C1. Digital Signal Processors

Space

Photo enhancement · Data compression · Remote sensing

Medical

CT/MRI/Ultrasound · ECG analysis · Image storage

Commercial

Multimedia compression · Movie effects · Video conferencing

Telephone

Voice compression · Echo reduction · Multiplexing

Military

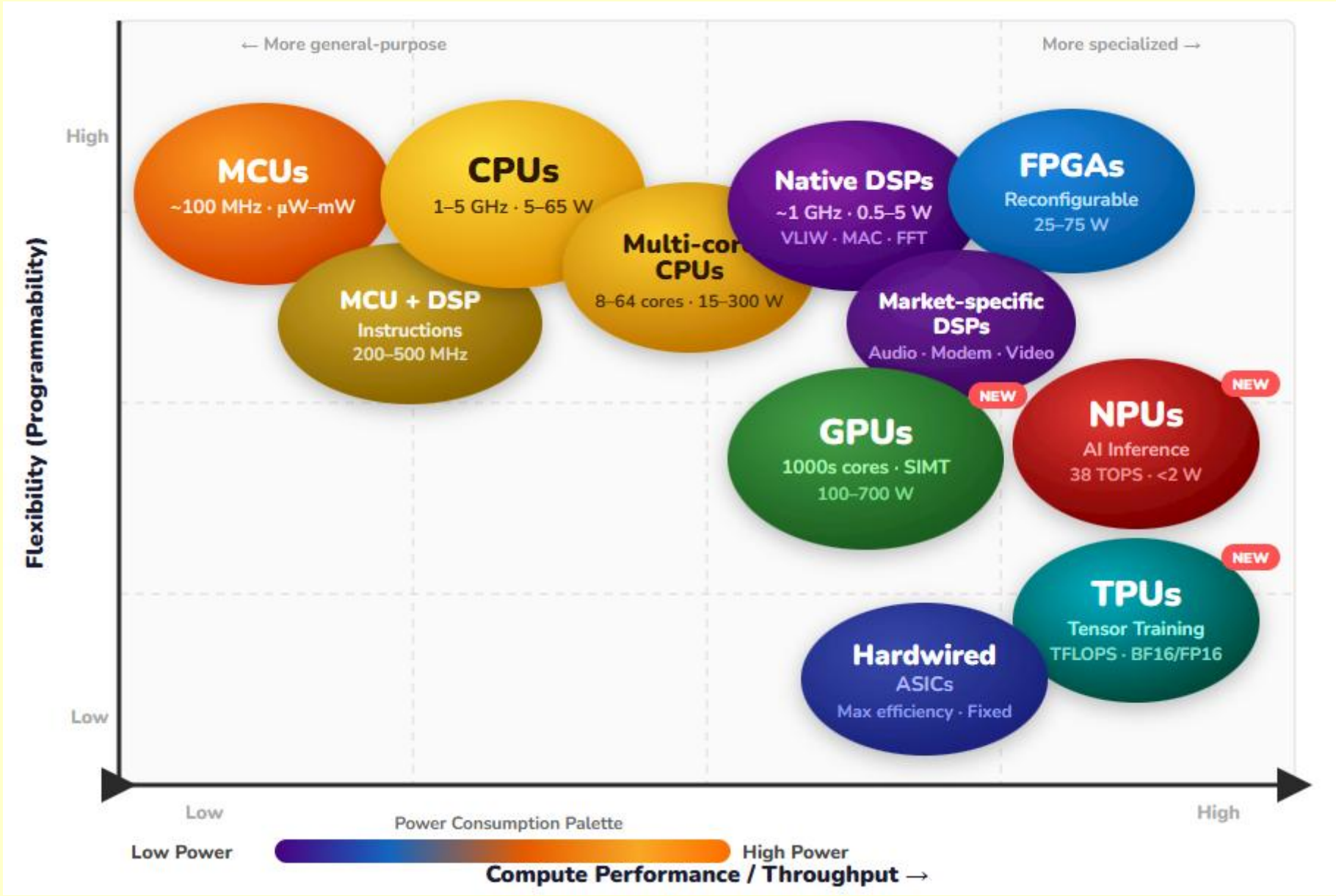
Radar · Sonar · Ordnance guidance · Secure comms

Industrial

Process monitoring · NDT · CAD tools · Oil prospecting

Scientific: Earthquake recording · Spectral analysis · Simulation · Modelling

Processor Landscape: Flexibility vs. Compute Performance
UPDATED 2024



References

- E. Lupu, R. Arsinte, T. Miclea - ***Procesoare digitale de semnal generatia TMS 320C2X. Prezentare si aplicatii*** Ed. PROMEDIA – la biblioteca
- R. Arsinte - Arhitecturi paralele si procesoare de semnal – la biblioteca
- E. Lupu, A.F. Suciu - ***Procesoare de semnal. Lucrari practice*** – UTPress – la biblioteca
- S. Emerich, E. Lupu, - ***Procesoare de semnal. Lucrari practice*** – Ed. Gutenberg

- ***http://www. ti.com***
- ***http://www.bdti.com***
- ***http://www.bdti.com/faq/dsp_faq.htm***

- SLIDES : Eugen LUPU Ph.D.

http://elupu.utcluj.ro/ password: dspe


1
SERIA
PRELUCRARI
NUMERICE
DE SEMNAL
01010101010101
01010101010101
01010101010101
PROCESOARE DIGITALE DE SEMNAL
DSP
SOFTWARE ITC
EUGEN LUPU
TIBERIU MICLEA
RADU ARSINTE
Generația TMS 320 C2X
prezentare și aplicații
EDITURA
PROMEDIA PLUS


Tempus
RADU ARSINTE
ARHITECTURI PARALELE
şi
PROCESOARE DE SEMNAL
Nucleu procesor
Bus Adrese
Bus Date
Memorie
EDITURA POLITEHNICA

There are more websites for speech/audio signal processing and recognition on the internet. Here we have a list of some of the usually employed websites, which can be linked for tutorials, discussions, papers, example code, etc.

http://www.phys.unsw.edu.au/~jw/dB.html
Introduction to the definition of Decibels for measuring energy/volume of speech/audio signals.
http://www.phys.unsw.edu.au/~jw/hearing.html
Introduction (including interactive demos) to curves of equal loudness.
http://www.phys.unsw.edu.au/music/
Homepage for "Music Acoustics".
http://www.phys.unsw.edu.au/~jw/musFAQ.html
FAQ for "Music Acoustics".
http://www.wotsit.org
File formats for various kinds, including audio and music.
http://www.speech.cs.cmu.edu/comp.speech/index.html
FAQ for the newsgroup "Comp.Speech".
http://www.bdti.com/faq/dsp_faq.htm
FAQ for the news group "Comp.DSP".
http://www.harmony-central.com/Effects/effects-explained.html
Introduction to audio effects, including many examples.
http://www.slideshare.net/fcharlot/digital-signal-processor-evolution-over-the-last-30-years

ASSESMENT

F_MARK = 30% PROJECT + 35% MIDTERM EX +35%EXAM

Ex(2) >4.5, PRJ >5, LAB – passed

Introduction — Course Outline

1. Why process signals digitally?
2. Definition of a real-time application
3. Why use Digital Signal Processing processors?
4. What are the typical DSP algorithms?
5. Parameters when choosing a DSP processor
6. Main Architectural Features of DSPs
7. DSP Market
8. Texas Instruments TMS320 family and trends in typical DSp applications

Why Use Digital Signal Processing?

⚠ Analogue Challenges (R, L, C, OA)

- ✗ Component tolerances (±5–20%)
- ✗ Temperature drift affects performance
- ✗ Voltage variation changes behaviour
- ✗ Component ageing degrades accuracy
- ✗ Mechanical vibration causes noise
- ✗ Difficult to modify / update
- ✗ Hard to reproduce identical circuits

✓ DSP Advantages(signal processors, algorithms)

- ✓ Easy to change, correct & update
- ✓ Highly reproducible results
- ✓ Reduced noise susceptibility
- ✓ Fewer chips → lower cost
- ✓ Shorter development time
- ✓ Lower power consumption
- ✓ Flexible — one algorithm, many apps

Why NOT Use Digital Signal Processing?

High-frequency signals are not always suitable for digital processing, two key reasons:

ADC Speed Limit

- Analogue-to-Digital Converters (ADC) cannot operate fast enough to sample very high-frequency signals without aliasing. The Nyquist criterion requires sampling at ≥ 2× the signal bandwidth.

Real-Time Complexity

- For extremely wideband or high-bitrate applications (e.g. mm-wave radar, optical comms), the computational load may exceed what any programmable DSP can handle within one sample period.

Rule of thumb: Use DSP where ADCs can keep up AND computation fits within the sample period.

Real-Time Processing

- Definition depends on the application: processing must complete before the next sample arrives.

Example: 100-tap FIR filter must complete between two samples:

$$y(n) = \sum_{k=0}^{99} a(k)x(n-k)$$

8 kHz speech	44.1 kHz audio	192 kHz pro	1 GHz radar
125 µs /sample	22.7 µs/sample	5.2 µs/sample	1 ns/sample

What are the typical DSP algorithms?

Algorithm	Equation
Finite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)$
Infinite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)+\sum_{k=1}^{N} b_k\, y(n-k)$
Convolution	$y(n)=\sum_{k=0}^{N} x(k)h(n-k)$
Discrete Fourier Transform	$X(k)=\sum_{n=0}^{N-1} x(n)\exp[-j(2\pi/N)nk]$
Discrete Cosine Transform	$F(u)=\sum_{x=0}^{N-1} c(u).f(x).\cos\left[\frac{\pi}{2N}u(2x+1)\right]$

- *The **Sum of Products (SOP)** is the key element in most DSP algorithms !!!*

Typical DSP Algorithms — Sum of Products (SOP) is Key

Finite Impulse Response (FIR)

$$y(n) = \Sigma\, a_k \cdot x(n-k) \quad [k=0..M]$$

M+1 multiplications + M additions per output sample. Always stable. Linear phase possible.

Infinite Impulse Response (IIR)

$$y(n) = \Sigma\, a_k \cdot x(n-k) + \Sigma\, b_k \cdot y(n-k)$$

Recursive filter — past outputs feed back. Fewer taps than FIR but stability must be verified.

Convolution

$$y(n) = \Sigma\, x(k) \cdot h(n-k) \quad [k=0..N]$$

Core operation for filtering, echo modelling, image processing and system identification.

Discrete Fourier Transform (DFT/FFT)

$$X(k) = \Sigma\, x(n) \cdot \exp[-j2\pi/N \cdot nk]$$

Frequency-domain analysis. FFT reduces $O(N^2)$ to $O(N \cdot \log_2 N)$ — 100× faster at N=1024.

Discrete Cosine Transform (DCT)

$$F(u) = \Sigma\, c(u) \cdot f(x) \cdot \cos[\pi/2N \cdot u(2x+1)]$$

Used in JPEG, MPEG, MP3, H.264 compression. Real-valued — no imaginary component.

★ The Sum of Products (SOP) operation is the fundamental building block of all DSP algorithms

Hardware vs. Microcode Multiplication

- DSP processors include dedicated hardware multipliers that complete a multiply and add (MAC) in ONE clock cycle.

Hardware Multiplier

Example: 4-bit unsigned 1011 × 1110

```
  1011
× 1110
------
10011010
```

Result in ONE clock cycle (dedicated combinatorial hardware)

Texas Instruments TMS32010 (1982)
First DSP with single-cycle hardware multiplier

Microcode Multiplication

Same 4-bit problem — multiple cycles:

Cycle 1	1011 × 0 = 0000
Cycle 2	1011 × 1 → 1011.
Cycle 3	1011 × 1 → 1011..
Cycle 4	1011 × 1 → 1011...
Cycle 5	Sum → 10011010

5 cycles for one multiplication!

What Is Special for DSPs? (Part 1)

Optimized for one program

- Unlike GPUs/CPUs, *DSPs typically run a single algorithm continuously*. No OS virtualization, no memory protection overhead — all cycles go to computation.

Hard real-time constraints

- You must account for every interrupt/ exception and subtract it from the available computing time. Exceptions ARE BAD in DSP — they steal computing time.

Continuous data stream

- DSPs process an infinite stream from the environment (e.g. audio at 48 kHz = one new sample every 20.8 µs). Computation must finish before the next sample.

Performance metric: MIPS/MFLOPS = MAC rate

- How busy the multiplier is, determines the judgment of DSPs.
- Standard benchmarks: FIR, IIR, FFT, convolution.

What Is Special for DSPs? (Part 2)

- The algorithms they run shape DSP architectures. Seven key features differentiate DSPs from general-purpose processors (GPP):

Fast Multipliers

Dedicated single-cycle MAC hardware

Multiple Ex. Units

Parallel ALU + MAC + shifter

Efficient Memory Access

Harvard architecture, dual-bus

Data Format

Fixed-point vs floating-point

Zero-Overhead Looping

Hardware loop control, no branch penalty

Streamlined I/O

DMA, serial ports, on-chip ADC/DAC

Specialised ISA

Parallel ops/tasks in a single instruction

Architectural Feature 1 — Fast Multipliers

- Multiplication is the most common operation in signal processing: convolution, FIR/IIR filtering, FFT…

1982	1990s	2000s	2010s	2020s
TI TMS32010	**16×16 MAC**	**32×32 MAC**	**SIMD MACs**	**Tensor cores**
First single-cycle hardware multiplier in a commercial DSP	16-bit × 16-bit = 32-bit result in 1 cycle, standard on all DSPs	32-bit precision for audio/radar; dual-MAC units emerge (2 MACs/cycle)	Vector MACs process 4–8 samples concurrently (VLIW + SIMD)	8/16-bit matrix MACs for AI inference, 1000+ MACs/cycle

- **All modern DSPs include *at least one dedicated single-cycle MAC unit*.**

Formula: y += a[k] * x[n-k] ← one cycle

- Higher-end devices feature 2–8 MAC units operating in parallel per clock cycle.

Architectural Feature 2 — Multiple Execution Units

- Real-time DSP requires HIGH computational throughput. Multiple independent execution units operate in parallel.

MAC Unit	ALU	Barrel Shifter	Address AGU	I/O DMA
Multiply-Accumulate Core DSP workhorse 32/64-bit accumulator	Arithmetic Logic Unit Add/sub/ Compare/ Bit logic operations	Bit-shift operations Scaling & normalising Single-cycle shifts	Address Generation Post-increment pointers Circular buffer control	Direct Memory Access Background transfers No CPU stall

Parallel execution in a single clock cycle:

MAC: y += a[k] * x[n-k] || ALU: n++ || Shift: scale result || AGU: update pointers

Architectural Feature 3 — Efficient Memory Access

- A single-cycle MAC requires three simultaneous memory accesses:
- instruction fetch + 2 data operands.

Von Neumann (General CPUs)

- Single shared bus
- Instruction OR data each cycle
- Cannot fetch instr. + 2 operands simultaneously
- Multiple cycles per MAC

→

Harvard (Early DSPs)

- Separate program & data buses
- Fetch instruction + 1 data operand per cycle
- Significant improvement over von Neumann
- Still limited for dual-operand MACs

→

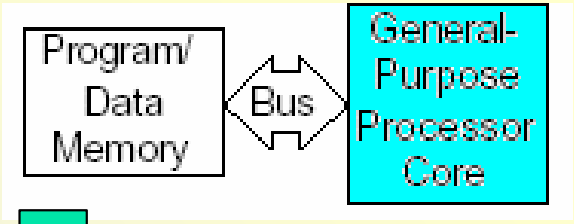
Modified Harvard (Modern DSPs)

- Separate program, data-A, data-B buses
- Fetch instr + 2 operands in ONE cycle
- Instruction cache for program memory
- 3–5× throughput vs. von Neumann

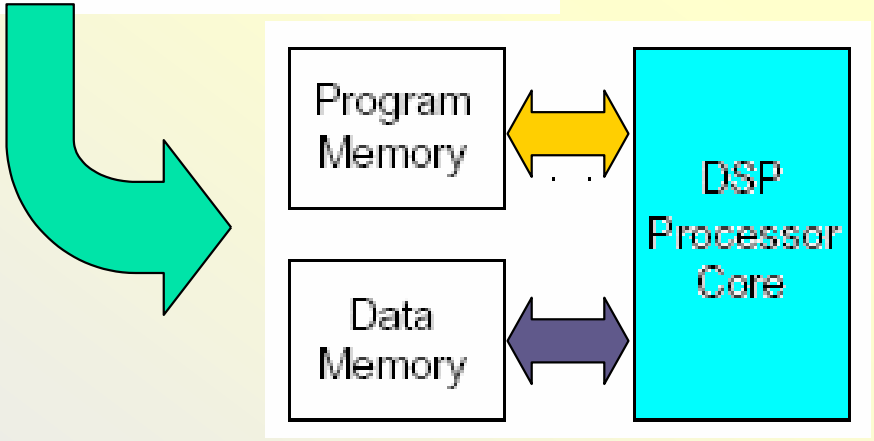
Modern DSPs also use: circular buffers · bit-reversed addressing · dedicated (AGU) Address Generation Units

Efficient Memory Access

General purpose processors



SISD (von Neumann)

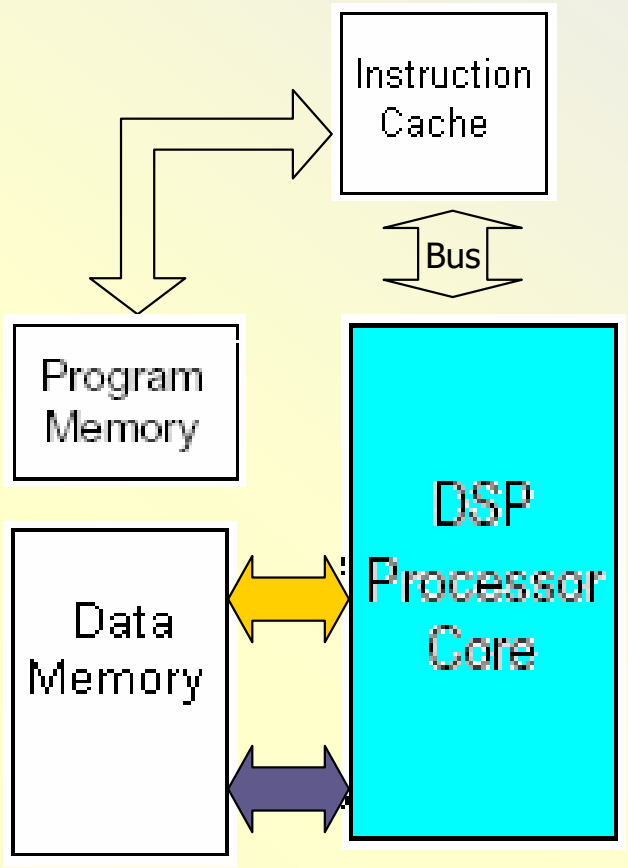


HARVARD

Early DSP processors

OR

More optimized DSP processors



HARVARD modified

High Memory Bandwidth — Addressing Techniques

- DSP memory access patterns are highly predictable. Specialized addressing modes exploit this for zero-overhead data fetch.

Register Indirect + Post-Increment

- The address pointer automatically increments after each access. Ideal for traversing data arrays without explicit loop counter updates.

Example (TMS320):
MAC *AR1+, *AR2+, A
— reads x[n], h[k], then * and increments both pointers in one instruction.

Circular Addressing

- Hardware wraps the pointer back to buffer start when it reaches the end.
  - Essential for delay lines in FIR/IIR filters.
  - No software overhead, the AGU handles the modulo operation in one clock cycle.
  - Supports multiple independent circular buffers concurrently.

Bit-Reversed Addressing

Swaps bit order of address index for radix-2 FFT butterfly re-ordering. Eliminates the $O(N \cdot \log_2 N)$ software bit-reversal pass.

Example: address 001 → accesses location 100. Implemented in dedicated AGU hardware — zero software cost.

Architectural Feature 4 — Data Format

- The data word width *determines accuracy, cost, power consumption, and hardware complexity*.

Property	Fixed-Point (16/24-bit)	Floating-Point (32-bit)
Representation	Integer or fractional Q-format, Q15	IEEE 754 mantissa + exponent
Dynamic range	96 dB (16-bit) / 144 dB (24-bit)	~1500 dB (32-bit)
Hardware cost	Simple — cheap silicon	Complex — expensive silicon
Power	Low — ideal for battery devices	High — ~10× more than fixed
Overflow risk	Must scale carefully	Very large range prevents overflow
Programming	Harder — programmer scales data	Easier — HW handles range
Use case	Mobile, IoT, consumer DSPs	Audio workstations, instruments

Most fixed-point DSPs use 16-bit words. Accumulator registers are wider (32/40-bit) to prevent overflow during MAC chains.

Architectural Feature 5 — Zero-Overhead Looping

- DSP algorithms are loop-dominated. Zero-overhead looping eliminates the branch penalty and counter update overhead.

❌ **Conventional Loop (CPU-style)**

```
for (k=0; k<N; k++)
{                              ; update counter
  y += a[k] * x[n-k];  ; MAC
  if (k==N) break;     ; check condition
  goto loop_start;     ; branch back
}
```

Overhead per iteration:
counter++ + compare + branch =
3 extra instructions wasted per loop cycle!

✓ **Zero-Overhead Loop (DSP)**

```
RPTB  loop_end-1      ; set hardware counter
MAC  *AR1+, *AR2+, A  ; loop body
loop_end:             ; hardware auto-repeats
```

Loop hardware:
- Dedicated loop counter register
- No branch instruction needed
- No condition check overhead
- Counter decrements silently in HW

Result: 100-tap FIR → 100 cycles exactly. No wasted cycles. Full MAC throughput.

Architectural Feature 6 — Streamlined I/O

- Specialized I/O reduces CPU interrupt overhead and keeps the DSP core computing instead of waiting for data.

DMA Controller

Transfers data between memory and peripherals in the background — CPU never stalls.

Sync Serial Ports

McBSP / SPI / I²S for audio codecs, ADCs, DACs — hardware-timed, interrupt-free.

Parallel Host Port

High-speed parallel interface to host processors for command/ status exchange.

On-chip ADC/DAC

Integrated converters eliminate external components and reduce board-level latency.

Timers & Clocks

Programmable timers for sample-rate generation and watchdog supervision.

Bit I/O Ports

GPIO lines for control signals, indicator LEDs, mode selection switches.

On-chip peripherals often operate during core SLEEP states — data arrives ready in buffer when the DSP wakes.

Architectural Feature 7 — Specialised Instruction Set

- DSP ISAs are compact and encode maximum hardware parallelism. Historically, are written in assembly - not C.

Goal 1: Maximum hardware utilization

Programmers specify parallel operations in a single instruction.
Example*: MAC + memory fetch + address update all in ONE opcode.*
Reduces cycle count for inner loops dramatically.

Goal 2: Minimize program memory footprint

Memory = cost.
Instructions kept short via:
- Mode bits rather than explicit encoding
- Restricting operations to specific registers
- Limiting instruction-level parallelism combinations

Result: complex but space-efficient instruction words.

Assembly Language is still used in production DSP code for inner loops:

Pure C	Fast to write, 70-80% optimal
C + Intrinsics	85-95% optimal, portable
Assembly	100% optimal, smallest ROM

DSP Applications Development Environment

DEBUGGING In-chip emulators: Pod-based | Scan-based (JTAG, OnCE)

🔧 Standard IDE Approach

Integrated Development Environment (IDE)

- TI Code Composer Studio (CCS) — most popular
- Languages: Assembler · C · C++ · Ada
- RTOS integration (TI-RTOS, FreeRTOS)
- Built-in profiler & cycle-accurate simulator
- Hardware-in-the-loop debugging via JTAG
- URL: https://www.ti.com/tool/CCSTUDIO

🚀 Advanced / Rapid Prototyping

MATLAB DSP System Toolbox

- Auto-generate C/assembly from Simulink
- RIDE / Hyperception visual real-time design
- Python (scipy.signal) for algorithm research
- FPGA-in-the-loop for ultra-high-speed testing
- JTAG + Lauterbach TRACE32 trace capture

Choosing a DSP Processor

- The right DSP depends on the application. *A choice for one task may be a poor choice for another.*

Arithmetic Format	Data Width	Speed (MIPS)	Memory
Fixed vs floating-point Q-format resolution	16 / 24 / 32-bit Accumulator guard bits	Clock frequency MPEG/FIR benchmarks	On-chip RAM/ROM size External bus bandwidth

Development Ease	Multiprocessor	Power Consumption	Cost
Compiler quality IDE & debugger tools	Inter-DSP links Shared memory support	mW/MIPS metric Sleep/wake modes	Unit price Development toolchain

Main Architectural Features of DSPs — Summary

Data Path

- ✓ Fixed-point arithmetic
- ✓ Dedicated MAC units
- ✓ 32/40-bit accumulators

Memory System

- ✓ Harvard architecture
- ✓ Multiple data memory blocks
- ✓ Instruction cache

Addressing Modes

- ✓ Bit-reversed (FFT)
- ✓ Circular buffers (FIR delay line)
- ✓ Post-increment indirect

Instruction Execution

- ✓ Zero-overhead loops
- ✓ VLIW parallel ops
- ✓ MAC in single cycle

Peripherals

- ✓ Serial ports (McBSP/SPI/I2S)
- ✓ DMA for background transfers
- ✓ On-chip ADC/DAC/timers

DSP Performance Metrics — Often Misleading

- Vendor-quoted figures are typically peak values under ideal conditions.
- Use benchmark data (kernel / application) for real comparisons.

Max Clock Frequency	**[MHz]**	Higher ≠ better if pipeline is inefficient
Power Consumption	**[W or W/MIPS]**	Critical for battery/embedded — compare W/MIPS
Execution Throughput	**[MIPS, MOPS]**	Peak MIPS ≠ sustained MIPS on real algorithms
Memory Bandwidth	**[MB/s]**	Bus width × clock; bottleneck for large FFTs
Memory Latency	**[clock cycles]**	Cache misses destroy real-time determinism
I/O Capacity	**[Port count]**	Serial/parallel ports, DMA channels available

⚠ **Use kernel benchmarks (FIR, FFT, IIR) on your specific algorithm — not headline MIPS figures.**

DSP Market Overview

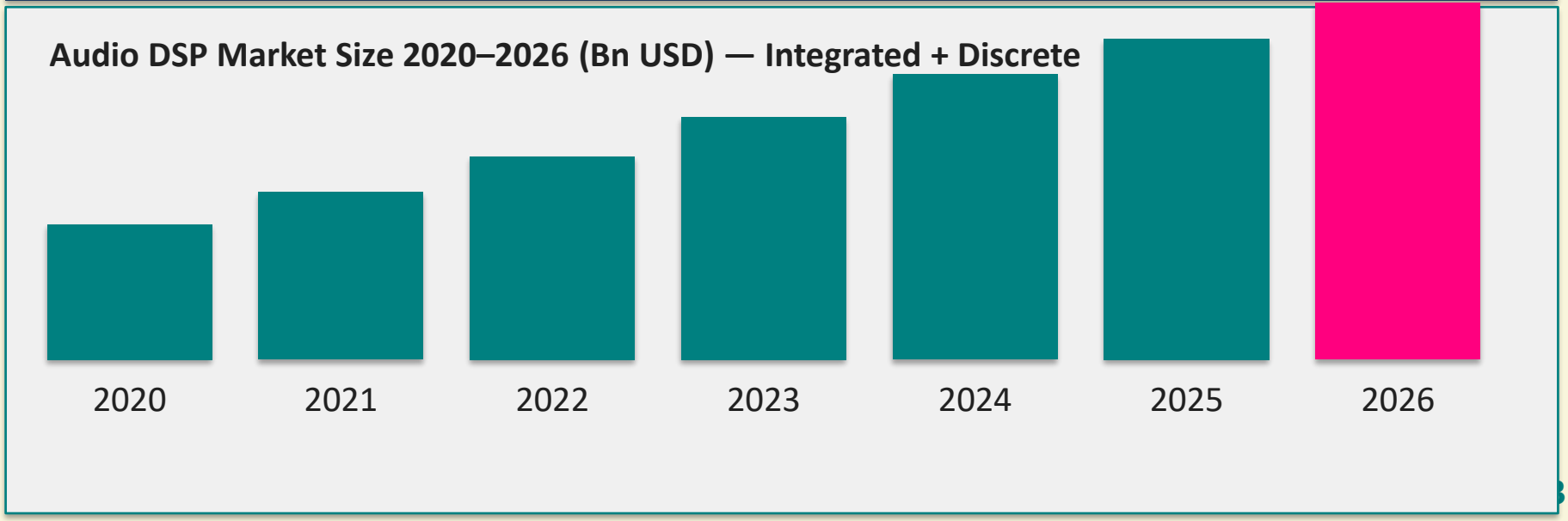
$17.7 Bn

Audio DSP market (2026 forecast)
Source: KBV Research

~25%	~15%	~18%	~12%
Phones	IoT Devices	Home Entertain.	True Wireless

~10%	~8%	~7%	~5%
Smart Homes	Wearables	Computers	Others

KEY MANUFACTURERS	Texas Instr. (48%)	Lucent Tech. (25%)	Motorola (11%)	Analog Devices (10%)	Others (5%)



Audio DSP Market — Key Manufacturers

- A few large semiconductor companies dominate the DSP market, with Texas Instr. historically leading.

Texas Instruments	Analog Devices	Qualcomm	NXP Semiconductor
TMS320 family leader>48% market share (1999)	SHARC & Blackfin families High-perf audio/comms	Hexagon DSP in Snapdragon Mobile & AI edge	i.MX application processors Automotive & IoT

CEVA Inc.	Renesas Electronics	Intel / Altera	Xilinx / AMD
Licensable DSP/AI IP cores used in 100s of chips	RA/RX families Industrial & automotive	FPGA with DSP blocks High-speed data centers	Versal AI engine FPGA+DSP hybrid SoCs

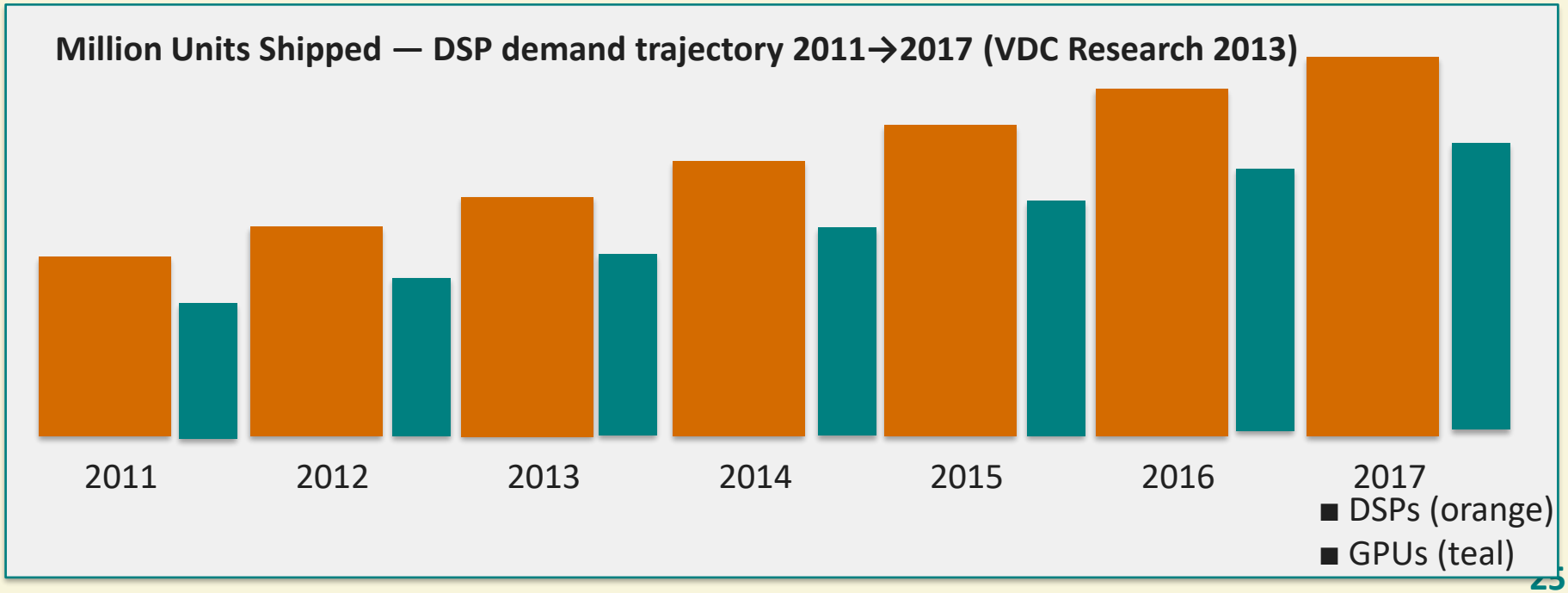
Broadcom	Samsung Electronics	Toshiba	Infineon Technologies
Cable/DSL modems Server storage DSPs	Exynos DSP cores Consumer SoCs	Bridging DSP ICs Power electronics	Automotive DSPs Functional safety

Company	Est. Share	Strength
Texas Instruments	~19.5%	TMS320 family, automotive & industrial
Analog Devices	~16.2%	SHARC/Blackfin cores, RF & audio DSP
Qualcomm	~15.3%	Hexagon DSP embedded in Snapdragon SoCs
Intel	~7.8%	Xtensa-based & AI DSP cores
NXP	~5.4%	Automotive & IoT
Broadcom	~4.8%	Optical PAM4 DSP for AI datacenters
Renesas	~3.9%	5G telecom
STMicroelectronics	~3.2%	Mixed-signal & audio
Cirrus Logic	~2.7%	Premium audio specialist
Others	~21.2%	Infineon, MediaTek, Samsung, CEVA IP…

Forecasted Embedded Processor Shipments

- DSPs are the fastest-growing embedded processor category.
- Compound Annual Growth Rate (CAGR) driven by wireless, AI and IoT.

DSPs	Media Processors	GPUs	MCUs
Fastest CAGR IoT · 5G · AI edge	Second fastest Video · streaming	Moderate CAGR ML acceleration	Steady growth Embedded control



Texas Instruments TMS320 : Different Needs, Multiple Families

'C2000 ('C20x, 'C24x)	'C5000 ('C54x, 'C55x)	'C6000 ('C62x, 'C67x)
Lowest Cost Control Systems	**Best MIPS per Watt / Dollar / Size**	**Maximum Performance Multi-Channel**
• Motor control	• Wireless telephones	• Communications infrastructure
• Digital power supplies	• Modems / Telephony	• xDSL broadband
• Storage devices	• VoIP / Video call	• Imaging & video
• Process control	• Portable media	• Radar/sonar

Earlier generations: 'C1x, 'C2x, 'C3x, 'C4x, 'C5x, 'C8x — legacy devices still in production for cost-sensitive applications

TI C5000 Family Roadmap — Power-Optimised DSPs

- The C5000 family targets maximum MIPS/mWatt: ideal for wireless handsets, portable audio, and VoIP equipment.

Telecom / VoIP / Infrastructure	Portable Media / Communication / Audio	Ultra-Embedded / Low-Power / Biometrics
▶ C5510 (320–400 MIPS)	▶ C5503 (400 MIPS)	▶ C5401 (50 MIPS)
▶ C5502 (400–600 MIPS)	▶ C5507 (400 MIPS)	▶ C5402 (100–160 MIPS)
▶ C5501 (600 MIPS)	▶ C5509A (400 MIPS + USB)	▶ C5404 (120 MIPS)
★ **C55x Next (future)**	★ **C55x Next (future)**	★ **C5000 Next (future)**

Y-axis: Power efficiency / peripheral integration increases upward. Production (white) vs. Future (coloured) devices.

TI ARM® Cortex™-A8 MPU + DSP SoC Roadmap

- TI's DaVinci and Sitara families combine ARM application processors with DSP cores for multimedia and embedded Linux applications.

OMAP 3530/25

Cortex-A8 + DSP + GPU

Production — mobile multimedia SoC

OMAP 3503/15

Cortex-A8 / Cortex-A8+GPU

Production — industrial/medical

DaVinci DM Next

DSP + Cortex-A8 + GPU

In development — video encode/decode

Sitara AM37xx

Cortex-A8 + GPU

In development — HMI + motor control

Sitara AM3517

Cortex-A8 + GPU

Sampling — industrial Linux platform

TI processors share common foundation software and tools to ensure portability and compatibility across families.

TI ARM9® MPU and C674x DSP Roadmap

- The OMAP-L13x and C674x families target cost-sensitive, battery-operated and industrial applications.

C674x Pure DSP Family

C6743	Fixed/Float DSP — entry-level	●
C6745/7	Fixed/Float DSP — mid-range	●
C6746/8/2	Fixed/Float DSP — high-perf	●
C674x Next	Fixed/Float	★

OMAP-L Series (ARM9 + C674x)

OMAP-L137	C674x + ARM9
OMAP-L138	C674x + ARM9
Sitara AM Next (ARM9+Audio)	ARM9 + Audio block
Sitara AM Next (ARM9+FF)	ARM9 + Fixed function
DSP + ARM Next	C674x + Cortex-A8

All TI processors share common SDK, compiler, and debugger toolchain for code portability.

Architectures for Developing DSp Applications

- Five implementation options for DSp systems - each trades flexibility against performance and cost.

ASIC Application-Specific IC	FPGA Field Programmable Gate Array	ASSP App-Specific Standard Products	GPP General-Purpose Processor	DSP Digital Signal Processor
Maximum performance & efficiency	Reconfigurable hardware logic	Fixed-function chip for a market	Maximum programmability (C++)	Optimized for MAC-dominated tasks
Full custom silicon, zero flexibility	More efficient than SW, less than ASIC	Large volume, moderate flexibility	Market-agnostic, easy to develop	Good balance of efficiency & flexibility
Highest NRE cost, lowest unit cost	Harder to program (VHDL/Verilog)	Between ASIC cost and DSP versatility	Highest power for given performance	Assembly/C + vendor IDE tools
Used for: WiFi chips, codec ICs	Used for: prototyping, baseband	Used for: cable modems, set-top boxes	Used for: PC audio, server workloads	Used for: phones, radar, hearing aids…

← Flexibility / Programmability Performance / Efficiency →

DSP Application — Mobile Phone Signal Chain

TRANSMIT PATH — Low sampling rate: 8 kSPS / 13-bit precision

A/D Converter → Voice Coding → Channel Coding & Interleaving → Encryption → Modulation → D/A Converter

◄— DSP Processing Core —►

RECEIVE PATH — High sampling rate: 20 MSPS / < 8-bit

D/A Converter → Voice Decoding → Channel Decoding & Deinterleaving → Decryption → Demod. & Equalisation → A/D Converter

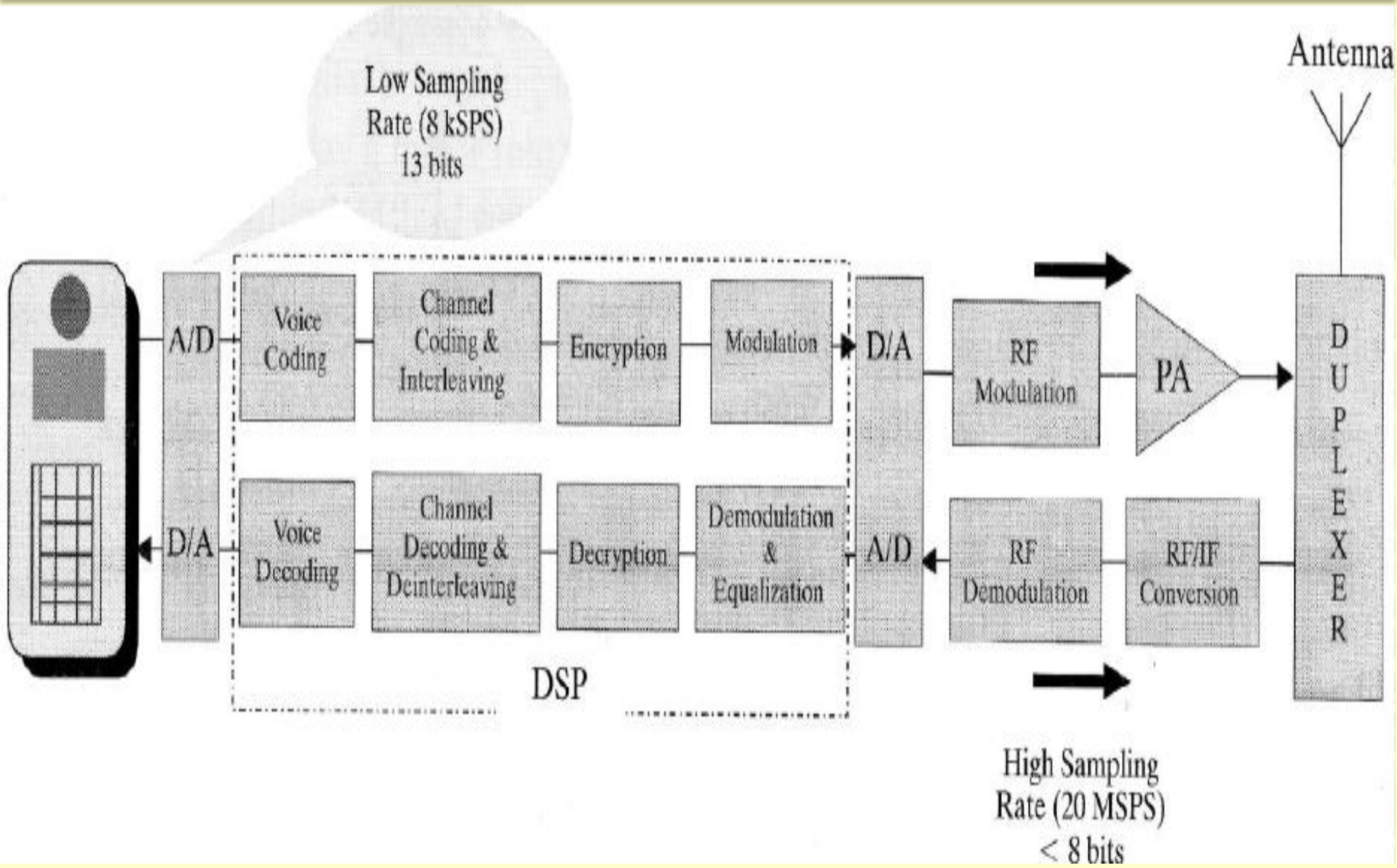
RF Frontend (Analogue)

RF Modulation / Demodulation → Power Amplifier → Duplex Antenna

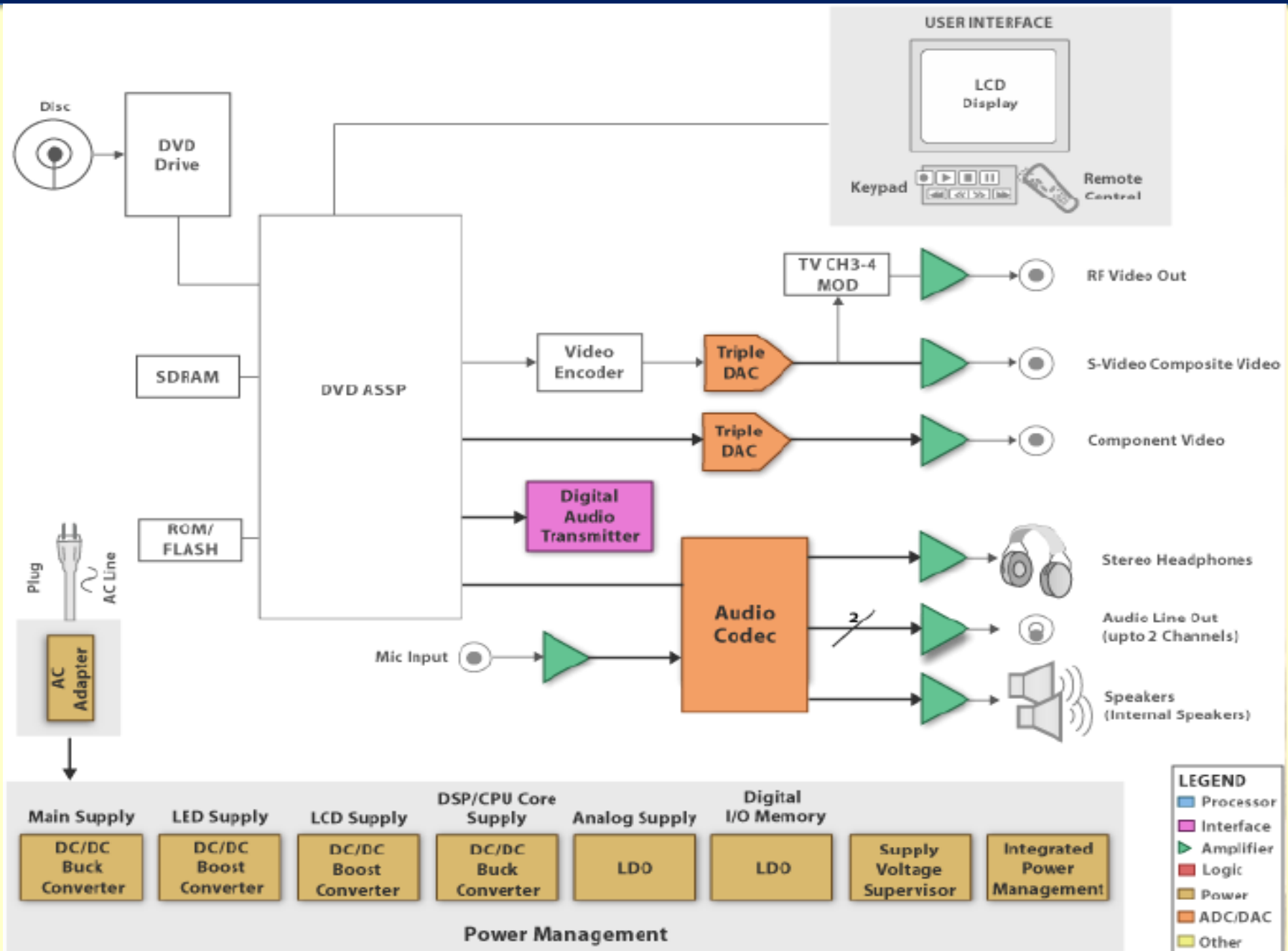
Why DSP for mobile?

All coding, encryption, modulation, echo cancel and equalization run simultaneously on ONE DSP core in < 125 µs (8 kHz voice)

DSP Application — Mobile Phone Signal Chain



DSP Application — DVD Player Architecture



DSP in Medical Imaging

Ultrasound

Beamforming delay-and-sum
30+ fps B-mode imaging
Doppler blood flow detection
Custom DSP ASICs + FPGA

MRI

3D FFT k-space reconstruction
Full pipeline from RF to image
Artifact removal (ghosting)
fMRI time-series analysis

ECG Analysis

QRS detection (matched filter)
Pan-Tompkins algorithm
Arrhythmia classification NN
Wearable: < 1 mW DSP core

Key Challenges

Real-time constraints
High precision required
Regulatory certification
Wearable power limits

Future Trends

AI-assisted diagnostics on DSP/NPU
Portable ultrasound on mobile SoC
Implantable DSPs for neural interfaces
Edge AI reduces cloud dependency

DSP in Audio Processing

Active Noise Cancellation

FIR/IIR anti-noise filter
Feedforward + feedback topology
< 1 ms latency requirement
Used in headphones, aircraft

Echo Cancellation

LMS / RLS adaptive filter
Convergence in < 1000 samples
Full-duplex speakerphone
VoIP & conference systems

Audio Equalisation

10–31 band IIR biquad filters
Parametric / graphic EQ
Car audio, home cinema
Venue acoustic correction

Speech Compression

G.711, G.729, AMR codecs
LPC (Linear Predictive Coding)
8× compression vs PCM
Mobile voice calls

Spatial Audio

Convolution reverb (HRTF)
Thousands of taps per channel
3D sound field synthesis
VR headsets & gaming

Pitch & Time Stretch

PSOLA / Phase vocoder
Independent pitch & tempo
Music production & karaoke
Speech rate conversion

DSP Applications — Algorithms & System Uses

DSP Algorithm	System Application
Speech Coding	Digital cellular, personal comms, cordless phones, multimedia, secure comms
Speech Encryption	Cellular phones, secure military communications, privacy systems
Speech Recognition	Advanced user interfaces, automotive voice control, robotics, call centres
Speech Synthesis	Navigation, text-to-speech, accessibility tools, voice assistants
High-Fidelity Audio	Consumer audio, digital broadcast, professional studios, multimedia PCs
Modems	Cable TV, wireless computing, navigation, digital audio/signalling, fax
Noise Cancellation	Professional audio, headphones (ANC), vehicular audio, industrial
Audio Equalisation	Consumer & professional audio, vehicular audio, music production
Image Compression	Digital photography, video conferencing, streaming (JPEG/H.264)
Beamforming	Medical ultrasound, radar/sonar, smart speakers, 5G massive MIMO
Echo Cancellation	Speakerphones, hands-free cellular, conference systems
Spectral Estimation	Radar/sonar, signals intelligence, professional audio, music

Modern DSP Trends

AI & Machine Learning

- Neural network inference on-device
- Keyword spotting & wake-word detection
- TinyML: <1mW always-on audio AI
- On-device NLP for privacy

5G & Wireless

- Massive MIMO beamforming DSP
- mm-wave channel estimation
- OFDM modulation at 1000+ carriers
- Software-defined radio (SDR)

Heterogeneous Computing

- DSP + CPU + GPU in single SoC
- FPGA hybrid for reconfigurability
- NPU / tensor-core co-processors
- Unified memory architecture

Automotive & Safety

- ADAS sensor fusion DSP
- Radar signal processing (77 GHz)
- In-cabin voice + audio
- ISO 26262 ASIL-D safe DSP cores

DSP in AI / ML Edge Intelligence

- Modern DSPs boost the computer power for AI on devices, providing intelligence with very little energy.

Keyword Spotting

MFCC feature extraction
Small RNN / CNN classifier
< 1 mW power budget
Ex.: 'Hey Siri', 'OK Google'

Image Recognition

Convolution layers on DSP
INT8 quantization
MobileNet / EfficientNet-Lite
Real-time 30 fps inference

Predictive Maintenance

Vibration FFT analysis
Anomaly detection ML
Bearing fault classification
< 10 µs latency decision

Gesture & Radar Sensing

60 GHz micro-gesture radar
Range-Doppler DSP
Gesture CNN classifier
Smartphone / wearable

Medical Wearables

ECG / PPG signal processing
Arrhythmia detection NN
Continuous monitoring
< 0.5 mW on wrist

Natural Language

Beamforming + noise cancels
Online speaker diarization
Speech-to-text on DSP
Edge privacy — no cloud

DSP — Summary & Future Outlook

1. DSPs are optimised for MAC operations — the foundation of all signal processing.
2. Harvard architecture enables single-cycle MAC by separating instruction and data buses.
3. Fixed vs floating-point trade cost/power against dynamic range and ease of programming.
4. Zero-overhead looping, circular addressing, and VLIW give DSPs their speed edge.
5. DSPs are everywhere: phones, medical, radar, audio, comms, automotive, and AI edge.

AI-integrated DSP | **Neuromorphic computing** | **DSP for 6G** | **Quantum-enhanced SP** | **Sub-mW always-on AI**