

TMS320C64x™ Architecture Extensions Boost Performance for Broadband Communications and Imaging

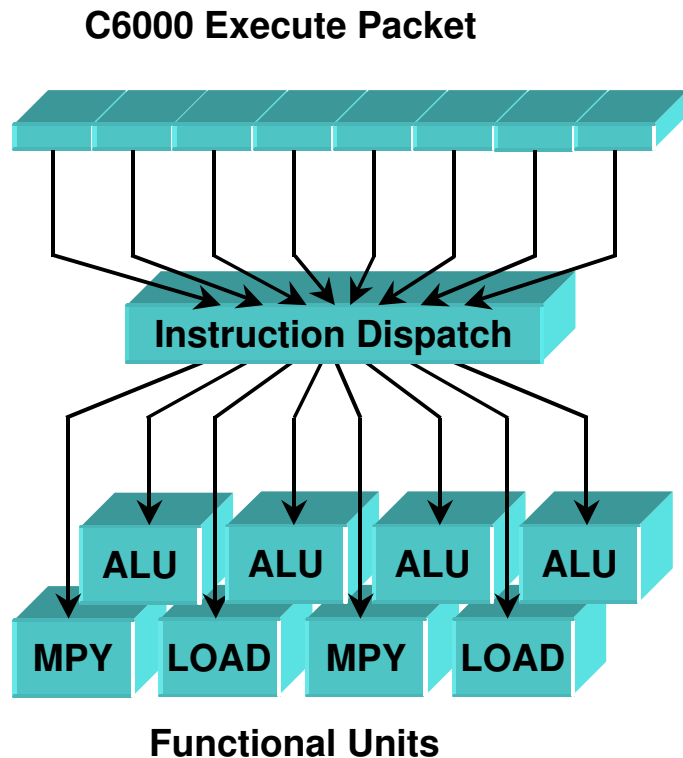
Jeremiah Golston

TMS320C64x™ DSP Applications Manager

Overview

- **C6000 VelociTI™ Advanced VLIW Architecture**
- **C64x VelociTI.2™ Architecture Extensions**
- **Algorithm Examples and Benchmarks**
- **Memory and Peripheral System**
- **Code Compatible Roadmap**

C6000 VelociTI™ Advanced VLIW Architecture



Speed Optimized Pipeline

- 6-stage instruction fetch/dispatch pipeline
- 5-stage load pipeline
- 2-stage multiply pipeline
- Allows highest DSP clock rate

Parallelism

- 8 new independent instructions can always be dispatched every cycle

Excellent Compiler Target

- Deterministic order and time of execution
- Single general purpose register file
- Simple independent instructions
- No special modes or status bits

C64x VelociTI.2™ Architecture Extension Requirements

100% Binary Compatibility with C62x

- No pipeline changes for existing instructions
- No modifications to existing opcodes

Maintain High Performance Clock Rate

- No new critical speed paths
- Minimize additional register porting

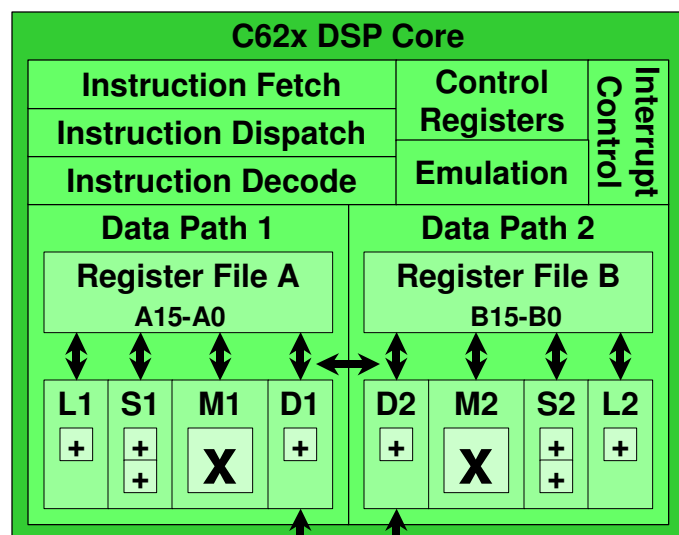
Retain compiler-friendly architecture

- Co-developed with the compiler team
- Seamlessly integrated into instruction set
- Operates on general purpose register file

Significant measurable architectural speed-up

- Minimum 2x cycle speed-up across

C64x DSP Core Enhancements



Current Performance

8 Instructions/Cycle
200-300 MHz
1600-2400 MIPs
400-600 MMACs

Extends Clock Rate

- Initial Devices 600-800 MHz

Increased Parallelism

- Packed Data Processing
 - Two 16-bit Multipliers per M Unit
 - Quad 8-Bit Arithmetic
 - Dual 16-bit Arithmetic

2x Register File

- 64 vs. 32 Registers

2x Data Bandwidth

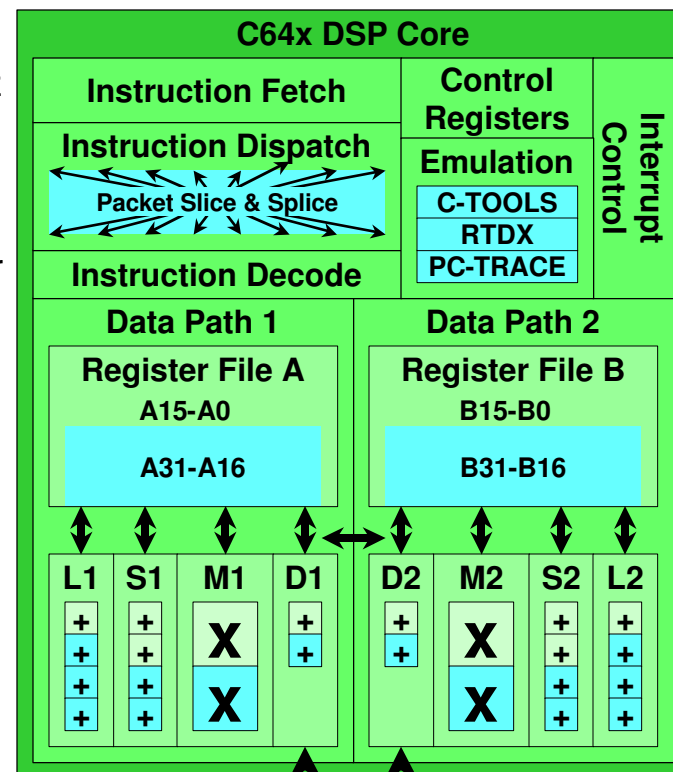
- 64-Bit Load/Store Paths

Codesize Reduction

- Execute Packets Can Span Fetch Packets

Advanced Emulation

- C-Tools Analysis
- Real Time Data Exchange
- Run-time PC Trace



Initial Performance

8 Instructions/Cycle
600-800 MHz
4800-6400 MIPs
2500-3200 16-Bit MMACs
4800-6400 8-Bit MMACs

C64x VelociTI.2™ Packed Data Processing Extensions

Instruction	Quad 8-Bit	Dual 16-Bit	Function
ADDx, SUBx	X	X	Adds/Subtracts
SADDx	X	X	Saturated Adds
MPYx, MPYx	X	X	Multiplies
DOTPx	X	X	Dot Products, $(a+bj)*(c+dj)$
DOTPxRx		X	Dot Products w/ Rounding
PACKx	X	X	Pack Operations
SPACKx	X	X	Saturated Pack Operations
UNPKx4	X	X	Unpack Operations
CMPx	X	X	Compares
MAXx/MINx	X	X	Max/Min Operations
SHRx2		X	Shifts
ABS2		X	Absolute Value
LDNx/STNx	X	X	Non-aligned Load/Stores

C64x Non-Aligned Memory Accesses

One 64-bit Non-Aligned Load or Store Per Cycle

- Uses load/store paths for both D units
- Accesses correct 64-bit from 128-bit access
- Remaining D unit still available for non-memory instructions

Available with Non-Scaled or Scaled Offset

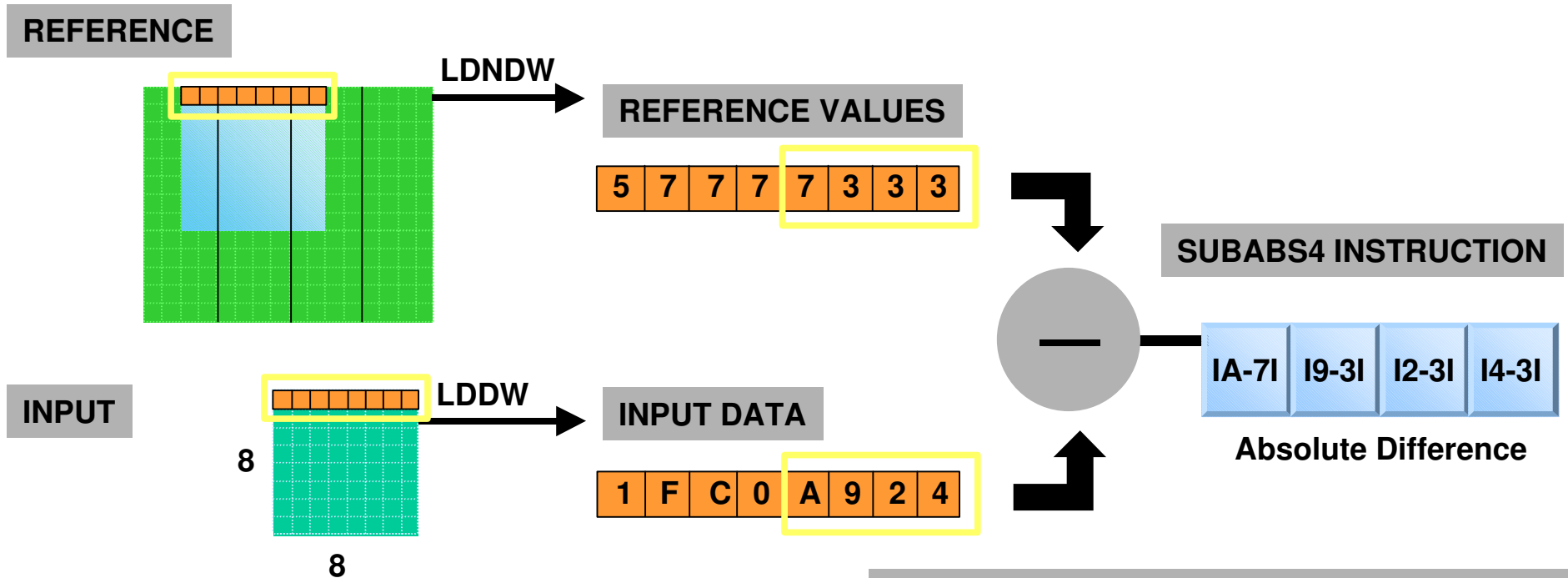
- Non-scaled offset provides access granularity for convolution-type operations
- Scaled offset by data size in bytes for compiler use in loop unrolling

Key to Compiler Optimizations for Packed Data Processing

C64x VelociTI.2™ Special Purpose Instructions

Instruction	Description	Typical Application
BITC4	Quad 8-Bit Bit Count	Machine Vision
GMPY4	Galois Field MPY	Reed Solomon Support
SHFL	Bit Interleaving	Convolutional Encoder
DEAL	Bit De-interleaving	Cable Modem
SWAP4	Byte Swap	Endian Swap
XPNDx	Bit Expansion	Graphics
MPYHIx,MPYLIx	Extended Precision 16x32 MPYs	Audio
AVGx	Quad 8-Bit, Dual 16-Bit Average	Motion Compensation
SUBABS4	Quad 8-Bit Absolute of Differences	Motion Estimation
SSHVL,SSHVR	Signed Variable Shift	GSM

Non-Aligned Loads & Packed Data Processing Accelerate Motion Estimation



LDNDW (Load Non-aligned Double Word)

- Accesses a double word on any byte alignment in internal memory in a single cycle
- Key to sustaining packed data processing performance in real-world algorithms

SUBABS4 (Quad Subtract and Absolute Value)

- Computes the absolute value of the difference between 4 reference frame pixels and 4 input frame pixels
- Two SUBABS4 operations can be performed each cycle

Non-Aligned Loads Enable Peak Performance

EXAMPLE

```

L_0:
  ADD      .D2X      B_curr_mad_0,      A_curr_mad_0,      B_curr_mad
|| DOTPU4   .M1      A_err_3210_r0,      A_k0x01010101,      A_mad_r0
|| DOTPU4   .M2      B_err_7654_r3,      B_k0x01010101,      B_mad_r3
|| ADD      .S2      B_mad_r6,          B_curr_mad_7,      B_curr_mad_6
|| ADD      .S1X     A_ref_d,          B_f,          A_ref_d
|| SUBABS4  .L2      B_src_7654_r2,      B_ref_7654_r2,      B_err_7654_r2
|| LDNDW    .D1      *A_ref_d(A_p3),      A_ref_7654_r3: A_ref_3210_r3
|| SUBABS4  .L1      A_src_3210_r7,      A_ref_3210_r7,      A_err_3210_r7

```

Sustains two SUBABS4 instructions every cycle

LDNDW operations allow this code to work on reference data with any alignment

8 parallel instructions used every cycle – Sustains peak parallelism in real applications

Galois Field Multiply Streamlines Reed-Solomon Error Correction

Key Operation for Reed Solomon Forward Error Correction

- Extensively used in broadband communications including ADSL, cable modem, wireless and digital television

Both C64x™ DSP M Units have a Galois Field Multiplier

- GMPY4 instruction performs 4 parallel operations on 8-bit packed data
- Total throughput of 8 Galois Field multiplies every cycle
- Programmable for all Galois Multiplies in fields $GF(2^i)$ for $i=1$ to 8 using any generator polynomial

Breakthrough Performance (eliminates need for ASICs or hardware accelerators)

- Decodes a (204, 188, 8) packet in 1180 cycles
- 6 Mbits/s Reed Solomon decoding requires <0.6% of an 800 Mhz device

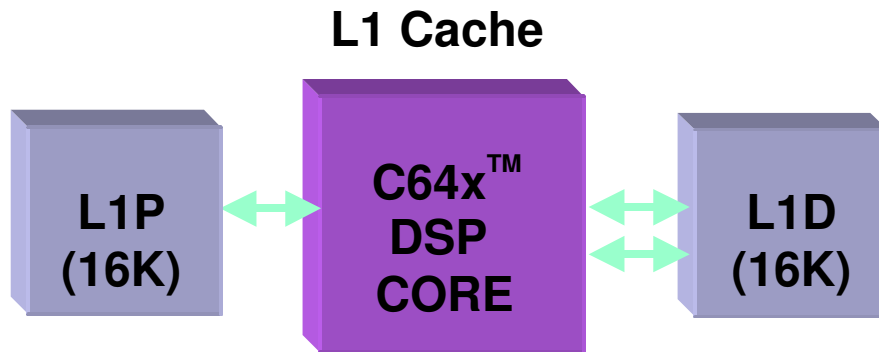
ALGORITHM	IMPROVEMENT/C62x™	CYCLES
Syndrome Accumulate	3.5x	470
Berlekamp-Massey	7.7x	246
Chien Search	4.8x	324
Forney	3.1x	140
TOTAL	4.7x	1,180

Sample Benchmark Results

DSP Kernels/Image Processing Kernels	Cycle Count		Performance	
	C62x™	C64x™	Cycle Improvement Ratio C64x™: C62x™	Total Improvement 750 MHz C64x™ vs 300 MHz C62x™
Correlation - 3x3 (8-bit)	4.5	1.28	3.5x	8.8x
	cycles/pixel			
FFT - Radix 4 - Complex (size = N log (N)) (16-bit)	12.7	6.0	2.1x	5.3x
	cycles/data			
Median Filter - 3x3 (8-bit)	9.0	2.1	4.3x	10.7x
	cycles/pixel			
Motion Estimation - 8x8 MAD (8-bit)	0.953	0.126	7.6x	19.0x
	cycles/pixel			
Polyphase Filter - Image Scaling (8-bit)	0.77	0.33	2.3x	5.8x
	cycles/output/filter tap			
Reed Solomon Decode: Syndrome Accumulation (204,188,8) Packet	1680	470	3.5x	8.8x
	cycles/packet			
Viterbi Decode (GSM) (16 states)	38.25	14 ^Ψ	2.7x	6.8x
	cycles/output			

Ψ includes traceback

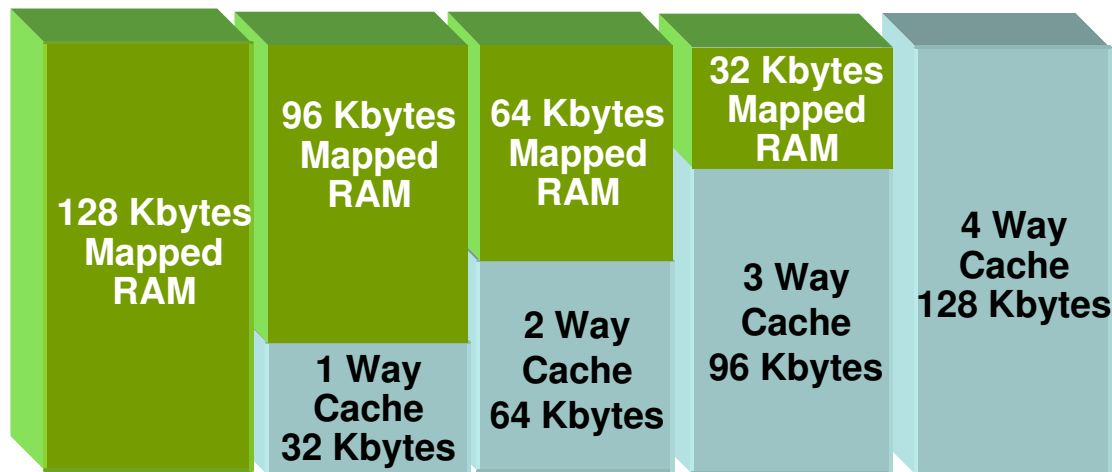
C64x DSP L1/L2 Cache Sustains High Clock Rate Performance



2-Level Cache Allows Single Cycle Access at 1.1 GHz

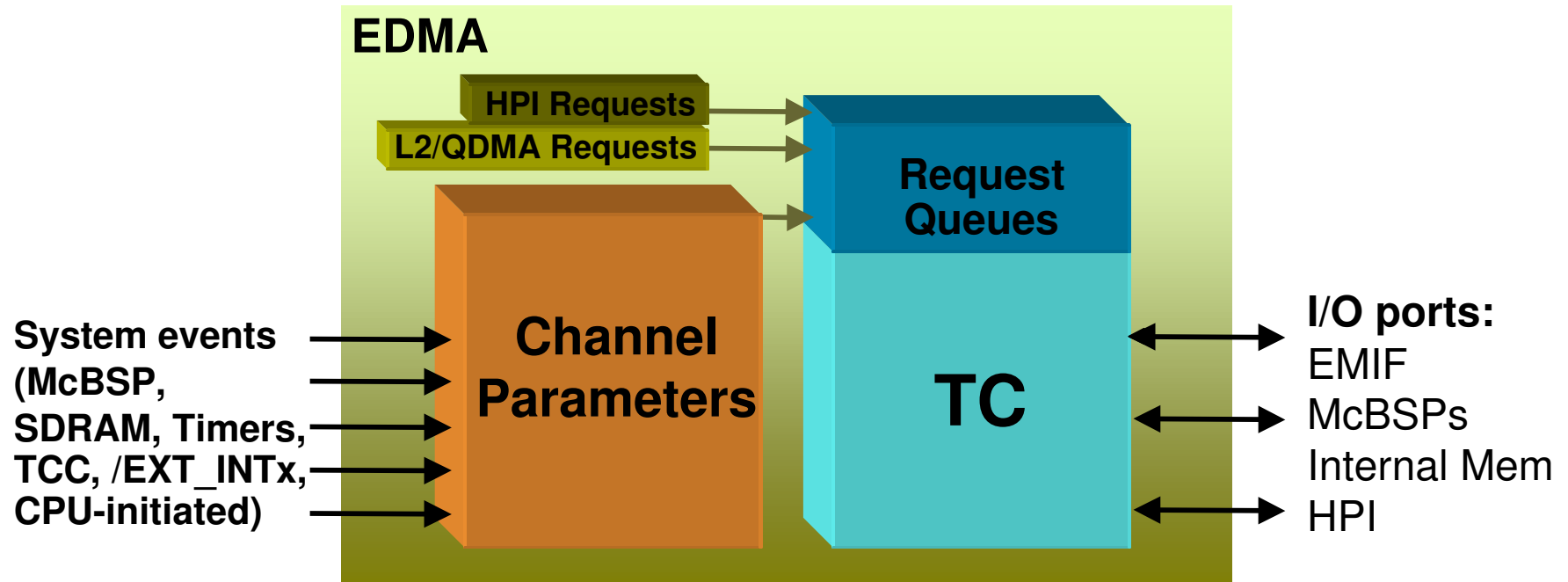
Dedicated L1s provide high clock-rate performance

128K L2 Cache Memory Configurations



Unified L2 Optimized for Streaming Data and General Purpose Code

Flexible, Programmable EDMA Maximizes Bandwidth Options



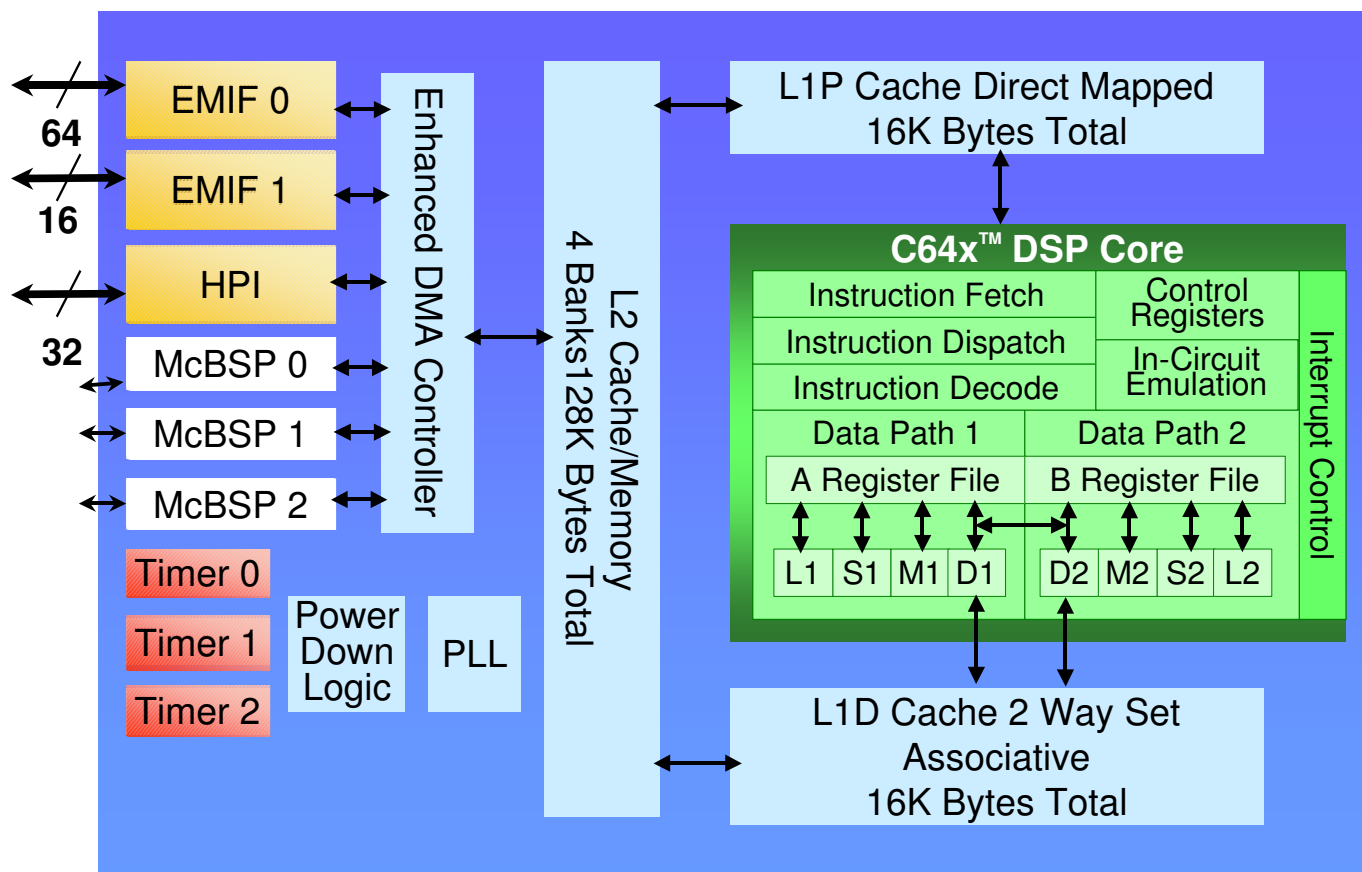
Maximizes Bandwidth Utilization

Over 2.6 Gigabytes of bandwidth
Cycle-by-cycle interleaving of transfers

Maximizes Concurrency

32 Channels with up to 85 total linkable parameter sets
4 Independent priority transfer queues
Scalable to support multiple peripherals

64x DSP Core Fueled by Gigabytes of Bandwidth for Ultimate Performance



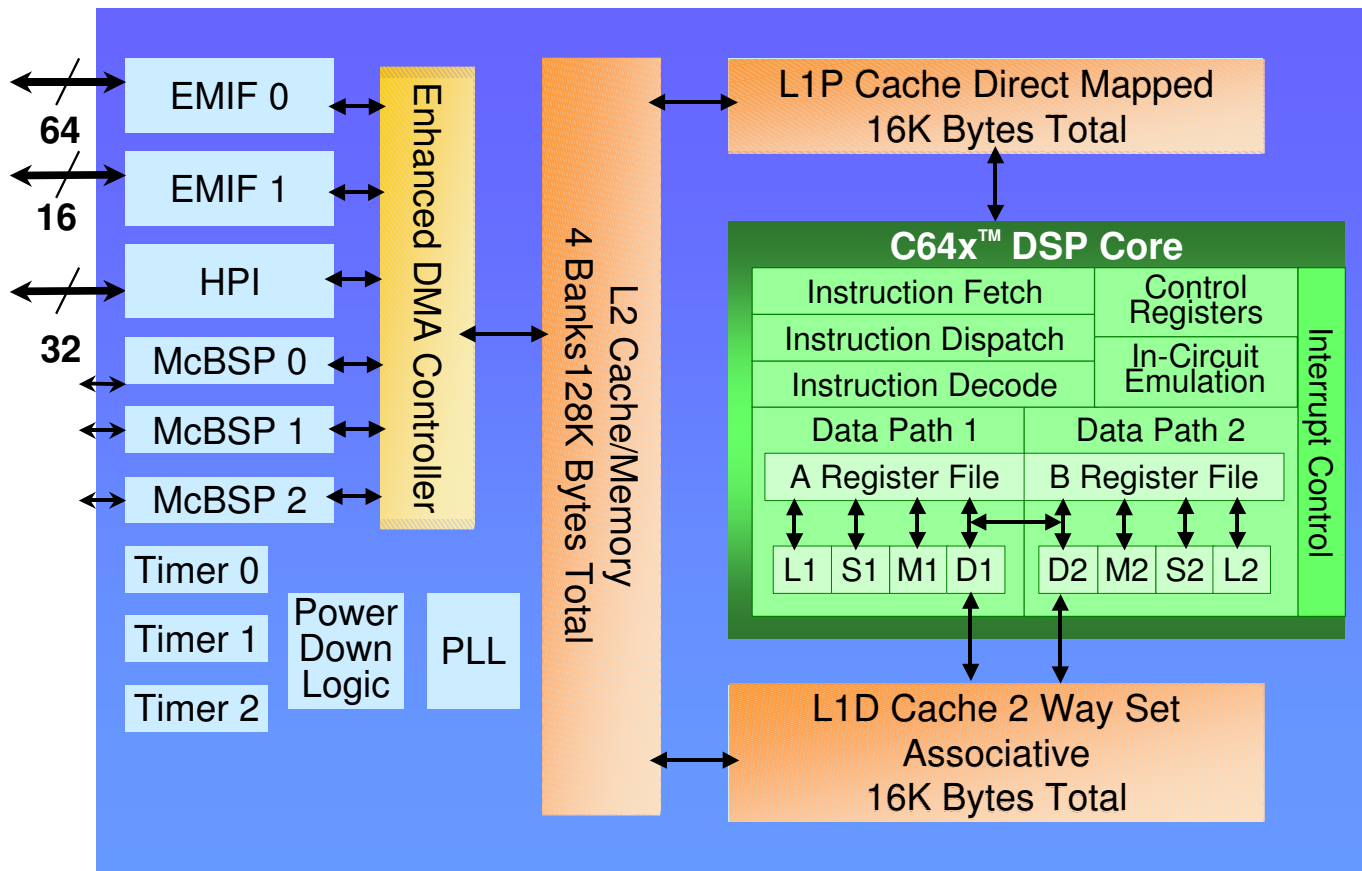
Three External Buses Providing Over 1.8 Gigabytes Bandwidth

- 64-Bit EMIF for Memory
- 1 16-Bit EMIF for I/O
- 1 32-Bit HPI

Three McBSPs with 128 Channel Support

Three Timers

64x DSP Core Fueled by Gigabytes of Bandwidth for Ultimate Performance



Enhanced DMA (EDMA) —
Over 2.6 Gigabytes of Sustained DMA Bandwidth

- 32-Channels
- Highly efficient transfer engine

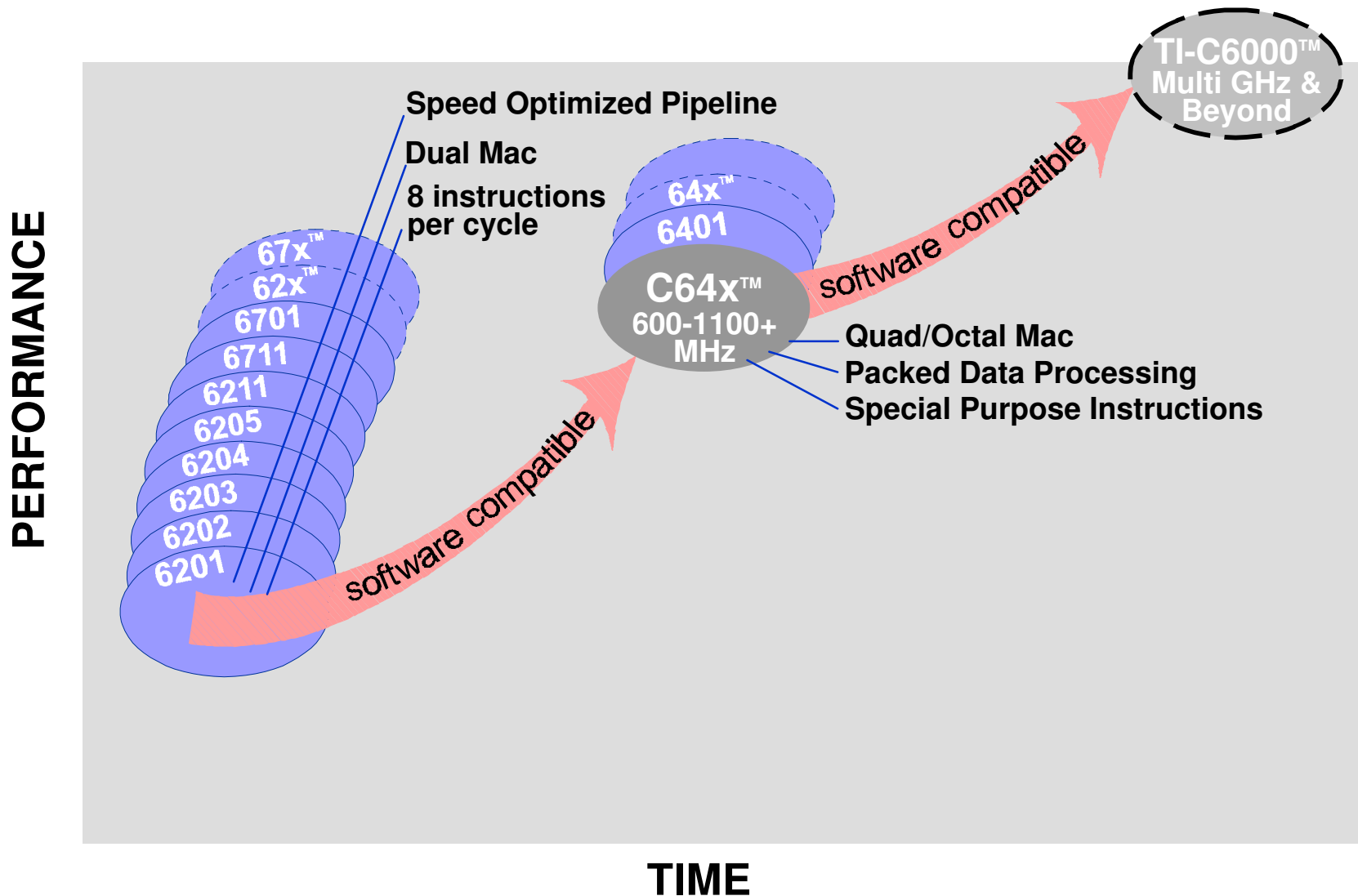
L1/L2 Cache Architecture

Additional Peripherals Under Development

Initial Samples at 600 to 800 MHz

C64x™
DSP

Software Compatible C6000™ DSP Platform Provides Range of Performance Leadership



THE WORLD LEADER IN DSP AND ANALOG

 TEXAS INSTRUMENTS

Conclusion

TI C64x™



C6000 VelociTI™ Advanced VLIW Architecture

ENABLES HIGHEST DSP CLOCK RATE



C64x VelociTI.2™ Architecture Extensions

BOOST BROADBAND COMM AND IMAGING



Packed Data Flow Support

SUSTAINS PEAK CORE PERFORMANCE



Efficient Memory and Peripheral System

MAXIMIZES SYSTEM OPTIONS



Code Compatible Roadmap

PROTECTS SOFTWARE INVESTMENT