C11
TI C7000 DSP Family

Architecture, Evolution, Family Members & Applications

(based on TI technical documentation & Optimization Guide)

Outline

- Part I: DSP Evolution - From C6000 to C7000
- Part II: C7000 Architecture Overview
- Part III: CPU Data Path & Functional Units
- Part IV: Streaming Engine & Matrix Multiply Accelerator
- Part V: Pipeline & Instruction Set
- Part VI: Memory Architecture
- Part VII: C7x ISA Variants (C7100, C7120, C7504, C7524)
- Part VIII: SoC Family Members (Jacinto, Sitara, AM275x)
- Part IX: Applications (ADAS, Robotics, Audio, Radar)
- Part X: Software Ecosystem & Development Tools
- Summary & Homework

Part I

DSP Evolution - From C6000 to C7000

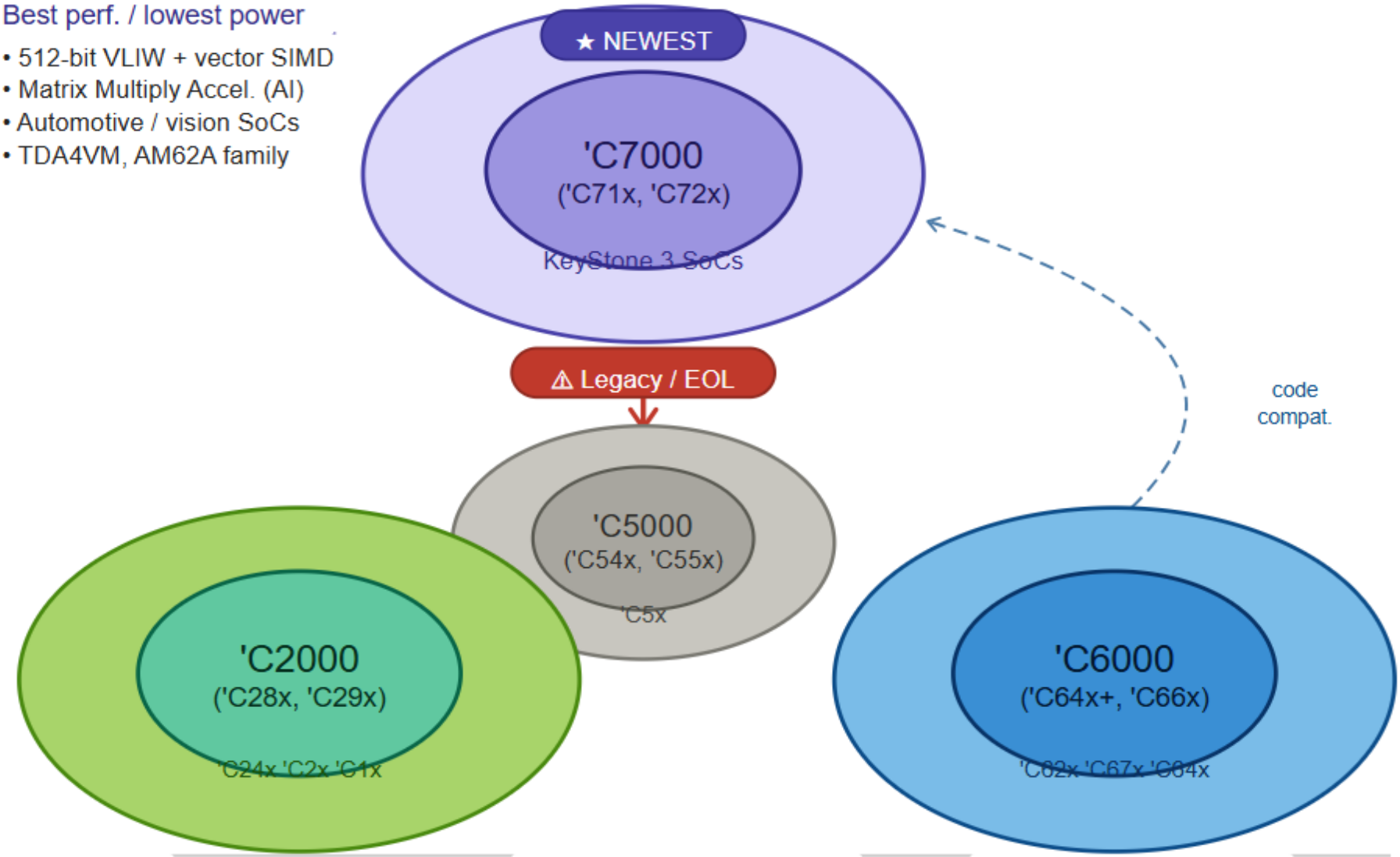
TI DSP Architecture Evolution

- 1983: TMS32010 - first TI DSP, 16-bit fixed-point
- 1997: TMS320C62x - first VLIW DSP (VelociTI), 8 functional units
- 2000: TMS320C64x - enhanced fixed-point, 2x MAC performance
- 2004: TMS320C64x+ - SPLOOP, compact instructions
- 2006: TMS320C674x - combined fixed+floating, low power
- 2010: TMS320C66x - 128-bit SIMD, IEEE 754 FP, multicore KeyStone
- 2019: C7x (C7100) - 512-bit vector VLIW, Streaming Engine, MMA
- 2023: C7x (C7504/C7524) - 256-bit vector, cost-optimized
- 2025: C7x in AM275x - automotive audio DSP SoCs

Different needs? Multiple families!

Best perf. / lowest power

- 512-bit VLIW + vector SIMD
- Matrix Multiply Accel. (AI)
- Automotive / vision SoCs
- TDA4VM, AM62A family



Lowest cost

Real-time control

- Motor & power conversion
- Industrial automation
- Edge AI / NPU (2024)

Legacy — no new designs

Was: wireless phones, VoIP

Was: modems / telephony

→ Migrate to Sitara AM57x

Maximum performance

Multi-channel / function

- 5G / radar / imaging
- KeyStone multicore SoCs
- xDSL, comm. infrastructure

Family	Status	Target / Positioning
C2000 (C28x, C29x)	✅ **Active** — expanding	Real-time control: motor drives, power conversion, industrial automation, automotive. Now includes NPU for edge AI.
C5000 (C54x, C55x)	⚠️ **Legacy / EOL**	No longer recommended for new designs. Replaced by ARM Cortex-M for low-power signal processing.
C6000 (C66x, KeyStone)	✅ **Active** — mature	High-performance fixed+float DSP: 4G/5G base stations, radar, medical imaging, SDR. Multi-core SoCs (up to 8× C66x + ARM).
C7000 (C71x)	✅ **Active** — newest	Highest performance VLIW DSP with vector (SIMD) instructions and MMA (Matrix Multiply Accelerator) for machine learning. Texas Instruments Embedded in automotive/vision SoCs (TDA4VM, AM62A).
Sitara AM57x	✅ **Active**	ARM + C66x DSP heterogeneous SoC — replaces OMAP for embedded Linux/RTOS desi

C5000 vs C7000 — Side-by-Side Comparison

Feature	‘C5000 (C55x) — Legacy	‘C7000 (C71x/C72x) — Current
CPU Architecture	16-bit fixed-point, modified Harvard, dual MAC	**VLIW, 512-bit vector SIMD, fixed + float unified**
Peak performance	up to 600 MIPS (C5510 @ 300 MHz)	**>40 GFLOPS/core (C7120 @ 1 GHz)**
Data width	16-bit words; 40-bit accumulators	**512-bit vector; 64-bit scalar; FP16/32/64**
AI / ML inference	None — software FIR/IIR filters only	**Matrix Multiply Accelerator (MMA) — INT8, FP16**
Target applications	VoIP, modems, portable audio, wireless handsets	**ADAS, camera pipelines, radar, AI inference**
Design status	**⚠ Legacy / EOL — not for new designs**	**✅ Active — recommended for new designs**

Summary — TI C5000 DSP Family

C54x Family — Key Points

- 16-bit fixed-point Harvard architecture; 40-bit accumulators; 17x17 MAC in one cycle.
- Key peripherals: McBSP (serial audio/codec), HPI (host port), 6-channel DMA, PLL, UART.
- Range: C5401 (50 MIPS, 1.8V) to C5441 (532 MIPS, quad-core, 1.5V, 550 mW).
- Special variants: C54V90 (integrated V.90 modem), C54CST (client-side telephony ROM).

C55x Family — Key Points

- Binary-compatible with C54x; 55x lower power at equivalent performance.
- C5510: 400 MIPS, 24 KB I-cache, 160 KW SRAM, 3 McBSPs, 16-bit EMIF.
- C5501: 600 MIPS @ 300 MHz, I2C, hardware UART, APLL, 76 GPIO — aimed at portable imaging and audio.

OMAP — Key Points

- Heterogeneous SoC: ARM(application)+C55x/C64x (DSP) + GPU on a single die.
- Pioneered the ARM+DSP SoC model that is now universal in smartphones and embedded systems.
- **End-of-Life (2017):** use TI Sitara AM57x for new designs.

Why C5000 Matters Today?

- Architectural concepts (Harvard memory, VITERBI accelerator, DMA offload, circular buffers) remain foundational in all modern DSP cores.
- Power management techniques introduced in C55x (idle modes, voltage scaling, clock gating) are standard practice in all SoC design today.
- Reference: Texas Instruments, C5000 DSP Platform Overview; TI.com/processors/dsp

C7000 DSP — Architecture Overview

CPU Core

- VLIW architecture — up to 13 functional units per cycle, all pipelined.
- 512-bit wide vector (SIMD) data path: up to 64 operations per instruction depending on data type.
- Unified fixed-point + floating-point (single and double precision) in one instruction set.
- Scalar pipes: 64-bit ALU, multiplier, load/store, branch, predicate.

Matrix Multiply Accelerator (MMA)

- Hardware accelerator for deep-learning inference (INT8, FP16, FP32 MACs).
- Offloads CNN/DNN inference from the CPU vector units, improving TOPS/W efficiency.
- Used via TI Deep Learning (TIDL) library, part of Processor SDK.

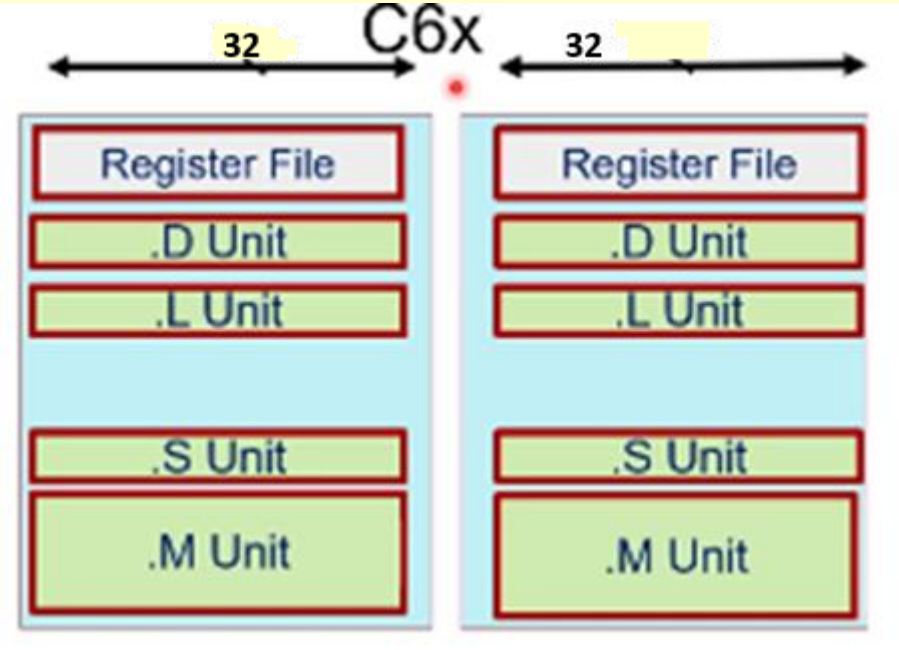
Memory Subsystem

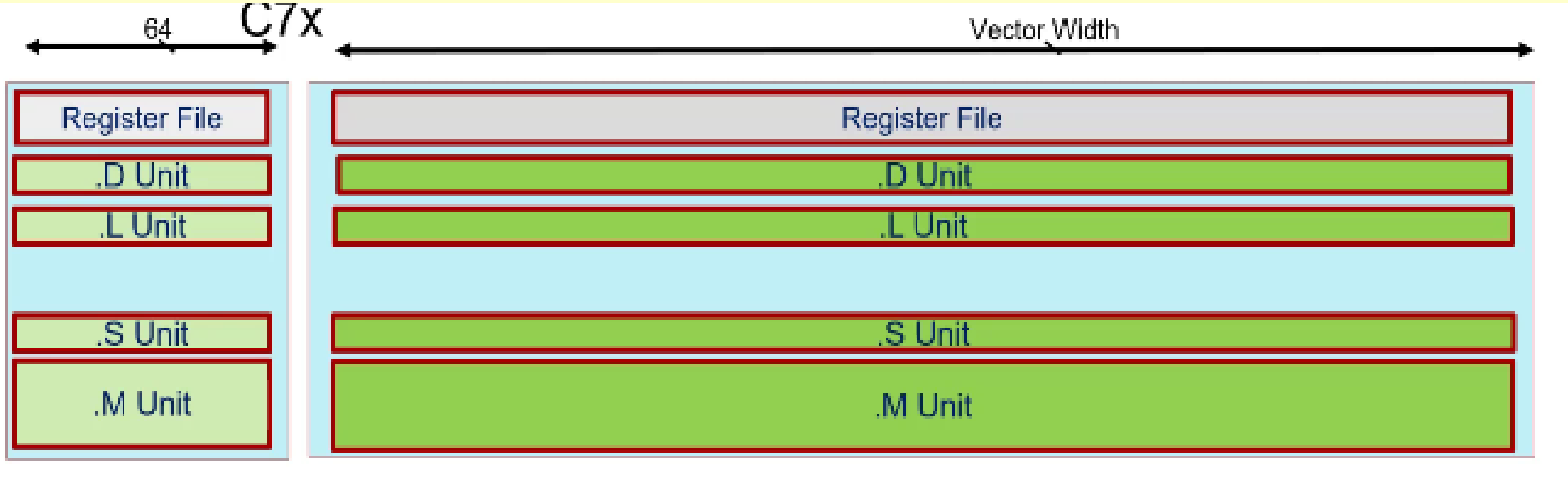
- Large L1D / L1P caches + unified L2 cache per core; L3 MSMC (Multi-core Shared Memory Controller) shared across cores.
- The hardware handles data movement, so programmers do not need to write DMA code for core operations.
- Histogram Unit: hardware accelerated histogram computation for imaging and computer vision.

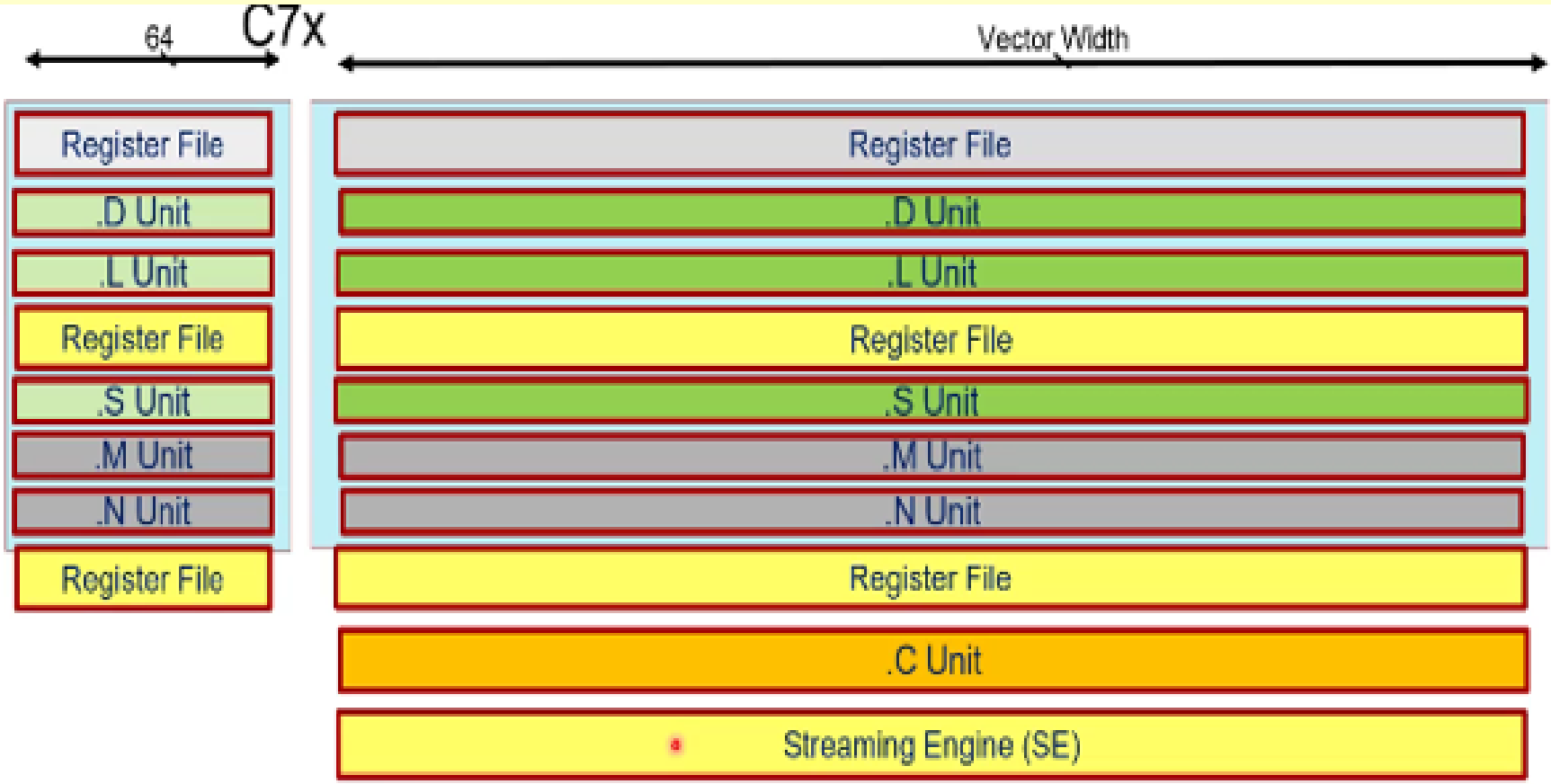
Development Tools

- TI C7000 CGT (cl7x) compiler: C/C++ with Open CL C extensions, auto-vectorizations for SIMD paths.
- Code Composer Studio 12+ (CCS) with C7000 host emulation: run C7x code on any Linux/Windows PC before hardware is available.
- Processor SDK with TIDL, Open VX and Vision Apps for rapid vision pipeline development.

DSP Evolution - From C6000 to C7000







- Expand B-side to vector width (256, 512, etc.)
- Split M-unit into two parts to maximize usage of available multipliers
- New .C Unit for vector permutation, horizontal adds, etc.
- Additional Registers, and Streaming Engine to supply vector data

Why C7000? Key Improvements over C6000.

- **Vector width: 512-bit (C7100/C7120) vs 64-bit (C66x) = 8x wider**
- **Functional units: 13 units vs 8 units (C66x)**
- **Register files: 64 vector regs (512-bit) + 64 scalar regs (64-bit)**
  - C66x had 2x32 general-purpose 32-bit registers
- **Streaming Engine: hardware-accelerated data prefetch, bypasses L1D**
  - Eliminates explicit LOAD instructions for regular patterns
- **Matrix Multiply Accelerator (MMA): dedicated AI/ML acceleration**
  - Up to 256 MACs/cycle for INT8, 64 MACs/cycle for FP32
- **Native data types: INT8, INT16, INT32, FP16, FP32, FP64**
- **Backward compatible with C6x code at source level**

Very Long Instruction Word (VLIW)

- Can encode up to 12 instructions in parallel
- Presumes compiler finds lots of parallelism. Scheduling loops into a software pipeline is the main way that happens.
- Lots of functional units
- An instruction for each unit can appear once in a group of instructions that execute in parallel

https://www.ti.com/video/6350279028112

C7000 in TI's Processor Roadmap

- **C7000 is part of KeyStone 3 architecture generation**
- **Not sold as standalone DSP - always inside a heterogeneous SoC**
  - Combined with ARM Cortex-A72/A53 + Cortex-R5F + accelerators
- **Target markets shifted from standalone DSP boards to:**
  - Automotive ADAS & autonomous driving (Jacinto 7)
  - Industrial robotics & machine vision (AM68A, AM69A)
  - Automotive audio systems (AM275x, AM62D)
  - Radar signal processing (AWRL6844)
- **The C7x DSP core is TI's compute workhorse for edge AI**
- **Positioned between general-purpose CPU and dedicated accelerator**

KeyStone I/II were DSP-centric architectures with ARM bolted on. KeyStone 3 is a heterogeneous-first architecture where C7x is one of many co-equal processing domains, all managed by central firmware and connected through a secure, QoS-aware fabric. This is what makes it possible to put C7x inside an automotive SoC alongside ARM cores, GPU, ISP, and video codecs — all safely isolated from each other.

Part II

C7000 Architecture Overview

C7000 Architecture - Key Features

- **VLIW (Very Long Instruction Word) architecture**
  - Same fundamental principle as C6000 VelociTI
- **13 fully pipelined functional units**
  - Up to 12 instructions can start execution every cycle
- **Split datapath: A-side (scalar) + B-side (vector)**
- **512-bit or 256-bit vector SIMD depending on variant**
- **Two Streaming Engines (SE0, SE1) for high-bandwidth data input**
- **Streaming Address Generator (SA) for high-bandwidth data output**
- **Matrix Multiply Accelerator (MMA)**
- **Hardware loop support (zero-overhead looping)**
- **Vector predication for conditional per-lane execution**

C7000 Block Diagram

- **C7x CorePac contains:**
  - C7x CPU core (scalar + vector data path)
  - L1 Program Memory Controller (32 KB L1P, all cache)
  - L1 Data Memory Controller (48 KB L1D, cache/SRAM)
  - L2 Unified Memory Controller (512 KB-1 MB, cache/SRAM)
  - Streaming Engine (SE) - 2 instances
  - CorePac MMU (CMMU)
- **Matrix Multiply Accelerator (MMA) - on select variants**
- **Power-down controller for low-power modes**
- **JTAG debug and trace capabilities**
- **Connected to SoC via the internal bus fabric (CBASS)**

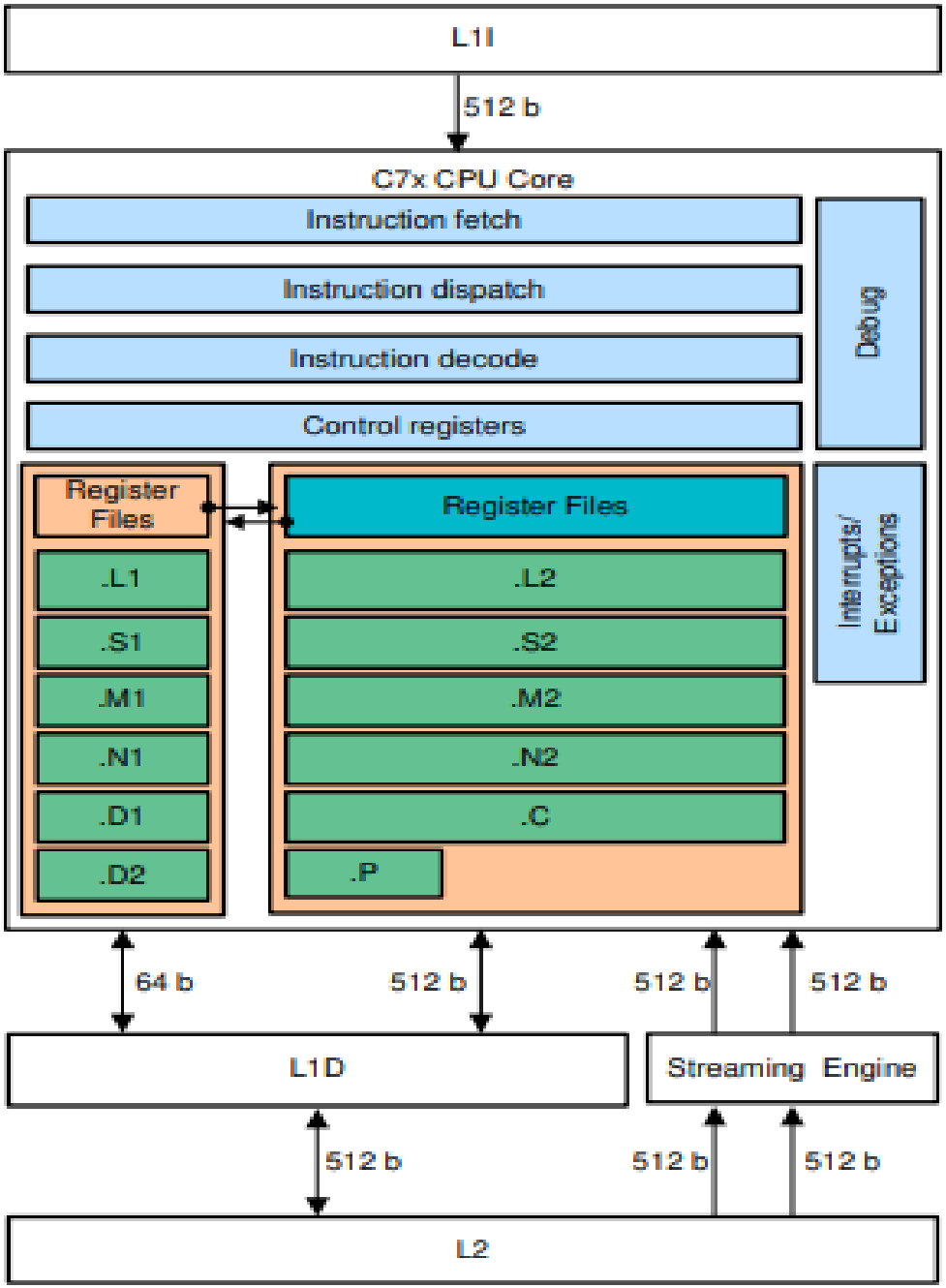
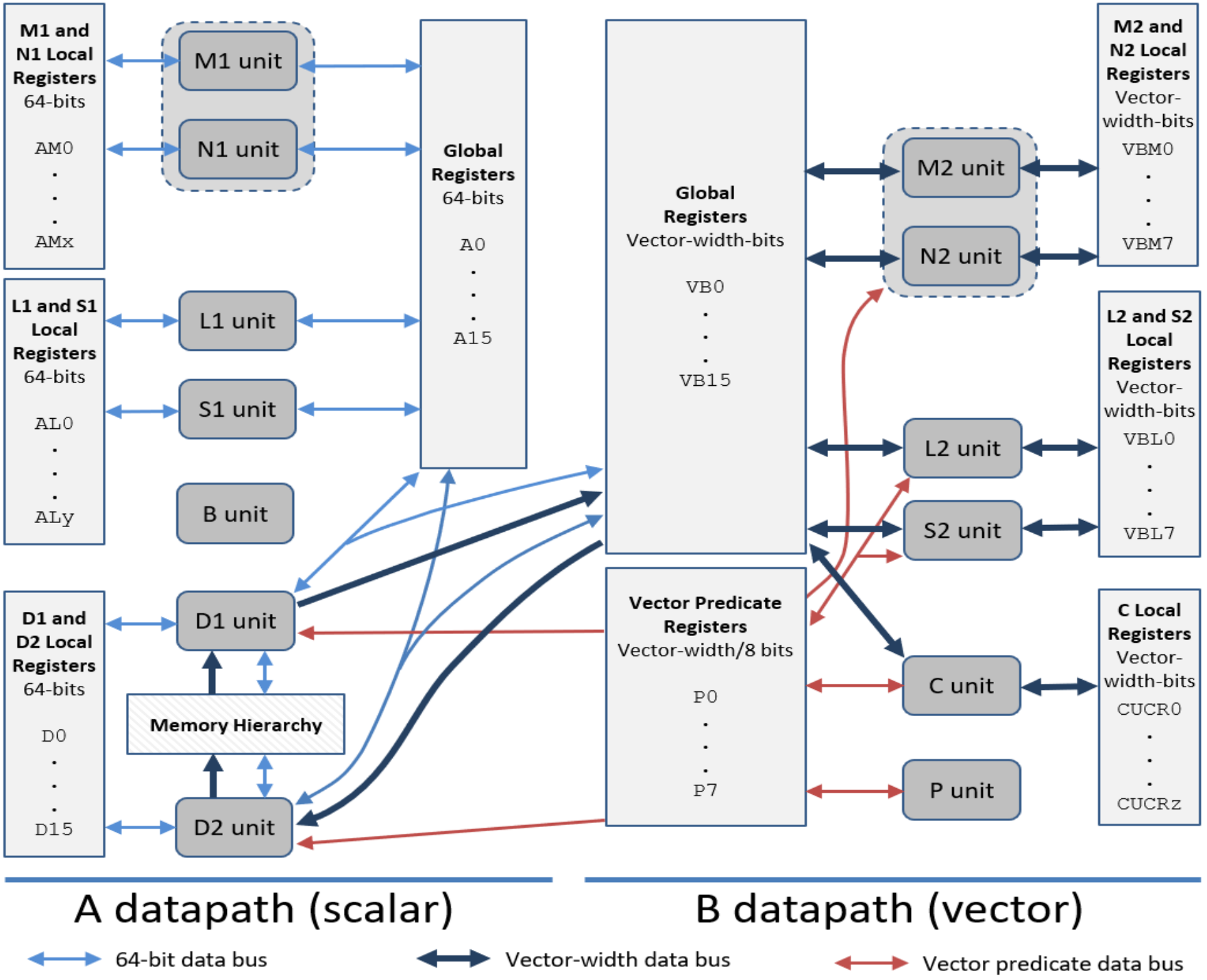


Figure 3-1. C7x DSP Block Diagram


M1 and N1 Local Registers 64-bits
AM0
AMx
M1 unit
N1 unit
Global Registers 64-bits
A0
A15
L1 and S1 Local Registers 64-bits
AL0
ALy
L1 unit
S1 unit
B unit
D1 and D2 Local Registers 64-bits
D0
D15
D1 unit
D2 unit
Memory Hierarchy
Global Registers Vector-width-bits
VB0
VB15
Vector Predicate Registers Vector-width/8 bits
P0
P7
M2 unit
N2 unit
L2 unit
S2 unit
C unit
P unit
M2 and N2 Local Registers Vector-width-bits
VBM0
VBM7
L2 and S2 Local Registers Vector-width-bits
VBL0
VBL7
C Local Registers Vector-width-bits
CUCR0
CUCRz
A datapath (scalar)
B datapath (vector)
64-bit data bus
Vector-width data bus
Vector predicate data bus

Split Datapath Architecture

- **A-side (Scalar) data path:**
  - Scalar computation (64-bit)
  - Load/store operations (both scalar and vector)
  - Control flow: branches, calls, returns
  - Units: L1, S1, D1, D2, M1, N1
- **B-side (Vector) data path:**
  - Vector math operations (512-bit or 256-bit wide)
  - Data permutation and rearrangement
  - Vector predication operations
  - Units: L2, S2, M2, N2, P (predicate)
- **One 64-bit cross-path per data path (A <-> B)**
  - Allows one read per cycle from opposite side register file

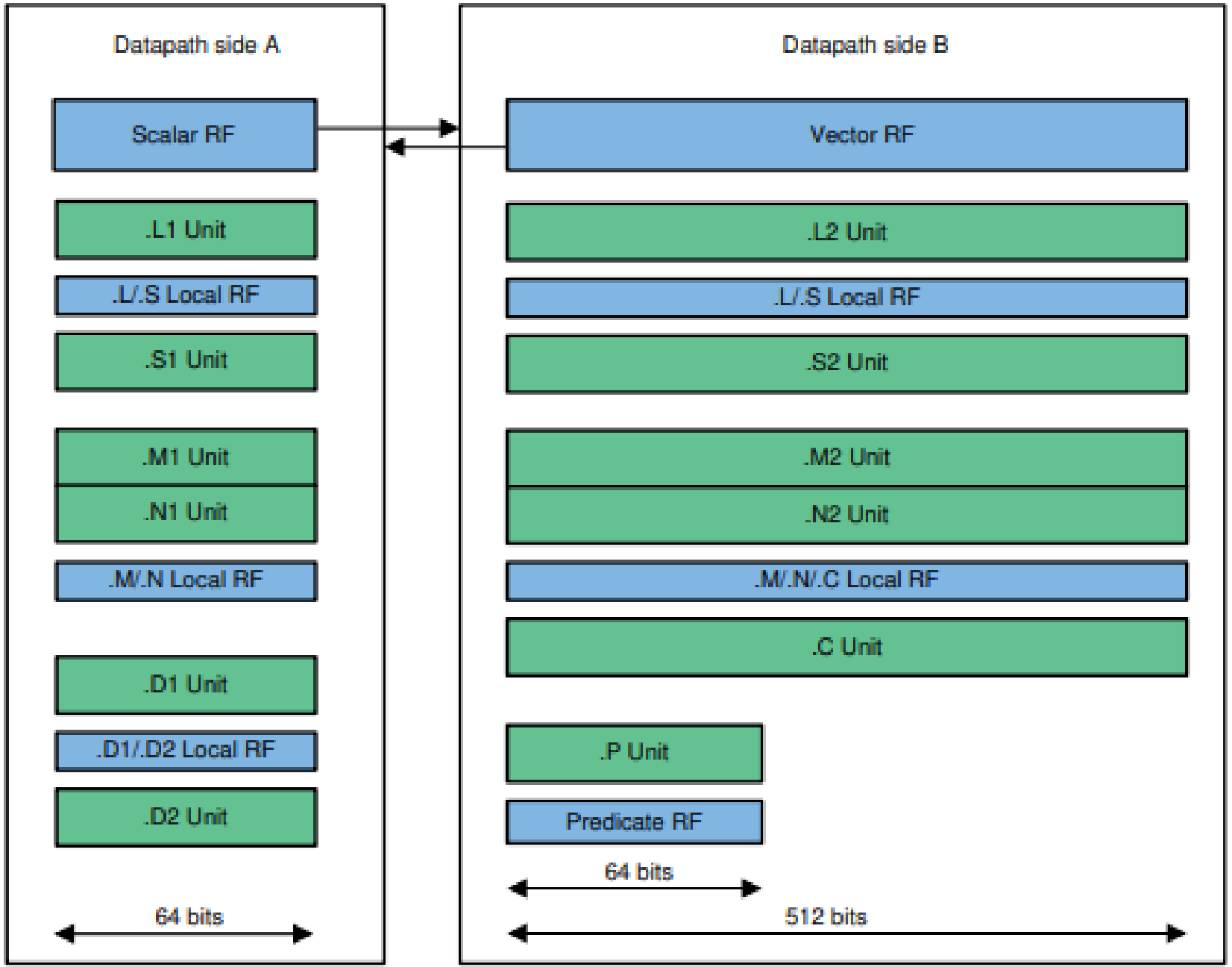


Figure 3-2. C7x Data Path Overview

C7x Register Files

- **Heterogeneous register files across datapaths:**
- **A-side (scalar) registers:**
  - A0-A63: 64-bit scalar general-purpose registers
  - Also used for addressing, loop counters, control
- **B-side (vector) registers:**
  - VB0-VB63: vector registers (512-bit on C7100/C7120)
  - Can be treated as vectors of INT8/16/32, FP16/32/64
- **Predicate registers:**
  - Used for per-lane masking in vector operations
  - Enable conditional execution at the element level
- **Comparison: C66x had 2 x 32 registers (32-bit each)**
- **C7x: 64 scalar (64-bit) + 64 vector (512-bit) = massive parallelism**

Functional Units - A-side (Scalar)

- **L1 unit: scalar ALU operations**
  - Add, subtract, compare, logical, shifts
  - Can also perform some vector operations
- **S1 unit: scalar ALU + branch/control flow**
  - Branches, calls, returns, loop control
  - Constant generation, bit manipulation
- **D1 unit: load/store unit #1**
  - 64-bit or vector-width loads from memory
  - 64-bit or vector-width stores to memory
- **D2 unit: load/store unit #2**
  - Allows two parallel memory accesses per cycle
- **M1 unit: scalar multiply unit**
  - Integer and floating-point multiply operations

Functional Units - B-side (Vector)

- **L2 unit: vector ALU operations**
  - Vector add/sub, compare, logical, shifts on full width
  - Operates on 8/16/32/64-bit element widths
- **S2 unit: vector ALU + permutation**
  - Vector shuffle, pack, unpack, interleave
  - Critical for FFT butterfly, matrix transpose
- **M2 unit: vector multiply unit**
  - Vector MAC (Multiply-Accumulate) operations
  - Complex multiply, dot product
- **N2 unit: additional vector compute**
  - Supplementary math operations, type conversion
- **P unit: predicate operations**
  - Compare, combine, manipulate predicate registers
- **Total: 13 units = up to 13 instructions per cycle**

C7x vs C66x Architecture Comparison

Feature	C66x	C7x (C7100)
VLIW Width	256-bit (8x32)	Variable (up to 13 instr)
Vector Width	64-bit	512-bit
Register Files	2x32 (32-bit)	64 scalar + 64 vector
Functional Units	8	13
Streaming Engine	No	Yes (2 instances)
MMA	No	Yes (optional)
FP Support	SP + DP	FP16 + SP + DP
SIMD Elements (INT8)	8	64
L1D Cache	32 KB	48 KB
L2 Memory	Up to 1 MB	512 KB - 1 MB

Part III

Streaming Engine & Matrix Multiply Accelerator

Streaming Engine (SE) - Overview

- Hardware unit for high-speed data loading from L2/higher memory
- Two instances: SE0 and SE1, operating independently
- Key benefits over traditional LOAD instructions:
  - Higher bandwidth from L2 to CPU than loads alone
  - Pre-fetches data to a location near the CPU for fast access
  - Bypasses L1D cache, - reduces cache capacity misses
  - Frees D1/D2 units for other load/store operations
- Compilers can automatically use SE via --auto_stream option
- Programmers can explicitly control SE via C7x intrinsics (c7x_strm.h)
- Ideal for: FFT, FIR/IIR filters, matrix multiplication, convolution
- SE delivers one vector-width element per cycle to functional units

Streaming Engine - Addressing Patterns

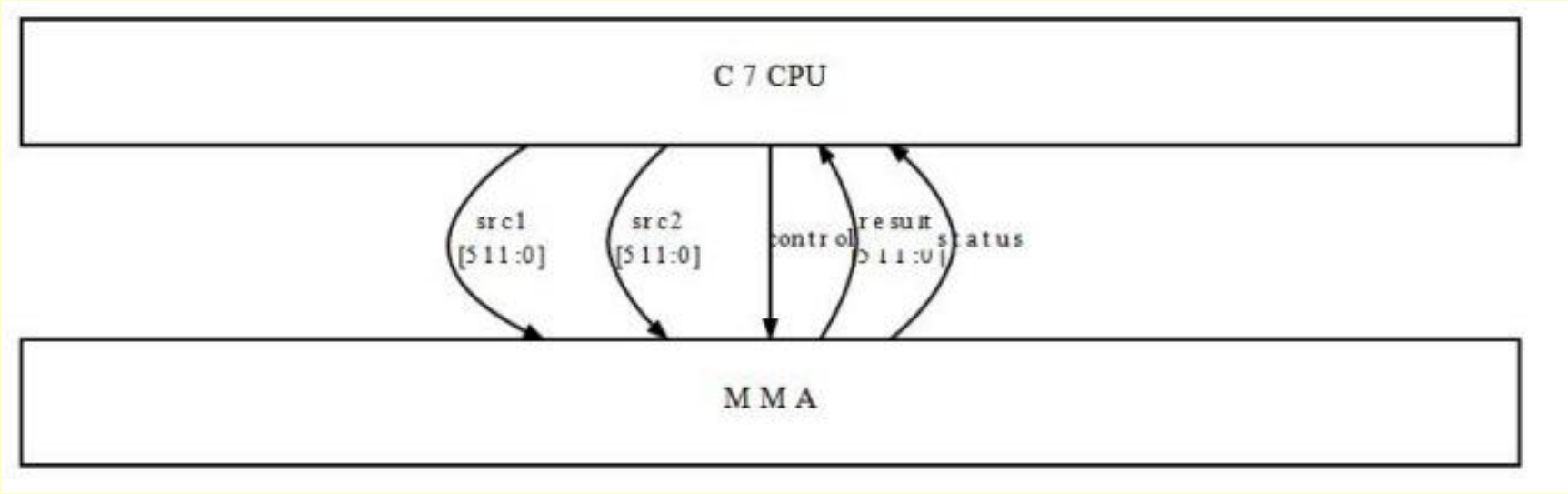
- SE supports multi-dimensional addressing patterns:
  - 1D: linear sequential access (array sweep)
  - 2D: row-by-row access (image processing)
  - 3D: block-by-block access (3D convolution)
  - Up to 6 dimensions supported
- Configurable parameters via __SE_TEMPLATE structure:
  - ELETYPE: element data type (INT8, INT16, FP32, etc.)
  - VECLEN: number of elements per vector fetch
  - DIMFMT: number of addressing dimensions
  - ICNT0-5: iteration counts for each dimension
  - DIM1-5: stride (in bytes) for each dimension
- SE pattern is opened with __SE_OPEN() and data read via __SE_REG()
- Pattern is closed with __SE_CLOSE() when done

Streaming Address Generator (SA)

- SA is the output counterpart to SE (Streaming Engine)
- *Generates store addresses in hardware-accelerated* patterns
- Same multi-dimensional addressing as SE (up to 6D)
- Configured via __SA_TEMPLATE structure
- Key difference from SE:
  - SE reads data from memory to CPU registers
  - SA generates addresses for store operations
- SA support varies by C7x variant:
  - C7504/C7524: full SA with context switching
  - C7100/C7120: SA support with some restrictions
- Combined SE+SA enables efficient in-place data transforms

Matrix Multiply Accelerator (MMA)

- Dedicated hardware accelerator for matrix multiplication
- Tightly coupled to C7x DSP core as a special functional unit
- Primary use: ***deep learning inference acceleration***
  - Convolutional Neural Networks (CNN)
  - Fully-connected layers, depthwise convolution
- Supported data types and throughput (per cycle):
  - INT8: up to 256 MACs/cycle
  - INT16: up to 64 MACs/cycle
  - FP16: up to 64 MACs/cycle (on C7120)
  - FP32: up to 16 MACs/cycle
- MMA performance in TDA4VM: 8 TOPS (INT8) at 1 GHz
- Use ***TI Deep Learning Library (TIDL)*** for optimized MMA access



- MMA Operands

- For data processing instructions, the MMA unit reads up to two operands.

The system can direct the operands to A vector, B matrix, C matrix storage, or configuration registers.

- The operand width is always 512 bits and matches the width of the C7x vector registers.
- MMA data processing instructions may use ***vector registers or streaming engines as operands***. When a streaming engine serves as an MMA operand, it behaves the same as it does for C7x instructions.
- The C7x CPU receives 512-bit results accompanied by status.

Part IV

Pipeline & Instruction Set

C7x CPU Pipeline

- Multi-stage pipeline, similar in concept to C6000 but enhanced:
- **Fetch phase:** fetches instruction packets from L1P cache
  - Variable-length fetch packets (unlike fixed 256-bit in C6000)
  - Each packet contains 1 to N instructions (header-based)
- **Dispatch phase:** routes instructions to functional units
- **Decode stage:** decodes opcodes and reads operands
- **Execute phase:** multi-stage execution in functional units
  - Single-cycle for most ALU operations
  - Multi-cycle for loads, multiplies, branches
- Pipeline depth: 9-16 stages, depending on operation.
- Branch prediction supported (unlike C6000 which had none)
- Hardware does not interlock – compiler manages scheduling

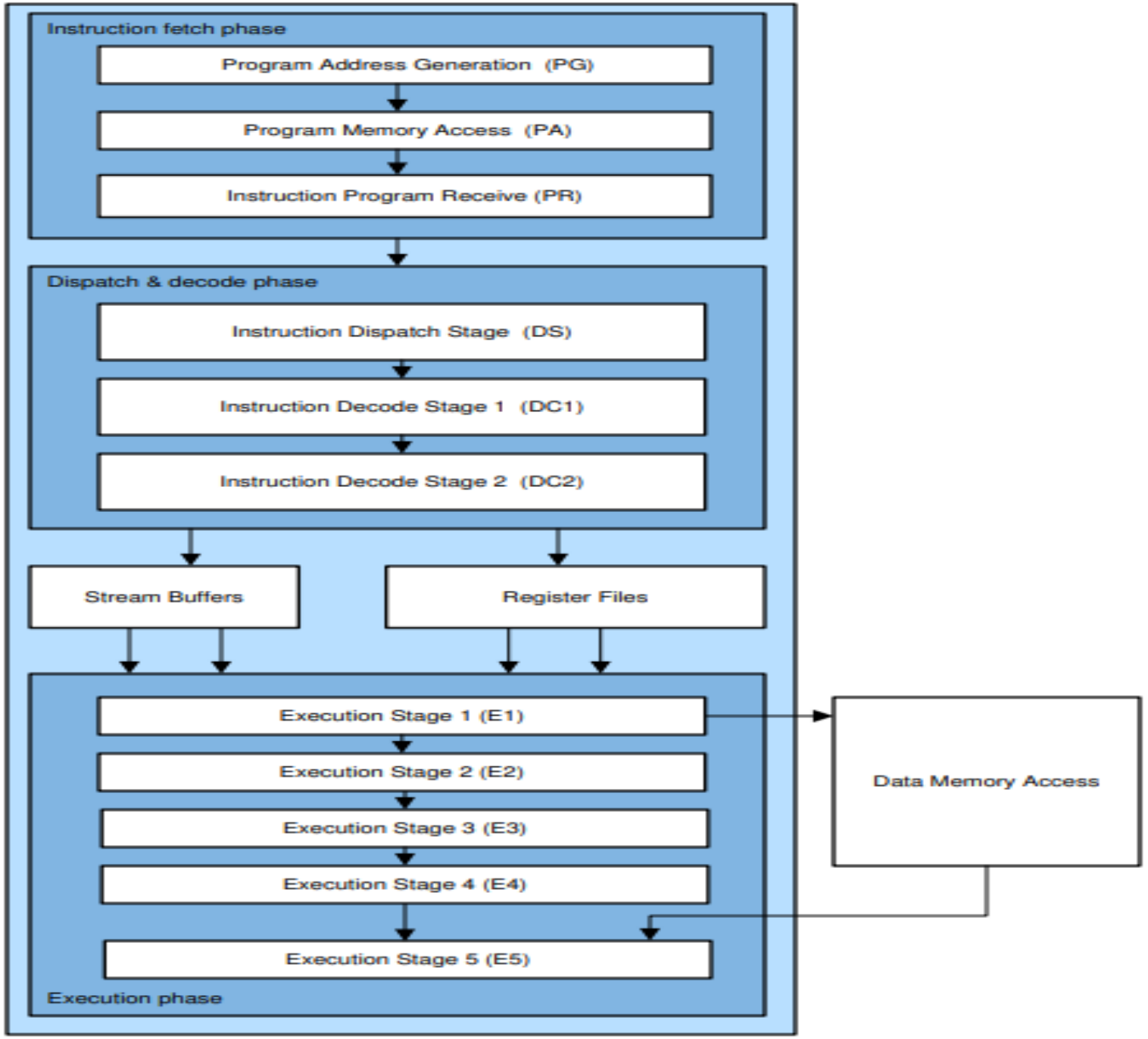


Figure 3-3. C7x Pipeline

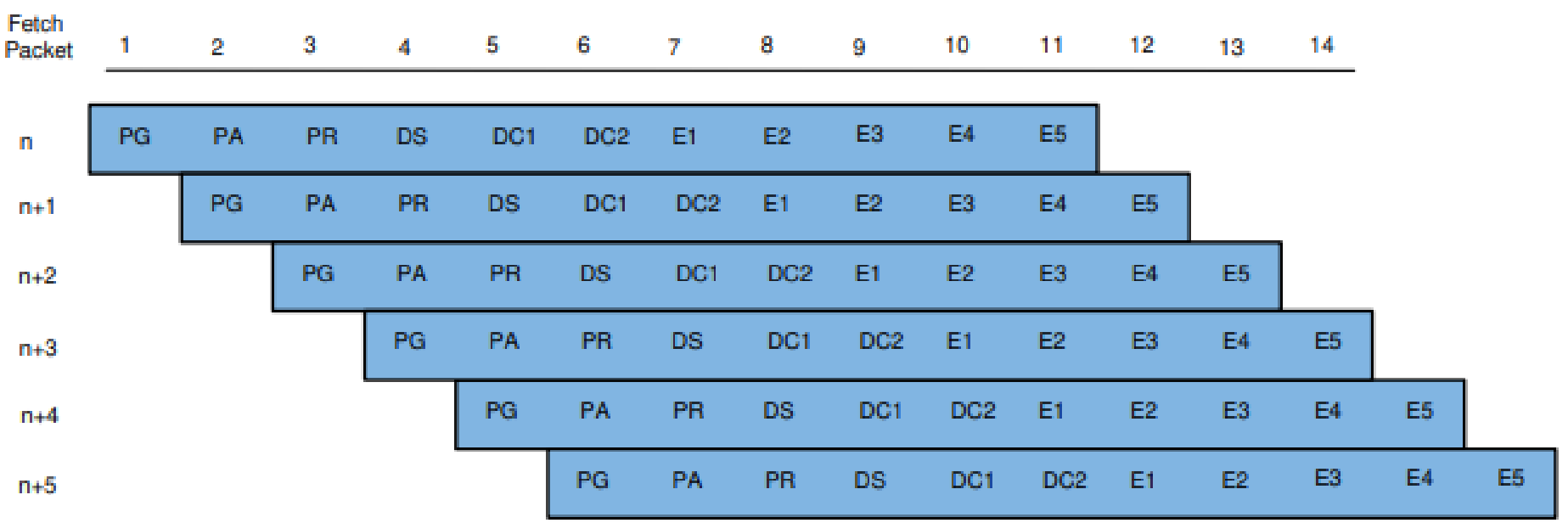


Figure 3-4. Pipeline Operation: One Execute Packet per Fetch Packet

C7x Instruction Packing

- C7x uses a different packing scheme than C6000:
- C6000: fixed 256-bit fetch packet, p-bit per instruction
- C7x: variable-length instruction packets with header
  - Packet header indicates the number of instructions (1-14)
  - More flexible than fixed 8-instruction packets
- Each instruction is 32 bit or 64-bit (some use extended encoding)
- Compact instructions (16-bit) for common operations
  - Reduces code size by up to 30%
- Parallel execution still determined at compile-time (VLIW)
- Compiler decides which instructions go into same packet
- No runtime scheduling overhead (unlike super-scalar)

C7x Instruction Categories

- **Scalar Operations (A-side):**
  - Integer arithmetic
  - Logical & shift
  - Branch & call
  - Load/Store
  - Scalar multiply
  - Bit manipulation
  - Compare & select
- **Vector Operations (B-side):**
  - Vector add/sub/mul
  - Vector MAC
  - Vector compare
  - Permute/shuffle
  - Pack/unpack
  - Type conversion
  - Complex arithmetic

Vector Data Types & SIMD Operations

- C7x supports multiple data types in vector registers:
- 512-bit vector (C7100/C7120) holds simultaneously:
  - 64 x INT8, or 32 x INT16, or 16 x INT32, or 8 x INT64
  - 32 x FP16, or 16 x FP32, or 8 x FP64
- 256-bit vector (C7504/C7524) holds:
  - 32 x INT8, or 16 x INT16, or 8 x INT32, or 4 x INT64
- SIMD operations process all elements in one instruction cycle
- VADDW – adds 16 x INT32 pairs in one operation (C7100)
- Vector predication: per-lane enable/disable via predicate register
  - Eliminates branches for conditional per-element processing
- OpenCL-style vector types available in C/C++ intrinsics

C7x Predication & Conditional Execution

- Two levels of predication on C7x:
- 1. Scalar predication (like C6000):
  - Any scalar instruction can be conditional on a register
  - Eliminates short branches, same as C6000 mechanism
- 2. Vector predication (new in C7x):
  - Operators can enable or disable each vector lane on their own.
  - Uses dedicated predicate registers (VP0-VPn)
  - Example: vector ADD with mask - only active lanes compute
- Vector predication eliminates scatter/gather overhead
- Critical for irregular data patterns and sparse computations
- Replaces branching in SIMD code for up to 64 elements at once

Key C7x Instructions for DSP

- VDOTPW: vector dot product (INT32 elements)
- VCMPY: vector complex multiply (FP32 or INT16)
- VFFTB: vector FFT butterfly operation
- VMATMPY: vector-matrix multiply
- VADDSP / VSUBSP / VMULSP: vector FP32 arithmetic
- VSORT: vector sort (for median filters, etc.)
- VPERM: vector permutation (arbitrary lane reordering)
- VPACK / VUNPK: pack and unpack between data widths
- VCONV: vector type conversion (e.g., INT16 to FP32)
- VBITCNT / VBITREV: bit-level vector operations
- VMAX / VMIN: vector element-wise maximum/minimum

Part V

Memory Architecture

C7x Memory Hierarchy

- L1 Program Memory (L1P): 32 KB, always cache
  - Instruction cache only no SRAM mode
- L1 Data Memory (L1D): 48 KB, configurable
  - You can split it into part cache and part SRAM.
  - Streaming Engine bypasses L1D for SE data
- L2 Unified Memory: 512 KB - 1 MB, configurable
  - Unified: holds both code and data
  - Configured as all-SRAM, all-cache, or split
- Multicore Shared Memory (MSMC/SRAM): 2-8 MB shared
- External DDR: LPDDR4 or DDR4 via EMIF
- DMA: UDMA (Unified DMA) for background data movement

Memory Protection & Virtual Memory

- CorePac Memory Management Unit (CMMU):
  - Virtual-to-physical address translation
  - Memory protection for multi-process environments
  - Supports Linux and RTOS memory management
- Memory Protection Unit (MPU):
  - Region-based access permissions (read/write/execute)
  - Privilege levels: supervisor and user modes
- Firewall protection:
  - SoC-level access control between different processors
  - Prevents unauthorized access from other cores
- Cache coherency: managed by hardware + software protocols
- Important for ADAS functional safety (ASIL-D) requirements

Part VI

C7x ISA Variants

C7x Core Variants

Variant	Vector Width	MMA	Primary SoCs
C7100	512-bit	Yes	TDA4VM, DRA829J
C7120	512-bit	Yes (enhanced)	TDA4VH, AM69A
C7504	256-bit	Yes	AM62A, AM62D
C7524	256-bit	Yes	AM275x (audio)

- C7100/C7120: high-performance variants (512-bit vectors)
  - Up to 80 GFLOPS DSP + 8-32 TOPS AI (with MMA)
- C7504/C7524: cost-optimized variants (256-bit vectors)
  - Up to 40 GFLOPS DSP + 2 TOPS AI
- All share the same ISA and compiler tool chain (cl7x)

C7100 - High Performance Core

- 512-bit vector width, 13 functional units
- First C7x variant, used in Jacinto 7 TDA4VM (2019)
- CPU clock: up to 1.0 GHz
- DSP performance: up to 80 GFLOPS (FP32)
- MMA performance: up to 8 TOPS (INT8)
- L1P: 32 KB (cache), L1D: 48 KB, L2: 512 KB
- Two Streaming Engines (SE0, SE1)
- Streaming Address Generator (basic support)
- Key SoCs: TDA4VM, TDA4VE, DRA829J
- Applications: ADAS front camera, surround view, parking assist

C7120 - Enhanced Performance Core

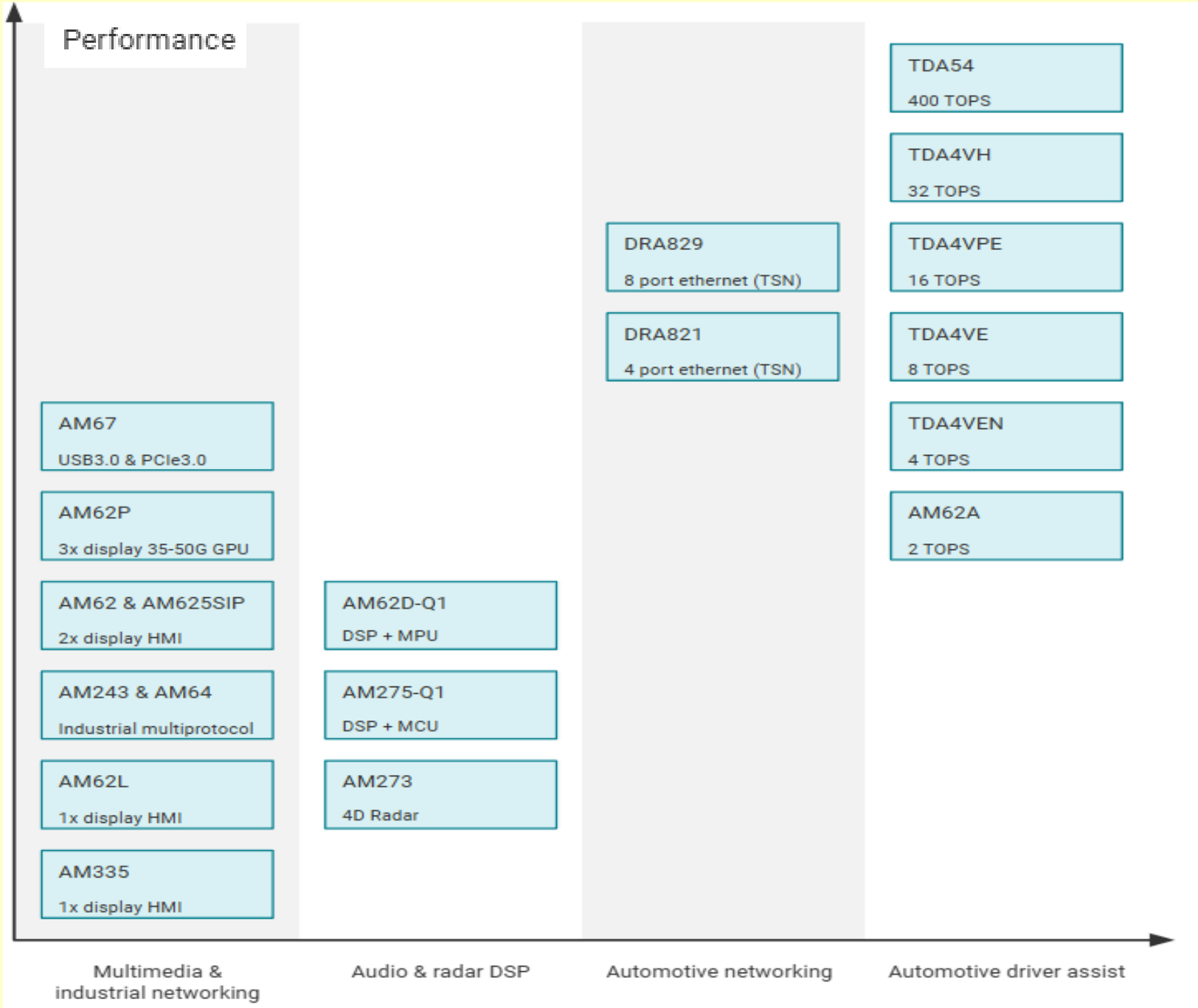
- 512-bit vector width, same architecture as C7100
- Enhanced MMA with FP16 support for better AI precision
- Higher MMA throughput: up to 32 TOPS (INT8)
- Multiple C7120 instances possible in a single SoC
  - AM69A: 4x C7120 cores + 4x C66x cores
- Improved power efficiency over C7100
- Key SoCs: TDA4VH (automotive), AM69A (industrial)
  - AM69A: 8x Cortex-A72 + 4x C7120 + 4x C66x
- Applications: L2+ autonomous driving, multi-camera fusion
- Supports higher resolution: 8MP cameras, 4-8 simultaneous sensors

C7504 / C7524 - Cost-Optimized Cores

- 256-bit vector width (half of C7100/C7120)
- Reduced silicon area and power consumption
- C7504 (in AM62A, AM62D):
  - 40 GFLOPS vector compute, 2 TOPS MMA
  - Paired with Cortex-A53 (quad-core)
  - Target: smart cameras, surveillance, edge AI
- C7524 (in AM275x):
  - Optimized for audio signal processing
  - Paired with Cortex-R5F (quad-core, no Cortex-A)
  - No DDR required - 10.75 MB on-chip SRAM
  - Target: automotive audio (Dolby Atmos, ANC, AVAS)
- Same compiler toolchain: use -mv7504 or -mv7524 flag

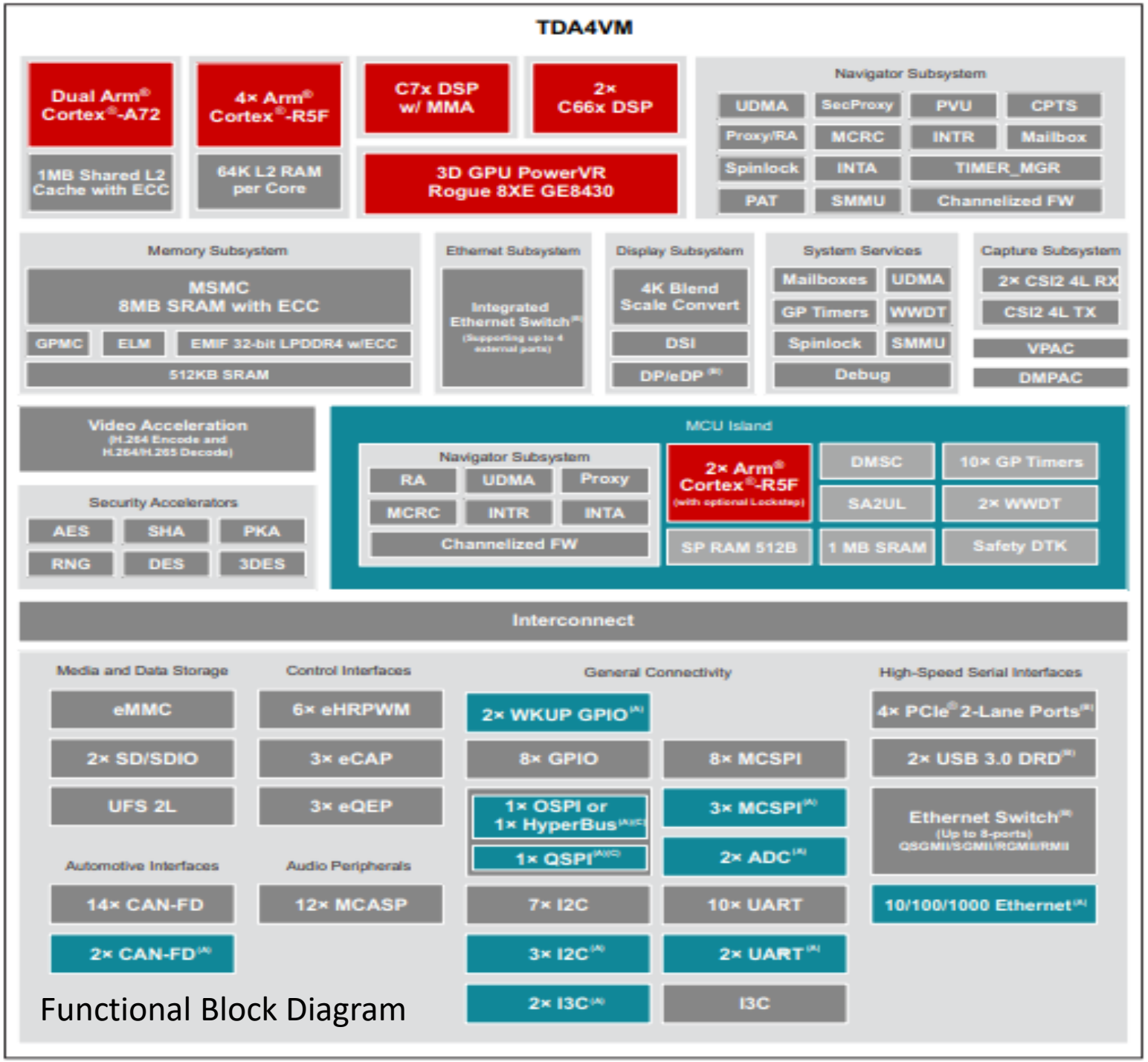
Part VII

SoC Family Members


Performance
TDA54
400 TOPS
TDA4VH
32 TOPS
DRA829
8 port ethernet (TSN)
TDA4VPE
16 TOPS
DRA821
4 port ethernet (TSN)
TDA4VE
8 TOPS
AM67
USB3.0 & PCIe3.0
TDA4VEN
4 TOPS
AM62P
3x display 35-50G GPU
AM62A
2 TOPS
AM62 & AM625SIP
2x display HMI
AM62D-Q1
DSP + MPU
AM243 & AM64
Industrial multiprotocol
AM275-Q1
DSP + MCU
AM62L
1x display HMI
AM273
4D Radar
AM335
1x display HMI
Multimedia & industrial networking
Audio & radar DSP
Automotive networking
Automotive driver assist

TI Jacinto 7 Platform - Automotive Grade

- Jacinto 7 is TI's automotive processor platform with C7x
- TDA4VM (flagship):
  - 2x Cortex-A72 + 6x Cortex-R5F + 1x C7100 + C66x
  - 8 TOPS AI, GPU (100 GFLOPS), VPAC with ISP
  - 5-20W power envelope
- TDA4VE (value):
  - Same architecture, reduced peripherals
- TDA4VH (high-end):
  - Multiple C7120 cores, higher AI performance (32 TOPS)
  - For L2+ autonomous driving applications
- All Jacinto 7 support ASIL-D functional safety
- Process: 16 nm FinFET



Functional Block Diagram

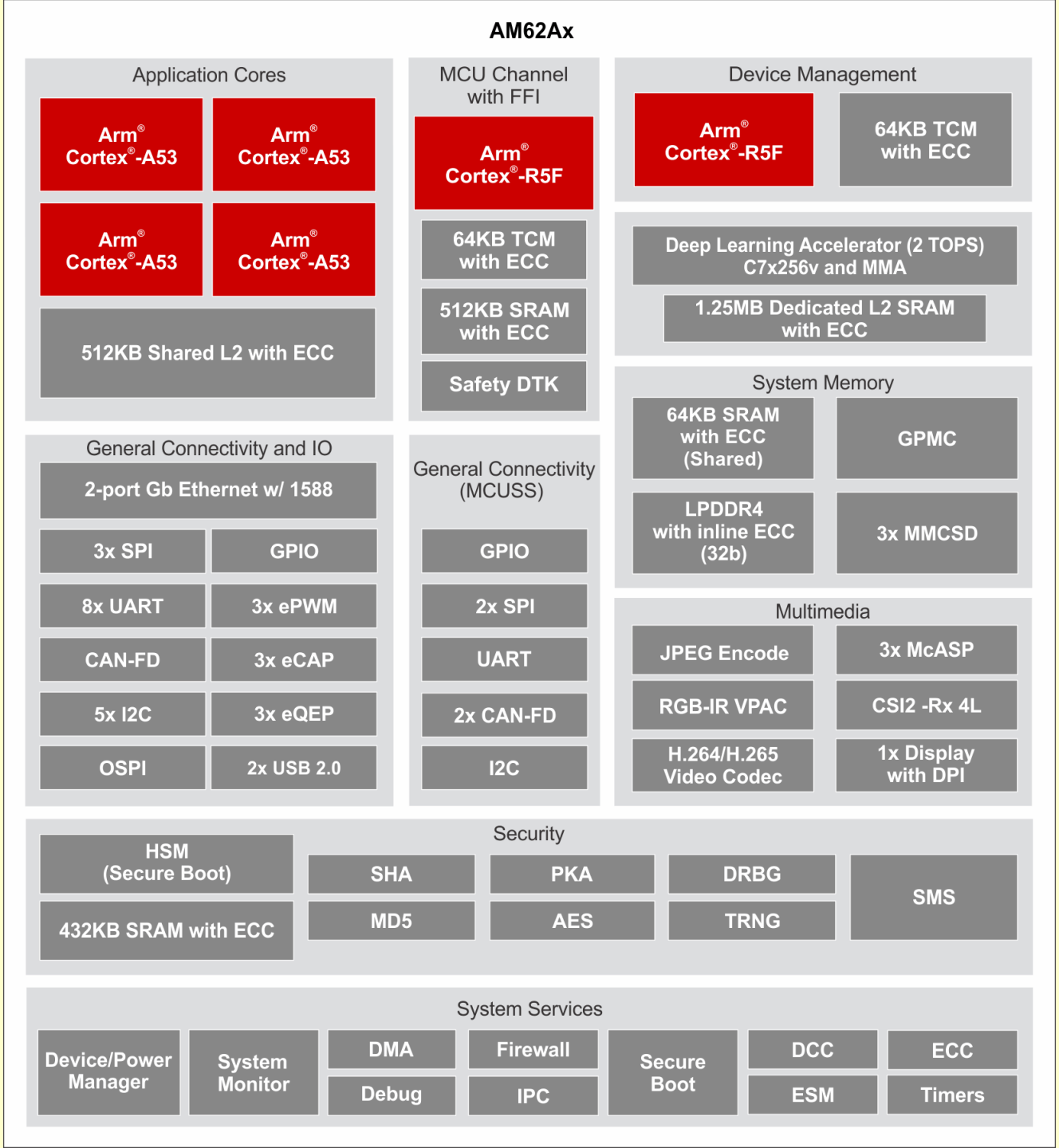
TDA4VM - Detailed Block Diagram

- Application cores: 2x ARM Cortex-A72 (up to 2 GHz)
- MCU Island: 2x ARM Cortex-R5F (isolated, safety)
- DSP cores: 1x C7x (C7100) + 1x C66x
- MMA: 8 TOPS deep learning accelerator (INT8)
- Vision Processing Accelerators (VPAC):
  - 7th-gen ISP (Image Signal Processor)
  - Lens Distortion Correction (LDC)
  - Motion & Stereo Disparity Engine (DMPAC)
- GPU: PowerVR 8XE GE8430 (100 GFLOPS)
- Video codecs: H.264/H.265 encode and decode
- I/O: CSI-2 (camera), PCIe Gen3, GbE switch, CAN-FD
- Security: hardware crypto, secure boot, HSM

AM62A / AM68A / AM69A - Vision & AI SoCs

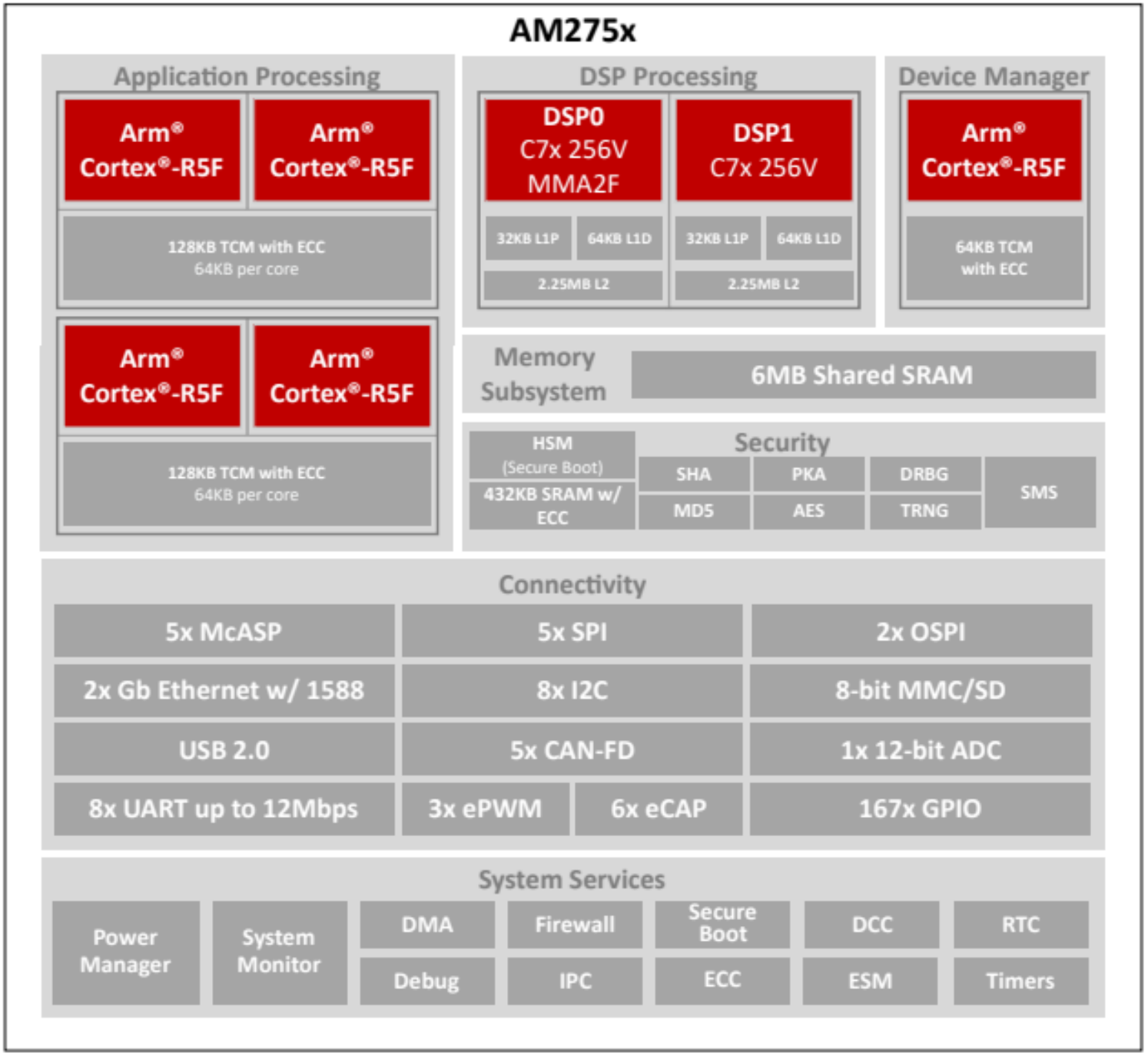
Feature	AM62A	AM68A	AM69A
App CPU	4x A53	2x A72	8x A72
C7x Cores	1x C7504	1x C7120	4x C7120
AI TOPS	2	8	32
C66x Cores	-	1	4
GPU	-	Yes	Yes
ISP	RGB-IR	Yes	Yes
Target	Smart camera	Machine vision	Multi-sensor AI
Price	$12+	$50+	$100+

Note: AM68A/AM69A are non-automotive equivalents of Jacinto TDA4x family


AM62Ax
Application Cores
Arm® Cortex®-A53
Arm® Cortex®-A53
Arm® Cortex®-A53
Arm® Cortex®-A53
512KB Shared L2 with ECC
MCU Channel with FFI
Arm® Cortex®-R5F
64KB TCM with ECC
512KB SRAM with ECC
Safety DTK
Device Management
Arm® Cortex®-R5F
64KB TCM with ECC
Deep Learning Accelerator (2 TOPS) C7x256v and MMA
1.25MB Dedicated L2 SRAM with ECC
System Memory
64KB SRAM with ECC (Shared)
GPMC
LPDDR4 with inline ECC (32b)
3x MMCSD
General Connectivity and IO
2-port Gb Ethernet w/ 1588
3x SPI
GPIO
8x UART
3x ePWM
CAN-FD
3x eCAP
5x I2C
3x eQEP
OSPI
2x USB 2.0
General Connectivity (MCUSS)
GPIO
2x SPI
UART
2x CAN-FD
I2C
Multimedia
JPEG Encode
3x McASP
RGB-IR VPAC
CSI2 -Rx 4L
H.264/H.265 Video Codec
1x Display with DPI
Security
HSM (Secure Boot)
432KB SRAM with ECC
SHA
MD5
PKA
AES
DRBG
TRNG
SMS
System Services
Device/Power Manager
System Monitor
DMA
Debug
Firewall
IPC
Secure Boot
DCC
ESM
ECC
Timers

AM275x - Automotive Audio DSP SoC

- Newest C7x-based SoC family (2025)
- Architecture: ARM Cortex-R5F + C7x DSP (C7524)
  - No Cortex-A core - MCU-class device, no Linux
  - Up to 2x C7x DSP cores (AM2754)
- Performance: up to 80 GFLOPS (dual C7x, 40 GFLOPS each)
- Memory: 10.75 MB on-chip SRAM - no external DDR needed!
  - Reduces BOM cost and simplifies PCB design
- Audio features: McASP (multi-channel audio serial port)
- Networking: 2-port Gigabit Ethernet with AVB/TSN support
- Security: integrated Hardware Security Module (HSM)
- Automotive qualified: -40 to +125 deg C
- Replaces 4+ chips with single SoC for audio amplifier systems


AM275x
Application Processing
Arm® Cortex®-R5F
Arm® Cortex®-R5F
128KB TCM with ECC
64KB per core
Arm® Cortex®-R5F
Arm® Cortex®-R5F
128KB TCM with ECC
64KB per core
DSP Processing
DSP0 C7x 256V MMA2F
DSP1 C7x 256V
32KB L1P
64KB L1D
2.25MB L2
32KB L1P
64KB L1D
2.25MB L2
Device Manager
Arm® Cortex®-R5F
64KB TCM with ECC
Memory Subsystem
6MB Shared SRAM
HSM (Secure Boot)
432KB SRAM w/ ECC
Security
SHA
PKA
DRBG
SMS
MD5
AES
TRNG
Connectivity
5x McASP
5x SPI
2x OSPI
2x Gb Ethernet w/ 1588
8x I2C
8-bit MMC/SD
USB 2.0
5x CAN-FD
1x 12-bit ADC
8x UART up to 12Mbps
3x ePWM
6x eCAP
167x GPIO
System Services
Power Manager
System Monitor
DMA
Firewall
Secure Boot
DCC
RTC
Debug
IPC
ECC
ESM
Timers

AM62D - Audio Processor with Cortex-A53

- Similar to AM275x but with Cortex-A53 application processor
- Architecture: Cortex-A53 + Cortex-R5F + C7x (C7504) + MMA
- Supports Linux OS for rich application development
- Audio capabilities:
  - Dolby Atmos spatial audio processing
  - Active Noise Cancellation (ANC)
  - Acoustic Vehicle Alert System (AVAS)
  - Sound synthesis and 3D audio rendering
- Networking: Ethernet Audio Video Bridging (eAVB), Dante
- 40 GFLOPS DSP + 2 TOPS AI (MMA)
- LPDDR4 support for larger audio models and data buffers
- Target: mid-to-high-end automotive audio systems

Radar Processors with C7x DSP

- TI radar SoCs integrate C7x for signal processing:
- AWRL6844 (2025): 60 GHz mmWave radar with C7x
  - In-cabin sensing: occupant detection, gesture recognition
  - Single-chip: radar transceiver + C7x DSP + ARM
  - Meets Euro NCAP 2025 requirements
- C7x in radar processing chain:
  - FFT processing on raw ADC data
  - CFAR detection, angle estimation
  - Point cloud generation
  - AI-based target classification via MMA
- Super-resolution algorithms using C7x + MMA
- Traditional radar processing on C7x DSP, AI on MMA

Part VIII

Applications

Application: ADAS & Autonomous Driving

- Primary application for C7x: Advanced Driver Assistance Systems
- Front camera ADAS (TDA4VM, 5-7W):
  - Lane departure warning, traffic sign recognition
  - Pedestrian detection, forward collision warning
  - Handles up to 8MP camera at 30fps
- Surround view & parking (TDA4VM, up to 20W):
  - 4-6 cameras (3MP each) simultaneously
  - Automated valet parking (AVP)
- L2+ autonomous driving (TDA4VH/AM69A):
  - Multi-sensor fusion: camera + radar + LiDAR + ultrasonic
  - 32 TOPS for complex DNN inference
- Amazon Proteus robot uses TDA4VM for autonomous navigation

Application: Industrial Robotics & Machine Vision

- AM68A and AM69A target industrial applications:
- Autonomous Mobile Robots (AMR):
  - Real-time obstacle detection and path planning
  - CNN-based object detection on C7x/MMA
  - ROS 2 support via TI Robotics SDK
- Machine vision inspection:
  - Defect detection, measurement, classification
  - High-speed camera (GigE Vision) + AI inference
- Collaborative robots (cobots):
  - Safety-critical perception (ASIL-D capable)
  - Human pose estimation for safe interaction
- TI Robotics SDK: ROS 2 + optimized C7x/MMA libraries
  - Supports perception, mapping, localization blocks

Application: Automotive Audio

- AM275x and AM62D *revolutionize in-vehicle audio*:
- C7x DSP delivers 4x the performance of traditional audio DSPs
- Supported audio features:
  - Dolby Atmos spatial audio rendering
  - Active Noise Cancellation (ANC) - road/engine noise
  - Acoustic Vehicle Alert System (AVAS) for EVs
  - Sound synthesis and virtual engine sounds
  - Voice enhancement and echo cancellation
  - Ethernet AVB audio networking
- Single-chip replaces multi-chip audio DSP solutions
- AI-enhanced audio: noise classification, scene detection via MMA
- Dolby partnership: "TI's C7x DSP enables Dolby Atmos in any car"

Application: Smart Cameras & Surveillance

- AM62A targets cost-sensitive edge AI cameras:
- Features: 1-2 TOPS AI, ISP with RGB-IR support
- Typical use cases:
  - Video surveillance with real-time analytics
  - Retail analytics: people counting, queue detection
  - Smart doorbell cameras
  - Industrial monitoring and inspection
- C7x processes both traditional CV and DNN inference
- Low power: operates within 2-5W envelope
- On-device inference: privacy-preserving (no cloud needed)
- Linux support: GStreamer + OpenCV + TI Edge AI stack
- Starting at $12 per chip in volume

Part IX

Software Ecosystem & Development Tools

Software Development Ecosystem

- Code Composer Studio (CCS): TI's IDE for C7x development
- C7000 C/C++ Compiler (cl7x):
  - Supports C7100, C7120, C7504, C7524 via -mv7xxx flag
  - Automatic vectorization and software pipelining
  - Automatic Streaming Engine usage (--auto_stream)
- Processor SDK Linux: complete Linux BSP for Cortex-A + C7x
- Processor SDK RTOS: FreeRTOS/SafeRTOS for real-time on C7x
- TI Deep Learning (TIDL): optimized DNN inference on C7x + MMA
  - Supports TFLite, ONNX Runtime, TVM
- TI Edge AI Studio: cloud-based model benchmarking
- TISP library: signal processing nodes (audio, DSP, FFT, math)

C7x Optimized Libraries

- **DSPLIB:**
  - FFT (real/complex)
  - FIR/IIR filters
  - Convolution
  - Correlation
- **MATHLIB:**
  - Trigonometry
  - Exponential/Log
  - Division, sqrt
  - 15 math nodes
- **AUDIOLIB:**
  - 31 audio nodes
  - Parametric EQ
  - Dynamic range
  - Spatial audio
- **TIDL (Deep Learning):**
  - CNN inference
  - Object detection
  - Segmentation
  - Pose estimation

Development Boards & Getting Started

- **Starter Kits (SK) from TI:**
  - SK-TDA4VM: Jacinto 7 starter kit for ADAS prototyping
  - SK-AM62A-LP: low-power vision AI starter kit
  - SK-AM68: machine vision and general AI processing
  - SK-AM69: high-performance multi-sensor AI
- **Evaluation Modules (EVM):**
  - J721EXSOMXEVM: TDA4VM System-on-Module
  - AM275x EVM: automotive audio development
- **Software entry points:**
  - TI Edge AI Studio (cloud): model benchmarking in minutes
  - Processor SDK: full development environment
  - Robotics SDK: ROS 2 ready-to-run demos
- **Pricing: SK boards from ~$99 USD**

Summary - TI C7000 DSP Family

- C7000 is TI's latest VLIW DSP - evolution of C6000 heritage
- Key innovations: 512/256-bit vectors, 13 FUs, Streaming Engine, MMA
- Always embedded in heterogeneous SoCs (not standalone)
- Four ISA variants: C7100, C7120, C7504, C7524
- Targets multiple markets:
  - Automotive ADAS (Jacinto 7: TDA4VM/VH/VE)
  - Vision AI & robotics (AM62A, AM68A, AM69A)
  - Automotive audio (AM275x, AM62D)
  - Radar processing (AWRL6844)
- Performance: up to 80 GFLOPS (DSP) + 32 TOPS (AI)
- Software: CCS, TIDL, Processor SDK, Robotics SDK, TISP
- C7x VLIW principles are the same as C6000 - a great learning path

References

- TI C7000 C/C++ Optimization Guide (online, 2024)
- C71x DSP CPU, ISA, and MMA Technical Reference Manual (SPRUIP0)
- TDA4VM Datasheet (TI, SPRSP18)
- AM275x Datasheet (TI, 2025)
- AM62A/AM68A/AM69A Datasheets (TI)
- TI Robotics SDK Documentation (software-dl.ti.com)
- TI Edge AI Studio (ti.com/edgeai)
- TI C7000-CGT Compiler Release Notes (v5.0.x)
- TISP Library - github.com/TexasInstruments/C7x_tisp
- CES 2025: TI Automotive Audio Press Release