C2

OUTLINE:

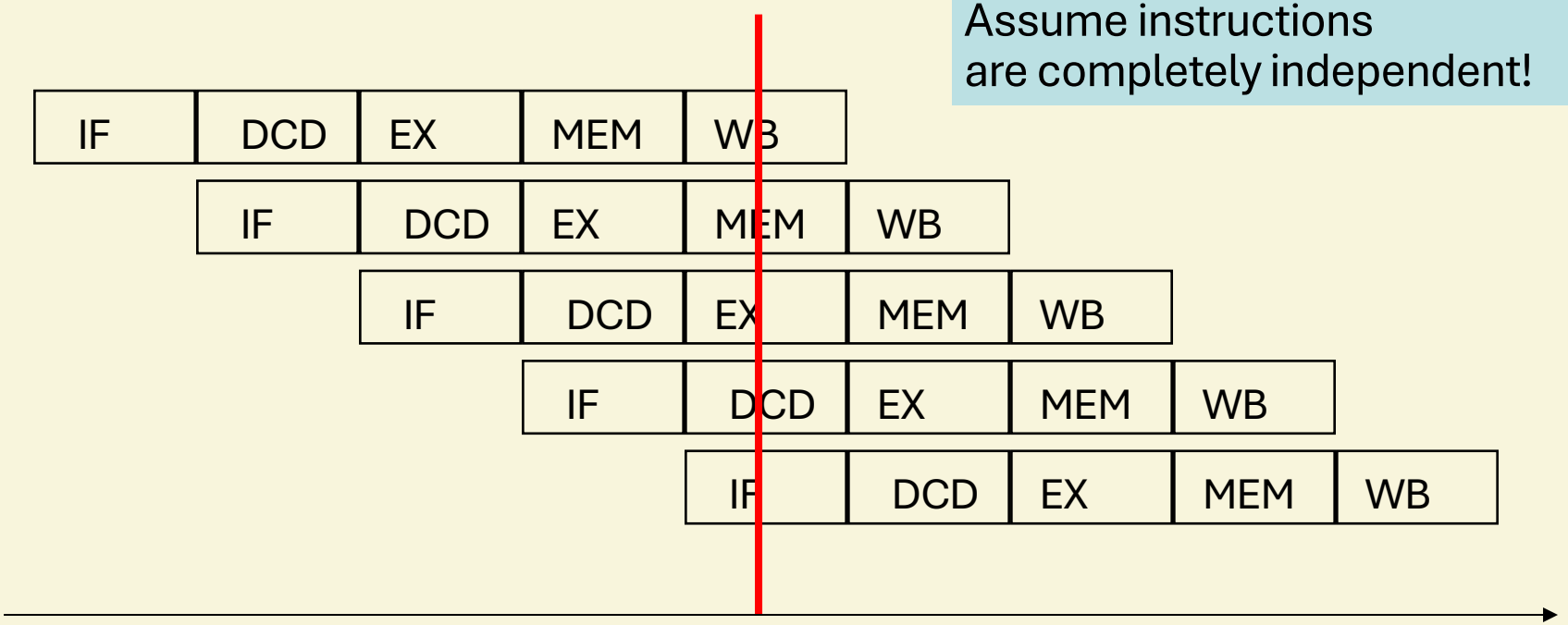
1. The concept of parallelism as it applies to the processing of data. Amdal law.
2. The Current DSP Architectures
3. Alternatives to DSP processors
4. DSP Benchmarking – SPEC /BDTI
5. TI DSP families
6. Mainstream DSP processor Families

1. Parallelism in data processing

- Most high-performance computers exhibit a great deal of concurrency
- However, it is not desirable to call every modern computer a parallel computer
- Pipelining and parallelism are 2 methods used to achieve concurrency
  - Pipelining increases concurrency by dividing a computation into a nr of steps
  - Parallelism is the use of multiple resources to increase concurrency

Ideal Pipelining

Assume instructions
are completely independent!



Maximum Speedup ≤ Number of stages

$$\text{Speedup} \leq \frac{\text{Time for un-pipelined operation}}{\text{Time for longest stage}}$$

Parallelism

- How to increase the number of operations that can be performed by each instruction?
  - Add execution units (multiplier, adder etc.)
    - Enhance the instr. set to use the additional hardware
    - Increase the instr. width.
    - Use wider buses to keep μP fed with data
  - Add SIMD (Single Instruction Multiple Data) capabilities

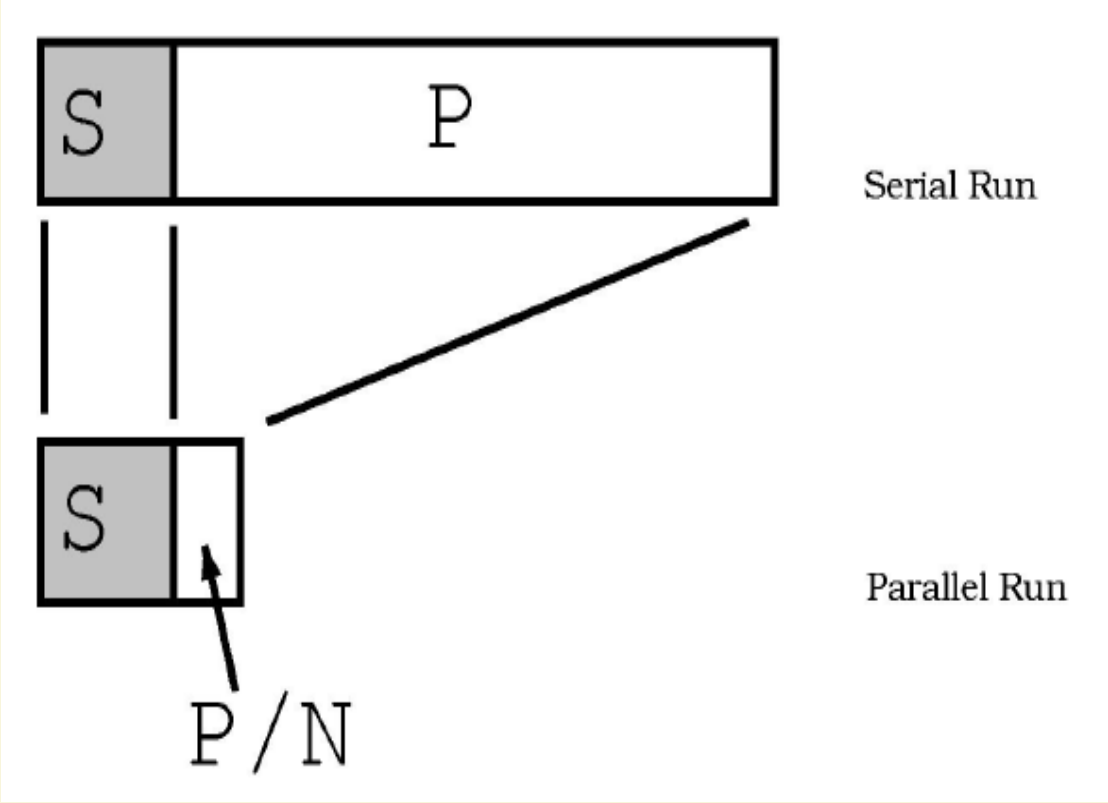
Parallelism

= More instructions / clock cycle

- How to increase the number of instrs. that are issued and executed in every clock cycle?
  - Use VLIW techniques
  - Use superscalar techniques
- *VLIW & super-scalar architectures* typically use simple, RISC-like based instr. rather than the complex, compound instr. traditionally used in DSP processors
- Why not use a super-scalar processor architecture ?
  - Super-scalar is not predictable
  - Very Long Instruction Words (VLIW) is !

Amdahl's Law (1967 – IBM)

$$Speedup \leq \frac{S+P}{S+P/N}$$



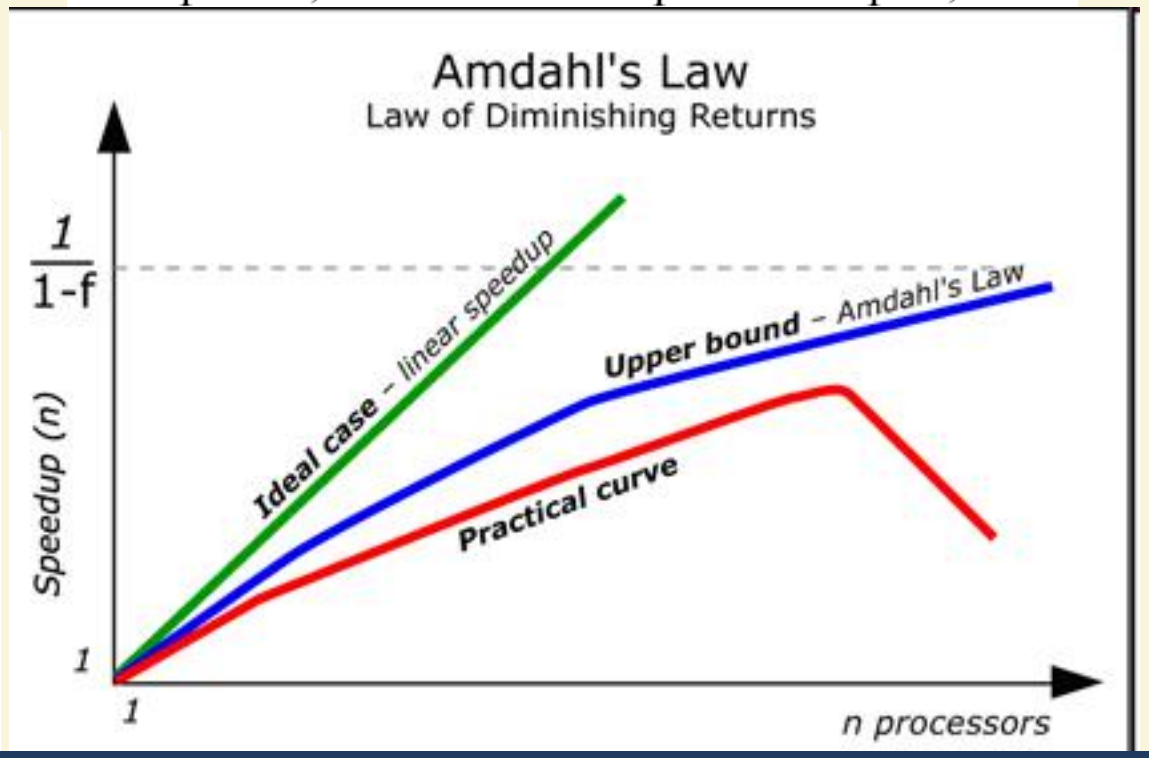
- S (%) is the percentage of instructions sequential in nature,
- P=1-S(%) is the percentage of parallelizable instructions, and
- N is the number of processors used in a parallel calculation

Amdahl's Law (1967 – IBM)

$$
\begin{aligned}
Speedup &\leq \frac{S+1-S}{S+\frac{1-S}{N}} \\
&\leq \frac{1}{S+\frac{1-S}{N}}. \\
Speedup &\leq \frac{1}{S+\frac{1-S}{N}} \\
&\leq \frac{1}{0+\frac{1-0}{N}} \\
&\leq \frac{1}{\frac{1}{N}} \\
&\leq N.
\end{aligned}
$$

- If we assume that the problem is infinitely parallelizable (S = 0), then we get the following upper-limit for speedup:

Ex. p=10 , 10% secv. 90% paral. → Sp=5,26



Lee's generalized Amdahl's Law

$$t_1 = \sum_{k=1}^{p} q_k t_1 \qquad t_p = \sum_{k=1}^{p} \frac{q_k t_1}{k}$$

$$q_k = \frac{1}{p}$$

$$S_p = \frac{t_1}{t_p} = \frac{\sum_{k=1}^{p} q_k t_1}{\sum_{k=1}^{p} \frac{q_k t_1}{k}}$$

$$S_p = \frac{\sum_{k=1}^{p} q_k}{\sum_{k=1}^{p} \frac{q_k}{k}} = \frac{1}{\sum_{k=1}^{p} \frac{q_k}{k}}$$

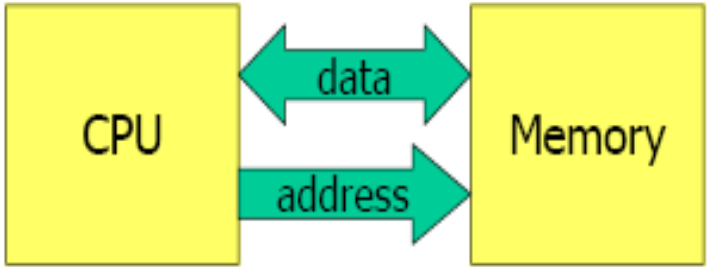
$$S_p = \frac{1}{\frac{1}{p}\sum_{k=1}^{p} \frac{1}{k}} = \frac{p}{\sum_{k=1}^{p} \frac{1}{k}}$$

$$S_p \leq \frac{p}{\log_2(p)}$$

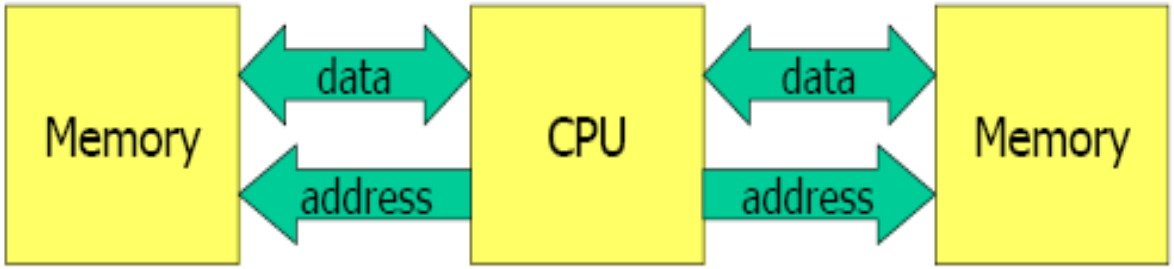
2. The Current DSP Architectures

Summary of DSP Hardware Implementations

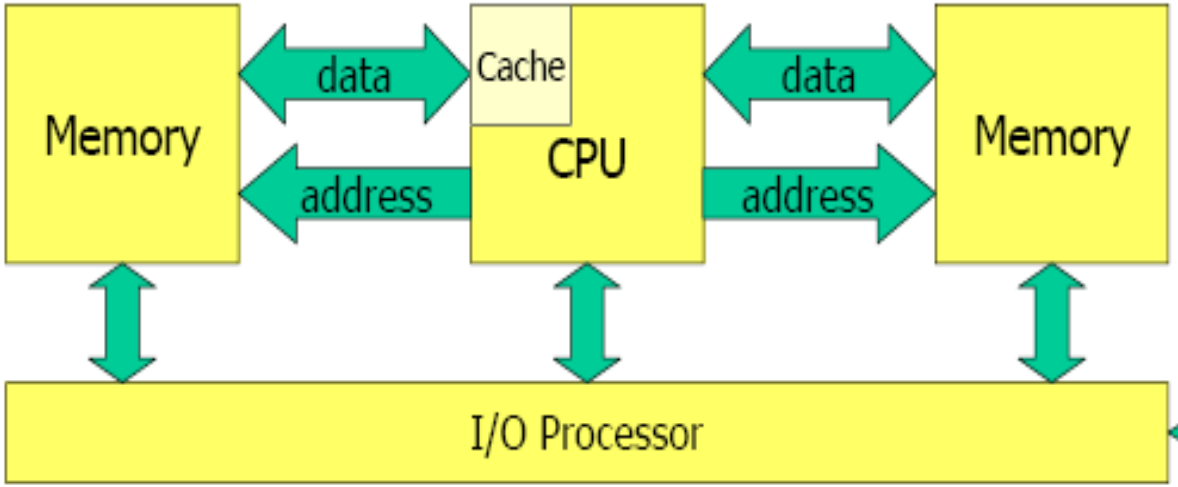
Property	ASIC	FPGA	µP/µC	DSP Processor	DSP+HW Accel.	GPU / NPU ★ NEW 2025
Flexibility	None	**High ↑** HLS tools	High	High	Medium	**Medium-High** CUDA/OpenCL
Design time	**Very Long ↓** 3nm cost++	**Short-Med ↑** via HLS	Short	Short	Short	**Short** PyTorch/CUDA
Power consumption	**Lowest ↑** 3nm/5nm nodes	Low-Medium	Medium-High	Low-Medium	Low-Medium	**High** 100-400W GPU
Performance	**Highest** task-specific	**Very High ↑** Versal AI Eng.	Low-Medium	Medium-High C7x: 2 TFLOPS	High	**Highest ↑** 1000+ TOPS NPU
Development cost	**Very High ↓** $100M+ 3nm	Medium	Low	Low	Low	**Low-Medium** APIs/frameworks
Production cost	**Low** high volume only	Medium	Low-Medium	Low-Medium	Medium	**High** $500-$10k/unit



Von Neumann
(single memory)



Harvard Architecture
(dual memory)

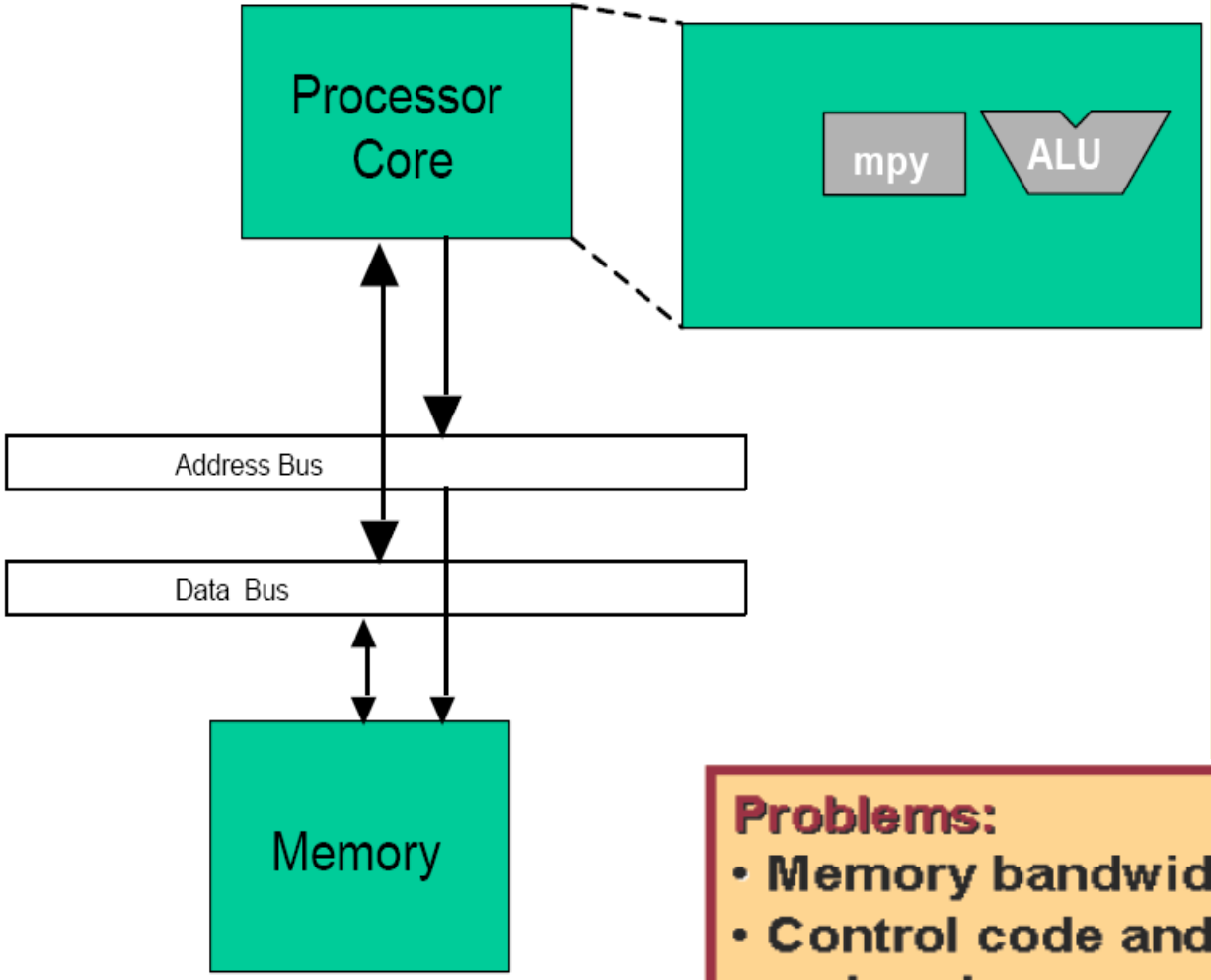


Super Harvard Architecture
(dual memory, cache,
I/O Processor)

Processor: SHARC

Von NEUMANN Machine (SISD)

- One memory space

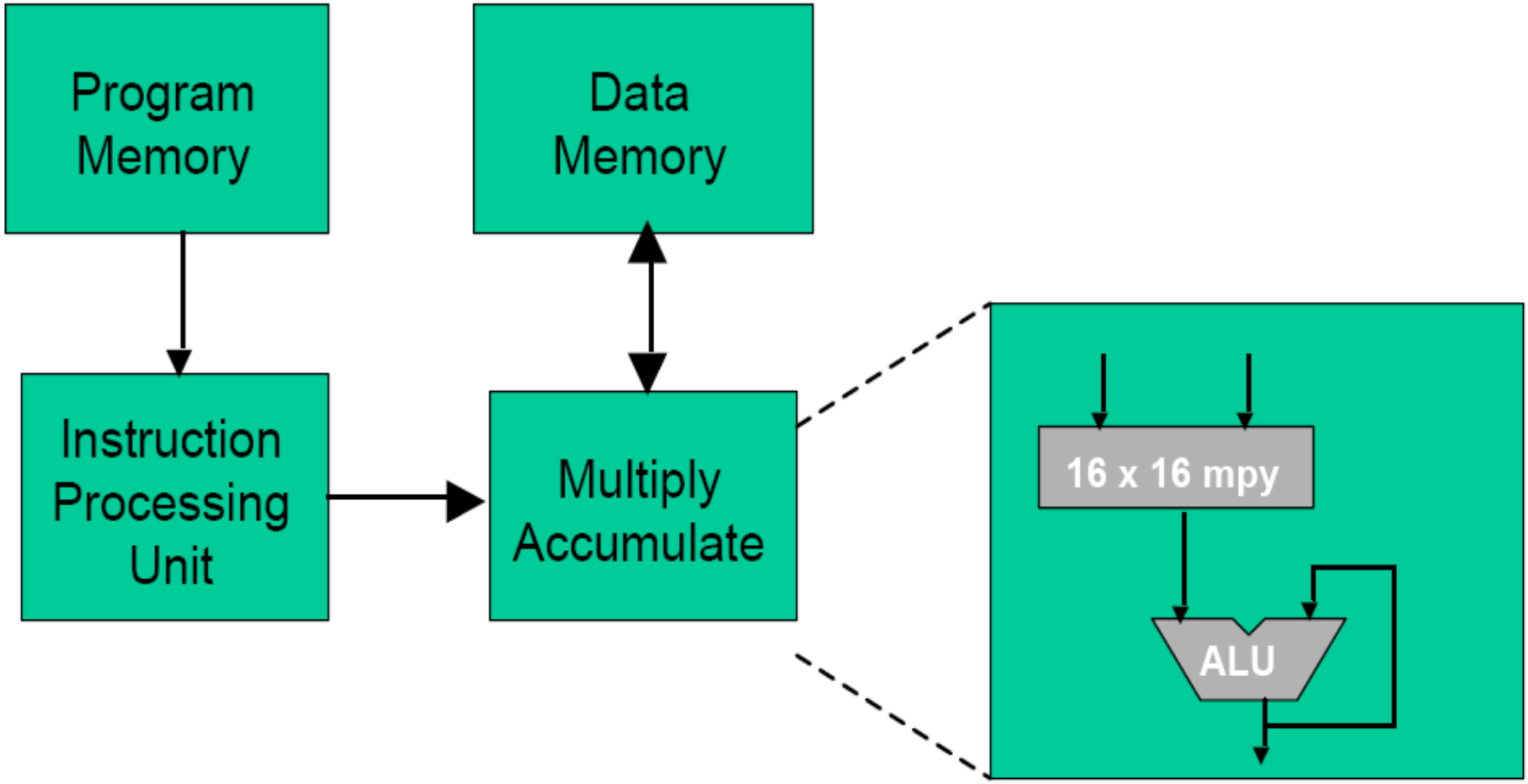


Problems:

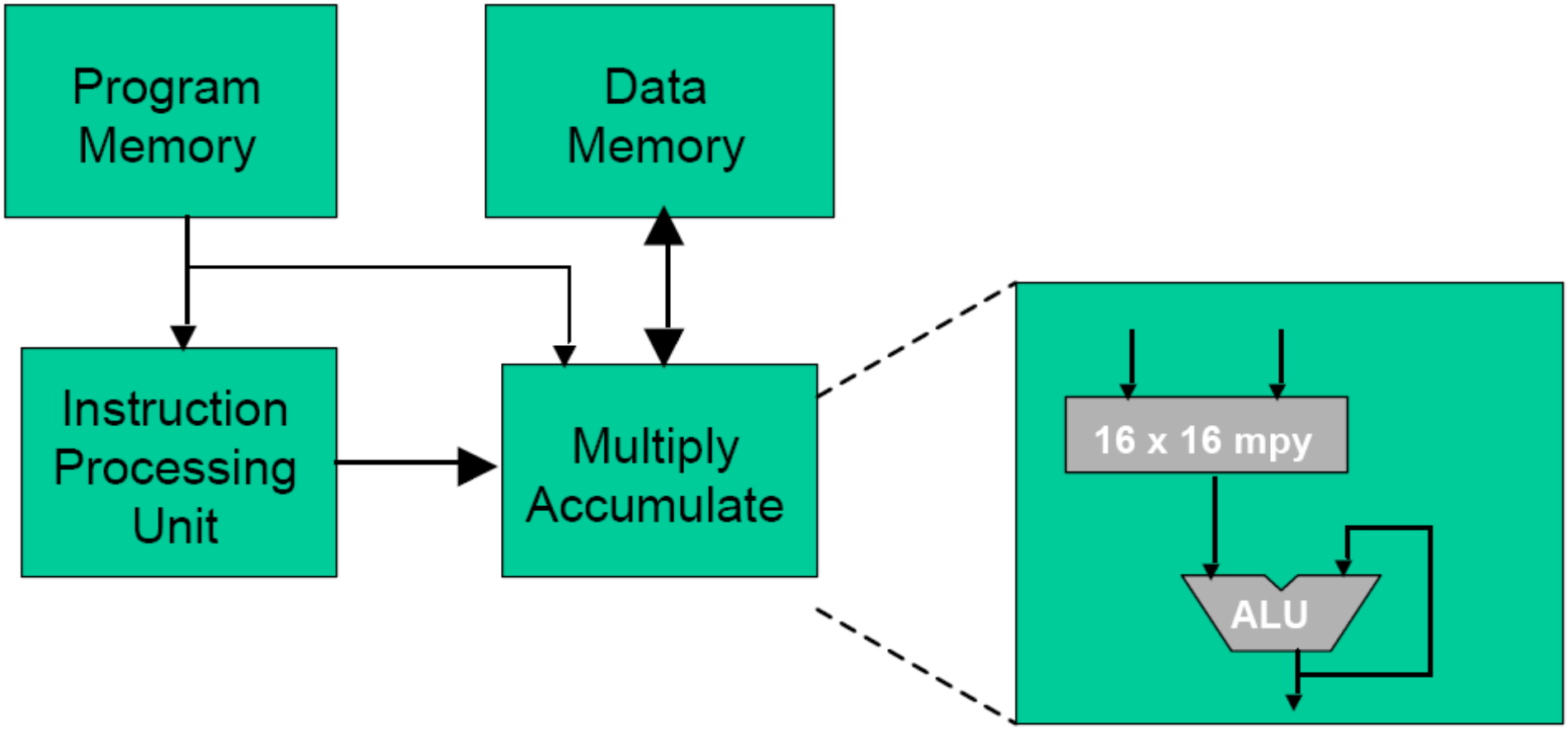
- Memory bandwidth bottleneck
- Control code and addressing overhead
- Possibly slow multiply

Basic Harvard Architecture

- Separate data memory from program memory!



Modified Harvard Architecture



Program bus to get instruction
Or to get coefficients (often stored in ROM)

ARCHITECTURE COMPARISON

Traditional vs. Modern DSPs

	TRADITIONAL DSP	MODERN DSP
Instructions	Highly compound	RISC or combination of RISC and compound
Multi-issue	N/A	Typically VLIW; a few superscalar
Issue Width	1	2 - 8
SIMD	Limited — e.g., only a dual-16-bit add	Extensive — e.g., 1×32, 2×16, or 4×8 for most arithmetic ops
Coprocessors / Accelerators	Rare	Communications & multimedia hardware common

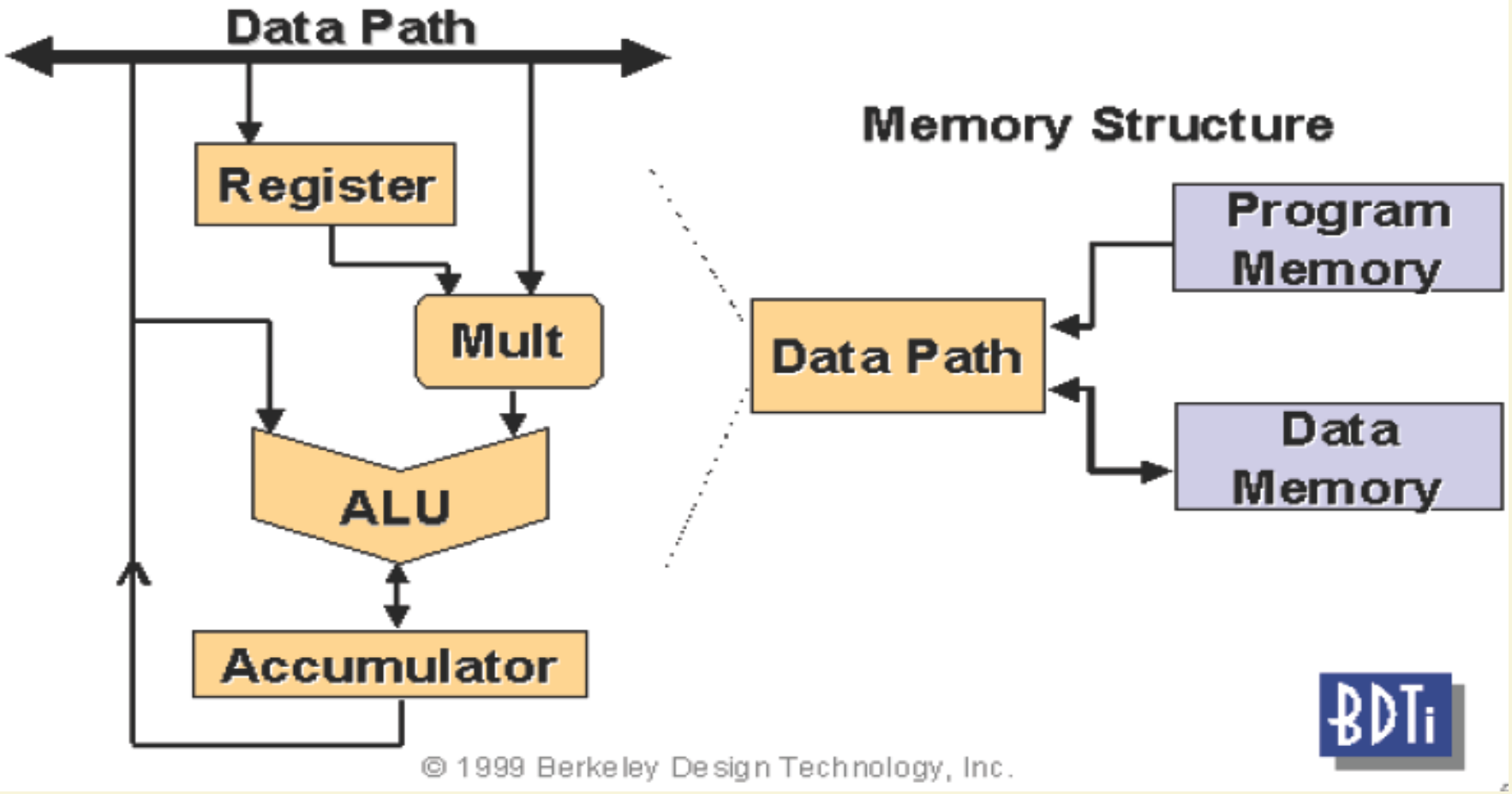
2.1 Conventional DSP Processors

- 1980s. architecture
- One inst. per cycle
- Include a single multiplier or MAC unit, an ALU, few exec. units
- 20-50Mhz operation frequency

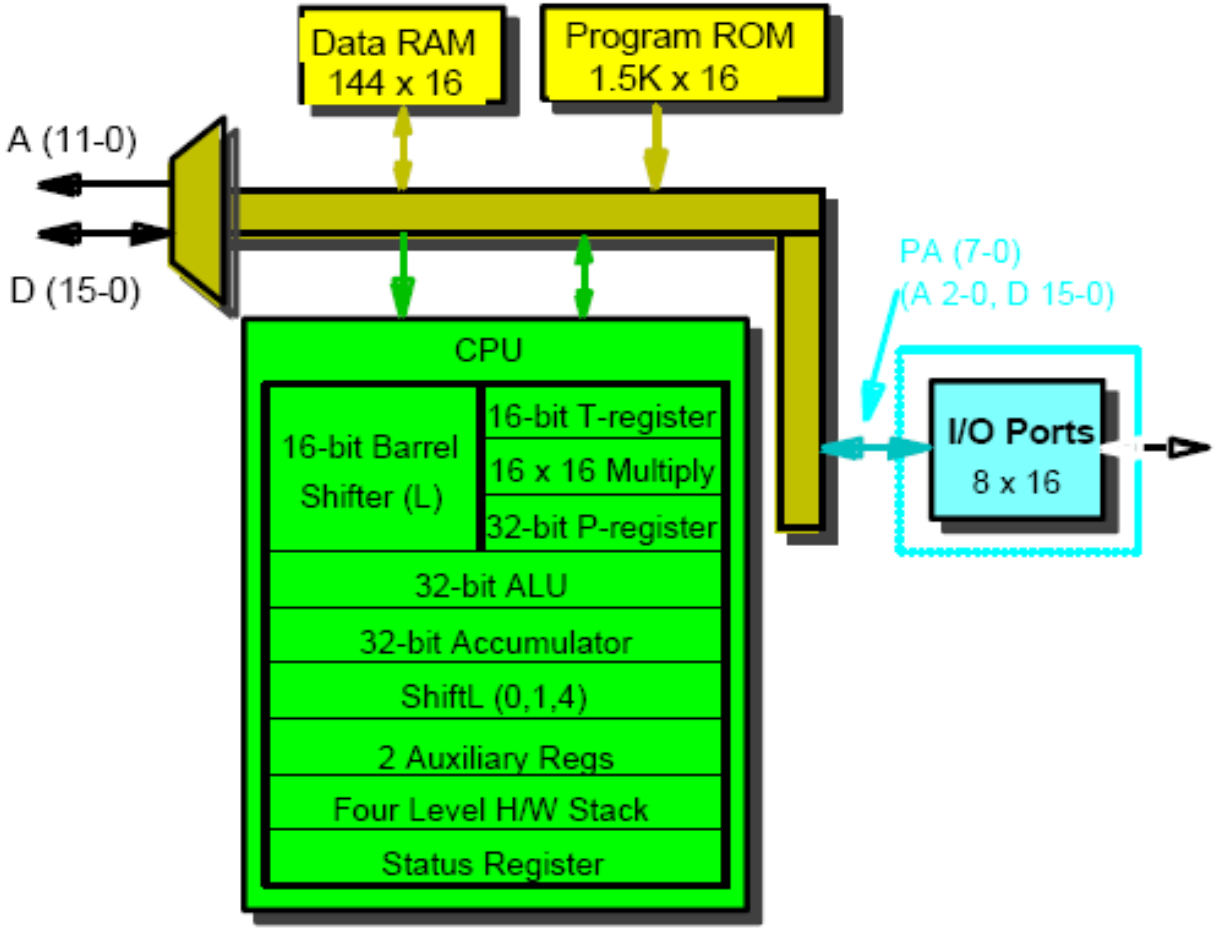
e.g.

- Analog Devices- ADSP-21xx family
- Texas Instruments- TMS320C1xx family
- Freescale (formerly Motorola)- DSP560xx family [Motorola DSP division became Freescale Semiconductor in 2004, acquired by NXP in 2015]

Early DSP Architecture



TMS320C10 (1982)

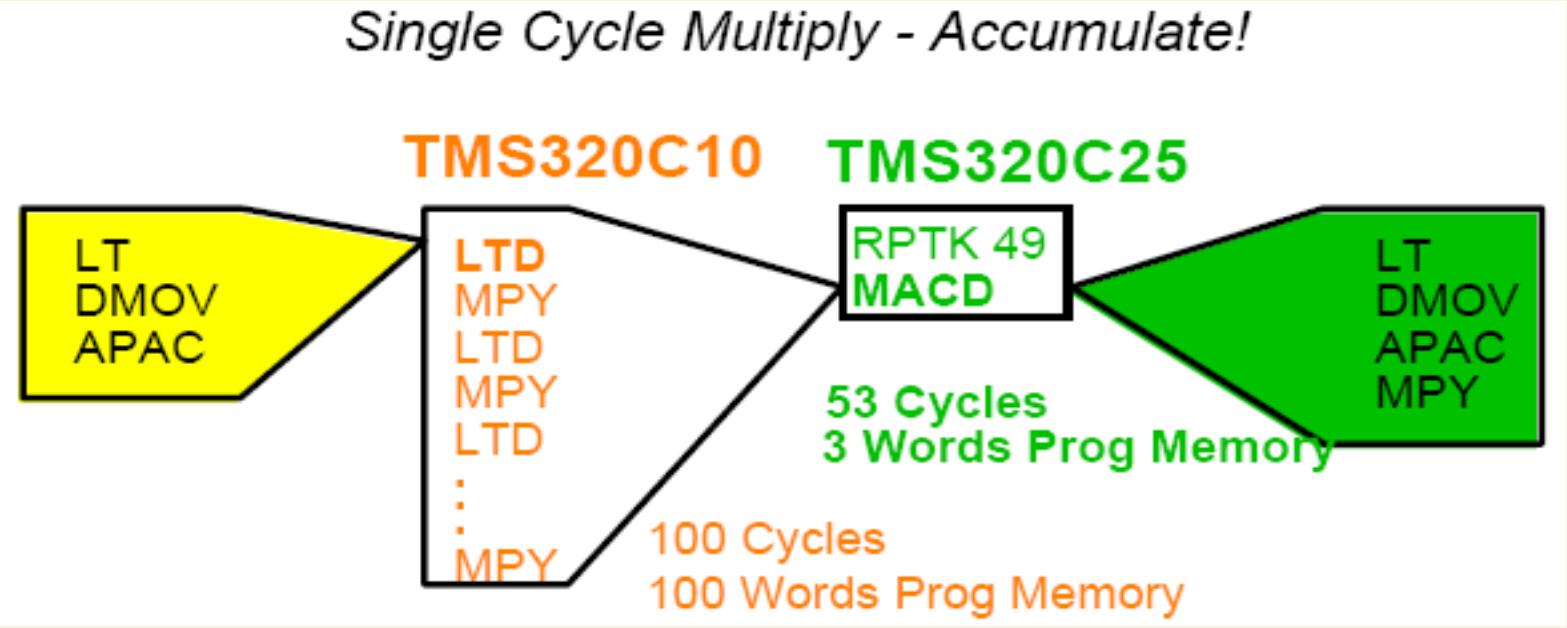


- **160/200ns Instruction cycle time**
- 4K word external address reach
- **60 general purpose and DSP specific instructions**
- **Single cycle multiply**
- 16-bit Barrel Shifter
- External interrupt and polled input pins
- Eight 16-bit I/O ports
- 40-pin DIP/44-pin PLCC

Courtesy: Texas Instruments

Conventional DSP Processors

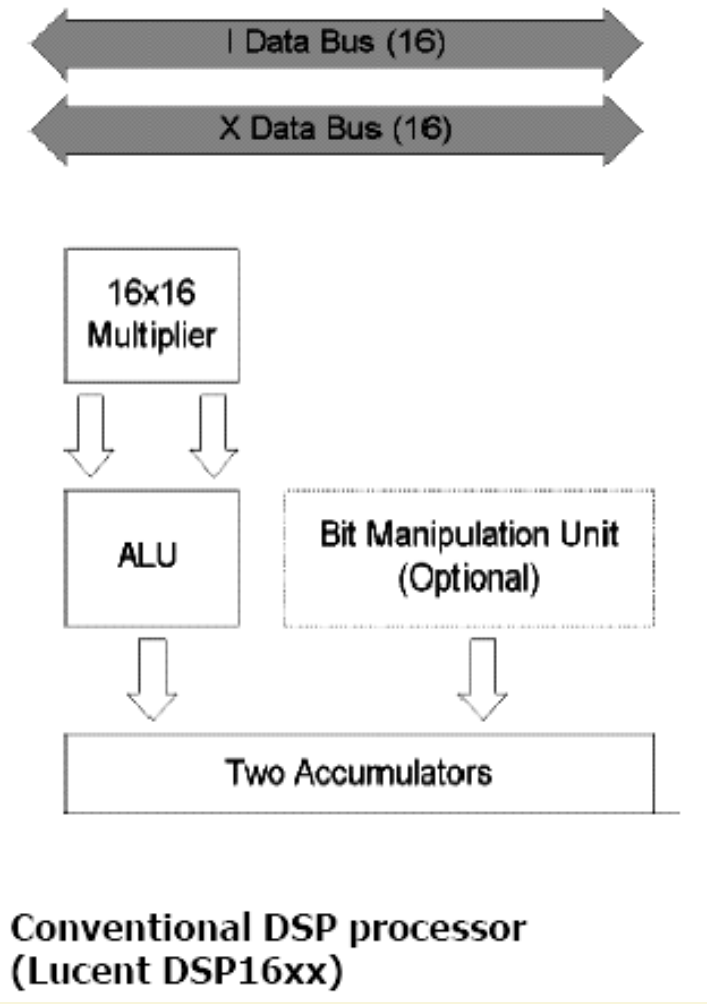
- ***Midrange Processors*:**
- Freescale/NXP DSP563xx (from Motorola), TI TMS320C25
- Increased clock speed (100-150MHz)
- Additional barrel shifter or instruction cache
- Deeper pipeline
- Low-cost, low-energy consuming
- Used in wireless telecommunications applications, high-speed modems



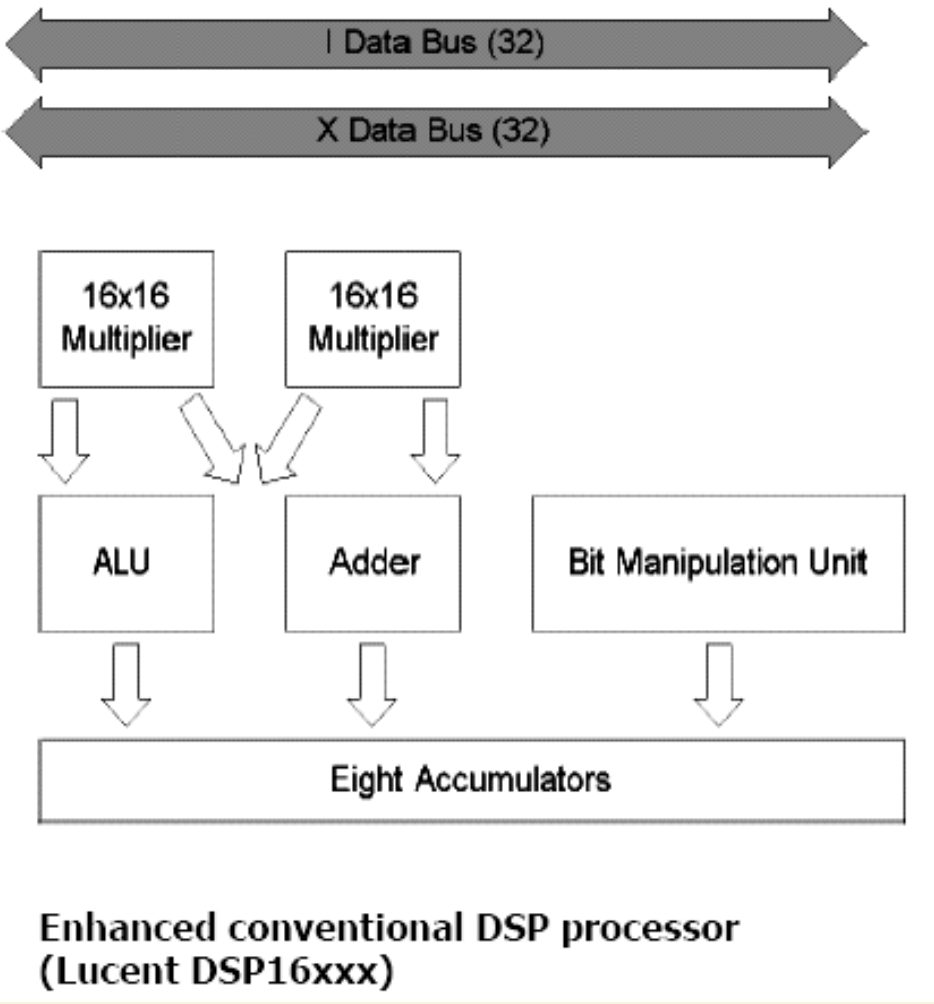
2.2 Enhanced DSP Processors

- *Parallel execution units:*
  - Second multiplier, adder units
  - SIMD capabilities
- *Extended instructions sets*
  - More operations in a single instruction
- *2 MACs per clock cycle*
- *Wider data buses* (more data words per clock cycle)
- *Co-processors*
  - Viterbi decoding, FIR filtering, etc.

2.2 Enhanced DSP Processors



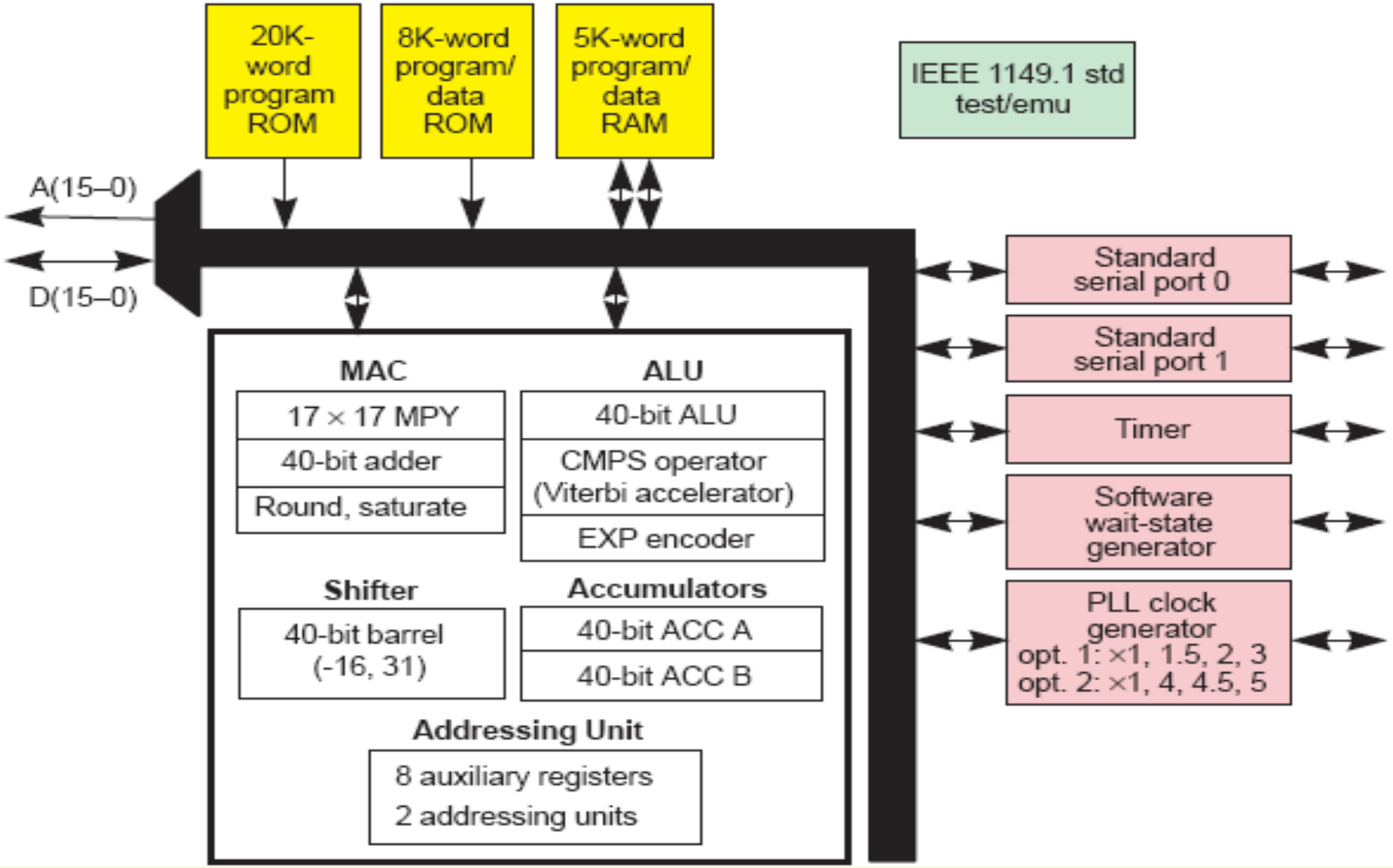
Conventional DSP processor (Lucent DSP16xx)



Enhanced conventional DSP processor (Lucent DSP16xxx)

- Figures compare the execution units and buses of a conventional DSP (the Agere Systems/formerly Lucent Technologies DSP16xx) to an enhanced-conventional DSP that extends the DSP16xx architecture (the Agere Systems DSP16xxx)

TMS320LC541 Block Diagram



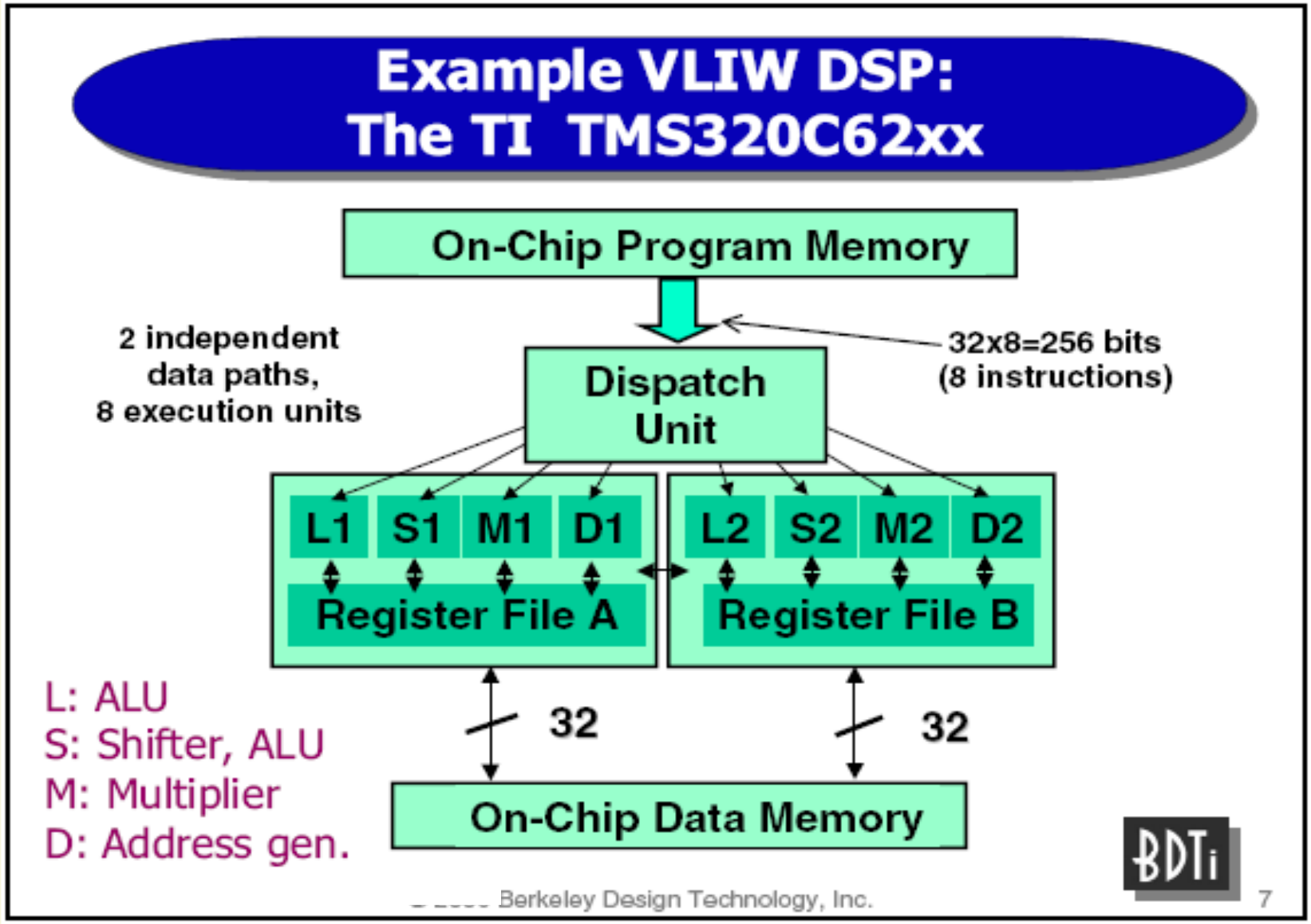
2.3 Multi Issue Architectures

- Goal: *Make programming easier!* (Previous ones were difficult in assembly)
- VLIW (very long instr. word) and super-scalar architectures execute (issue) multiple instructions in parallel.
- Use very simple instructions
- Issuing & executing instructions in parallel groups
- Using simple instructions:
  - Simpler instruction decoding and execution
- Now all major vendors (TI, Analog Devices, NXP/Freescale, Lucent/Agere) employ multi-issue architectures.
- All current multi-issue DSP processors use **VLIW**
- VLIW approach (TMS320C62xx/C64xx/C66x/C7x; MSC 140; Infineon Carmel – CLIW..)

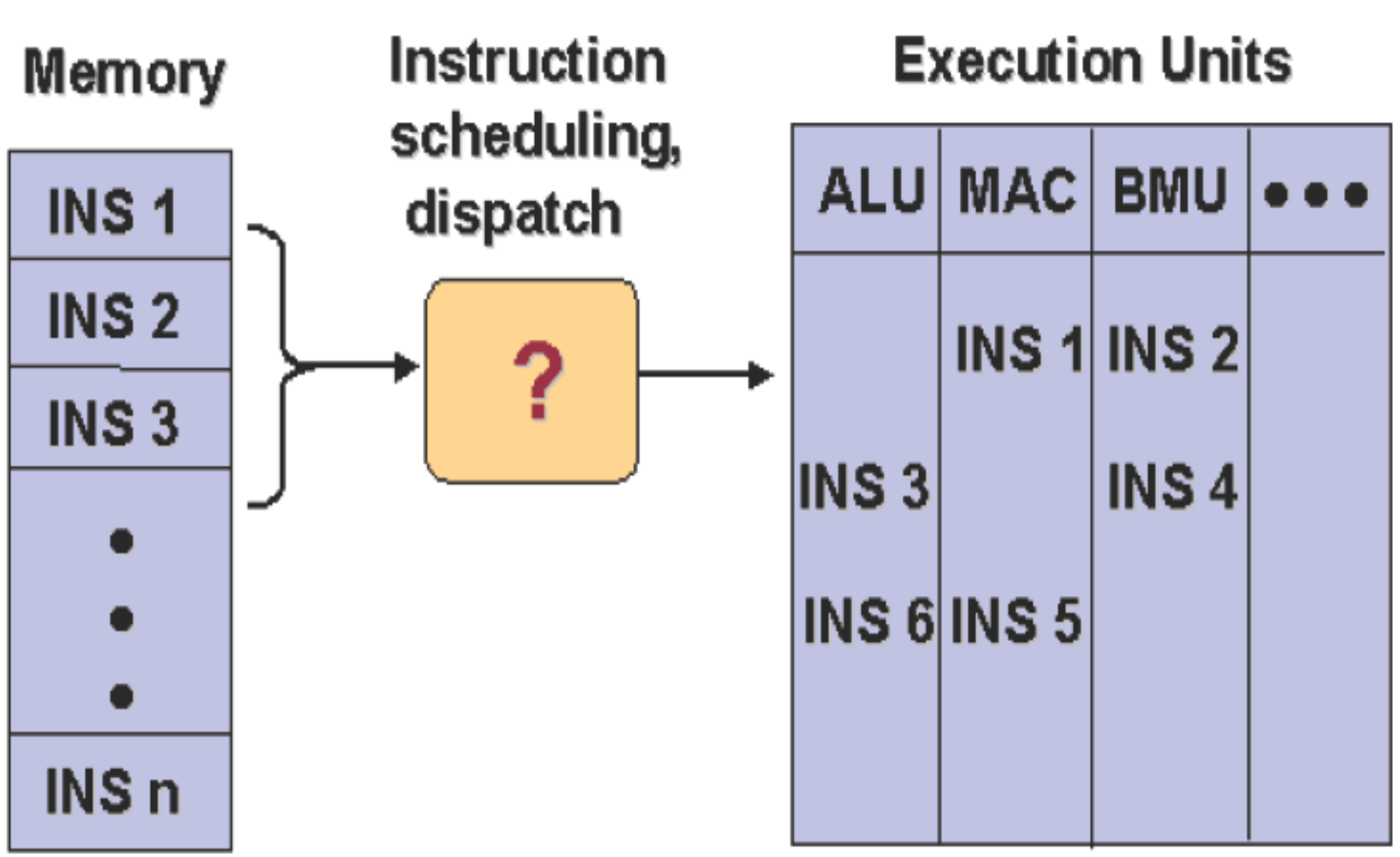
2.3. Multi Issue Architectures

- VLIW >>A class of architectures that execute multiple instr. in parallel
- VLIW provide many execs. units each is executing its own instruction
- Typically, issue: maximum of 4 - 8 instructions per cycle
- The programmer specifies the instructions to be executed in parallel
- The program's instructions group during assembly/compilation.
- 32 bits instruction word rather than 16 bits

2.3. Multi Issue Architectures



2.3. Multi Issue Architectures



2.3. Multi Issue Architectures

- **L Units:** (L1 for data path one, L2 for data path two)
  - each contain a 40-bit integer ALU
  - They are used for 32/40-bit arithmetic and compare operations,
  - 32-bit logical operations, normalization, and bit count operations.
  - All L-unit operations execute in a single instruction cycle

- **S Units:** (S1 for data path one, S2 for data path two)
  - each contain a 32-bit integer ALU and a 40-bit shifter
  - Used to perform 32-bit arithmetic, logical and bit field operations, and 32/40-bit shifts, branching, constant generation, and register transfers to and from control registers.
  - All S-unit operations execute in a single instruction cycle

2.3. Multi Issue Architectures.

M Units:(M1 for data path one, M2 for data path two)

capable of performing 16x16->32-bit multiplications

D Units:(D1 for data path one, D2 for data path two)

each contains a 32-bit adder/subtracter

used *for address generation,* including linear and circular address calculations

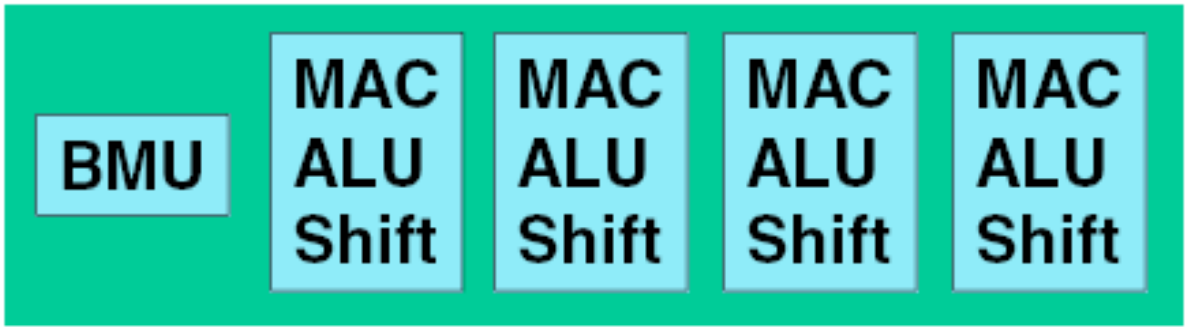
In the best case, all units operate in parallel, and the processor performs:

- four arithmetic operations
- two multiplications
- two address calculations in one instruction cycle !!!

StarCore SC140 Core

The First Implementation of the SC100

- Up to 6 instructions per cycle
  - Fewer than 'C62xx, but more powerful
    - 4 MACs/cycle vs 2 for 'C62xx
- Short (5-stage) pipeline
- Current development chip operates at 300 MHz at 1.5 volts



2.3. Multi Issue Architectures

- Wider instructions
- Advantages:
  - Use of larger, more uniform register sets
  - Specify which functional unit will exec. the instruction
  - Instructions have few restrictions on register usage and addressing modes making easier to program in assembly language
- Disadvantages:
  - Simple instructions so,
    - require more instructions to perform a task
      - But words are wider
        - High program memory usage, high chip cost!

2.3. Multi Issue Architectures

- VLIW processors use *wide buses* to access data memory
- Faster and simpler, leading to efficient compiler code generation
- But suffers from energy consumption !!!
- Can’t be used in cellular phone (mobile applications)

Summary Comparison

Major VLIW DSP Architectures

Processor	Issue width	Data memory bandwidth (16-bit words)	Instruction Size	Clock (MHz)	Pipeline Depth	Notable characteristics
TMS320C62xx	8	4 words/cycle	32 bits	300	11	First VLIW-based DSP processor
SC140	6	8 words/cycle	16 bits w/ 16-bit prefixes	300	5	Targeting compact code, low energy
SC110	3	4 words/cycle	16 bits w/ 16-bit prefixes	300*	5	Scaled-down SC140
Carmel	2, 6	4 words/cycle	24/48 bits	180	8	CLIW instructions, 4 AGUs
TMS320C64xx	8	8 words/cycle	32 bits	600*	11	Enhanced 'C62xx
TMS320C55xx	2	3 words/cycle	8-48 bits	160*	7	Based on 'C54xx

*Projected

2.4. SIMD Technique

- Improves performance on some algorithms by allowing the processor to execute multiple instances of the same operation in parallel using different data
- Increase performance for vector operations
- Programmer must arrange data in memory
- *SIMD is only effective in algorithms that can process data in parallel !!!*

2.5 Multi-Core DSP Architectures

Symmetric and Heterogeneous Multi-Core DSP Systems

- Multi-core DSP: Multiple identical DSP cores on one chip, sharing memory/peripherals
- TI TMS320C6678: 8 x C66x cores at 1.25 GHz = 160 GMACs total
- **Advantages:**
  - Linear performance scaling for parallelizable algorithms (Amdahl permitting)
  - Shared L2/L3 cache reduces memory bandwidth bottlenecks
  - One chip replaces multiple single-core DSP boards
- **Challenges:**
  - Inter-core communication overhead: EDMA3, SRIO, HyperLink buses
  - Load balancing: Uneven task distribution wastes cycles
  - Programming models: OpenMP, TI SYS/BIOS, OpenCL for DSP

3. Alternatives to DSP processors

High Performance CPUs

- GPP: Intel Core/Xeon (x86-64), AMD Ryzen/EPYC, ARM Cortex-A series…
- Added SIMD based instruction sets
  - E.g: SSE/AVX/AVX-512 for Intel (MMX and SSE are legacy),
  - VSX/AltiVec for PowerPC/POWER, NEON/SVE for ARM
  - AVX2/AVX-512 for AMD (3DNow!)
- Those have:
  - 256/512-bit SIMD data bus, 64-bit general registers, 256/512-bit vector ALU.
  - 4 times the performance on 16-bit data (data size most often used in DSP)
- Modern CPUs operate at 3-5GHz (up to 10x faster than DSP processors) with multi-core and turbo boost

3. Alternatives to DSP processors
High Performance CPUs

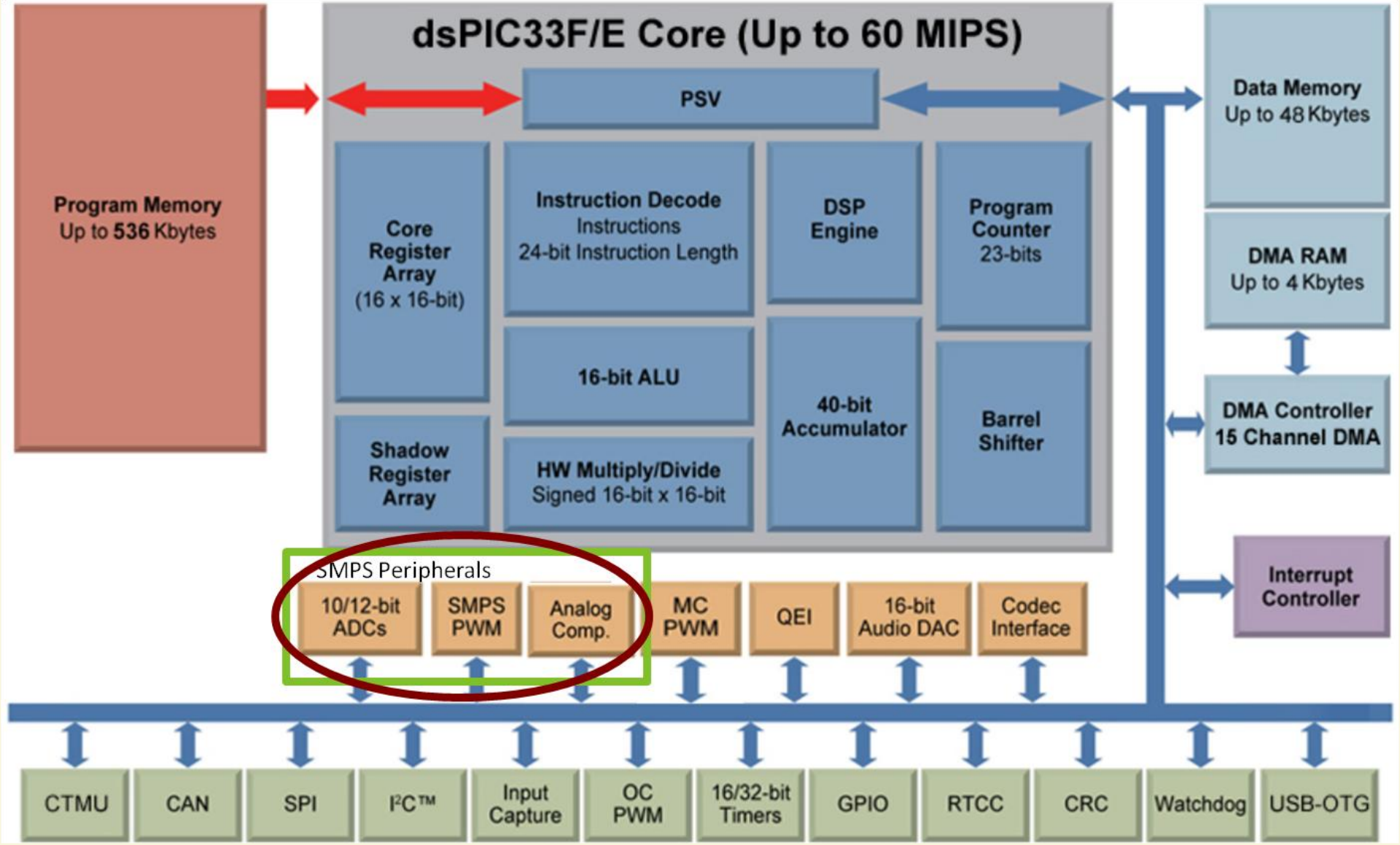
- So, why do we use DSP processors at all?
  - DSP processors provide the best mix of:
    - *Performance*
    - *Power consumption*
    - *Price*
  - Availability of development tools
  - For real-time applications, those CPU can be problematic for their dynamic features.

3. Alternatives to DSP processors

DSP/Microcontroller Hybrids

- Used in applications that require a mixture of “control-oriented software” & “DSP software”
  - E.g : digital cellular phone
- Microcontrollers are good at controlling but bad in DSP task
- DSP processors: Opposite of microcontroller.
- Combination performed using two separate processors:
  - A microcontroller + DSP processor.
- Some vendors (Hitachi, ARM,…) added DSP functionality to microprocessor designs
- OMAP/Sitara – Texas Instr. (ARM Cortex-A9/A15 + DSP: C64x+/C66x);

  Likewise Jacinto (automotive) and AM2x/AM6x SoC families


dsPIC33F/E Core (Up to 60 MIPS)
Program Memory
Up to 536 Kbytes
PSV
Core Register Array (16 x 16-bit)
Instruction Decode Instructions 24-bit Instruction Length
DSP Engine
Program Counter 23-bits
16-bit ALU
40-bit Accumulator
Barrel Shifter
Shadow Register Array
HW Multiply/Divide Signed 16-bit x 16-bit
Data Memory Up to 48 Kbytes
DMA RAM Up to 4 Kbytes
DMA Controller 15 Channel DMA
Interrupt Controller
SMPS Peripherals
10/12-bit ADCs
SMPS PWM
Analog Comp.
MC PWM
QEI
16-bit Audio DAC
Codec Interface
CTMU
CAN
SPI
I²C™
Input Capture
OC PWM
16/32-bit Timers
GPIO
RTCC
CRC
Watchdog
USB-OTG

ARM Cortex + NEON/SVE: Mobile DSP

ARM Architecture as DSP Platform in Smartphones and IoT

- NEON: ARM's 128-bit SIMD extension — present in all Cortex-A processors since A8
- NEON executes 8x16-bit, 4x32-bit, or 2x64-bit operations per cycle
- SVE2 (Scalable Vector Extension): 128–2048-bit vectors, introduced in Cortex-A55/A78/X1
- **Why ARM replaced dedicated DSPs in mobile:**
  - A single SoC (e.g. Apple A18 Pro, Qualcomm Snapdragon 8 Gen 3) includes CPU+GPU+DSP+NPU
  - Hexa/Octa-core CPUs with NEON match DSP performance in most audio/video tasks
  - Still used separately: Ultra-low-power context sensing (Qualcomm Hexagon DSP)
- Qualcomm Hexagon DSP: Integrated in all Snapdragon SoCs for audio, camera, sensor hub

Modern High-Performance CPUs

Updated: Intel Core/Xeon, AMD Ryzen/EPYC, ARM Cortex-A

- Modern x86-64 CPUs: Intel Core Ultra, AMD Ryzen 9 — operate at 3.5–5.5 GHz (base/boost)
- Intel AVX-512: 512-bit SIMD, 16 float32 ops/cycle — dominant in HPC/DSP workloads
- AMD AVX2 (256-bit): Strong performance in signal processing and ML inference
- ARM Cortex-A78/X3: NEON 128-bit + SVE2 scalable SIMD — widely used in mobile DSP
- **Why dedicated DSPs still exist despite powerful CPUs:**
  - Power efficiency: DSPs deliver 10–100x better MOPS/mW than general CPUs
  - Real-time determinism: DSPs have predictable interrupt latency; CPUs do not
  - Cost: Embedded DSPs cost $1–$10 vs $50–$500 for a full CPU

GPU as DSP processor

GP-GPU Computing for Massive Parallel Signal Processing

- Modern GPUs: Thousands of cores running the same operation on different data (SIMD at scale)
- NVIDIA CUDA platform: C6000+ DSP kernels can run on GPU for offline/batch processing
- **Application areas:**
  - FFT processing: cuFFT library delivers 10–100x speedup vs single-core CPU
  - Software-Defined Radio (SDR): GPU replaces multiple DSP boards
  - 5G base station signal processing (NVIDIA Aerial platform)
- **Limitations for real-time DSP:**
  - High power consumption (100–400W TDP) — unsuitable for mobile/IoT
  - PCIe latency -- not suitable for hard real-time (< 1ms) requirements

FPGA for DSP

Field-Programmable Gate Arrays as Flexible DSP Engines

- FPGA: Reconfigurable hardware — can implement custom DSP data paths at gate level
- Key advantages over DSP processors:
  - True parallelism: Thousands of DSP blocks (multiplier-accumulators) running simultaneously
  - Deterministic timing: No OS, no cache misses — guaranteed latency
  - Custom precision: 8-bit, 16-bit, 32-bit — no fixed word length
- Leading FPGA vendors: Xilinx/AMD (Versal AI Core), Intel/Altera (Agilex)
- Key Limitation: Requires hardware design expertise (VHDL/Verilog/HLS)
- HLS (High-Level Synthesis) tools partially close the gap: Vitis HLS (AMD), Intel HLS Compiler
- Used in: Radar, sonar, software radio, medical imaging (MRI/ultrasound)

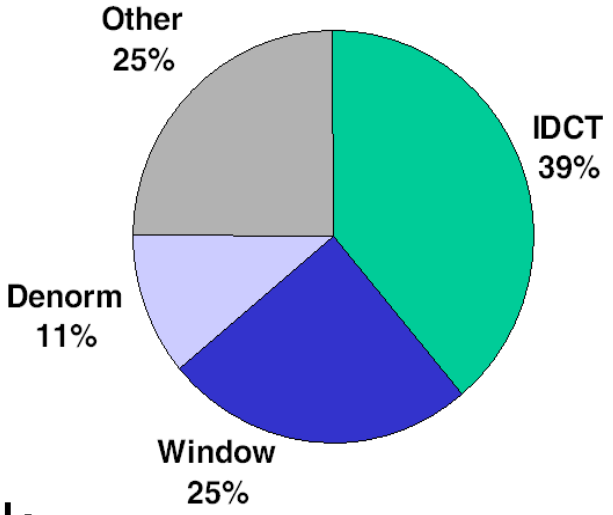
4. DSP Benchmarking - *DSPstone*

- Benchmark is a relative measure of the performance of a DSP in a particular application.
- You can measure DSP processor performance in many ways.
- The *most common metric* is *the time required for a processor to accomplish a defined task*.
- Traditional Approaches to Performance Measurement :
  - use of MIPS, MOPS, and MACS
- To benchmark computer systems is *to use complete applications,* or even *suites of applications*.
- Application benchmarking allows to be measured for a processor:
  - *energy consumption, execution time, the memory usage…*
- This approach is used by the *Standard Performance Evaluation Corp*. SPEC benchmarks for GPP and systems.
- In DSP, examples of applications cover: speech coders (CELP, VSELP, GSM etc.), modems (V.34, V.90, etc.), disk drive servo control programs, …

Algorithm Kernel Benchmarks

A Natural Approach for DSP Benchmarks

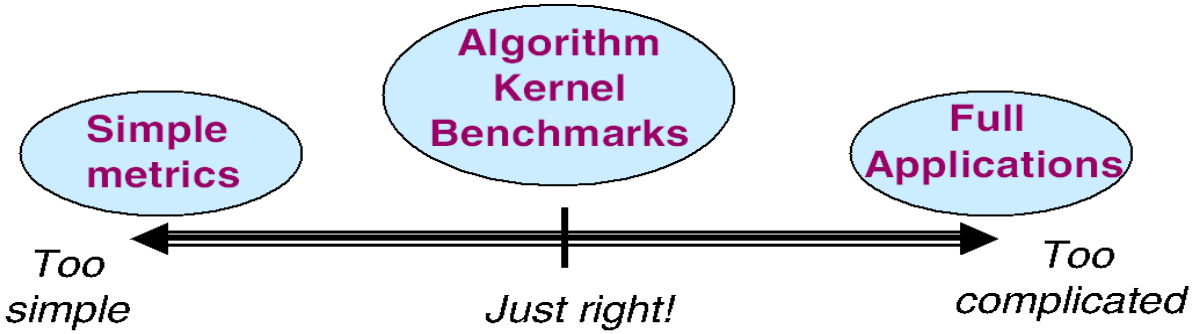
- DSP algorithm kernels are the most computationally intensive portions of DSP applications.
- Example algorithm kernels include
  - FFTs
  - IIR filters
  - Viterbi decoders …
- Application-relevant algorithm kernels are strong predictors of overall performance.



Why Use Algorithm Kernels?

Algorithm kernels are good benchmark candidates because they are:

- Relevant
- Practical to specify and implement
- Relatively simple to optimize



The BDTI DSP Kernel Benchmarks™		
Benchmark	**Description**	**Example Application**
Real Block FIR	Finite impulse response filter that operates on a block of real (not complex) data.	Speech processing (e.g., G.728 speech coding).
Complex Block FIR	FIR filter that operates on a block of complex data.	Modem channel equalization.
Real Single-Sample FIR	FIR filter that operates on a single sample of real data.	Speech processing, general filtering.
LMS Adaptive FIR	Least-mean-square adaptive filter; operates on a single sample of real data.	Channel equalization, servo control, linear predictive coding.
IIR	Infinite impulse response filter that operates on a single sample of real data.	Audio processing, general filtering.
Vector Dot Product	Sum of the pointwise multiplication of two vectors.	Convolution, correlation, matrix multiplication, multi-dimensional signal processing.
Vector Add	Pointwise addition of two vectors, producing a third vector.	Graphics, combining audio signals or images.
Vector Maximum	Find the value and location of the maximum value in a vector.	Error control coding, algorithms using block floating-point.
Viterbi Decoder	Decode a block of bits that has been convolutionally encoded.	Error control coding.
Control	A sequence of control operations (test, branch, push, pop, and bit manipulation).	Virtually all DSP applications include some control code.
256-Point In-Place FFT	Fast Fourier Transform converts a time-domain signal to the frequency domain.	Radar, sonar, MPEG audio compression, spectral analysis.
Bit Unpack	Unpacks variable-length data from a bit stream.	Audio decompression, protocol handling.

- The BDTI DSP Kernel Benchmarks comprise 12 digital signal processing algorithm kernels.
- Developers implement each kernel in hand-optimized assembly language on the target processor.

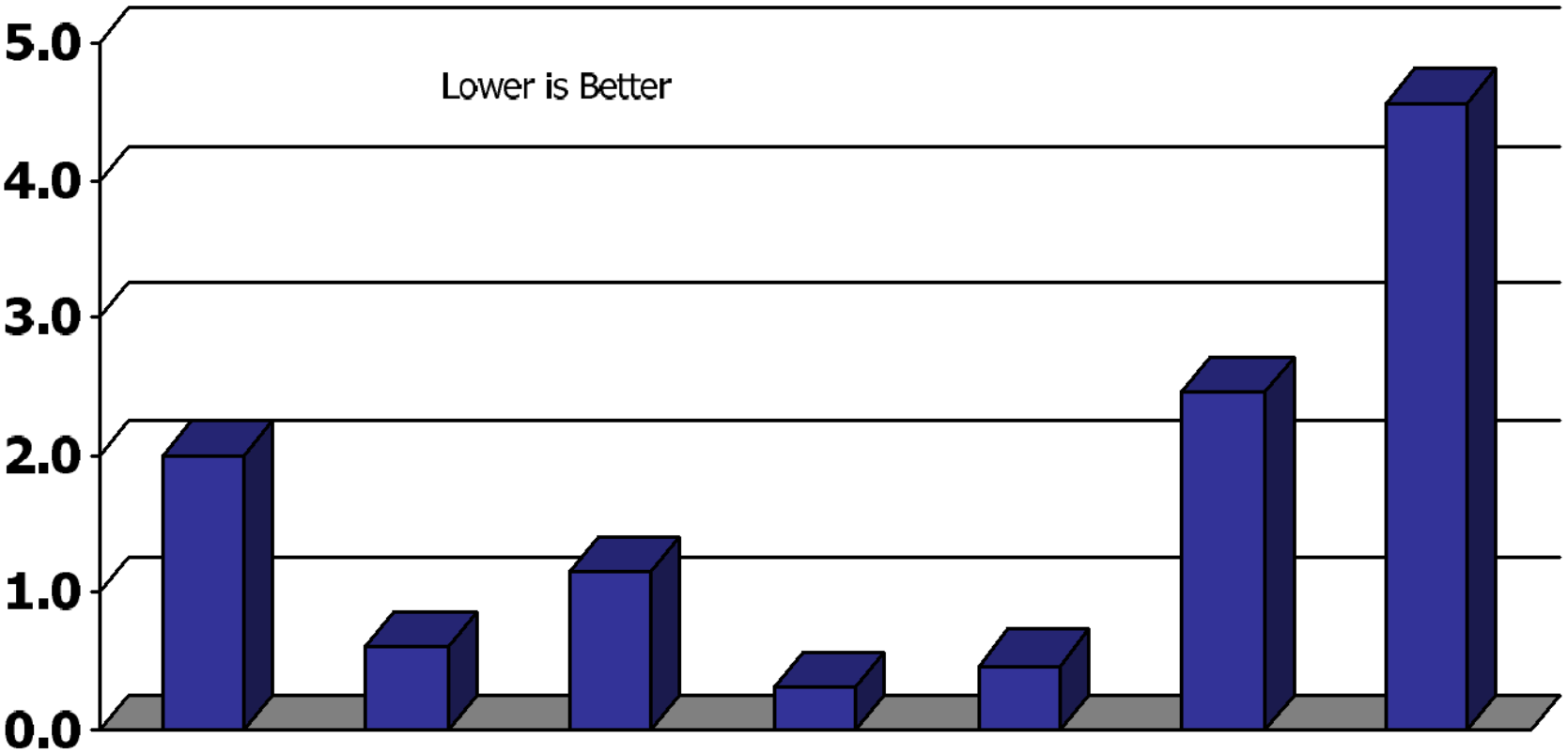
DSP Benchmarking

Ex. of benchmark (by BDTI)

- execution time (FIR, SoP, Viterbi decoder,..)
- energy consumption (FIR)
- Memory usage (control, FIR)

Block FIR Filter Benchmark

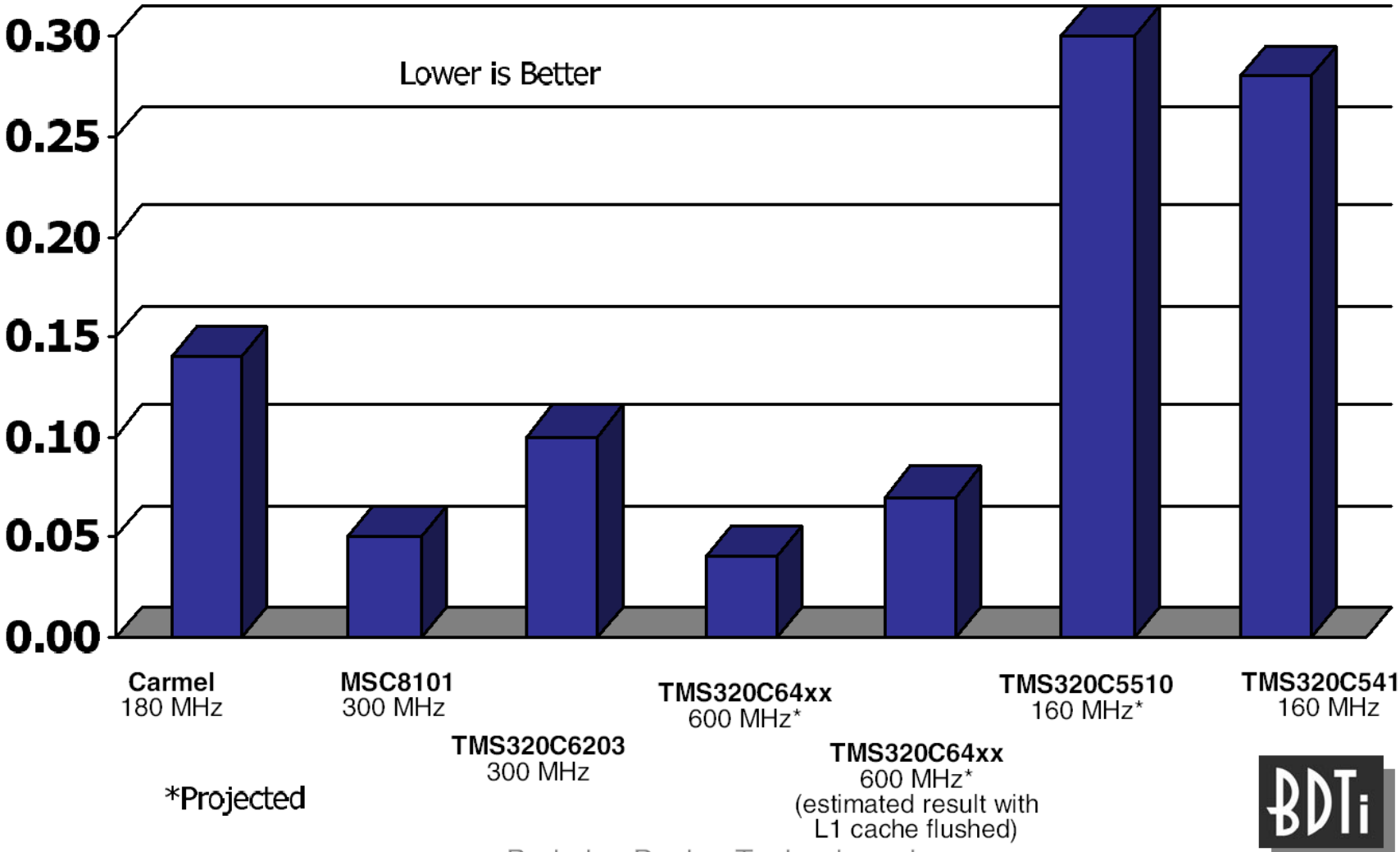
Execution Times (ms)



*Projected

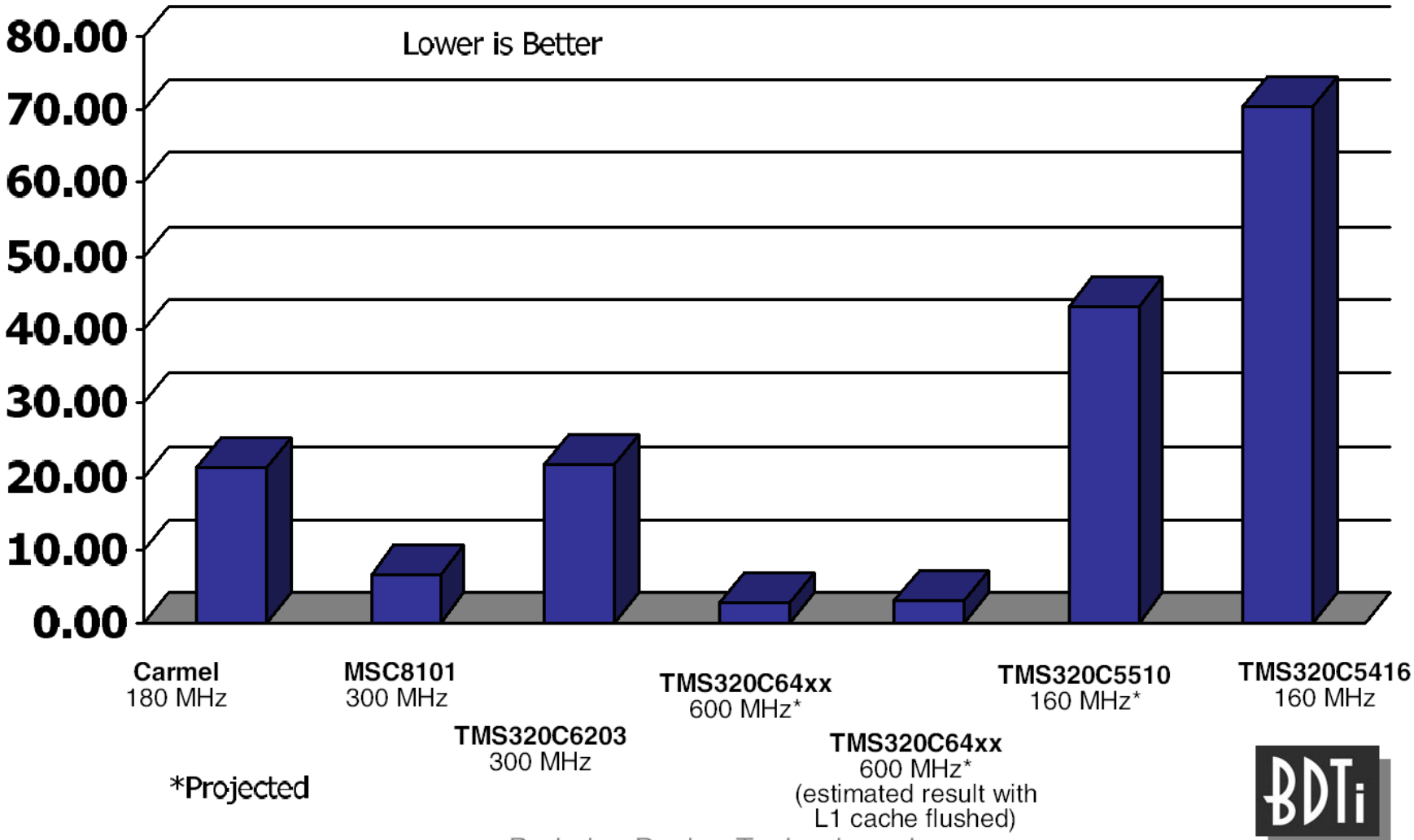
Vector Dot Product Benchmark

Execution Times (ms)



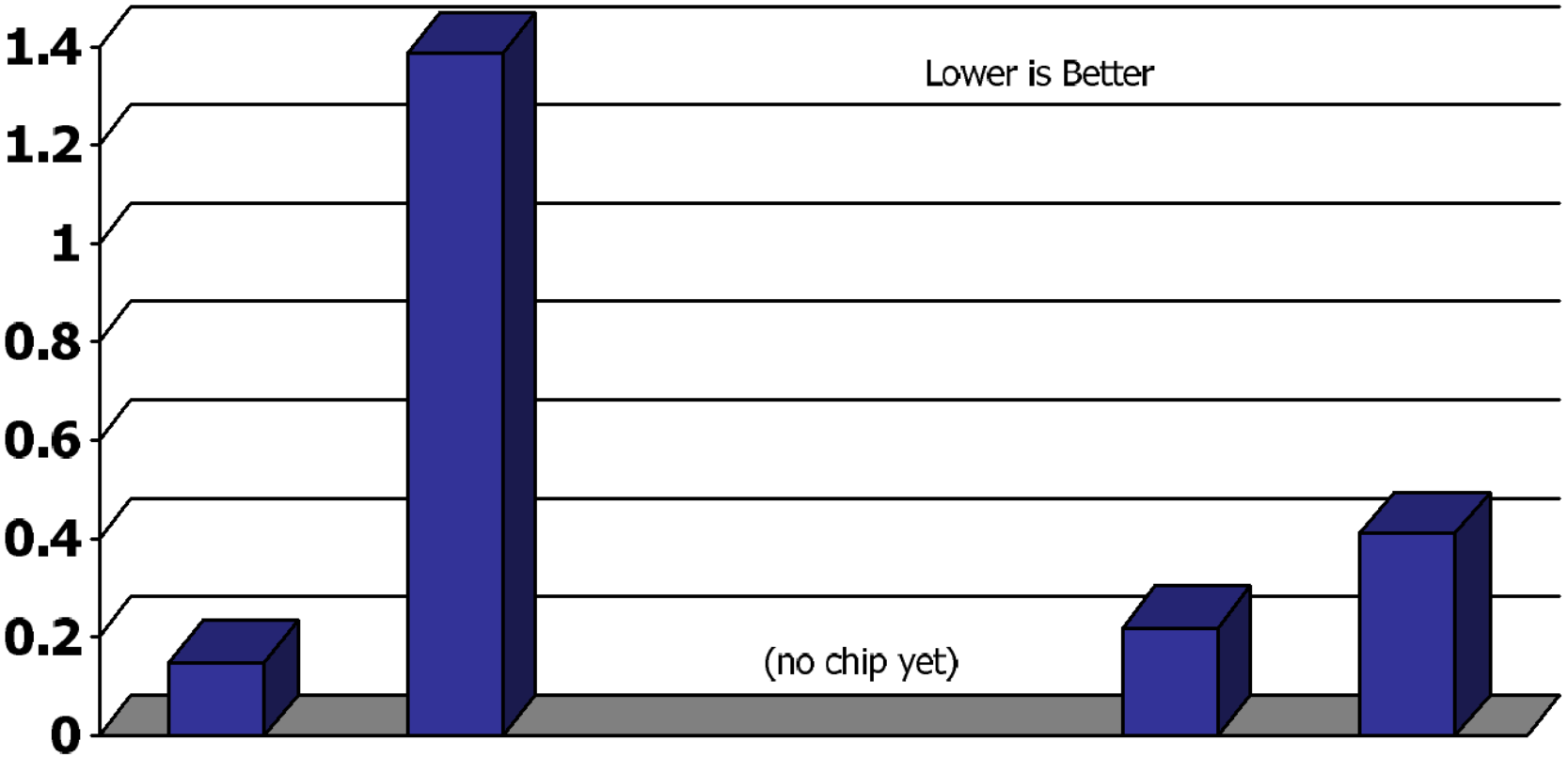
Viterbi Decoder Benchmark

Execution Times (ms)



Block FIR Filter Benchmark

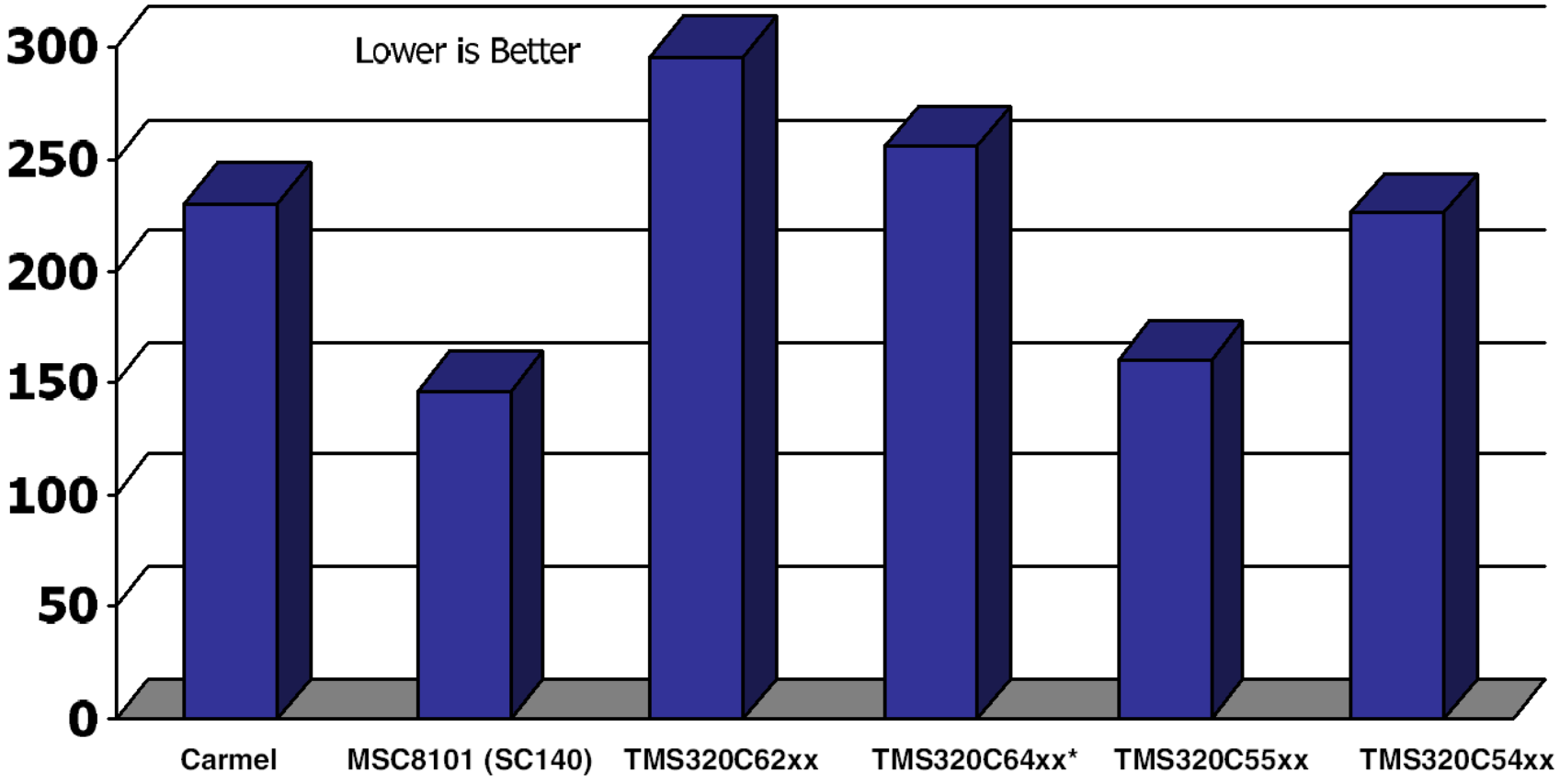
Estimated Energy Consumption (W-s)



*Projected

Control Benchmark

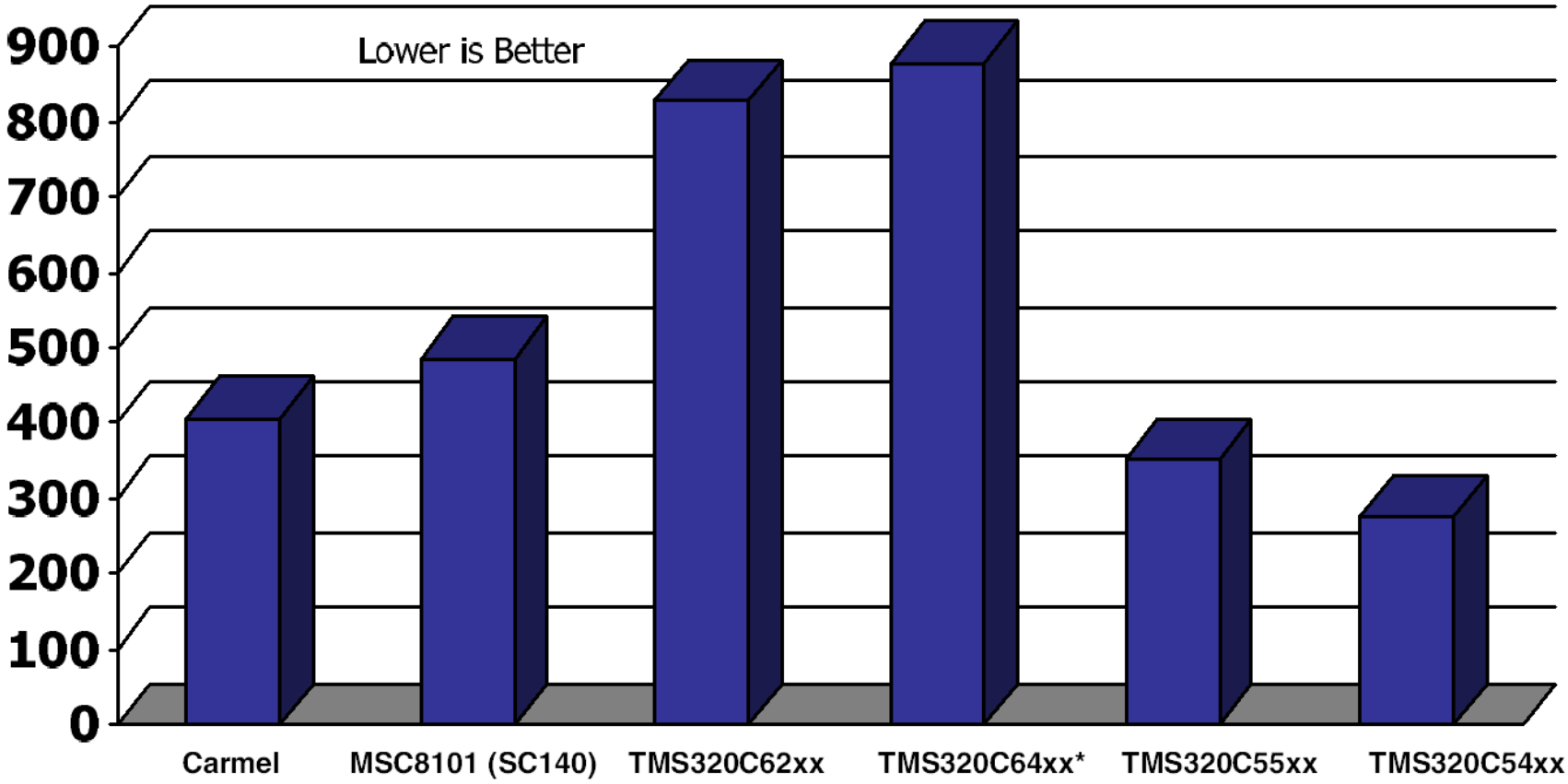
Total Memory Usage (Bytes)



*Projected

Block FIR Filter Benchmark

Total Memory Usage (Bytes)



*Projected

Speed Scores for Fixed-Point Packaged Processors (Higher is Better)

BDTImark2000™ and BDTIsimMark2000™ (Single-Core Scores)

Updated November 2013

Processor Family	Clock Rate (min-max)	BDTImark2000™, *BDTIsimMark2000™* (min-max)
ADI ADSP-218x	80 MHz	**240**
ADI ADSP-219x	100–160 MHz	**250–410**
ADI ADSP-BF5xx (Blackfin)	200–600 MHz	**1120–3360**
ADI ADSP-BF60x (Blackfin)	400–500 MHz	***5260–6580***
ADI ADSP-TS201S (TigerSHARC)	500–600 MHz	**5330–6400**
ADI ADSP-TS202S/203S (TigerSHARC)	500 MHz	**5130**
Freescale B4860 (SC3900)	1200 MHz	***37460***
Freescale DSP563xx[1]	150–275 MHz	**450–820**
Freescale DSP5672x[1]	200–250 MHz	**590–740**
Freescale DSP5685x/56F8xxx (56800E)	32–120 MHz	***90–340***
Freescale DSP56F8xx (56800)	60–80 MHz	**80–110**
Freescale MSC71xx (SC1400)	200–300 MHz	**2240–3370**
Freescale MSC814x (SC3400)	800–1000 MHz	**9520–11900**
Freescale MSC815x/825x (SC3850)	1000 MHz	**15420**
Freescale MSC815x/825x (SC3850)[2]	1200 MHz	***18500***
Freescale MSC81xx (SC140)	300–500 MHz	**3370–5610**
Marvell PXA255	200–400 MHz	**470–930**
Marvell PXA27x	312–624 MHz	**1070–2140**
Microchip dsPIC3x	16–70 MHz	***50–220***
NEC µPD77050 (SPXK5)	250 MHz	***1770***
Qualcomm Hexagon V2 (1 thread)[3]	67–100 MHz (per thread)	***1040–1550 (1 Thread)***
Qualcomm Hexagon V2 (6 threads)[3]	67–100 MHz (per thread)	***6240–9300 (Projected Best Case for 6 Threads)***
Qualcomm Hexagon V4 (1 thread)[3]	100–233 MHz (per thread)	***1810–4220 (1 Thread)***
Qualcomm Hexagon V4 (3 threads)[3]	100–233 MHz (per thread)	***5430–12660 (Projected Best Case for 3 Threads)***
Qualcomm Hexagon V5 (1 thread)[3]	100–267 MHz (per thread)	***1810–4840 (1 Thread)***
Qualcomm Hexagon V5 (3 threads)[3]	100–267 MHz (per thread)	***5430–14520 (Projected Best Case for 3 Threads)***
Texas Instruments C55x+[2]	400–500 MHz	***2530–3160***
Texas Instruments OMAP35x[4]	600–720 MHz	***4540–5450***
Texas Instruments TMS320C54x	50–160 MHz	**150–500**
Texas Instruments TMS320C55x	50–300 MHz	**240–1460**
Texas Instruments TMS320C62x	150–300 MHz	**960–1920**
Texas Instruments TMS320C64x	400–1000 MHz	**3650–9130**
Texas Instruments TMS320C64x+	400–1200 MHz	**4390–13170**
Texas Instruments TMS320C66x	850–1500 MHz	**11350–20030**
VeriSilicon VSI40x	120–200 MHz	**560–940**

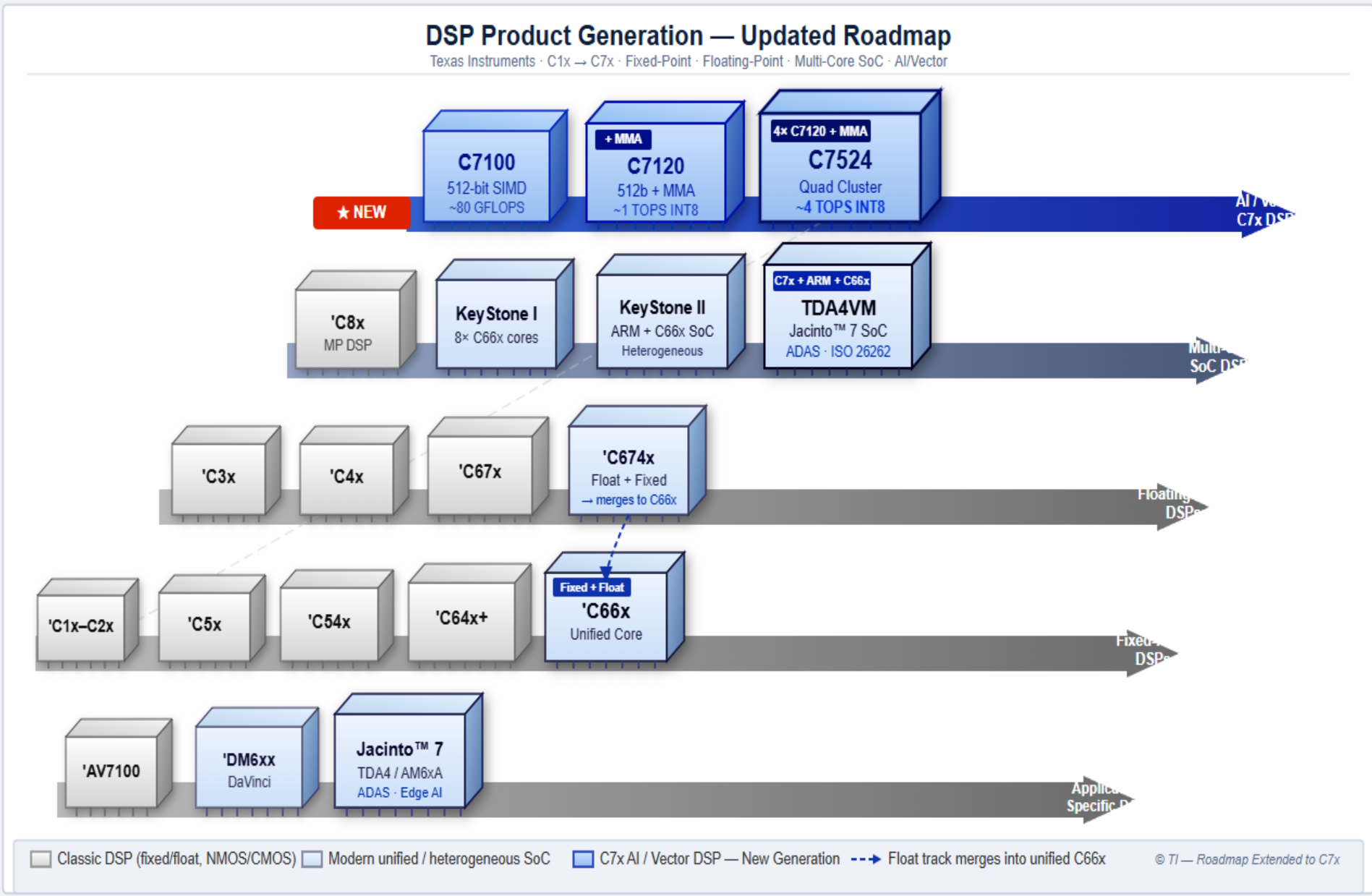
[1] Benchmarked with 24-bit fixed-point data; all other processors benchmarked with 16-bit fixed-point data
[2] Not available to the general market
[3] Lower range of score is official single-thread BDTIsimMark2000, higher score is projecteded best case score using the maximum number of available threads (not an offical BDTIsimMark2000 score).
[4] Metrics are for ARM Cortex-A8 core only ('C64x+ DSP is also available in some family members)

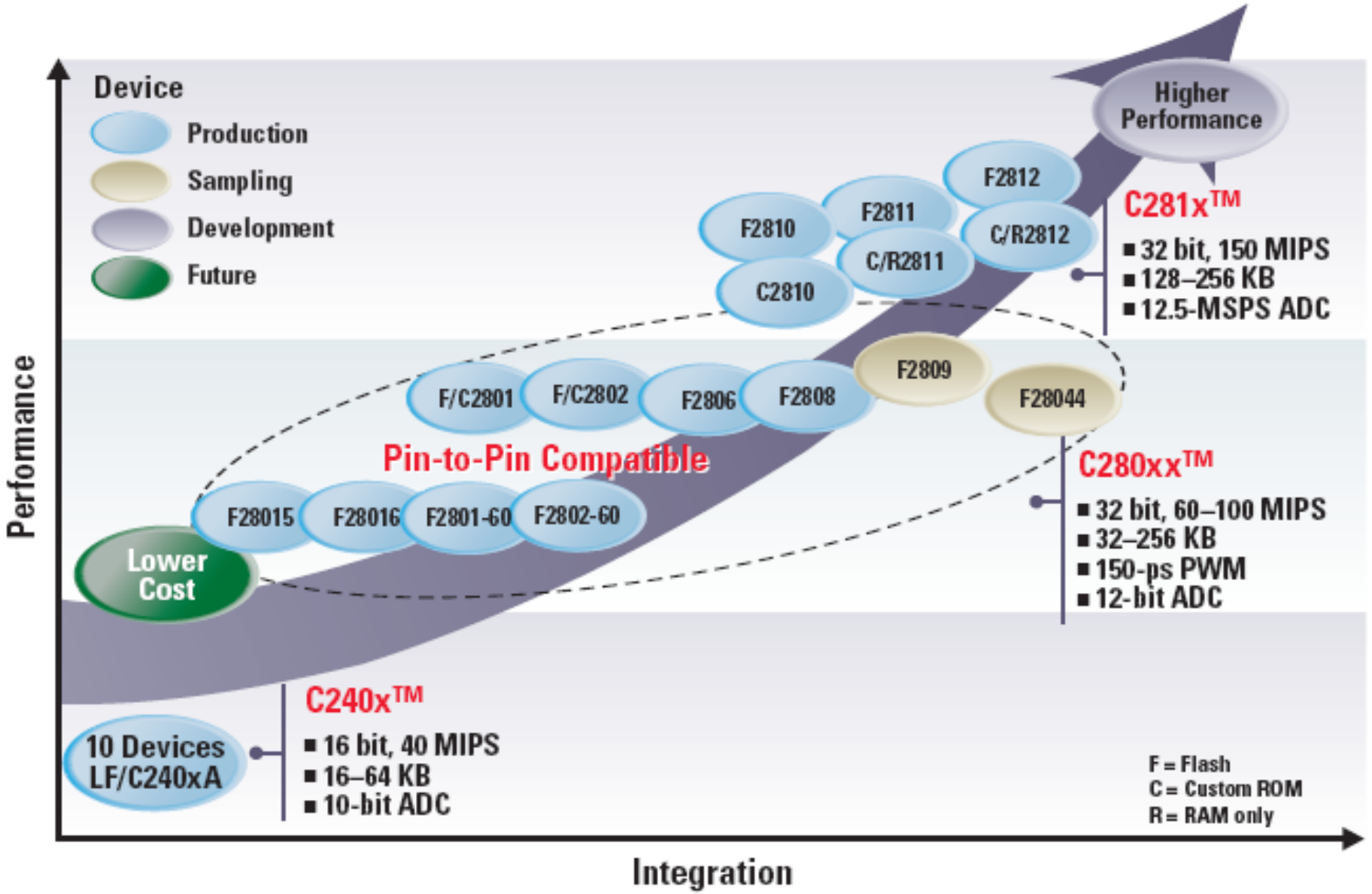
DSP Benchmarking – Modern Standards

Updated Performance Measurement Frameworks (2010s–2020s)

- BDTI (Berkeley Design Technology Inc.) — still the industry standard for DSP benchmarking:
  - –BDTImark2000: Composite score from 12 DSP kernels (FIR, IIR, FFT, Viterbi, etc.)
  - –BDTI DSP Efficiency (BDTImark/mW): Power-normalized score for IoT/mobile comparison
- MLPerf Inference (2019–present): Benchmarks for AI/ML on DSP and NPU hardware:
  - –Image classification (ResNet-50), object detection (SSD), NLP (BERT), voice (RNN-T)
- EEMBC (Embedded Microprocessor Benchmark Consortium):
  - –CoreMark: Integer/control benchmark for embedded MCU/DSP
  - –MLMark: Neural network inference on embedded devices
  - –ULPMark: Ultra-low-power benchmark relevant to IoT DSP applications

5. TI DSP Family

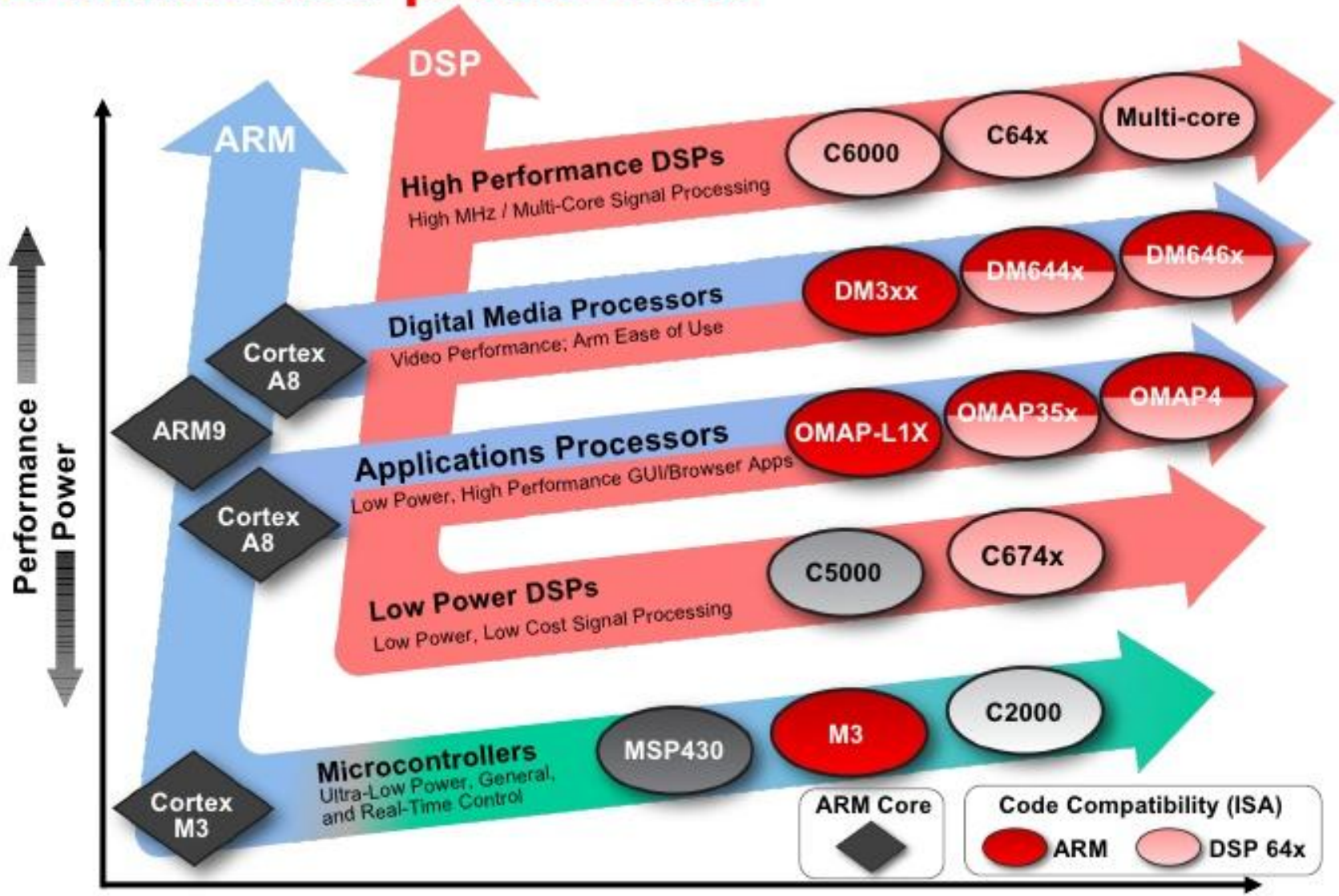




TMS320C2000 Digital Signal Controller Platform Roadmap

The C2000™ controller platform provides an optimized combination of DSP performance and MCU integration for digital control systems.

TI embedded processors



TEXAS INSTRUMENTS · HIGH-PERFORMANCE DSP

C7X DSP

PRODUCT FAMILY ROADMAP · ARCHITECTURE · CORES · SOC INTEGRATION · 2019 →

512-BIT VLIW+SIMD | DEEP LEARNING | AUTOMOTIVE ADAS | INDUSTRIAL VISION | EDGE AI | C66X HERITAGE

ISA EVOLUTION

▸ INSTRUCTION SET

512-bit VLIW + SIMD Vector ISA

8 functional units per cycle · Up to 16 operations/cycle
Superscalar dispatch · Predicated execution
Extends C6000 ISA lineage with new 512-bit vector ops

▸ DATA TYPES

FP64 · FP32 · FP16 BF16 · INT8 · INT16

Full IEEE-754 FP32/FP64 support
FP16 + BF16 for deep learning inference
INT8/INT16 SIMD for high-throughput inference

▸ MEMORY HIERARCHY

L1D / L1P Cache L2 SRAM + Cache

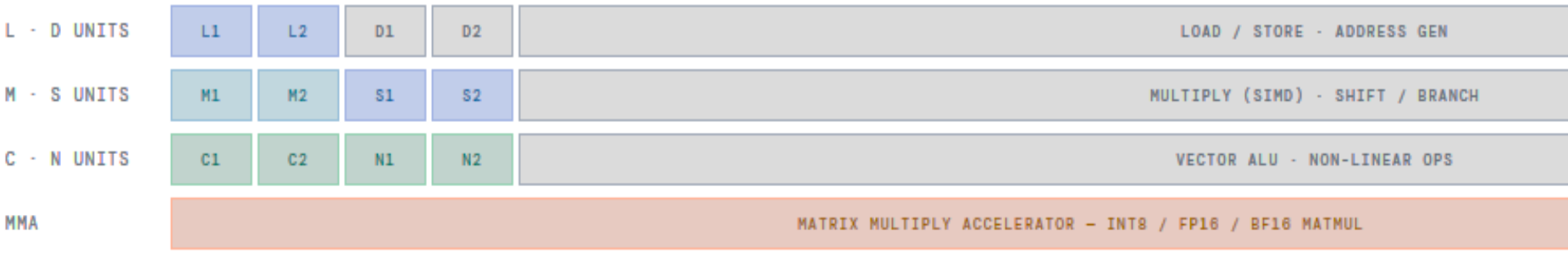
L1D: 32 KB · L1P: 32 KB per core
L2: 512 KB – 3 MB (variant-dependent)
DMA (UDMA) · Shared L3 via SoC interconnect

▸ MMA ACCELERATOR

Matrix Multiply Accel. Deep Learning Engine

Tightly coupled to C7x core pipeline
8×8 INT8 matrix ops · FP16/BF16 matmul
Dramatically boosts TOPS for CNN inference

▸ C7X FUNCTIONAL UNIT PIPELINE (PER CORE · 512-BIT WIDE)



DSP CORE ROADMAP

2019

C7100

GEN 1 · BASELINE

VECTOR WIDTH 512-bit	FP32 PEAK ~80 GFLOPS
L2 CACHE 512 KB	MMA No
INT8 THROUGHPUT ~256 GOPS	DATA TYPES FP32/16 INT8/16

512B SIMD · C66X COMPATIBLE · FULL FP64

2021

C7120

GEN 2 · MMA

VECTOR WIDTH 512-bit	FP32 PEAK ~80 GFLOPS
MMA ✓ Integrated	INT8 W/ MMA ~1 TOPS
L2 CACHE 512 KB	BF16 Yes

MMA ACCEL · BF16 · ~1 TOPS INT8 · DL INFERENCE

2022

C7504

QUAD C7100 CLUSTER

CORES 4 × C7100	FP32 PEAK ~320 GFLOPS
INT8 PEAK ~1 TOPS	MMA No
SHARED L3 Yes	TOPOLOGY Cluster mesh

4× CORES · HIGH THROUGHPUT · CLUSTER BUS

C7524

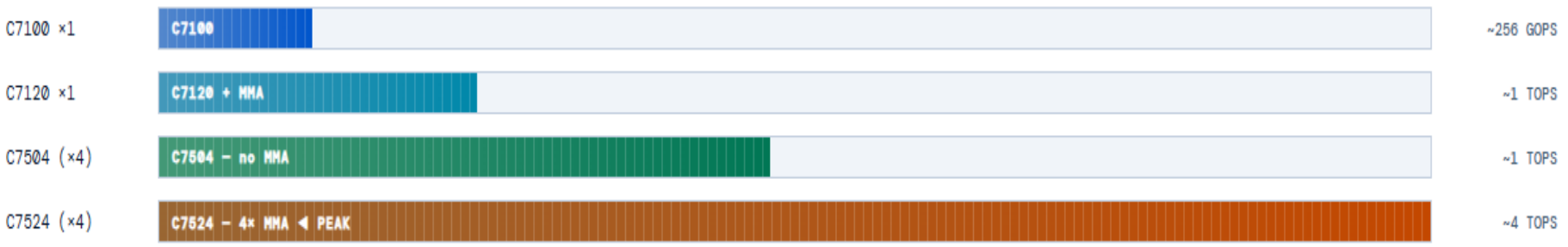
QUAD C7120 + MMA

CORES 4 × C7120	FP32 PEAK ~320 GFLOPS
MMA 4 × MMA	INT8 (MMA×4) ~4 TOPS
L2 TOTAL 4 × 512 KB	BF16 Yes

4× MMA · ~4 TOPS · HIGHEST PERF · BF16 / FP16

PERFORMANCE COMPARISON

RELATIVE INT8 INFERENCE THROUGHPUT (NORMALIZED)



SOC INTEGRATION

TDA4VM

AUTOMOTIVE

C7X CORE	**2 × C7100 + MMA**
HOST CPU	6× ARM Cortex-A72
SAFETY	ISO 26262 ASIL-B/D
VISION	MVI · 4× C66x
FAMILY	Jacinto™ 7
USE CASE	ADAS · Fusion

TDA4VH

AUTOMOTIVE

C7X CORE	**4 × C7120 + MMA**
HOST CPU	8× Cortex-A72
SAFETY	ISO 26262 ASIL-B/D
AI PERF	**~20+ TOPS system**
FAMILY	Jacinto™ 7
USE CASE	Central ADAS

TDA4AL

AUTOMOTIVE

C7X CORE	**1 × C7100 + MMA**
HOST CPU	4× Cortex-A72
SAFETY	ISO 26262 ASIL-B
TARGET	Cost-optimized
FAMILY	Jacinto™ 7
USE CASE	Entry ADAS

AM62A

INDUSTRIAL

C7X CORE	**1 × C7120 + MMA**
HOST CPU	4× Cortex-A53
SAFETY	IEC 62443
VISION	ISP · Camera
FAMILY	Sitara™
USE CASE	Industrial Vision

AM68A	INDUSTRIAL
C7X CORE	2 × C7120 + MMA
HOST CPU	8× Cortex-A72
SAFETY	Functional Safety
AI PERF	~8 TOPS system
FAMILY	Sitara™
USE CASE	Robotics · Vision AI

AM69A	EDGE AI
C7X CORE	4 × C7120 + MMA
HOST CPU	8× Cortex-A72
SAFETY	Functional Safety
AI PERF	~32 TOPS system
FAMILY	Sitara™
USE CASE	High-End Edge AI

- The C7x DSP core is programmed via TI's **TI Processor SDK** and **TIDL** (TI Deep Learning) library. Development tools include Code Composer Studio (CCS), Open CL, and optimized BLAS/FFT libraries. MMA acceleration is exposed via TIDL's inference engine for CNN/DNN workloads.
- C7x maintains backward ISA compatibility with C6000 intrinsic, enabling code reuse. Performance figures are approximate and depend on clock frequency, memory bandwidth, and workload characteristics.

In the context of TI's C7x DSP, **MMA** stands for **Matrix Multiply Accelerator**.

It's a dedicated hardware block coupled to the C7x core pipeline, designed to accelerate the heavy matrix multiplication operations that dominate deep learning workloads (CNNs, DNNs, transformers).

Here's what it does:

What it computes: It performs the operation **D = A × B + C** — a multiply accumulate on matrices — which is the fundamental operation in every neural network layer (convolution, fully connected, attention).

Supported data types:

- INT8 × INT8 (highest throughput — used for inference)
- FP16 × FP16
- BF16 × BF16 (Brain Float 16, common in ML training/inference)

Why it matters: Without MMA, the C7x core can do INT8 SIMD via its 512-bit vector units at ~256 GOPS. With MMA enabled, that jumps to **~1 TOPS per core** — roughly a 4× boost for inference workloads — because the MMA hardware executes an entire matrix tile operation in a single clock cycle rather than breaking it into scalar multiplications.

How it's used: TI's **TIDL** (TI Deep Learning) library and the **TFLite / ONNX** runtime delegate inference layers automatically to the MMA when available. Programmers don't usually invoke it directly.

It's conceptually similar to NVIDIA's **Tensor Cores** or Google's **TPU matrix units**, just scaled for embedded/automotive power budgets rather than data-center performance.

TMS320 DSP for Multimedia - Application-to-Family Mapping — Updated 2025

Application	'C55x ultra-low pwr	'C66x VLIW HPC	'C674x fix+float	'C28x real-time ctrl	'C7x ★ AI/SIMD	Mixed Signal Simple link	Hetero. SoC ★ ARM+DSP
5G / Wireless Comms (evolved from Modems)	X	X	X		X	X	X
Audio Processing voice, music, hearing aids	X	X	X		X	X	X
Speech / NLP codec, ASR, wake-word	X	X	X		X	X	X
★ IoT Edge Processing (replaces DTAD, obsolete)	X			X	X	X	X
Computer Vision / Imaging (evolved from Graphics/Imaging)		X	X		X		X
Video Encode/Decode H.265/H.266, AV1		X	X		X	X	X
Videoconference / WebRTC SIP, RTP, AEC, noise cancel		X	X		X	X	X
★ Industrial Motor Control (C2000 strength)			X	X		X	
★ Automotive ADAS radar, lidar, camera fusion		X	X		X		X
★ AI / ML Inference TinyML, on-device DNN		X			X		X

6. Mainstream DSP processor Families

Updated 2025 — Fixed-point, Floating-point & AI-capable DSP families

Property	TI C55x Low-Power	TI C66x VLIW-HPC	TI C7x AI/SIMD	SHARC+ (ADI)	Hexagon (Qualcomm)	HiFi 5 (Cadence)	CEVA-XM8	MSC8xxx (NXP)
Manufacturer	Texas Instruments	Texas Instruments	Texas Instruments	Analog Devices	Qualcomm	Cadence (IP core)	CEVA (IP core)	NXP
Architecture	Fixed-pt. VLIW-2	Fixed-pt VLIW-8 (8-wide)	Float SIMD 512-bit VLIW-8	Float SuperSHARC 40-bit	Fixed-pt HVX vector	Fixed-pt SIMD 64-bit	Float NN+DSP hybrid	Fixed-pt VLIW-6
Max Clock	200 MHz	1.25 GHz	1 GHz (C7504)	450 MHz (SC589)	~1 GHz (in SoC)	1 GHz+	1 GHz+	1 GHz
Peak Perf.	400 MMAC/s	160 GFLOPS (C6678×8)	2 TFLOPS (C7504)	3.6 GFLOPS (SC589)	1+ TOPS (HVX)	Up to 1 GFLOPS	Up to 4 TOPS (NN)	12.8 GMAC/s (per core)
Word Length	16/32-bit	32/64-bit	32/64-bit + 512-bit vec	32/40-bit float	8–32-bit	16/24/32-bit	8–32-bit (flex)	16/32-bit
Power	0.05–0.5 W (ultra-low)	0.5–5 W (per core)	2–8 W (SoC)	0.4–1 W	Embedded in AP SoC	< 1 W (typical)	< 1 W (typical)	2–5 W
Process Node	130 nm (legacy)	40 nm	7 nm (TDA4x SoC)	16 nm (SC5xx)	5–4 nm (Snapdragon)	Licensee dependent	Licensee dependent	28 nm
Key Device (2025)	TMS320 C5535 (IoT/audio)	TMS320 C6678 (8-core)	TDA4VM / AM69 / C7504	ADSP-SC589 (audio SoC)	Snapdragon X-series (mobile)	HiFi5 DSP (in SoCs)	CEVA-XM8 (AI+DSP)	LS1043A / MSC8257
Primary Application	IoT / Audio Hearing aids Wearables	Radar / SDR HPC / MIMO Multi-	ADAS / Computer Vision / AI	Pro Audio Industrial Speech	Mobile AP 5G modem AR/VR	Smart Speaker Voice UI	AI inference IoT / Cam Drone	Telecom Basestation Packet proc

Modern TI DSP Families (2020s)

Texas Instruments Current DSP Product Lines

- **C7000 series (C7x): Latest-gen fixed/floating-point VLIW DSP**
  - –512-bit SIMD, 2 TFLOPS, used in TDA4x SoCs for ADAS/automotive
- **C6000 series (C66x/C674x): High-performance fixed/float DSP**
  - –8-core C6678 @ 1.25GHz, used in 4G LTE base stations, imaging
- **C5500 series: Ultra-low power (0.05 mW/MHz) for audio/voice**
  - –Battery-powered devices: hearing aids, voice wake-up, wearables
- **C2000 series: Real-time control MCU+DSP hybrid**
  - –Motor control, power conversion, solar inverters — C28x + CLA co-processor
- Sitara/Jacinto SoCs: ARM + C66x/C7x heterogeneous computing for automotive/ industrial

DSP in 5G Communications

Digital Signal Processing as the Core of 5G NR Infrastructure

- **5G New Radio (NR) places extreme demands on DSP processing:**
  - OFDM modulation with up to 3300 subcarriers at mmWave (60 GHz)
  - Massive MIMO: 64-256 antenna beamforming — real-time matrix operations
  - LDPC and Polar codes: New FEC codes requiring iterative DSP decoding
- **Key DSP operations in 5G base stations:**
  - FFT/IFFT: Channel estimation, OFDM modulation/demodulation
  - Turbo/LDPC decoding: Iterative belief-propagation algorithms
  - Beamforming: Matrix multiply for spatial filtering across antenna arrays
- Solutions: TI C6678 multi-core DSP + FPGA co-processors, or NVIDIA Aerial GPU platform

DSP and Machine Learning / AI Inference

Convergence of Traditional DSP and Neural Network Processing

- **ML inference (running trained neural networks) relies heavily on DSP-like operations:**
  - Convolution layers: Equivalent to 2D FIR filtering with multiple channels
  - Matrix multiplication (GEMM): Core of fully connected and attention layers
- SIMD/VLIW architectures ideally suited for INT8/FP16 neural network inference
- **Dedicated Neural Processing Units (NPUs) emerging:**
  - Apple Neural Engine (ANE): 38 TOPS in A17 Pro
  - Qualcomm Hexagon NPU: 45 TOPS in Snapdragon 8 Gen 3
  - TI C7x with MMA: Matrix Multiply Accelerator for on-chip AI inference
- Trend: DSP + AI inference converging — keyword spotting, speech, image processing on-device

Heterogeneous Computing for DSP

Combining CPU + DSP + GPU + FPGA + NPU on a Single SoC

- **Modern SoCs integrate multiple processing elements:**
  - Host CPU (ARM Cortex-A): OS, application control, user interface
  - DSP cores (C66x/C7x, Hexagon): Signal processing, audio, modem
  - GPU: Graphics, computer vision, neural network acceleration
  - FPGA fabric or MMA: Custom datapaths, low-latency hardware
- **Examples of heterogeneous DSP platforms:**
  - TI TDA4VM (ADAS): ARM + C7x DSP + MMA + GPU + dedicated vision cores
  - Xilinx Versal: ARM + FPGA + AI Engine array (DSP + AI tiles)
  - Qualcomm Snapdragon 8 Gen 3: CPU + Hexagon DSP + Adreno GPU + Hexagon NPU
- Programming challenge: Task partitioning, data movement, synchronization across domains

Power Efficiency in Modern DSP Design

Energy-Aware DSP for Mobile, IoT, and Edge Computing

- Power consumption is a critical constraint in modern DSP applications
- **Key metrics:**
  - MOPS/mW (Million Ops per milliwatt): Efficiency metric for DSPs
  - GFLOPS/W: For floating-point workloads (ML inference, radar)
- **Techniques to reduce DSP power consumption:**
  - Dynamic Voltage and Frequency Scaling (DVFS): Lower Vdd when load is low
  - Clock gating: Disable idle execution units cycle-by-cycle
  - Data precision reduction: INT8 inference uses 4x less energy than FP32
  - Near-threshold computing: Operate at minimum voltage for near-zero standby power
- TI C5535: 0.04 mW/MIPS — among the most energy-efficient DSPs for IoT/wearables

DSP in IoT and Edge Applications

Signal Processing at the Edge of the Network

- IoT generates massive sensor data requiring on-device DSP before cloud transmission:
- **Key IoT DSP tasks:**
  - Keyword spotting: Low-power DSP listening for “Hey Siri / OK Google” (always-on, <1 mW)
  - Vibration analysis: FFT on accelerometer data for predictive maintenance (Industry 4.0)
  - ECG/EEG processing: Real-time filtering in wearable health monitors
  - Beamforming: Smart speaker arrays (Amazon Echo) use DSP for noise cancellation
- **Ultra-low-power DSPs for IoT:**
  - TI C5535: 100 MHz, 0.04 mW/MIPS — audio processing from a coin cell battery
  - Analog Devices ADSP-SC5xx Sharc+: Floating-point DSP for industrial edge AI
- TinyML on DSP: Running neural networks under 256KB RAM for sensor analytics

Summary – DSP Architectures and Trends

DSP Architectures: From Conventional to Modern Systems

- Parallelism (pipelining + VLIW + SIMD) remains the core strategy for DSP performance
- Amdahl's Law: Sequential code fractions limit parallel speedup — architecture must minimize S%
- **DSP architecture evolution:**
  - 1980s Conventional → Enhanced → VLIW Multi-Issue → Multi-Core (C6678) → Heterogeneous SoC (TDA4x)
- Key players (updated): TI C6000/C7000, Analog Devices SHARC, NXP (Freescale), Qualcomm Hexagon
- Alternatives: GPUs for batch processing, FPGAs for custom latency-critical paths, ARM NEON for mobile
- Benchmarking: BDTImark2000, MLPerf, EEMBC — measure performance AND power efficiency
- **Future trends:**
  - DSP + AI/NPU convergence on SoC, TinyML at the edge, 5G/6G real-time beamforming