache system, DLP micro-mirror device and ALU and load store unit for a microprocessor. He is currently working on EDMA system for C64x DSP devices.



N. S. Nagaraj is the Director of Interconnect Modeling and Analysis Group at Texas Instruments (TI), Dallas, TX. He has been with TI since 1990 and developed parasitic extraction and signal integrity solutions for leading DSP and Wireless designs. He has published 25 papers, chaired three panel discussions, and given four tutorials in international conferences. His research interests include interconnect modeling, silicon correlation, signal integrity, and reliability verification for high-speed designs.



Tod Wolf received the B.S.E.E. and M.S.E.E. degrees from Wichita State University and the University of Alabama, Huntsville, in 1985 and 1987, respectively.

In 1988, he joined Texas Instruments, Dallas, as an Integrated Circuit and System Designer. He has worked on various designs such as cache tags, floating point units, JPEG, audio MPEG, cable modem, Reed–Solomon coders, and turbo decoders. Presently, he is involved in research of turbo decoders for wireless applications. He holds eight patents in the area of digital communications.

Tony T. Elappuparackal received the B.Tech. degree in electrical engineering from the University of Calicut, India, in 1994.

He worked for the R&D group Wipro Infotech during 1994–1997 on design and development of high-performance remote-access servers and RAID storage systems. He also worked for NCR Corporation on their chip-set design for Pentium-Pro-based multiprocessor high-end servers. He joined Texas Instruments, Dallas, in 1998 and has worked on the design and development of the high-performance C62X and C64x series of DSPs. His current responsibilities include design of the multiplier unit in the c64x datapath, verification, timing analysis, and parasitic analysis.