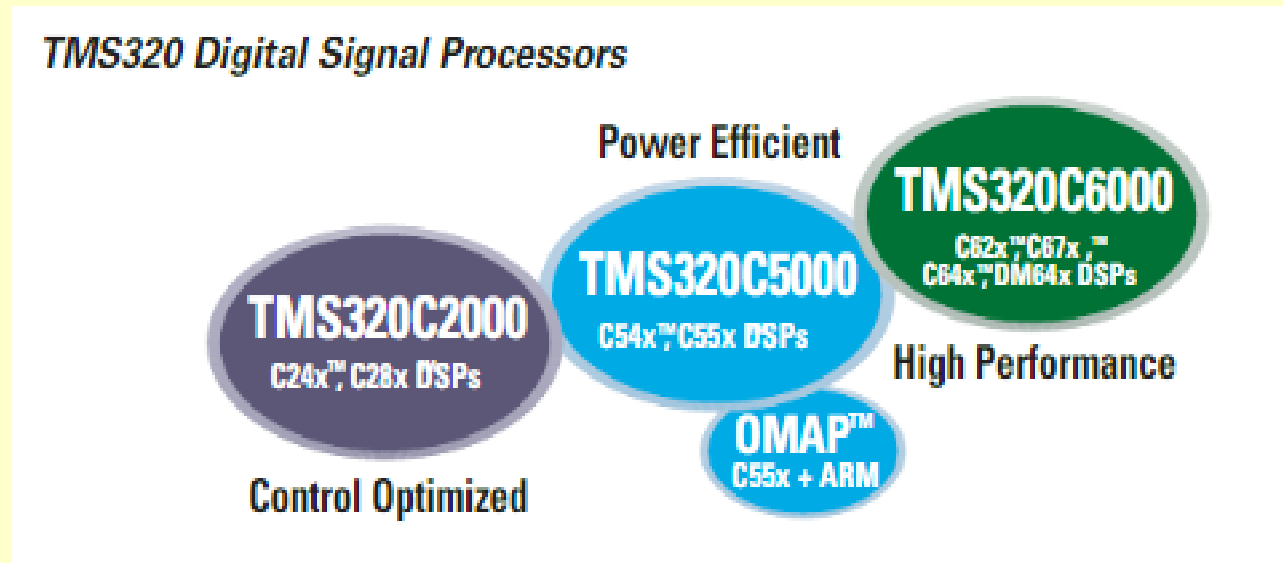


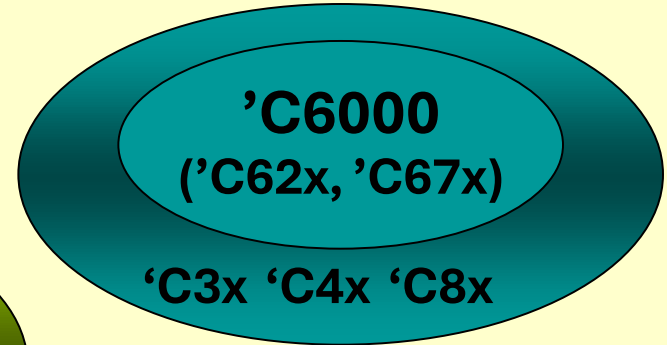
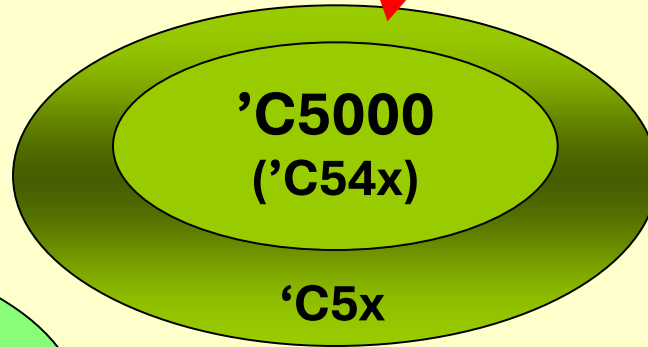
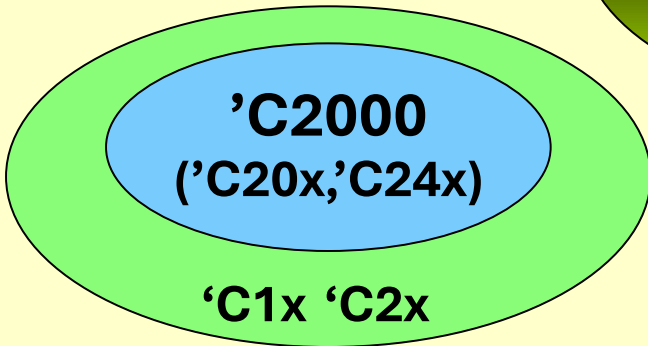
Lecture 8

TI C5000 DSP Family overview & OMAP



Reference: Texas Instruments

Different Needs? Multiple Families! (former approach)



Maximum Performance
Multi-Channel / Function

- Comm Infrastructure
- xDSL
- Imaging, Video

Best MIPS per:
Watt / Price / Size

- Wireless Telephones
- Modems / Telephony
- VoIP

Lowest Cost Control Systems

- Storage
- Motor Control

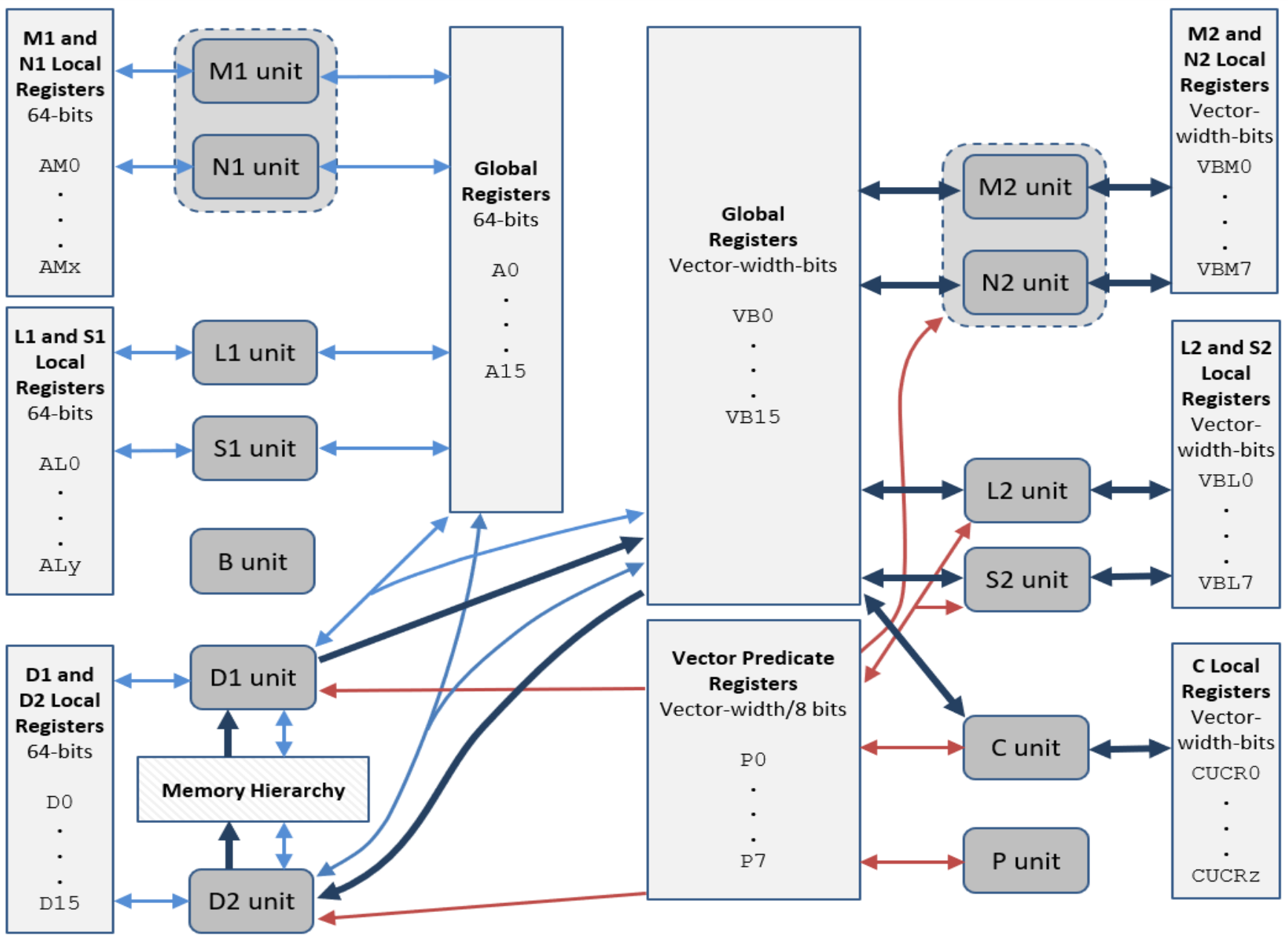
C2000 — still very active, now gaining AI . TI announced the TMS320F28P55x series of C2000 MCUs at Electronica 2024 — called the *industry's first real-time microcontrollers with an integrated neural processing unit (NPU)*. TI also revealed the new 64-bit F29H85x series built around its C29 DSP core targeting automotive applications.

C5000 —TI has absorbed C5000 devices like the C55x into legacy/EOL status, and TI *no longer positions them as a standalone active family*. The "Best MIPS/Watt" niche they once owned — wireless phones, modems, VoIP — *has been completely taken over by ARM Cortex-A/M Cores + SIMD extensions*.

C6000 — still active, but now extends to C7000.

The C7000 CPU DSP architecture is TI's *latest high-performance DSP*, featuring *VLIW design with wide vector instructions and multiple functional units*.

Texas Instruments launched the C7000 series in early 2020. They bundle the cores within SoCs such as the TDA4VM, which contains a C71x DSP, rather than making them available separately. [Texas Instruments](#) (Advanced Driver-Assistance Systems)



A datapath (scalar)

B datapath (vector)

↔ 64-bit data bus

↔ Vector-width data bus

↔ Vector predicate data bus

The **TDA4VM** is Texas Instruments' flagship **ADAS/autonomous driving SoC**, part of the **Jacinto 7** family. It's a heterogeneous multicore processor designed specifically for advanced vision and deep-learning inference at the edge — all while keeping power consumption in the 5–20 W range.

What's inside the TDA4VM:

The SoC is a textbook example of "right core for the right task" design. It packs together:

C71x DSP — this is the star from a signal-processing perspective. The C71x is TI's most advanced DSP generation, combining the traditional C66x fixed/floating-point DSP capabilities with a **vector processing unit** (512-bit SIMD). It can handle matrix multiplies, FFTs, and traditional DSP workloads but also accelerate parts of neural-network inference. It runs at up to ~1 GHz and delivers roughly 80 GMACs of scalar+vector compute. Think of it as a bridge between classical DSP and the emerging AI workload space.

MMA (Matrix Multiply Accelerator) — a dedicated deep-learning accelerator delivering up to ~8 TOPS (INT8). This is what handles the heavy CNN/DNN inference for object detection, semantic segmentation, and classification from camera feeds.

Dual Cortex-A72 — the application processors (64-bit, running Linux/QNX) that manage the high-level software stack, sensor fusion decision-making, visualization, and communication.

R5F cores (multiple) — there are several Cortex-R5F cores handling real-time, safety-critical tasks: MCU island for functional safety (ASIL-D capable), system-level control, and peripheral management.

VPAC (Vision Processing Accelerator) — dedicated hardware for lens distortion correction, noise filtering, stereo disparity, and optical flow.

DMPAC (Depth and Motion Processing Accelerator) — handles dense stereo depth estimation and motion compensation in hardware, offloading the main processors.

How the C71x fits in the processing pipeline:

In a typical ADAS camera pipeline on the TDA4VM, the data flows roughly like this: camera captures a frame → VPAC does lens correction, white balance, and noise reduction → the C71x DSP does pre-processing (resizing, color space conversion, custom filtering, feature extraction) → the MMA runs the neural network inference → the C71x or A72 does post-processing (non-maximum suppression, bounding-box refinement, tracking) → the R5F handles safety-critical output decisions and actuator interfaces.

The C71x is especially valuable because it's programmable — unlike the fixed-function MMA, you can write custom algorithms for it. This makes it the "glue" processor for anything that doesn't fit neatly into the CNN accelerator or the vision hardware.

Key specs and positioning:

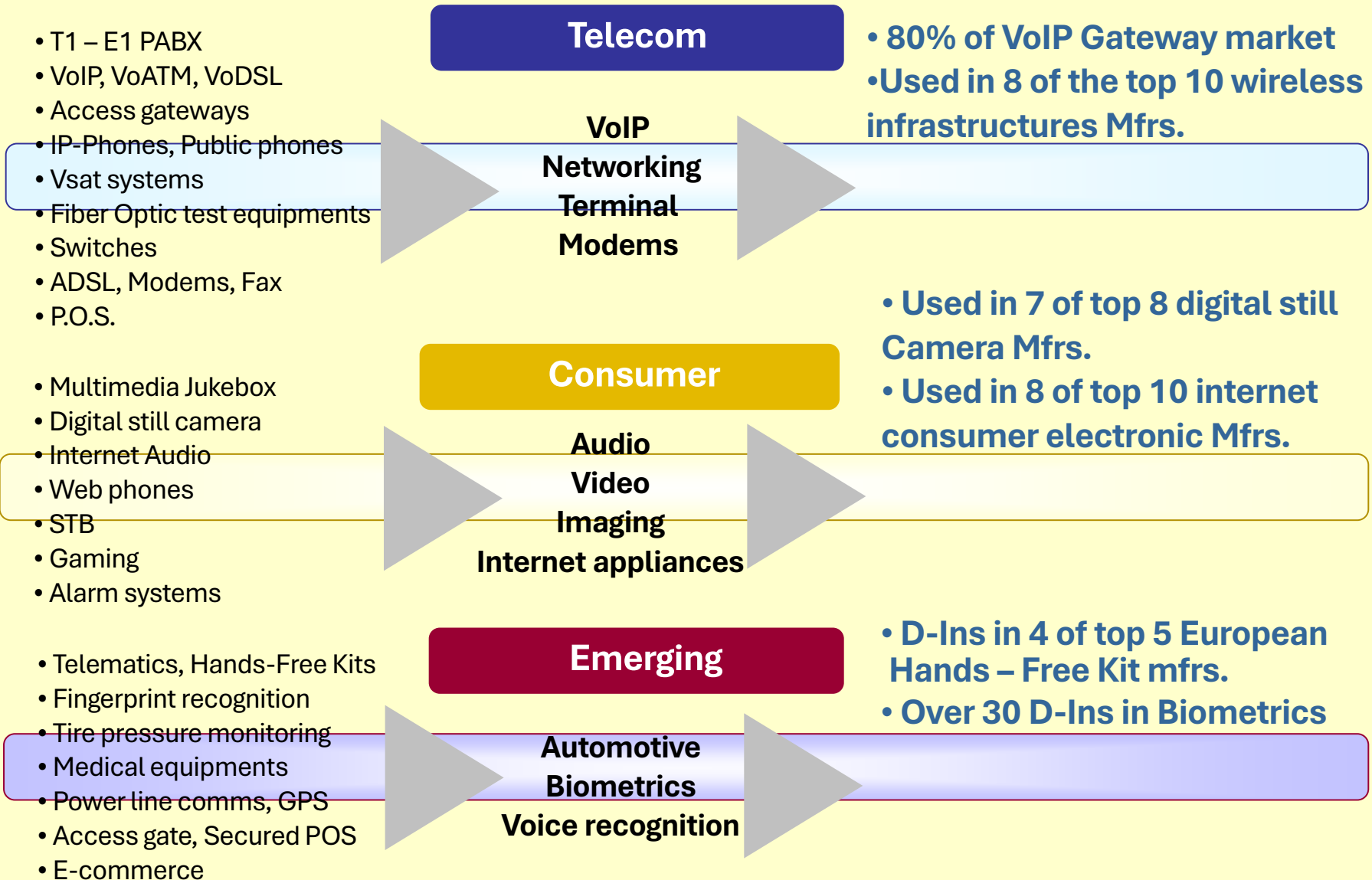
The TDA4VM targets Level 2+ to Level 3 autonomous driving with a power budget of around 5–20 W depending on workload — dramatically lower than GPU-based solutions like NVIDIA's Orin, which targets higher autonomy levels at 15–60 W. TI positions the TDA4VM for front cameras, surround-view, driver monitoring, parking assist, and sensor fusion applications where power efficiency is critical (think mass-market vehicles, not robotaxis).

In the context of what we discussed: the TDA4VM shows the evolution from standalone DSPs (like the C55x) toward deeply integrated SoCs where the DSP (C71x) coexists with application processors (A72), real-time controllers (R5F), and AI accelerators (MMA) — each optimized for its part of the workload.

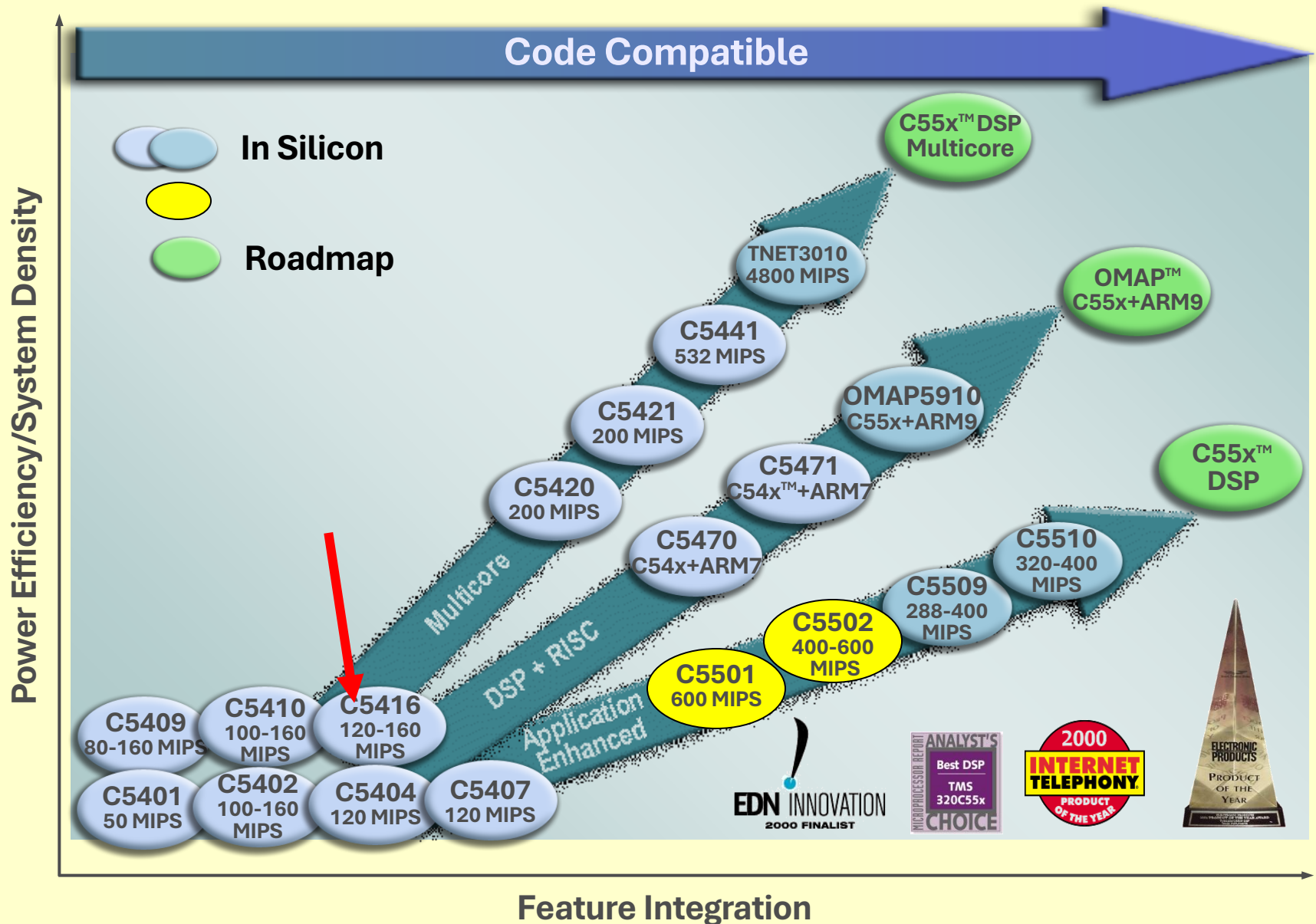
'C5000 DSP Market Segments Targeted

> 500 mil. C5000 DSPs shipped in more than 1000 different applications

End Equipments



TI Extends World's Most Popular DSP Platform

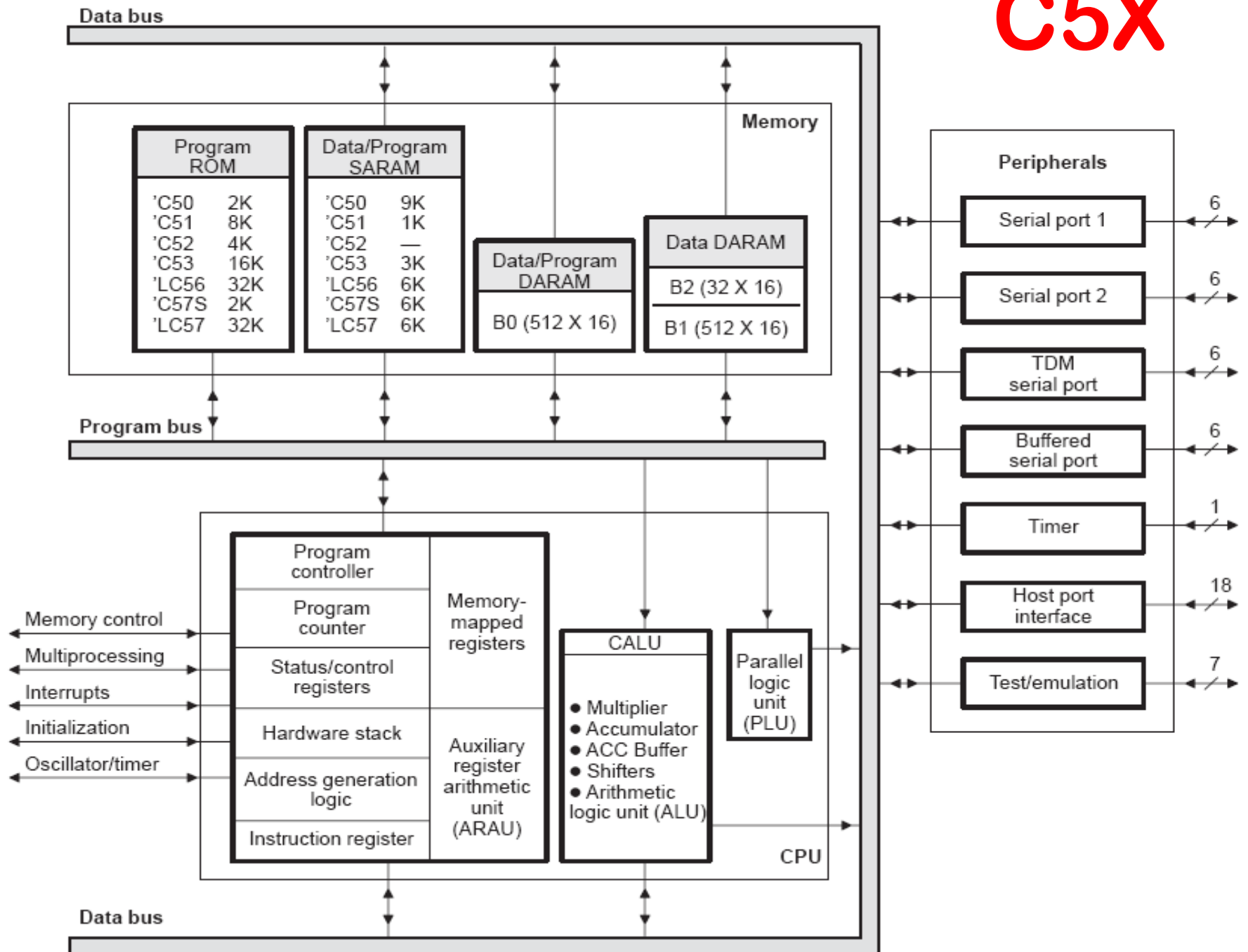


C54xx

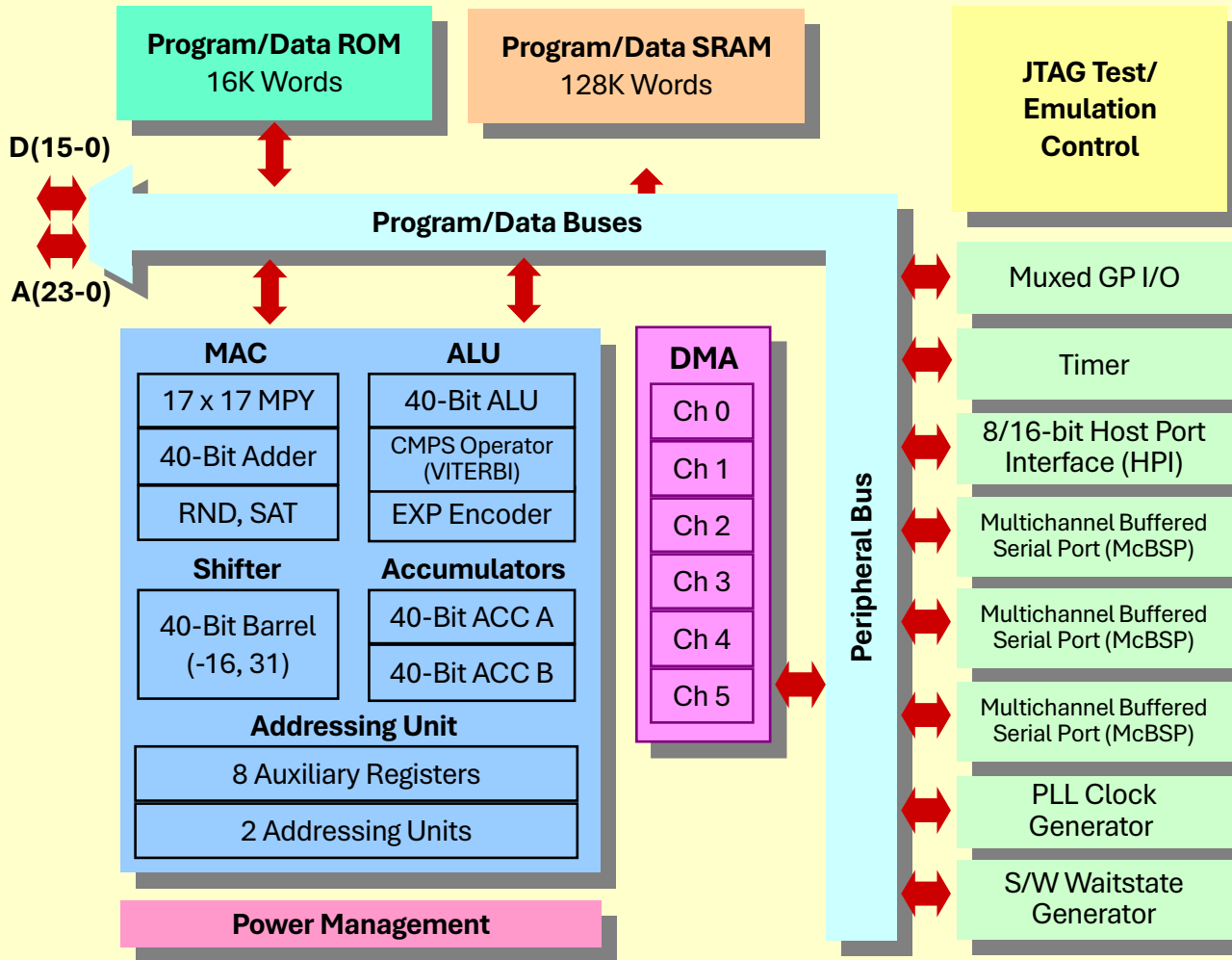


Figure 2-1. 'C5x Functional Block Diagram

C5X



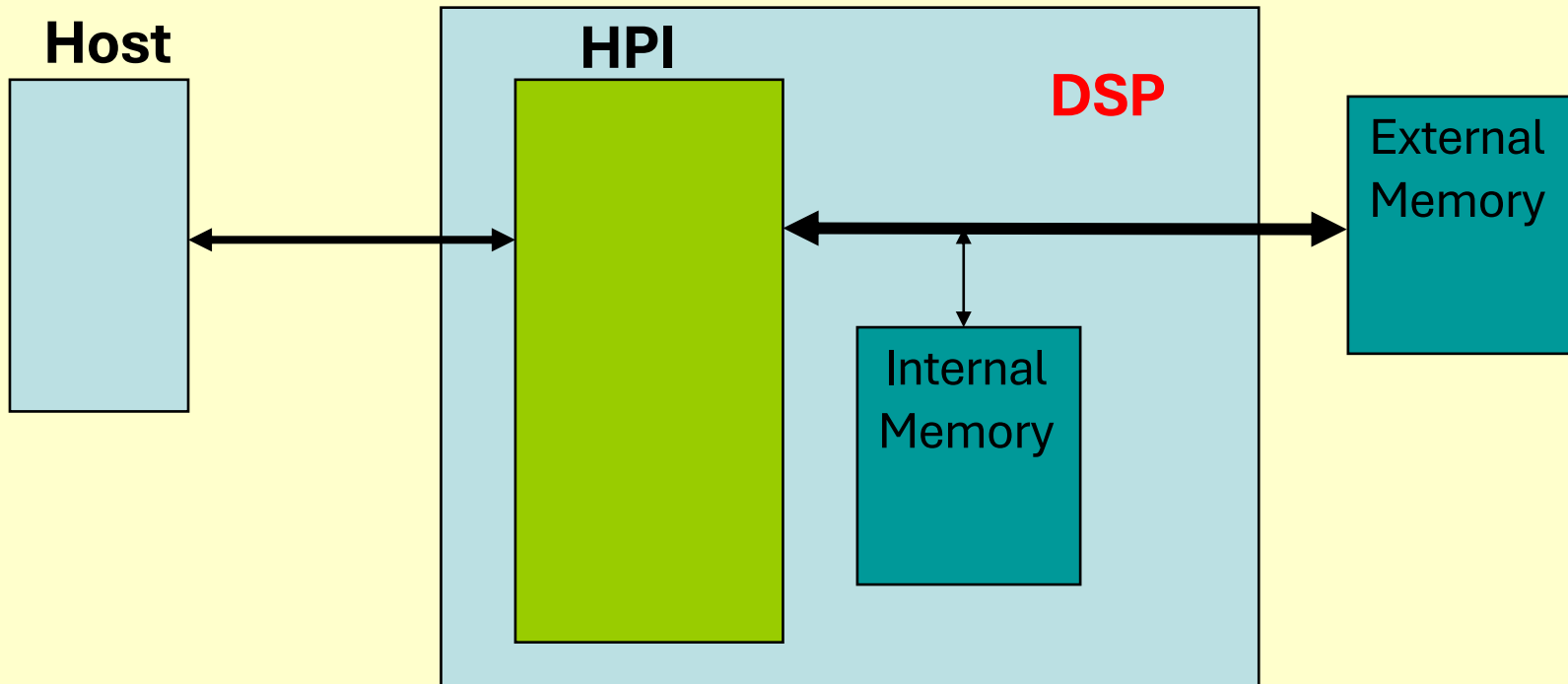
'C5416: 60% Core Performance Boost vs C'5409



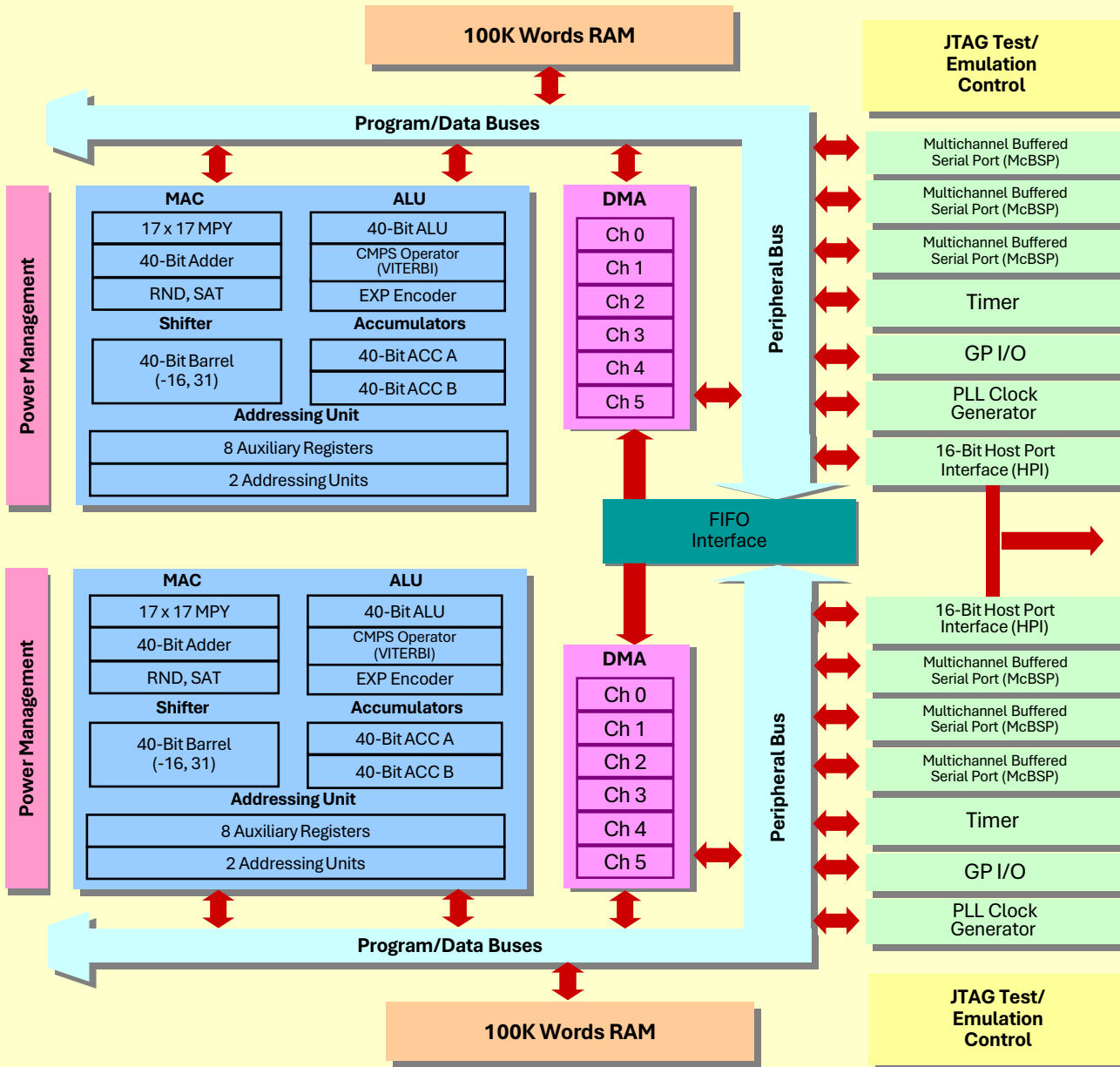
- High Performance
 - 160 MIPS
 - 128Kwords SRAM
 - 16K words ROM
 - 3 MchBSPs
 - 6-channel DMA
 - 8/16-bit HPI
 - Extended Program Addressing
- Low Power
 - 1.5-V core
 - 90 mW active @ 160 MIPS
- Small Size package
 - 144 TQFP
 - 144 microStar BGA

Host port interface (HPI)

- Provide a parallel port to connect the host processor of a PC, microcontroller or DSP directly to the DSP memory
- The host and DSP exchange data via the internal or external memory



'C5420-Multicore: Minimize mW & in² / channel

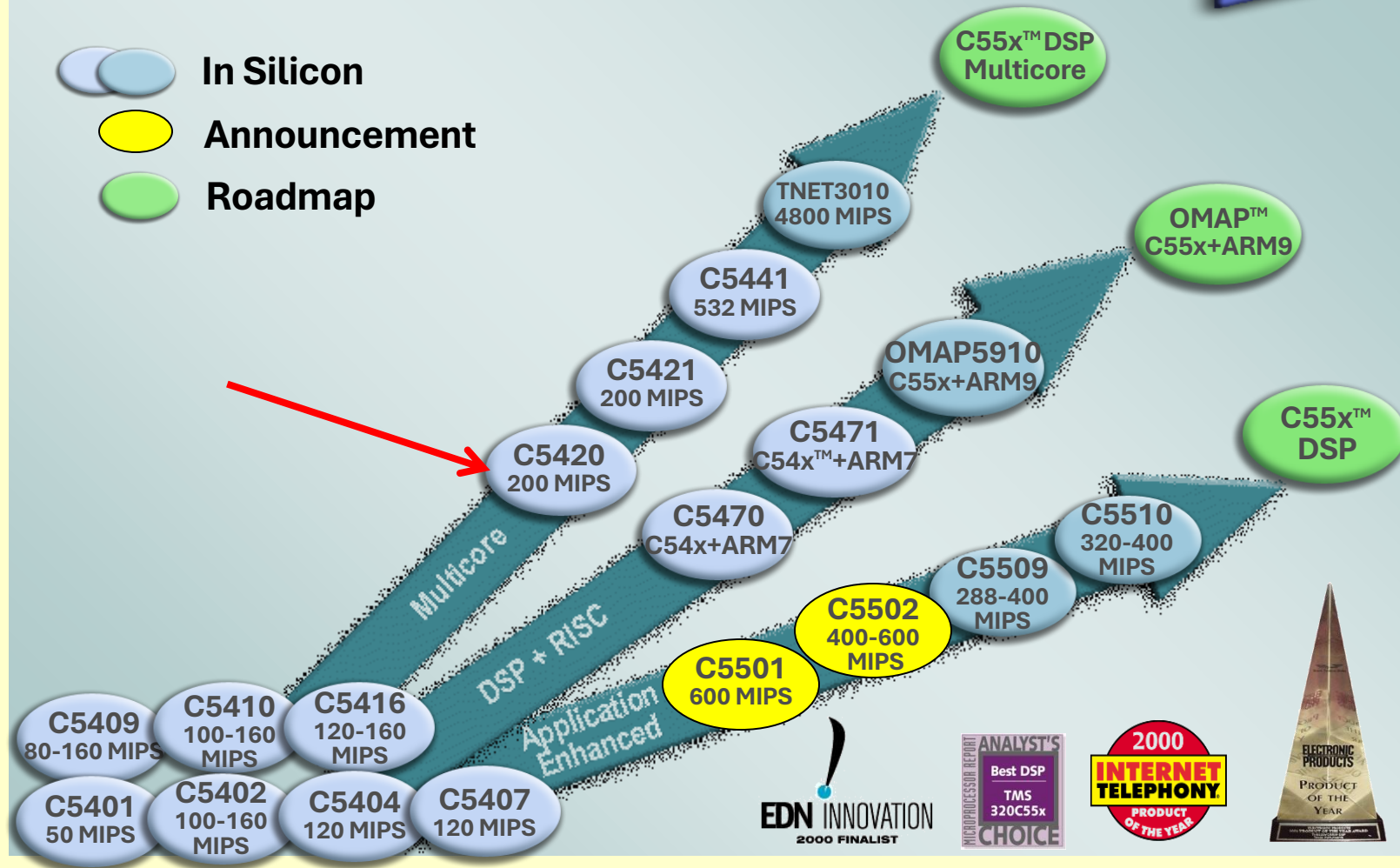


- Performance
 - 200 MIPS ✓
 - 200Kwords SRAM
 - 6 McBSPs
 - 12-channels DMA
 - 16-bit HPI
- Low Power
 - 1.8-V core ✓
 - 266 mW active power
 - Low power/channel
- Small Size Package
 - 144-pin TQFP
 - 144 microStar BGA

Power Efficiency/System Density

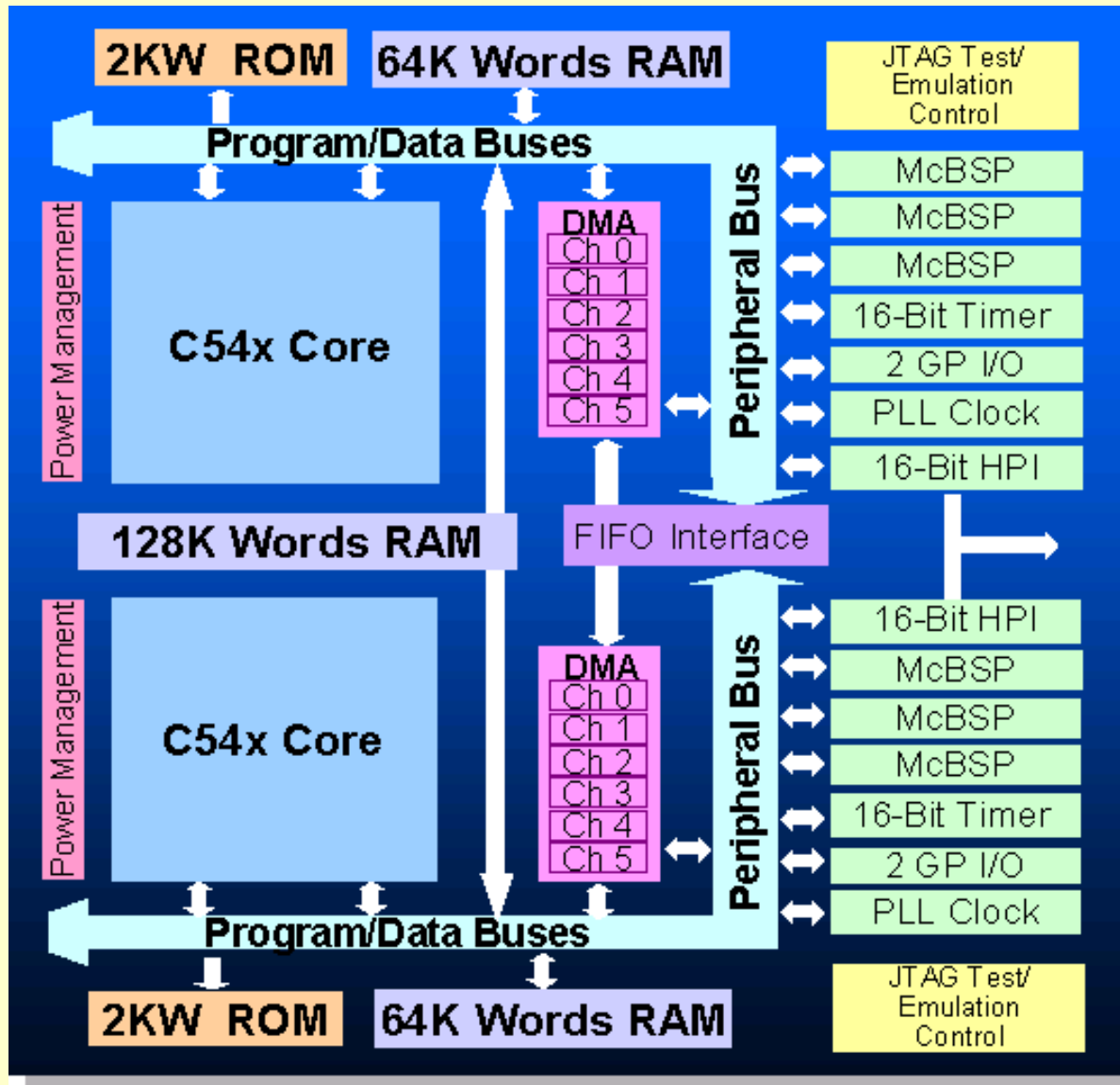
Code Compatible

-  In Silicon
-  Announcement
-  Roadmap



Feature Integration

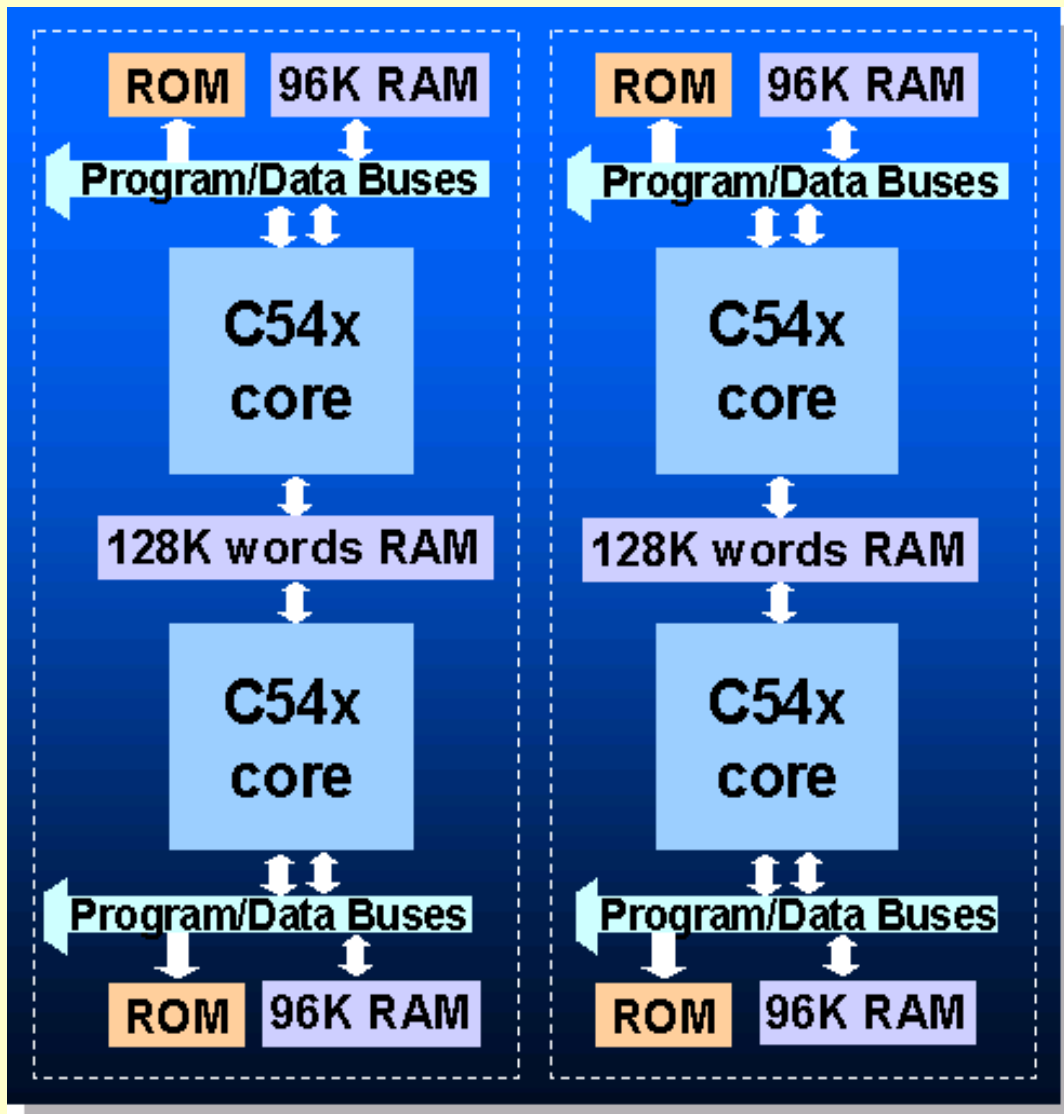
TMS320VC5421: Balances Channel Density for Strict Board-Level Power Budgets



- **Performance for increased density and flexibility**
 - 200 MIPS
 - 256Kw SRAM
 - 6 McBSPs with 128-channel selection
 - 12-channels DMA w/ external transfer (256Kw)
 - 16-bit HPI
- **Low power to meet board-level power budgets**
 - 1.8-V core
 - 160 mW active power at 200 MIPS
- **Small size for maximizing available space**
 - 144-pin TQFP
 - 144 microStar™ BGA

'C5441 – 4 core:

Performance for the Ultimate in Density and Flexibility



- **Performance:**
 - 133 MIPS /core (**532 MIPS**)
 - 640Kw SRAM
 - 4 McBSPs + 2 shared
 - 16-bit HPI
- **Ultra-low power/channel**
 - **1.5-V** core
 - **550 mW** active power ✓
at **532 MIPS** ✓
 - 0.15 um process technology
- **Small size**
 - 176 TQFP
 - 179 microStar™ BGA

C54V90 Modem Chipset

- **What is?** TMS320C54x™ DSP with a highly integrated two-chip V.90 embedded modem chipset:
- Supports V.90 and all prior ITU data and fax modes
- Ultra-low power consumption
- Parallel or serial interface
- Works with any OS – modem is host independent

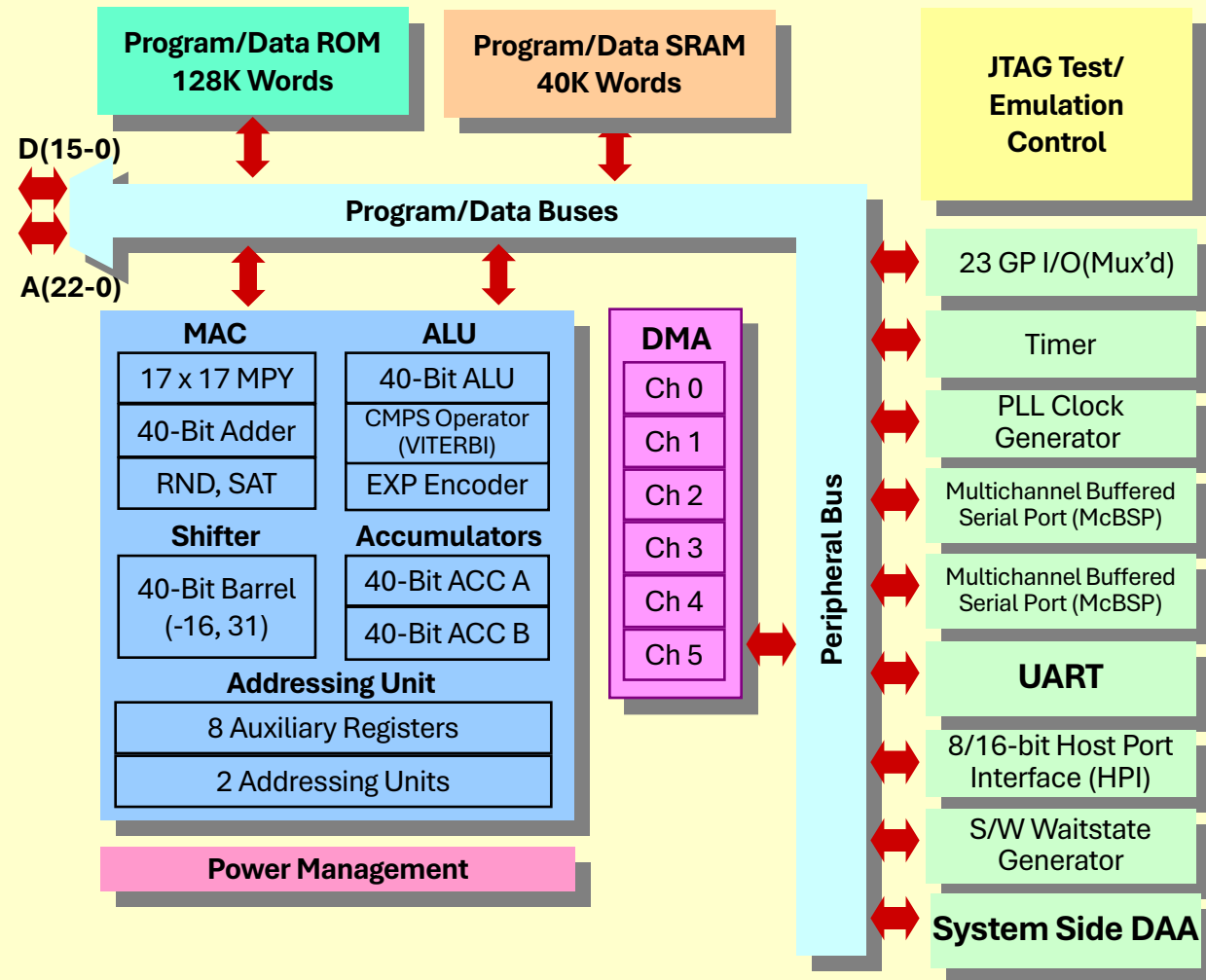


Possible Applications:

- Internet Appliances
- Set-Top Boxes (STBs)
- Gaming Consoles
- Digital Cameras
- Digital Music Players
- Electronic Books (eBooks)
- Personal Digital Assistants (PDAs)
- Home Networking etc.



C54V90 DSP

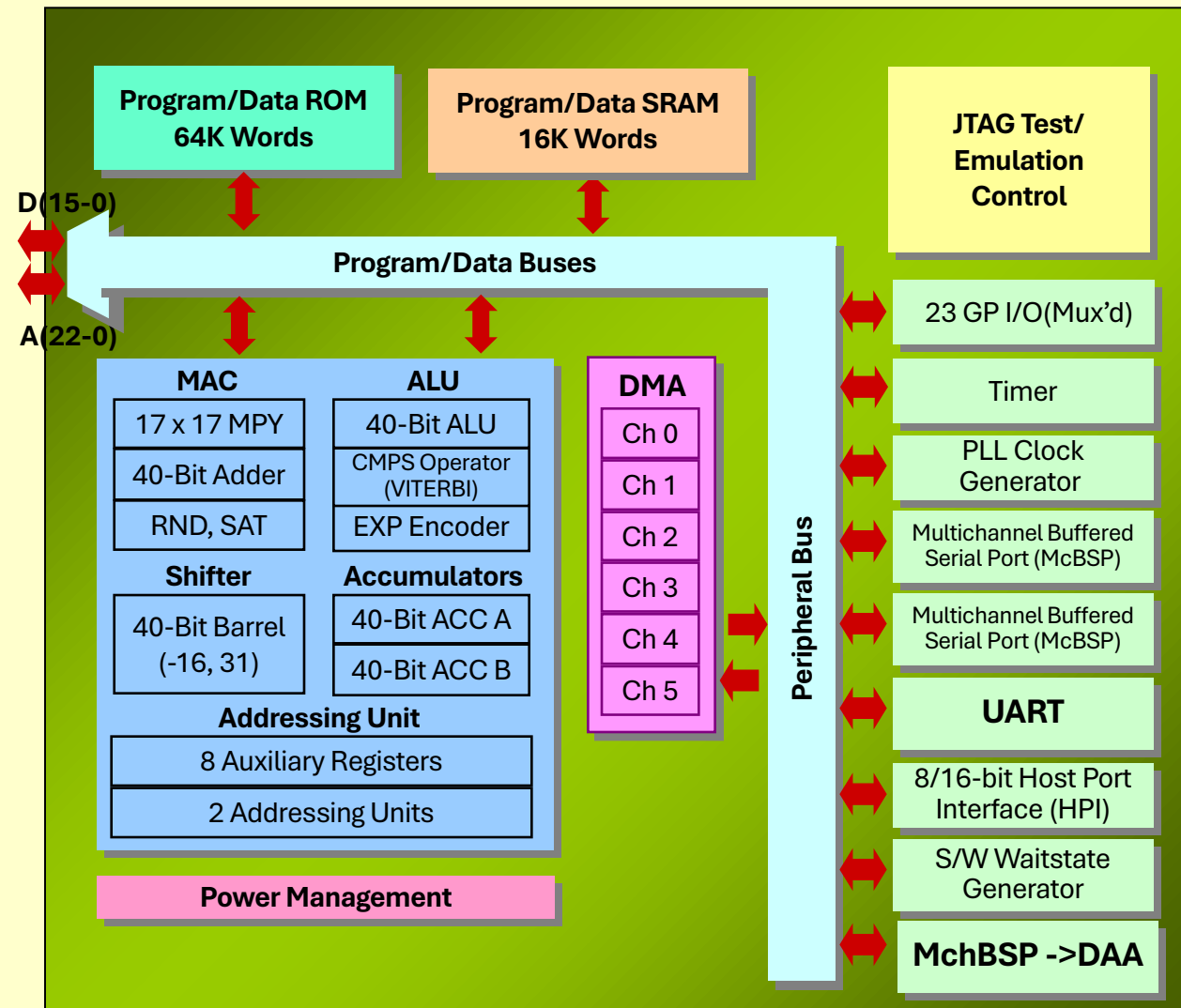


- Features
- **120 MIPS** Performance
- 40K words SRAM
- 128K words ROM
- 2 MchBSPs (full 128-ch)
- 6-channel DMA
- 8/16-bit HPI
- UART
- System Side DAA*
- 23 GPIO Pins (Mux'd)
- Power
- **25mW** for modem solution
- 1.5-V core (3.3V I/O; 5V for DAA)
- Three power-management modes

* DAA-(direct access arrangement) implements a standard telephone interface

C54CST DSP Product Summary

(Client-Side Telephony)



- **Features**
 - Up to **120 MIPS**
 - 16K words SRAM
 - 64K words ROM
 - 2 McBSPs
 - 6-channel DMA
 - 8/16-bit HPI
 - UART
 - System Side DAA (operates at 59 or 118Mhz)
 - 23 GPIO Pins (Mux'd)
- **Power**
 - **25mW** typical power
 - **1.5-V** core (3.3V I/O; 5V for DAA)
 - Three power-management modes

Note: C54CST memory size is 16Kw RAM and 64Kw ROM.

Software algorithms and framework developed by Spirit Corporation

Beagle-Board

Laptop-like performance

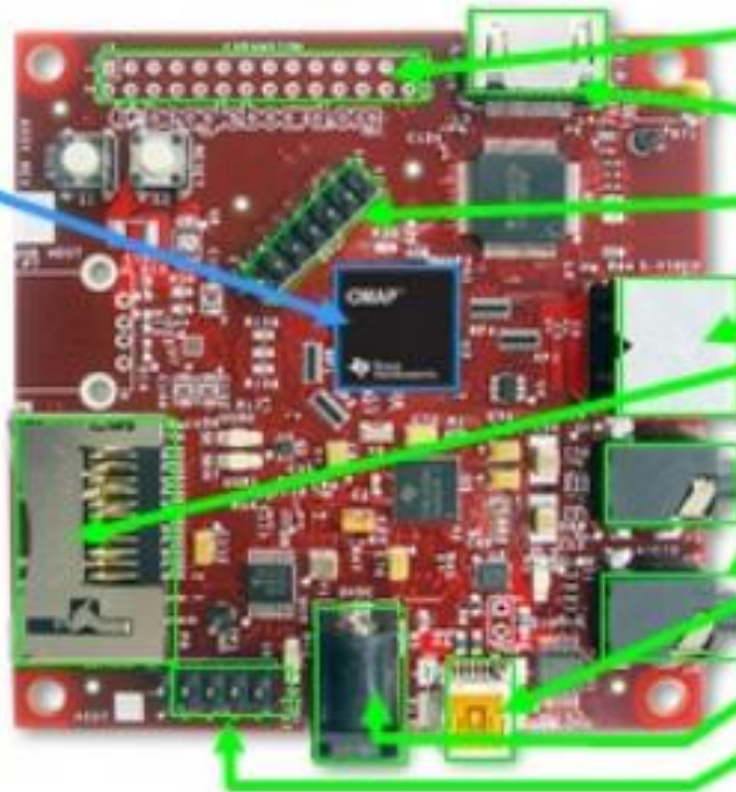
TI OMAP3530

- 600 MHz superscaler ARM[®] Cortex™-A8
- More than 1200 Dhrystone MIPS
- Up to 10 Million polygons per sec graphics
- HD video capable C64x+™ DSP core

Memory

- 128MB LPDDR RAM
- 256MB NAND flash

← 3" →



Flexible expansion

I²C, I²S, SPI,
MMC/SD

DVI-D

JTAG

S-Video

SD/MMC+

Stereo Out

Stereo In

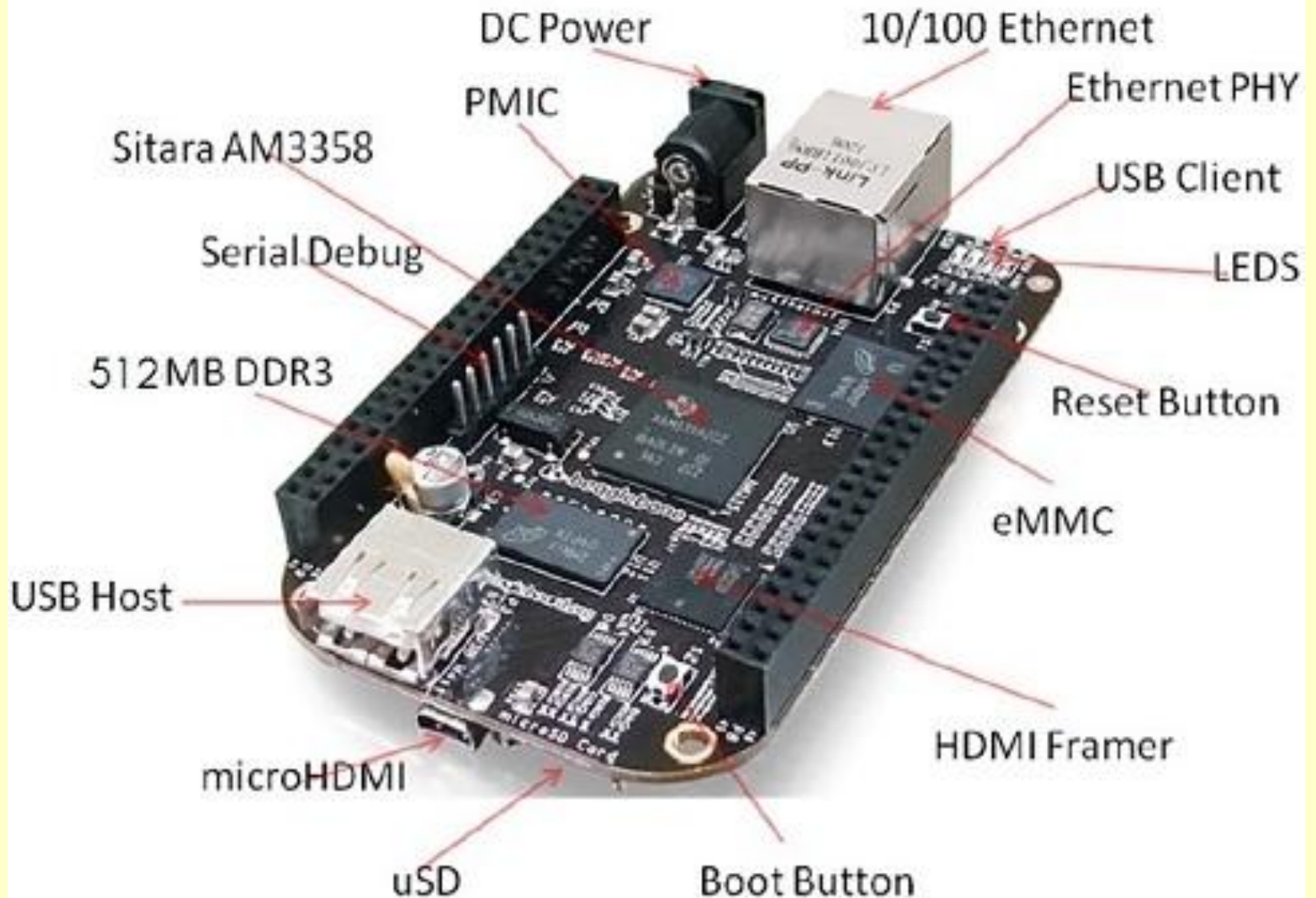
USB 2.0 HS OTG

Alternate Power

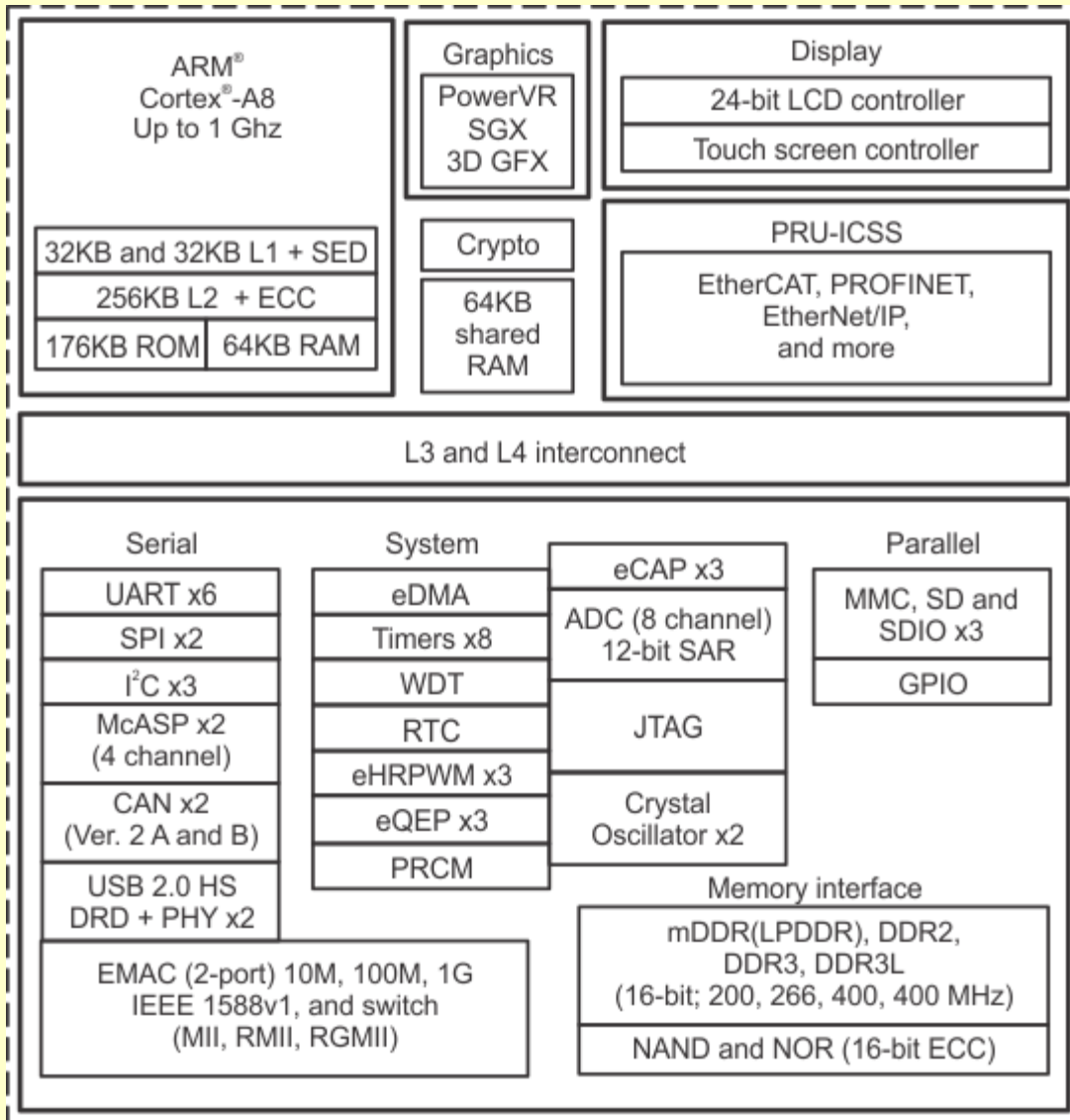
RS-232 Serial

- Built on the proven BeagleBoard.org[®] open-source Linux approach/Android, fills the gap between small SBCs and more powerful industrial computers.

Beagle-Bone



SITARA AM3358

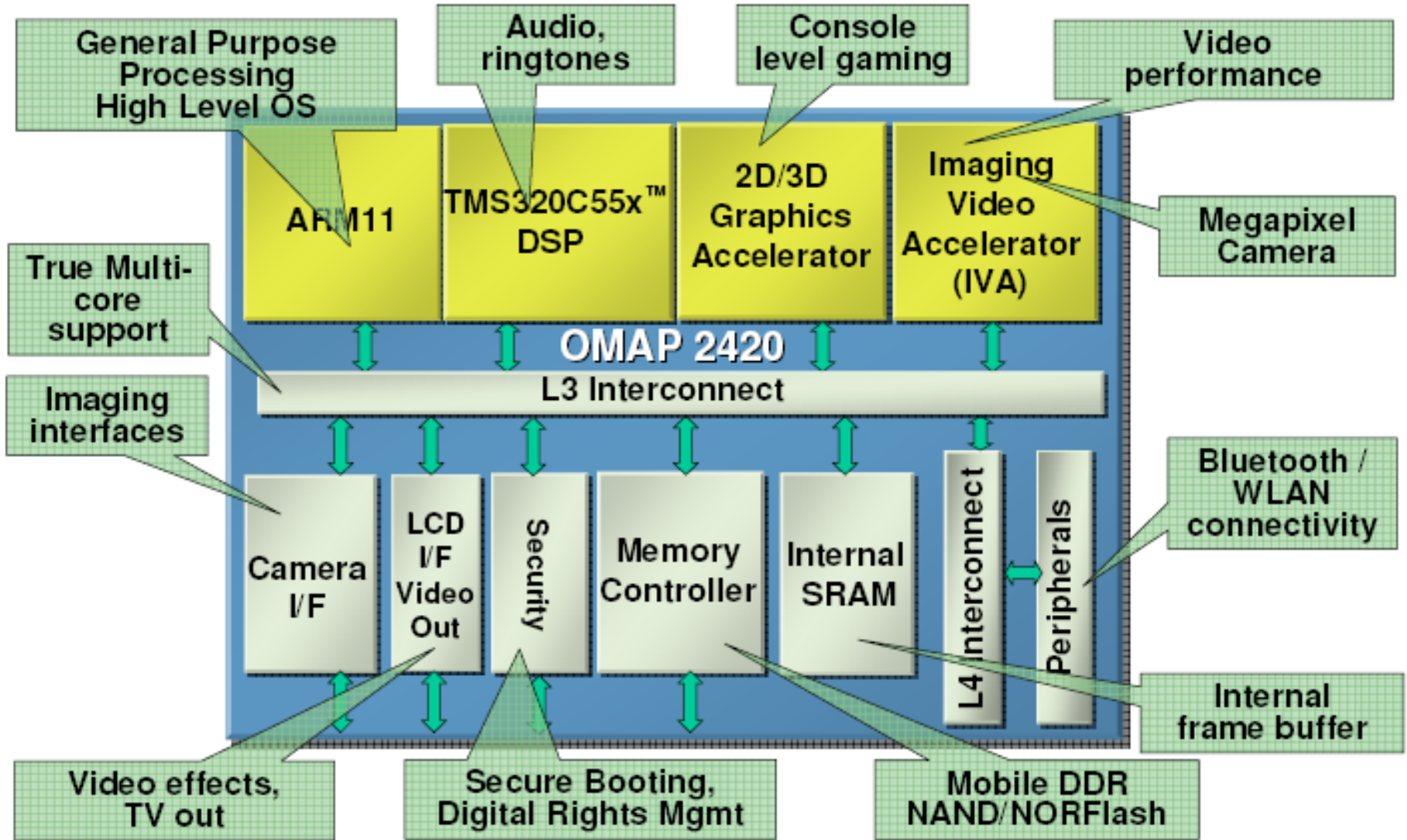


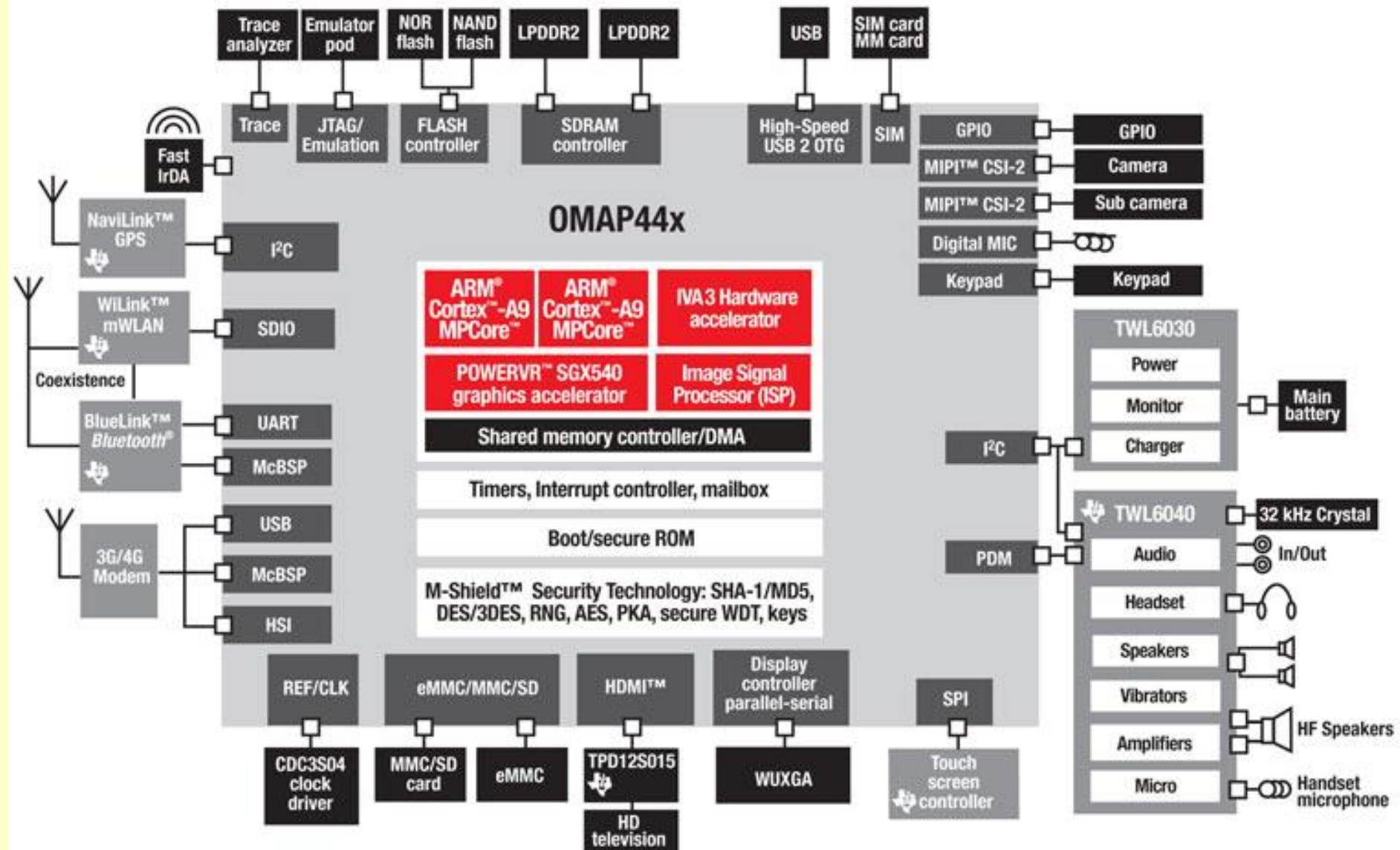
- ARM CPU -ARM Cortex
- A8 ARM MHz (Max.) 600, 800, 1000
- Co-processor(s)PRU-ICSS* CPU32-bit
- Graphics acceleration
- 3D Display type1 LCD
- Ethernet protocols, Ethernet/IP, Profinet, Sercos, ProfibusEthernet
- MAC2-Port 10/100 PRU EMAC, 2-Port 1Gb
- Switch features Networking
- Operating system Linux, RTOS

*Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS)



OMAP™ 2 Enables Cutting-Edge Applications





- The **OMAP 4460** is a fourth generation SoC (System on Chip) developed by **Texas Instruments** (TI) for use in *mobile-phones and tablets*.
- The chip includes **dual ARM Cortex-A9 CPU cores** running at **1.2 GHz up to 1.5 GHz** on an ARM v7 instruction set and ARM NEON SIMD (Single instruction, multiple data) engine.

Status LEDs

SD/MMC Card Slot

Serial /RS-232

Camera Connector

USB 2.0 OTG

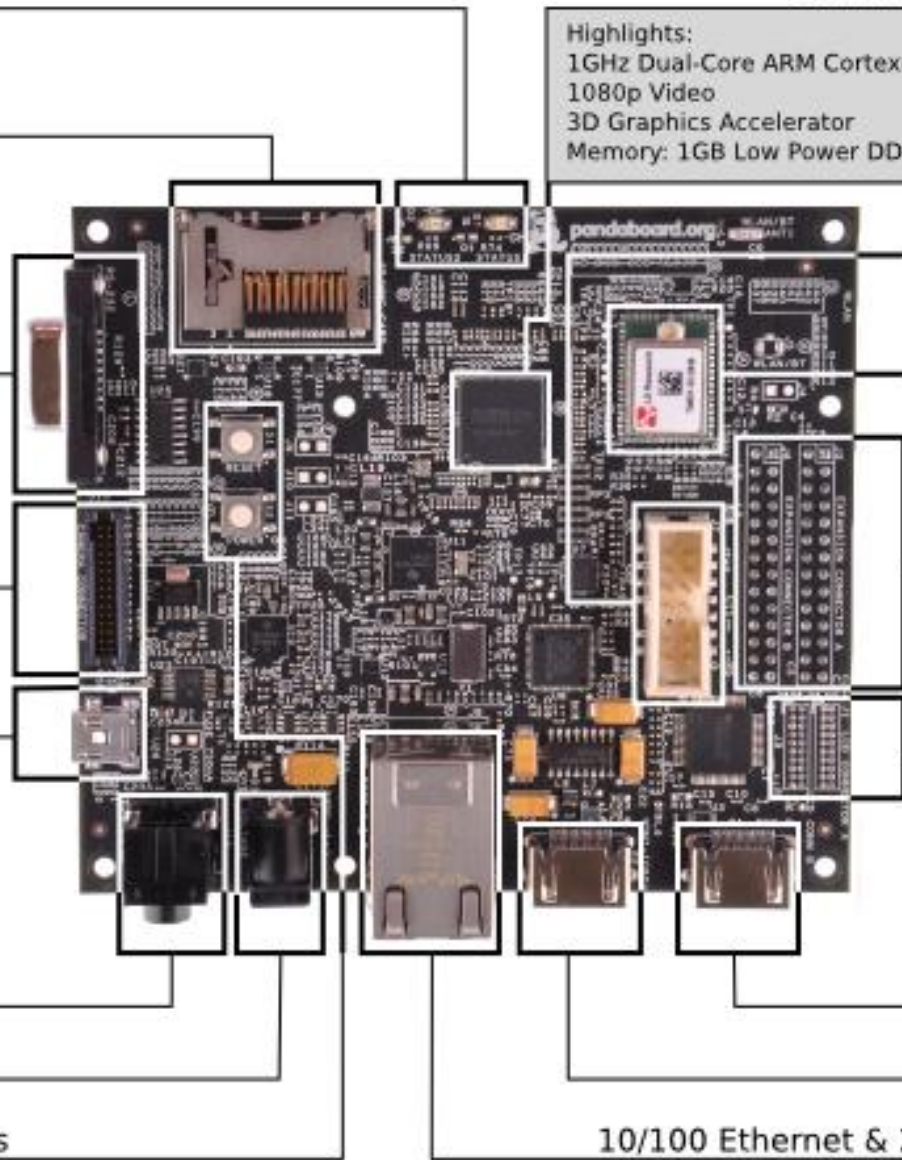
Audio in/out

Power Supply 5V

Power/Reset Buttons

OMAP4430 Processor

Highlights:
 1GHz Dual-Core ARM Cortex-A9 MPCore
 1080p Video
 3D Graphics Accelerator
 Memory: 1GB Low Power DDR2 RAM



JTAG

WLAN/Bluetooth

Expansion Connector

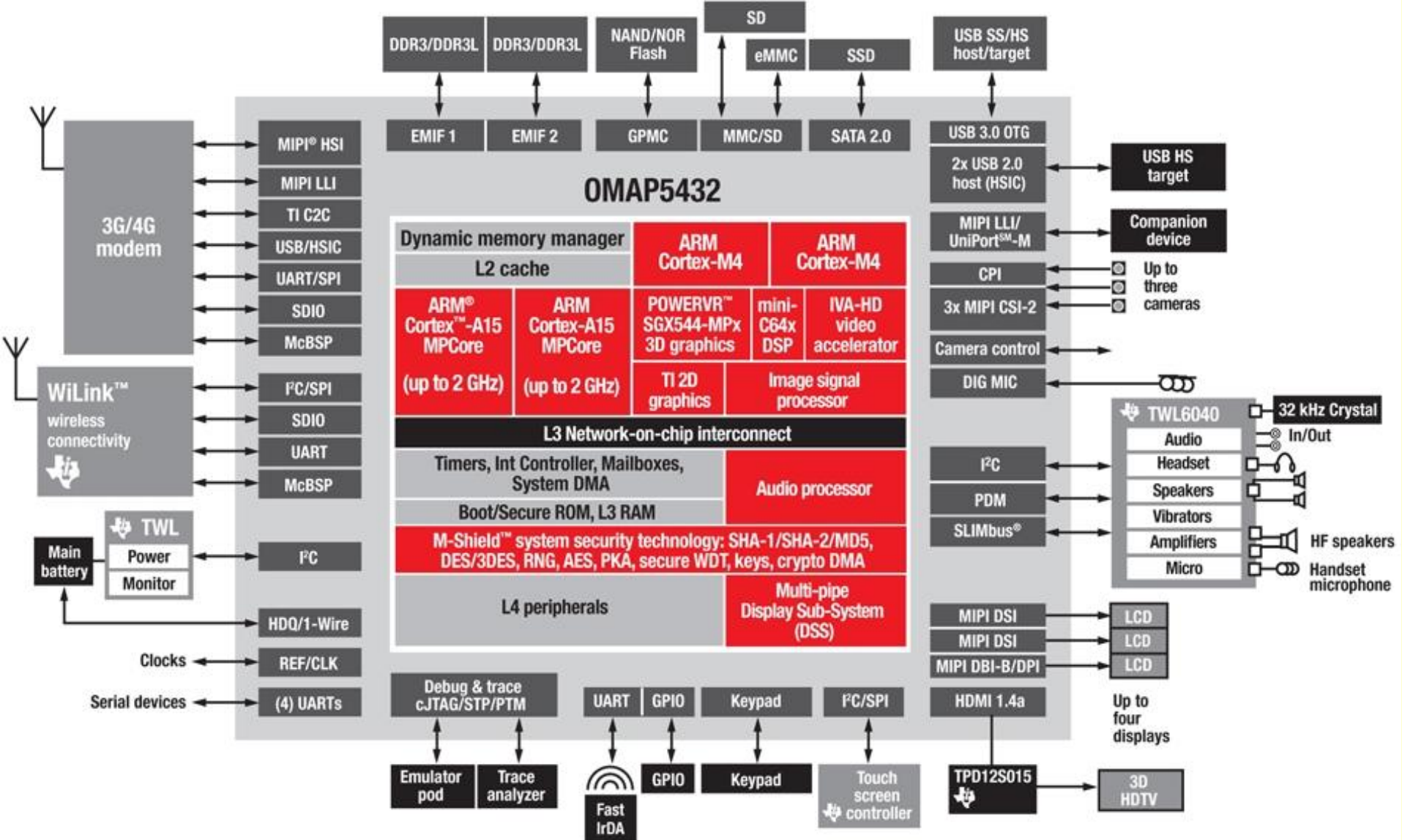
LCD Expansion

DVI Out

HDMI Out (Type A)

10/100 Ethernet & 2xUSB 2.0 Host ports

Board Dimensions: W:4.0" (101.6 mm) X H: 4.5" (114.3 mm)



With the OMAP 5 architecture at its core – including dual ARM® Cortex™-A15 cores, dual Cortex-M4 cores and -Imagination Technologies' POWERVR™ SGX544-MPx graphics cores – the DRA74x processor increases ARM CPU performance up to five times and graphics (GPU) performance close to seven times as compared to the previous "Jacinto 5" offering.

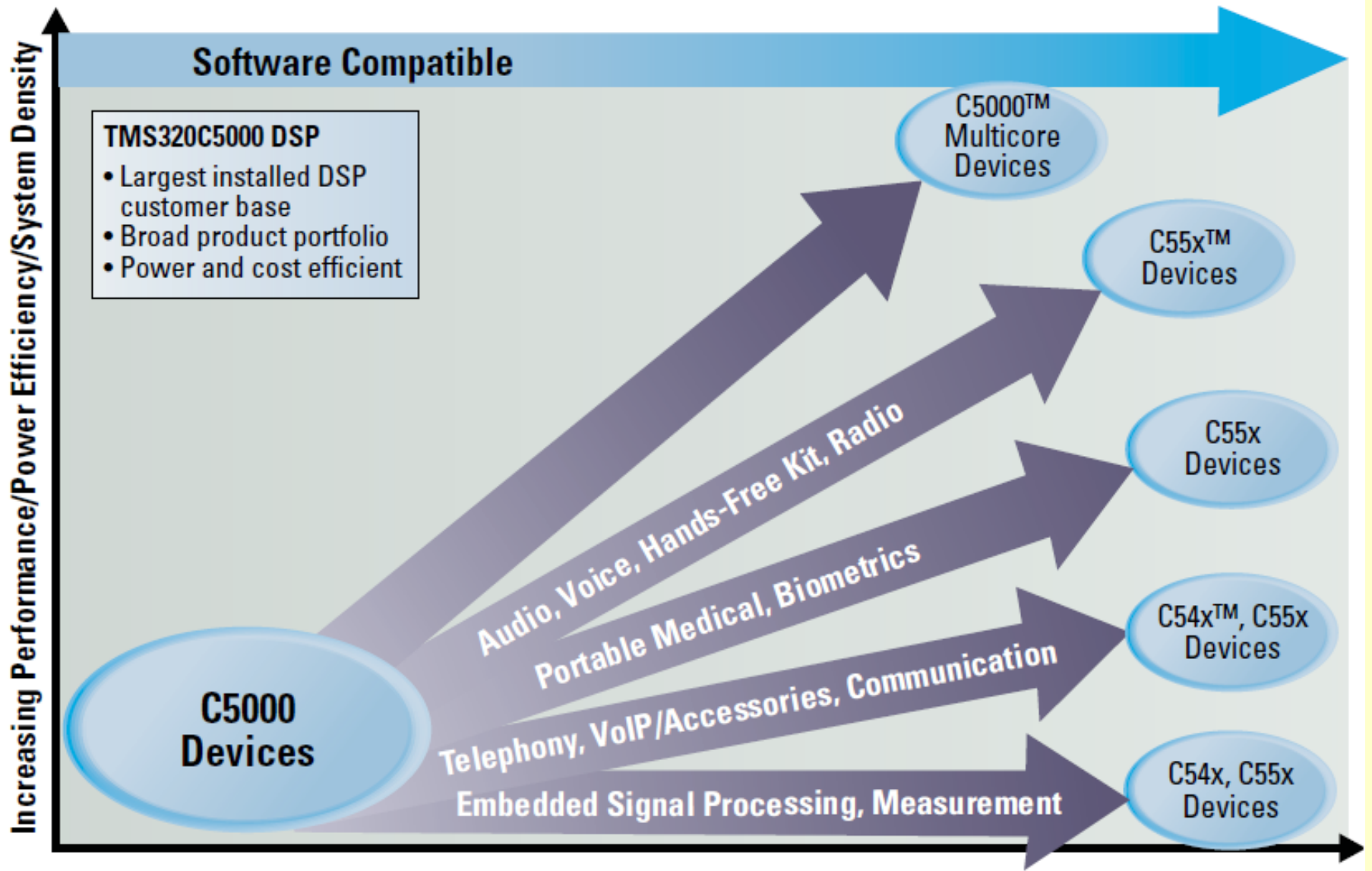
TMS320C54XX architecture

Overview:

- Common Architectural Features of DSPs
- What Constitutes a Good DSP?
- TMS320C54XX DSP Generation Block Diagram
- C54xx Architecture, C55xx
- Refereces: TI educational material ; sprs039.pdf

Courtesy to: Texas Instruments
C5000 Educational Materials

Study: <https://www.edn.com/benchmarking-digital-signal-processors-do-fpgas-cost-more/>



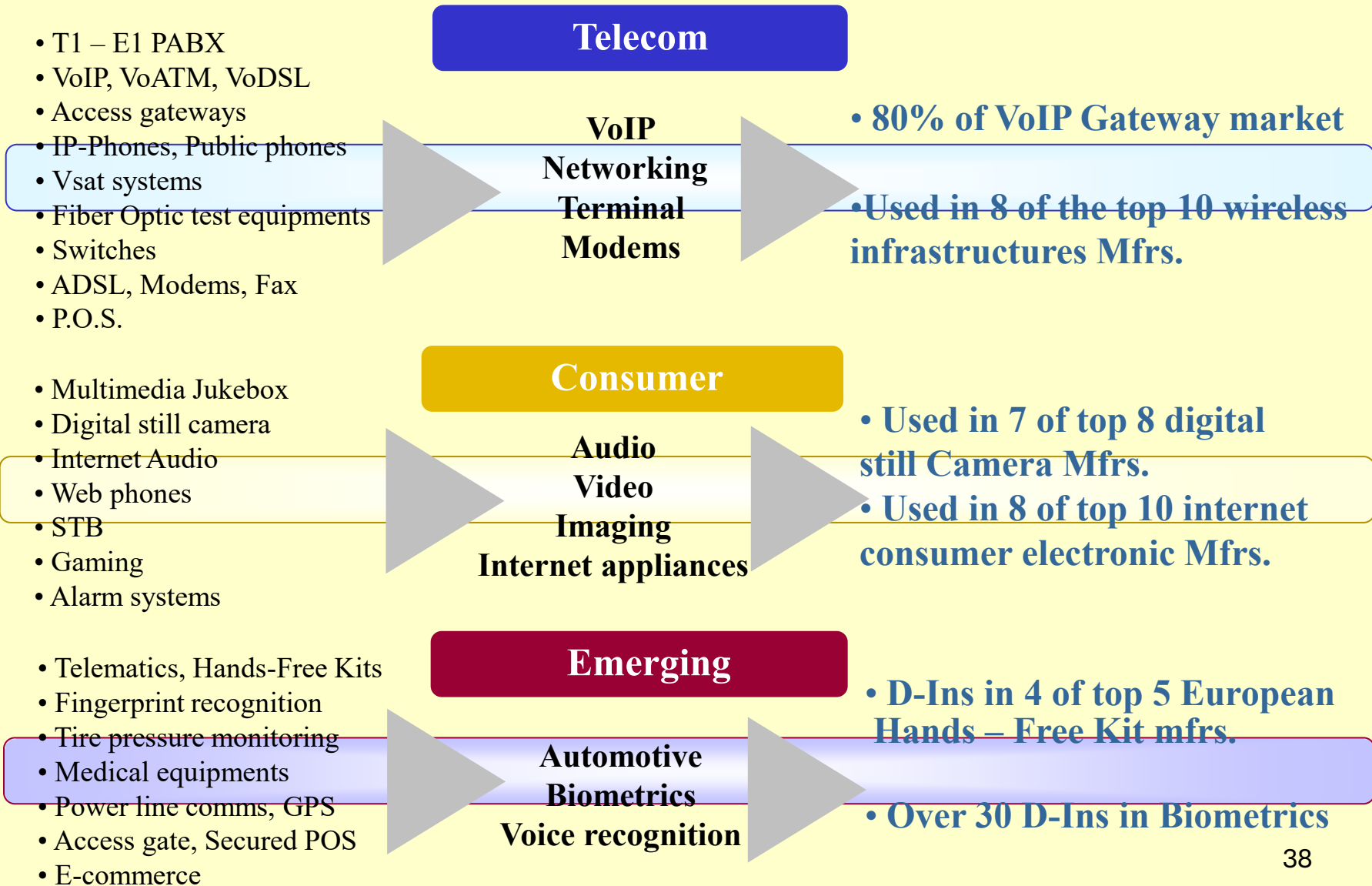
C5000™ DSP Platform Roadmap

The C5000 DSPs span the applications spectrum with core performance extended to 300 MHz.

'C5000 DSP Market Segments Targeted

> 500 mil. C5000 DSPs shipped in more than 1000 different applications

End Equipments



Common Architectural Features of DSPs

Data processing:



- ✓ **MAC- multiply/ accumulate Unit Executes one or more MAC in one clock cycle**
- ✓ **Internal registers for storing operands & results**
- ✓ **Multiple internal data memories blocks**
- ✓ **Auxiliary addressing registers (8 or more) with a dedicated ALU (ARAU)**
- ✓ **Multiple buses and dual access to data memory (DARAM)**
- ✓ **Separate access to program and data memories (Harvard architecture)**
- ✓ **DMA (Direct Memory Access)**

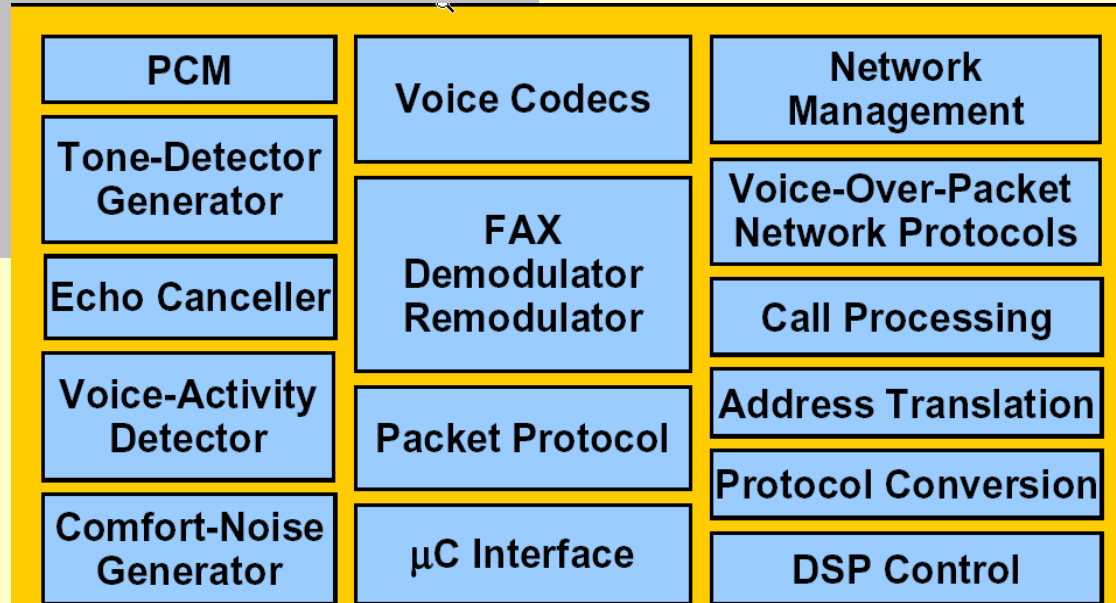
Common Architectural Features of DSPs

Real-time processing and Multi-tasking:

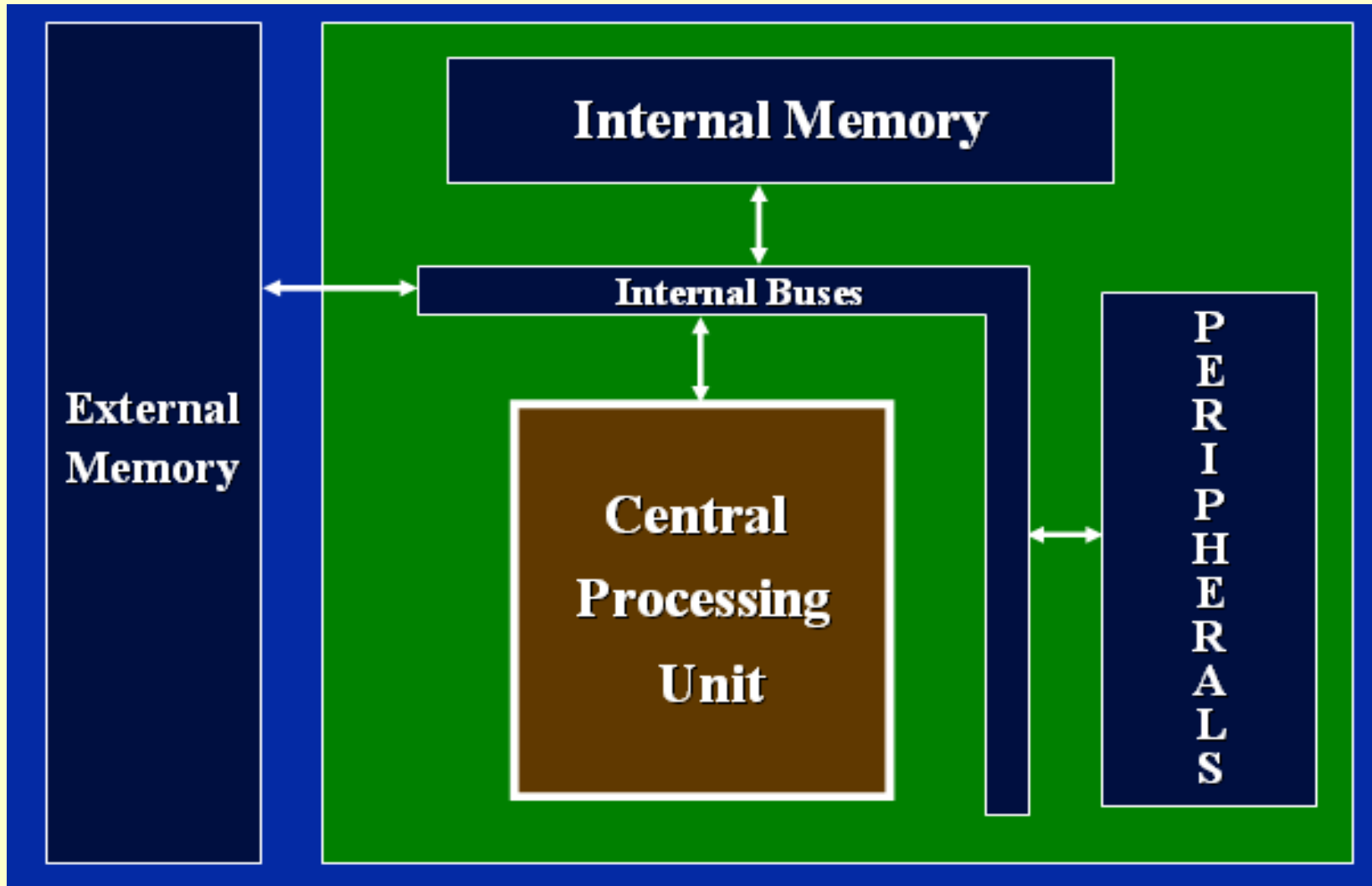
- ✓ Hardware & software interrupt handling
- ✓ Zero overhead loops
- ✓ Pipelined instructions
- ✓ Parallel units
- ✓ Specialised peripherals

Support for Complex Algorithms:

- ✓ Specialised instructions (Viterbi, LMS...)
- ✓ Bit reverse addressing (FFT)
- ✓ Circular buffers (FIR filters)



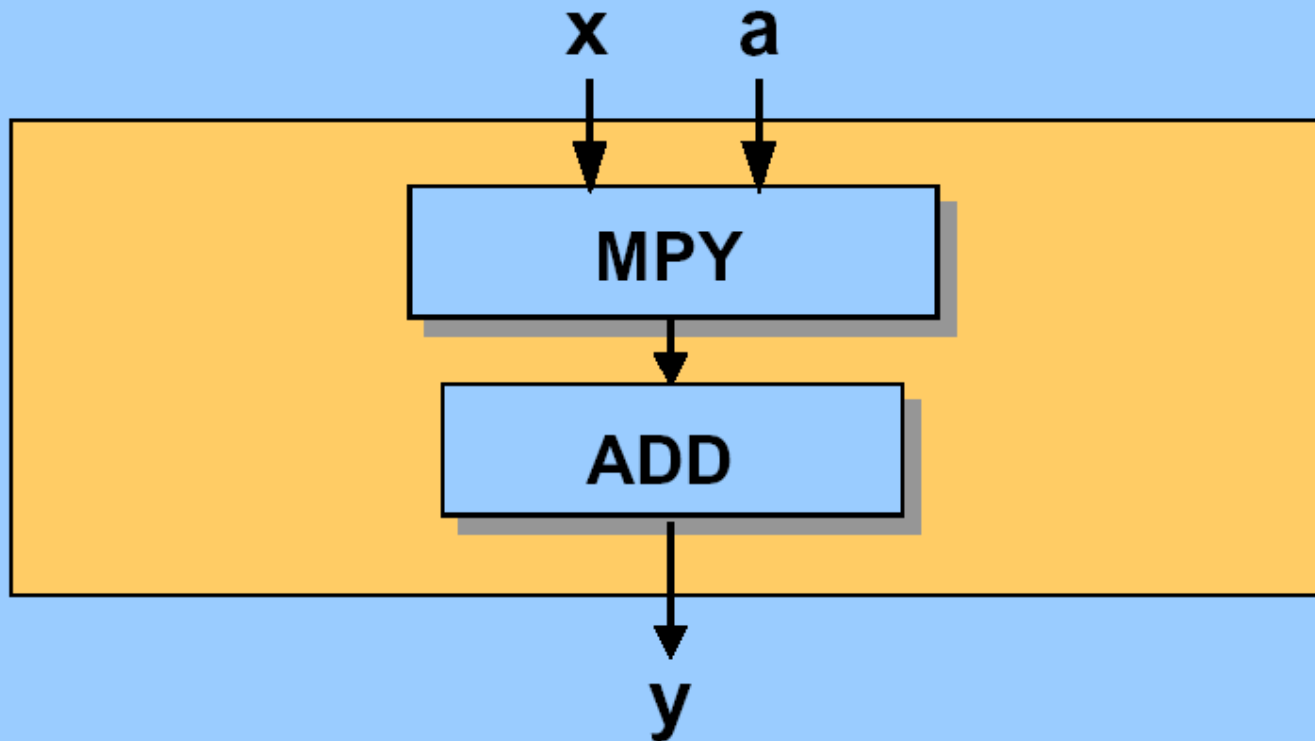
General DSP System Block Diagram



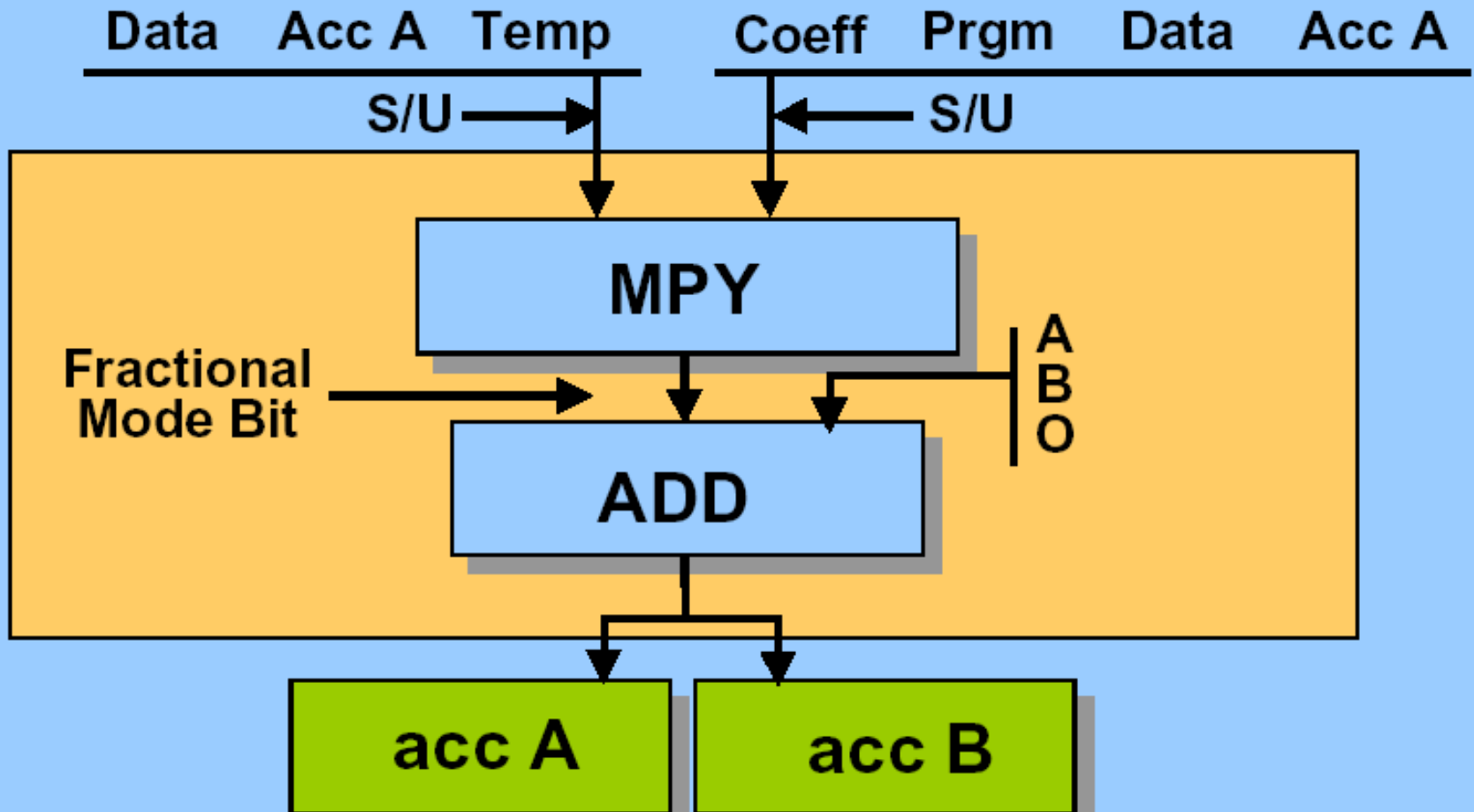
What Constitutes a Good DSP?

DSPs require 1 or more MACs in a single cycle

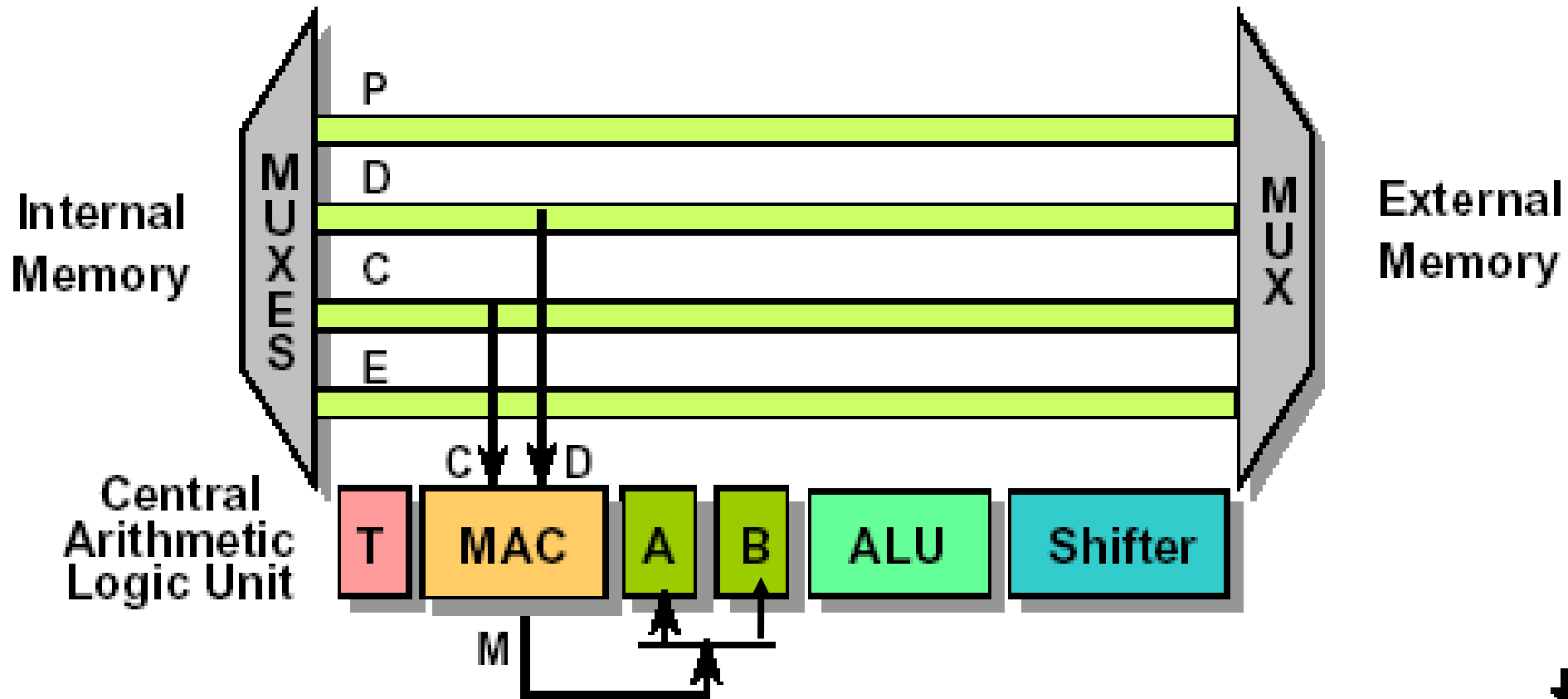
$$Y = \sum_{n=1}^{40} a_n * x_n$$



Multiply and Accumulate Unit



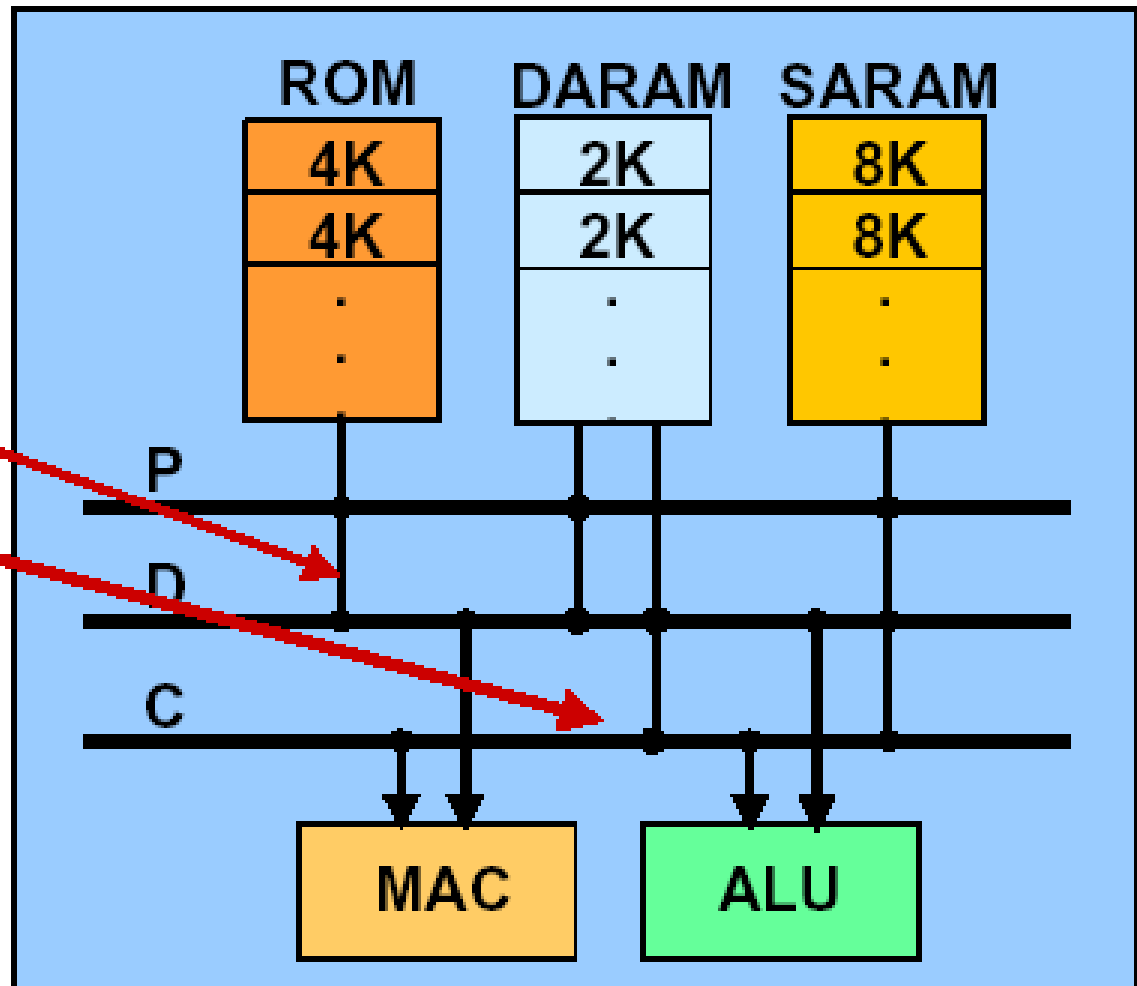
Multiple Buses



Internal Memory for Fast Access

ROM & RAM:
One access
per block
per cycle

DARAM:
Two accesses
per block
per cycle



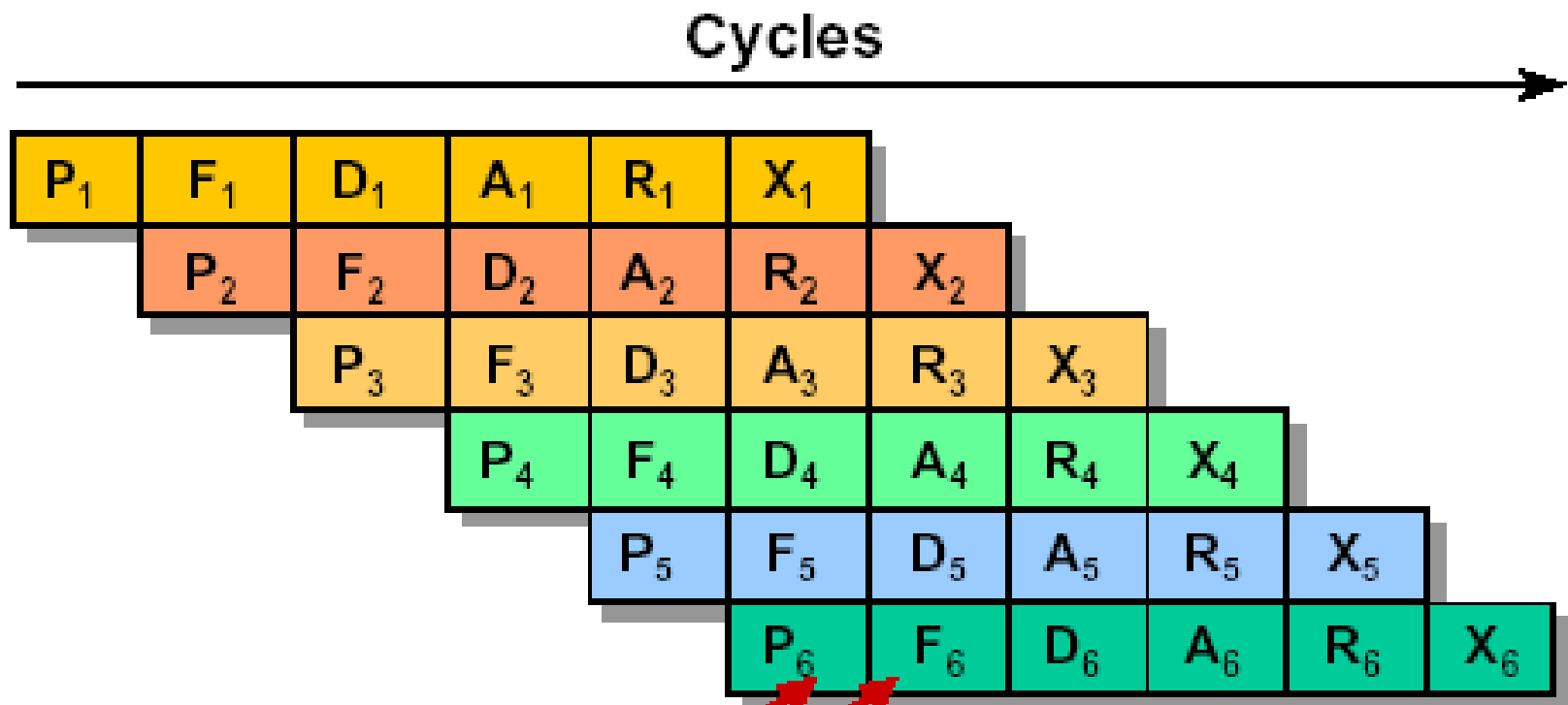
Instruction Pipeline for Fast Execution

- Instruction is broken into smaller tasks that can be executed in parallel



- ◆ Prefetch: Calculate address of instruction
- ◆ Fetch: Collect instruction
- ◆ Decode: Interpret instruction
- ◆ Access: Collect address of operand
- ◆ Read: Collect operand
- ◆ Execute: Perform operation

Sequential-parallel processing of instructions

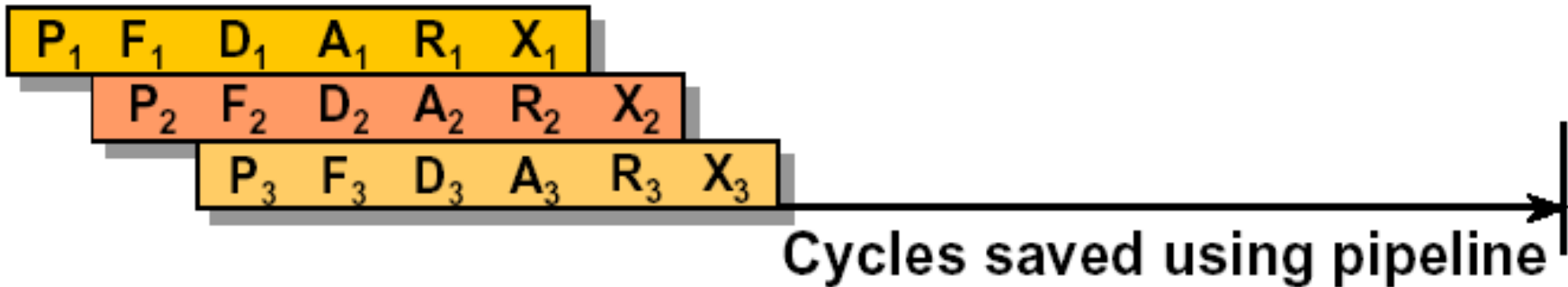
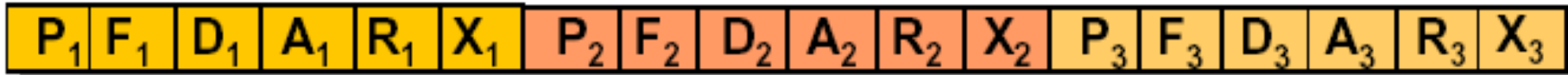


Fully loaded pipeline

Less Cycles per Instruction
Less Power Consumption

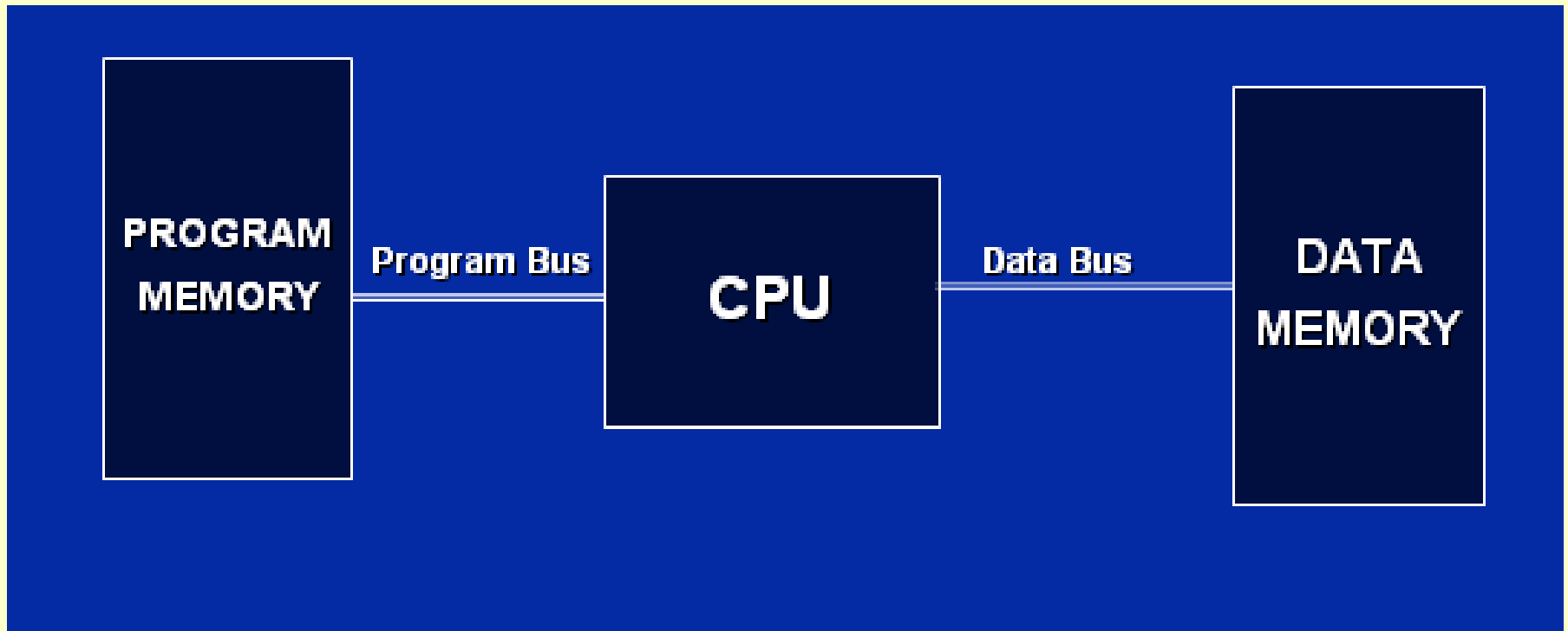


Cycles required without pipeline

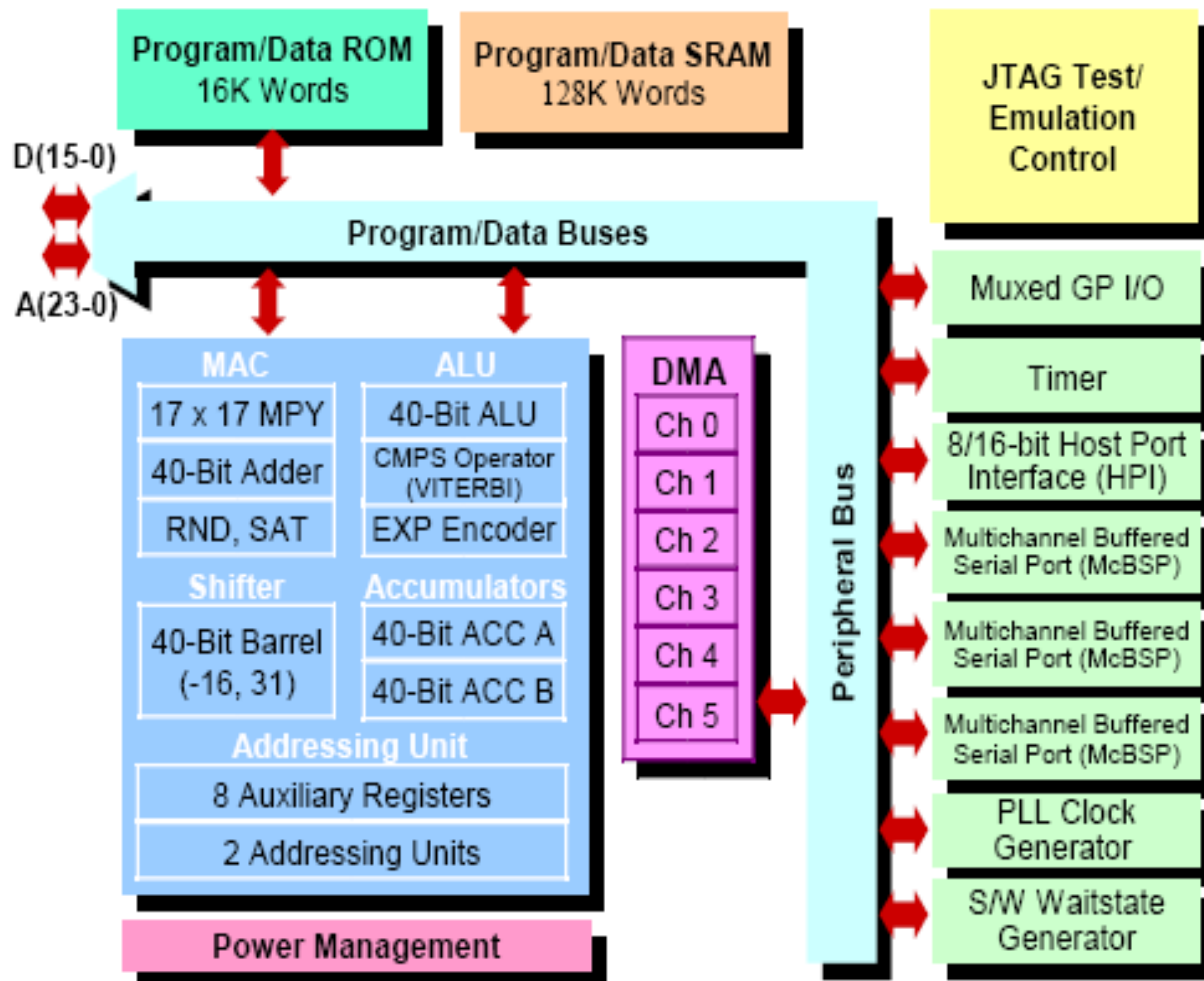


Texas Instruments C5000 Solutions

Basic Harvard Architecture 1st DSP Generation



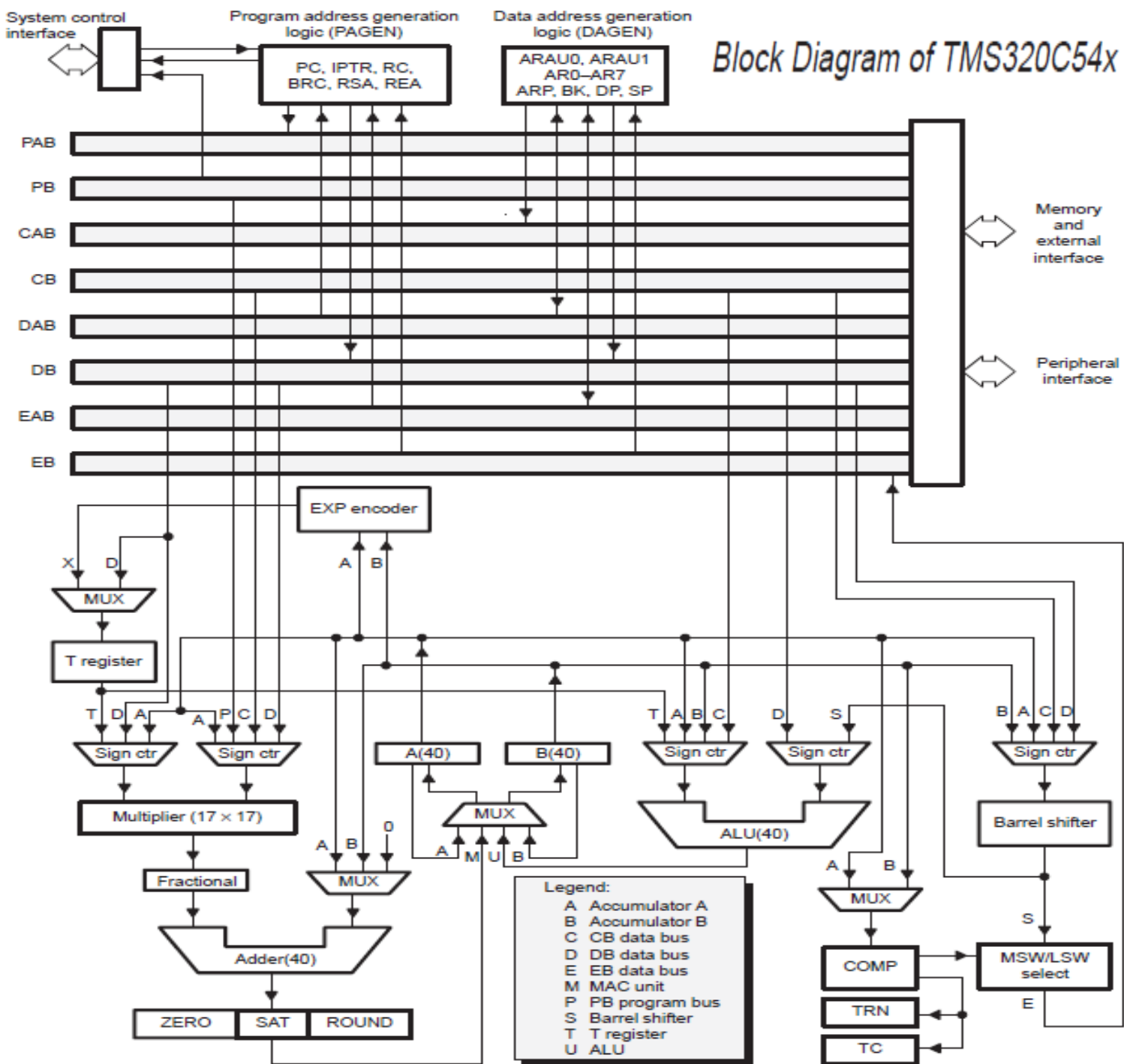
C54x Block Diagram



C5416 example

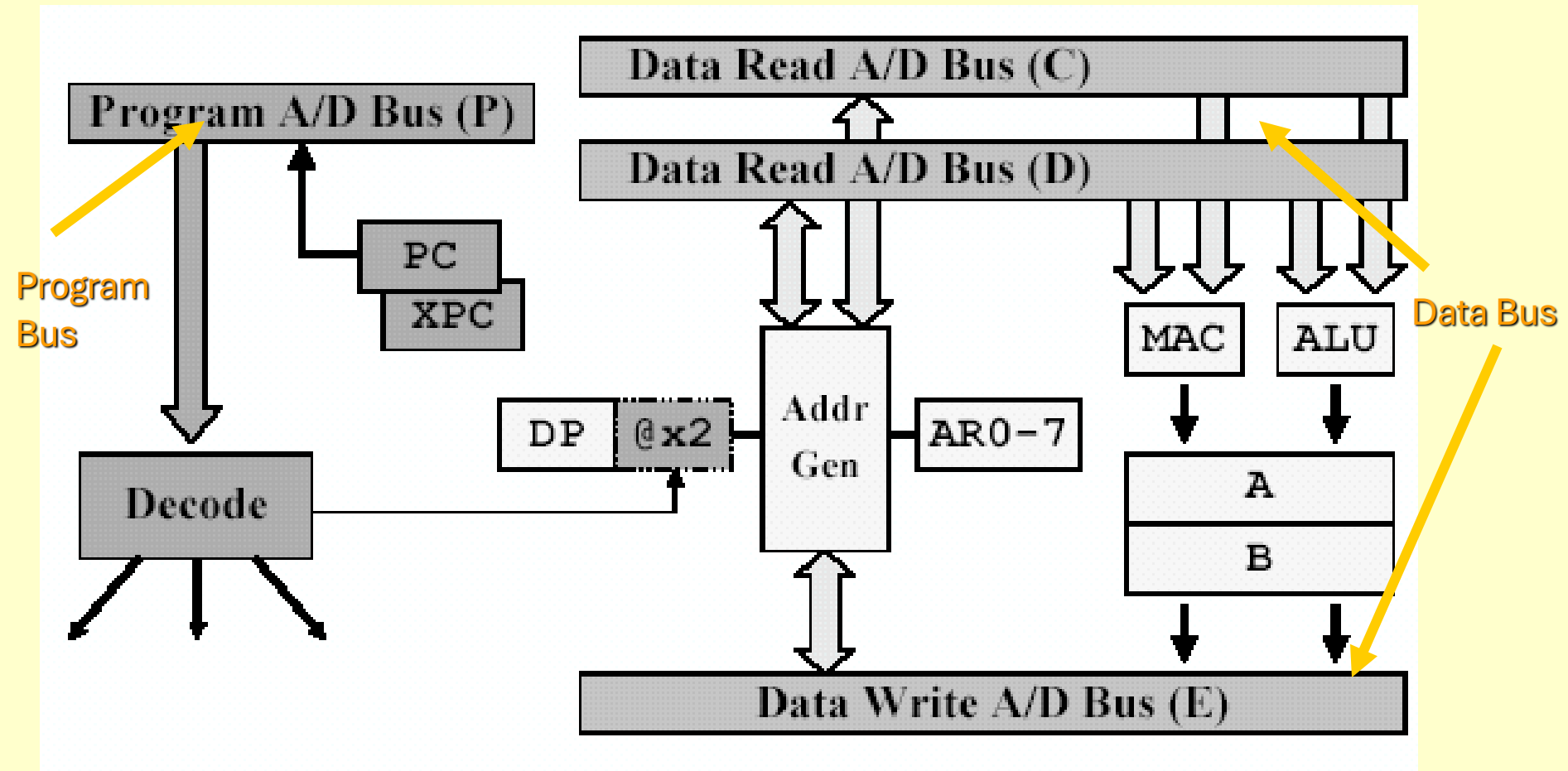
- 17x17 MAC Unit
- Saturation and Rounding Hardware
- Two 40-bit ACC's
- 40-bit ALU
- 40-bit Barrel Shifter
- Temporary Register
- Exponent Encoder
- Program and Data Address Generation Units
- Compare, Select and Store Unit
- 4 Internal Bus Pairs
- External Interface

Block Diagram of TMS320C54x Internal Hardware

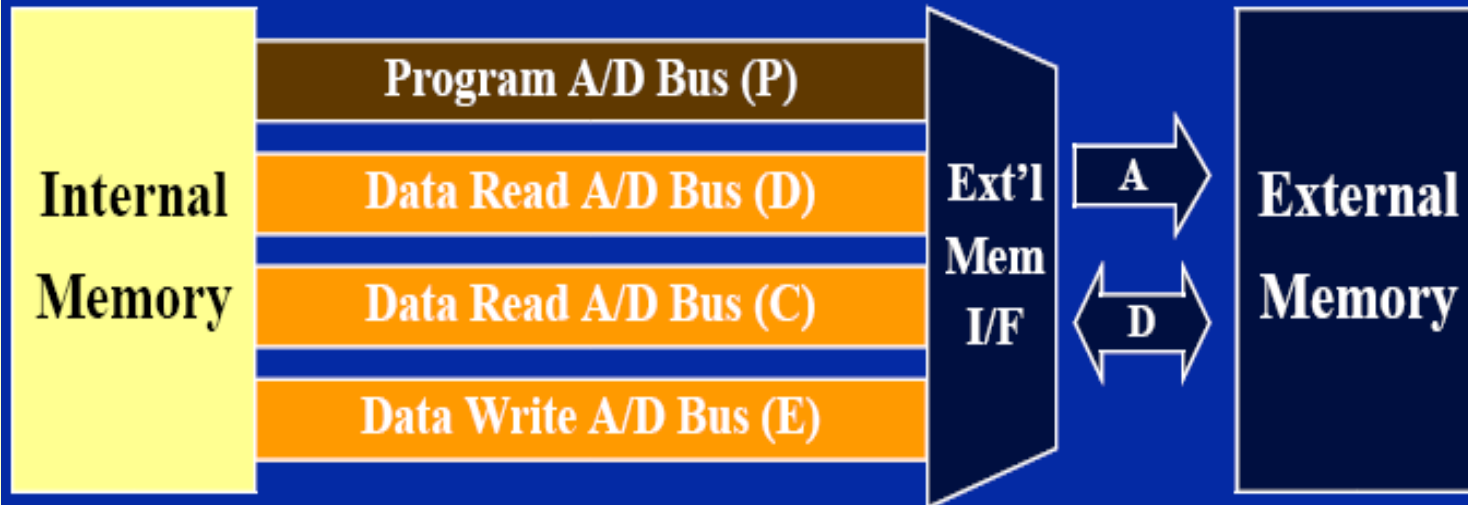


TMS320C54X DSP Adr. Generation Block Diagram

Conventional Harvard DSP architecture



C54x Pipeline



Internal: Up to 4 accesses / cycle

- External: 1 access / cycle
- up to 8M words program

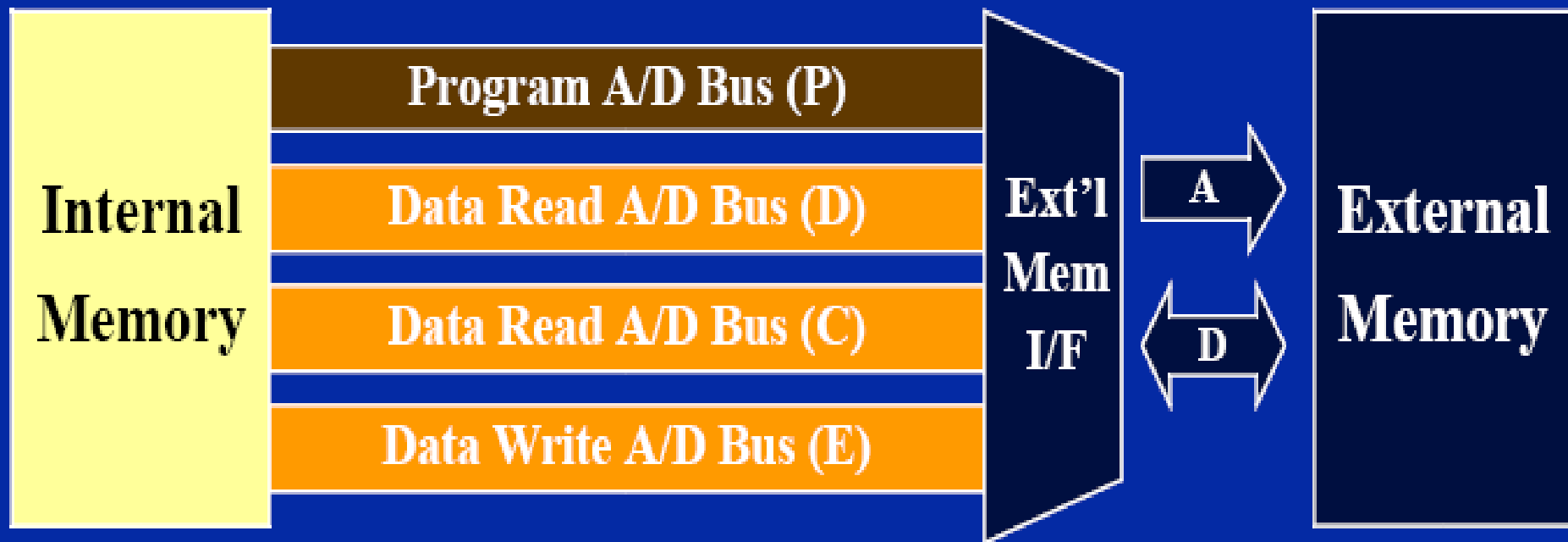
Pipeline Phases

P - generate program address
F - get opcode
D - decode instruction
A - generate read address
R - read operands
X - execute



Full Pipeline

C54x Memory and Buses



◆ Internal: Up to 4 accesses / cycle

◆ External: 1 access / cycle

◆ up to 8M words program

C54 Pipeline Phases

Pipeline Phases

P - generate program address

F - get opcode

D - decode instruction

A - generate read address

R - read operands

X - execute



Full Pipeline

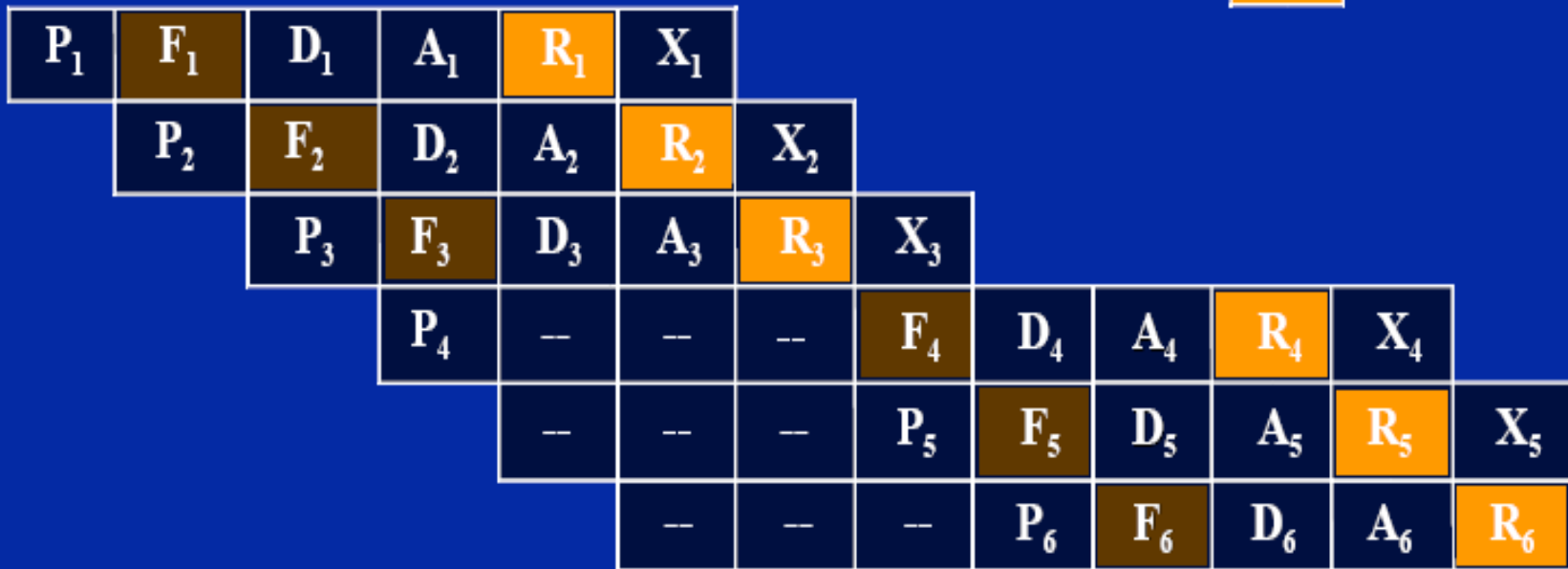
C54x Pipeline Bus/Hardware Use

P	Generate Program address	→ P_A	PC
F	Get opcode	← P_D	Program mem
D	Decode instruction		Decoder
A	Generate Read address	→ D_A/C_A	ARs, ARAU
R	Read operands	← D_D/C_D	Data mem
	Generate Write address	→ E_A	ARs, ARAU
X	Execute instruction		MAC, ALU
	Write result	→ E_D	Data mem

- ◆ When storing results back to memory, the “write” is broken into two phases:
 - generating the write address
 - writing the result
- ◆ Overlaid onto R & X phases

Pipeline Implications 1 of 3

What if all data and program are external?

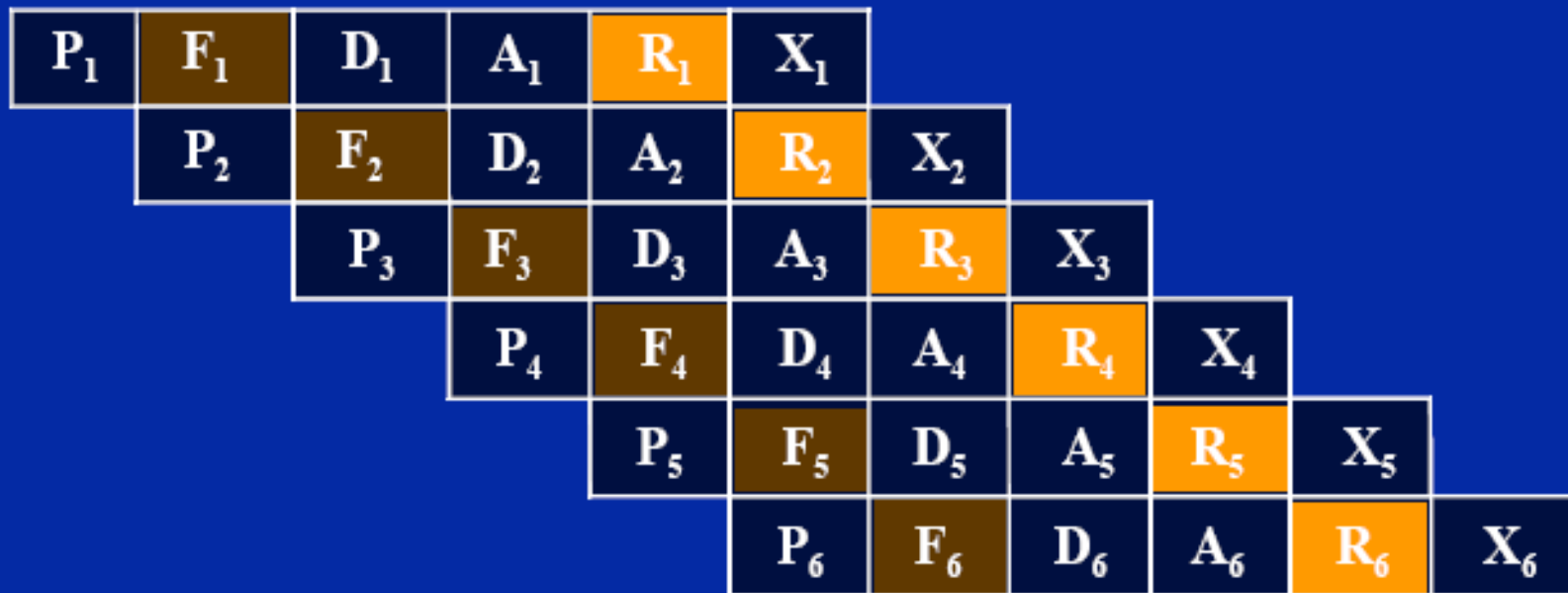


- ◆ External read conflicts with external fetch
- ◆ Can reduce performance by at least 50%

How would you avoid this situation?

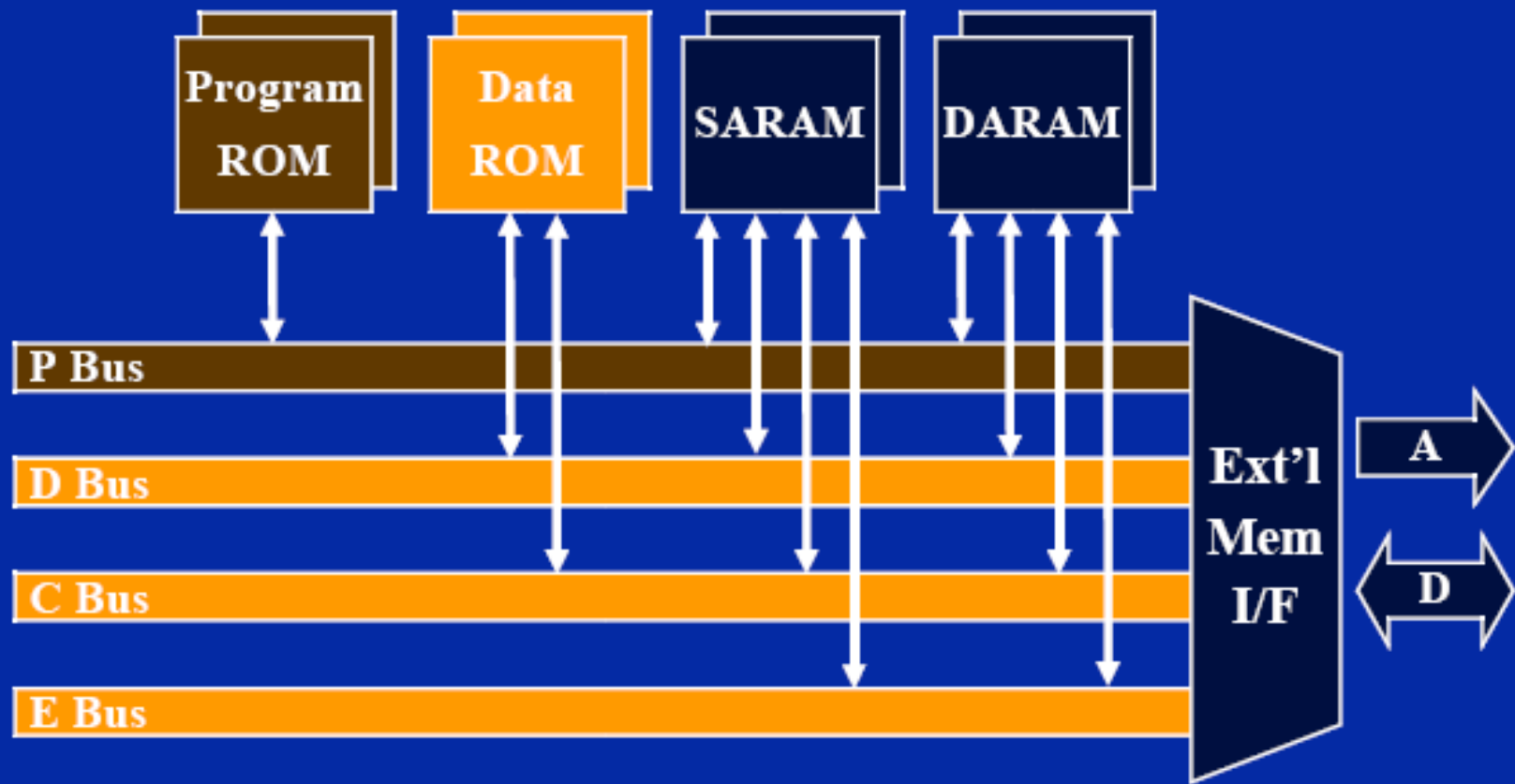
Pipeline Implications 2 of 3

When either
Program or Data
is located internally...
...fetch and read can
occur simultaneously



What if *both* program and data are located internally?

Pipeline Implications 3 of 3



There are no conflicts as long as you follow these rules:

- ◆ **ROM/SARAM - 1 access per block per cycle**
- ◆ **DARAM - 2 accesses per block per cycle**

Size and number of blocks vary based upon device - refer to memory map

Table 2. Bus Usage for Accesses

ACCESS TYPE	ADDRESS BUS				PROGRAM BUS	DATA BUS		
	PAB	CAB	DAB	EAB	PB	CB	DB	EB
Program read	√				√			
Program write	√							√
Data single read			√				√	
Data dual read		√	√			√	√	
Data long (32-bit) read		√(hw)	√(lw)			√(hw)	√(lw)	
Data single write				√				√
Data read/data write			√	√			√	√
Dual read/coefficient read	√	√	√		√	√	√	
Peripheral read			√				√	
Peripheral write				√				√

Legend:

hw = high 16-bit word

lw = low 16-bit word

Bus Structure

The '54x architecture is built around *eight major 16-bit buses* (four program/data buses and four address buses):

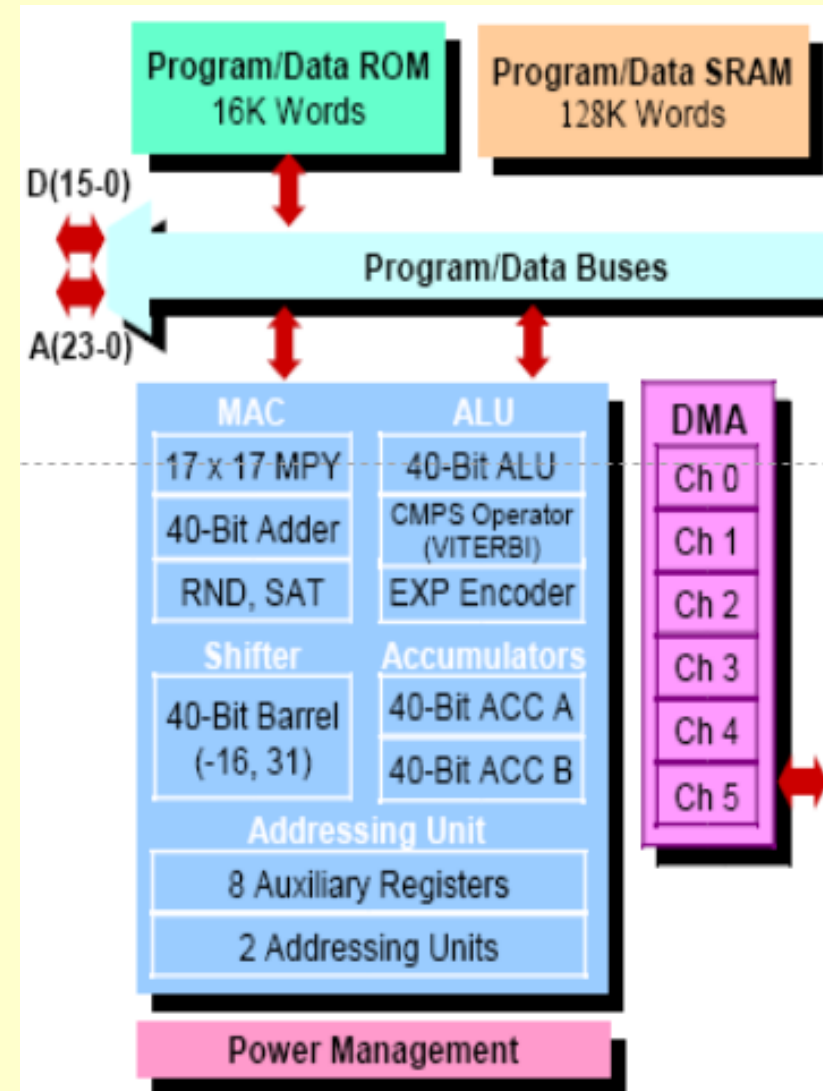
- The program bus (PB) carries the *instruction code* and *immediate operands* from *program memory*.
- Three data buses (CB, DB, and EB) interconnect to various elements, such as the CPU, data address generation logic, program address generation logic, on-chip peripherals, and data memory.
 - ✓ The CB and DB carry the operands that are read from data memory.
 - ✓ The EB carries the data to be written to memory.
- *Four address buses* (PAB, CAB, DAB and EAB) carry the addresses needed for instruction execution.
- The '54x can generate *up to 2 data-memory addresses/cycle* using the two auxiliary register arithmetic units (ARAU0 and ARAU1).

TMS320C54X Key Features

- **Central Processing Unit (CPU)**

The '54x CPU is common to all the '54x devices. The '54x CPU contains:

- 40-bit arithmetic logic unit (ALU)
- Two 40-bit accumulators
- 40-bit Barrel shifter
- 17×17 -bit multiplier
- 40-bit Adder
- Compare, select, and store unit (CSSU)
- Data address generation unit
- Program address generation unit

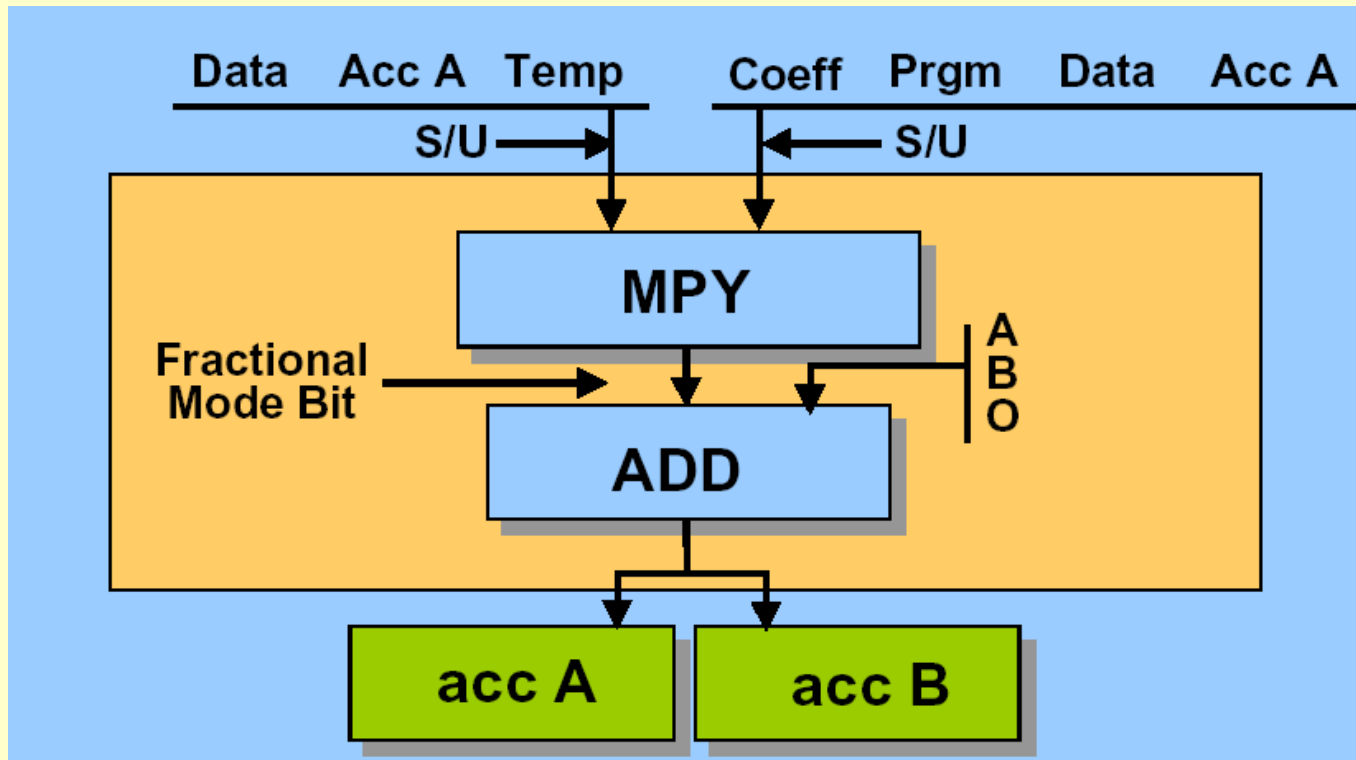


Arithmetic Logic Unit (ALU)

- The '54x performs 2s-complement arithmetic with a 40-bit (ALU) and two 40-bit accumulators (A and B). The ALU can also perform Boolean operations. The ALU uses these inputs:
 - 16-bit immediate value
 - 16-bit word from data memory
 - 16-bit value in the temporary register, T
 - 2 x 16-bit words from data memory
 - 32-bit word from data memory
 - 40-bit word from either accumulator
- The ALU can also function as *two 16-bit ALUs* and perform two 16-bit operations at once.

Accumulators

- *Accumulators A and B* store the output from the ALU or the multiplier/adder block. They can also provide a second input to the ALU; accumulator A can be an input to the multiplier/adder. Each accumulator is divided into three parts:
 - Guard bits (bits 39–32)
 - High-order word (bits 31–16)
 - Low-order word (bits 15–0)

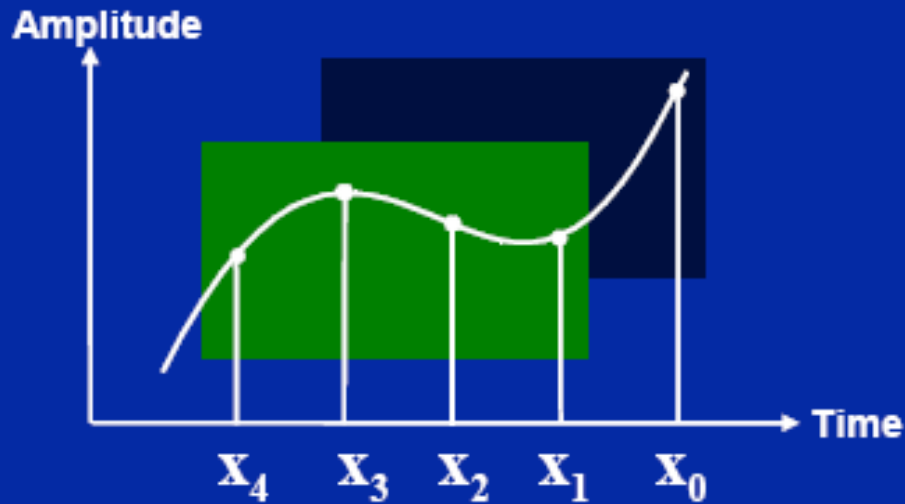


C54X Memory Blocks

- ◆ 192K words \times 16-bit addressable memory space in 3 blocks:
 - ◆ 64K-words program**
 - ◆ 64K-words data,
 - ◆ 64K-words I/O

**extended program memory depending on the C54X version.

Convolution with the C54x



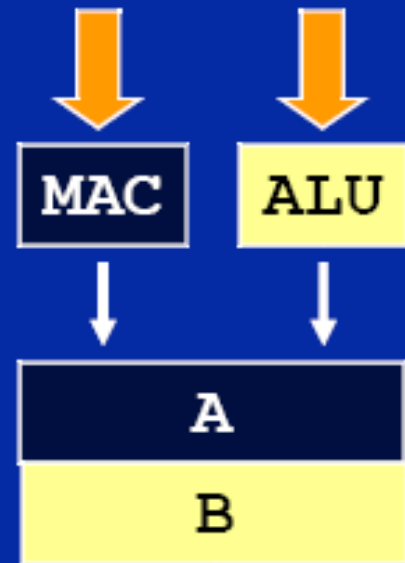
$$y_0 = \sum_{n=0}^3 a_n x_n$$

$$z = x_2 + x_4 + x_3 + x_1$$

MAC *AR2+, *AR3+, A

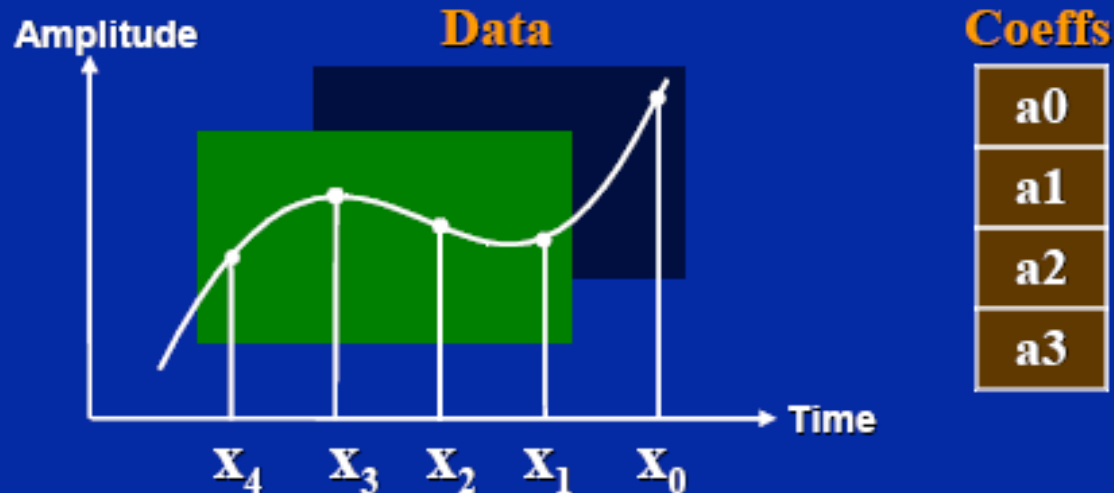
ADD @x2, B ...

Data Read Buses



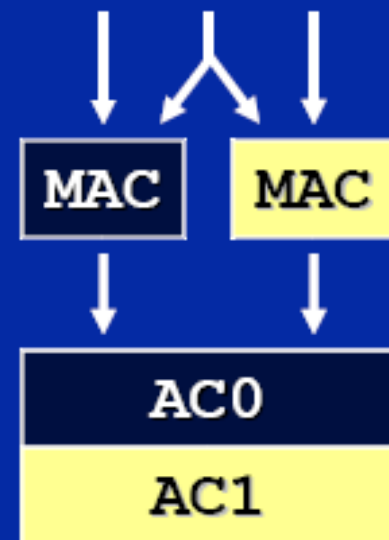
- ◆ Single-cycle MAC
- ◆ Single-cycle ADD

With the C55x it can be done faster!



◆ 2 taps/cycle

Data Read Buses



Results

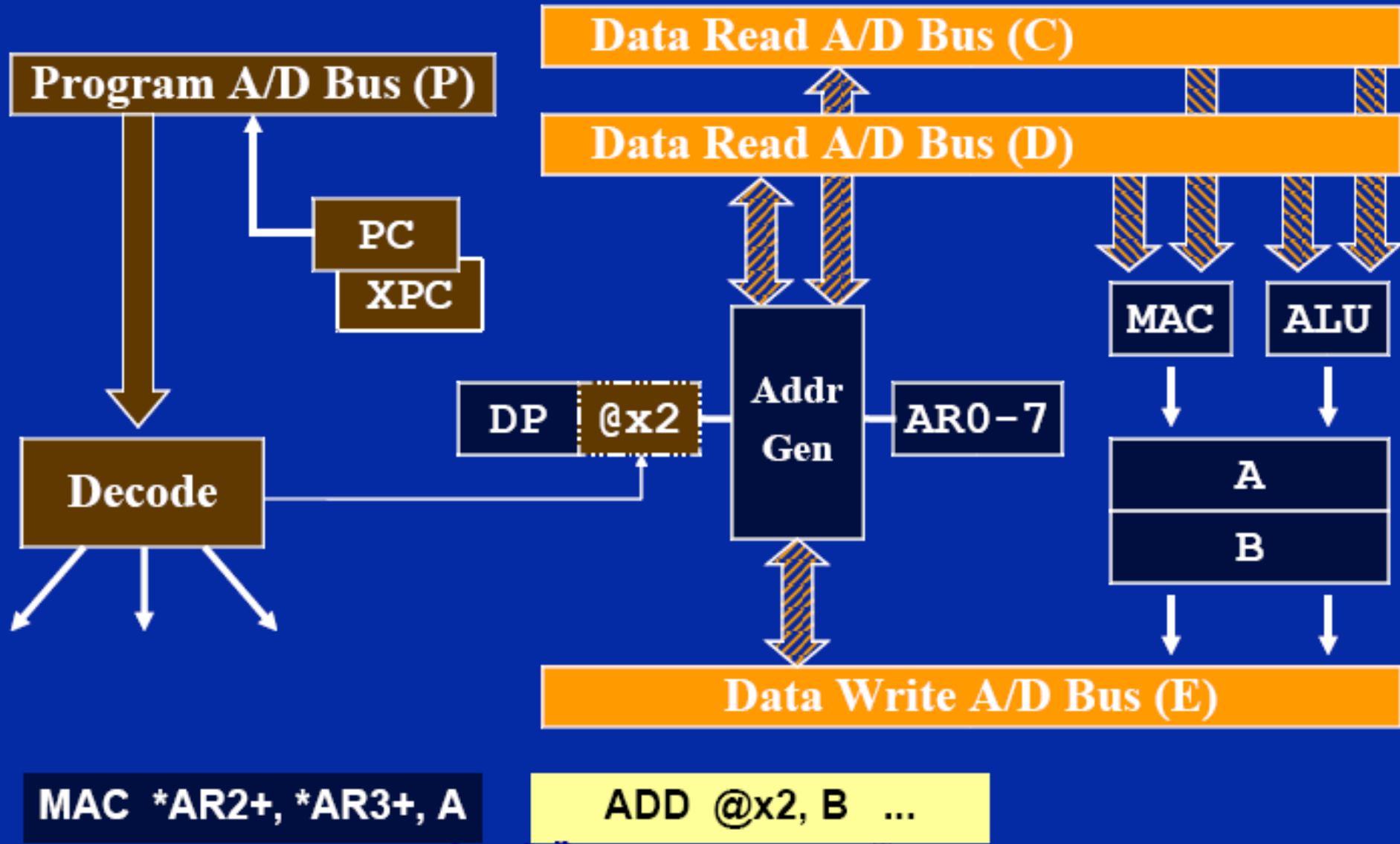
$$y_0 = a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3$$

$$y_1 = a_0x_1 + a_1x_2 + a_2x_3 + a_3x_4$$

C55x: MAC *AR2+, *CDP+, AC0 :: MAC *AR3+, *CDP+, AC1

C54x: MAC *AR2+, *AR3+, A

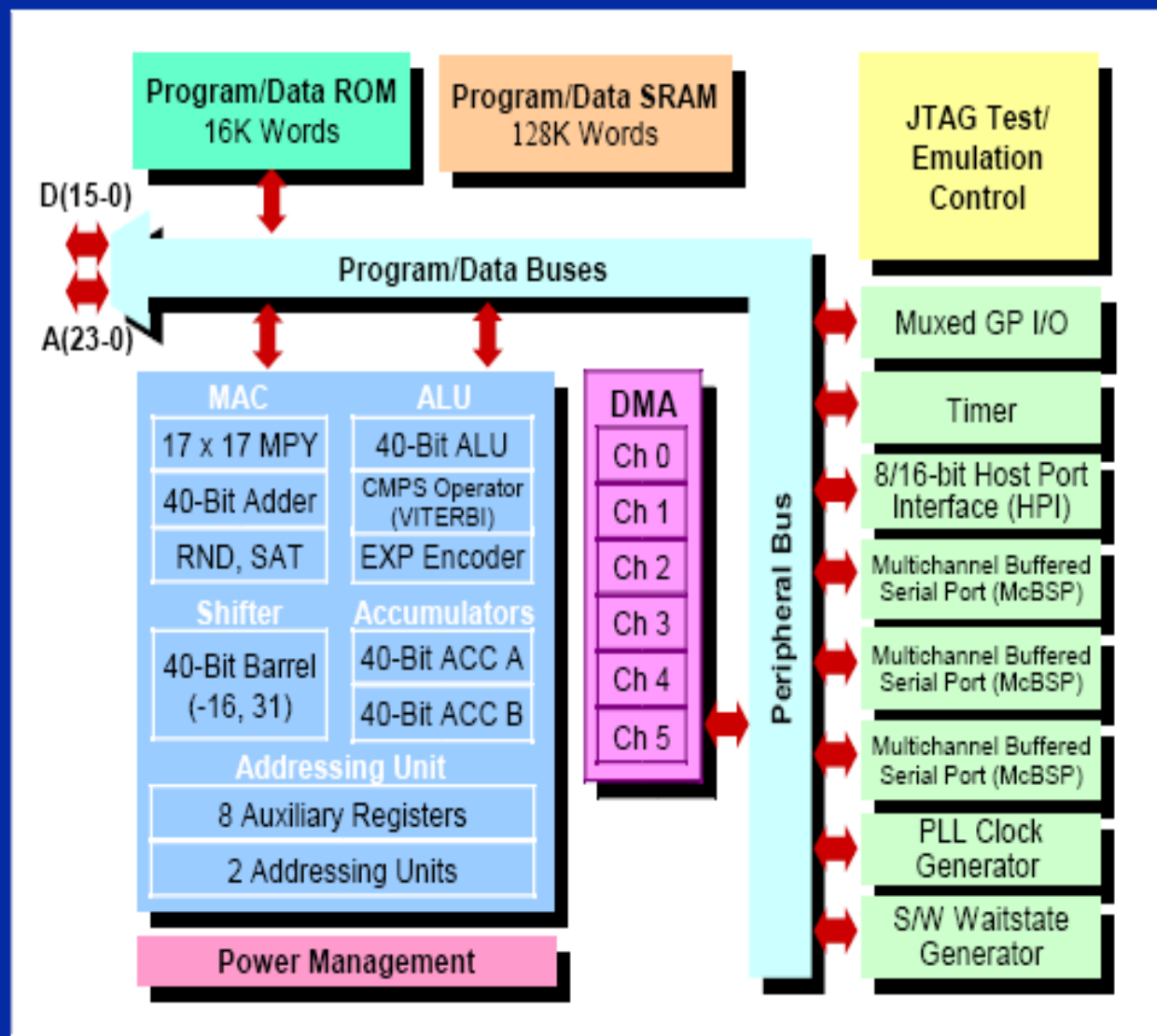
C54x Architecture



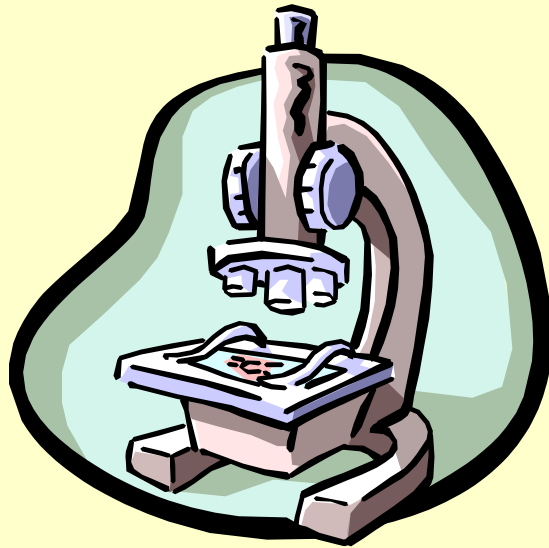
- program counter extension register (XPC)

C54x Block Diagram

- 17x17 MAC Unit
- Saturation and Rounding Hardware
- Two 40-bit ACC's
- 40-bit ALU
- 40-bit Barrel Shifter
- Temporary Register
- Exponent Encoder
- Program and Data Address Generation Units
- Compare, Select and Store Unit
- 4 Internal Bus Pairs
- External Interface

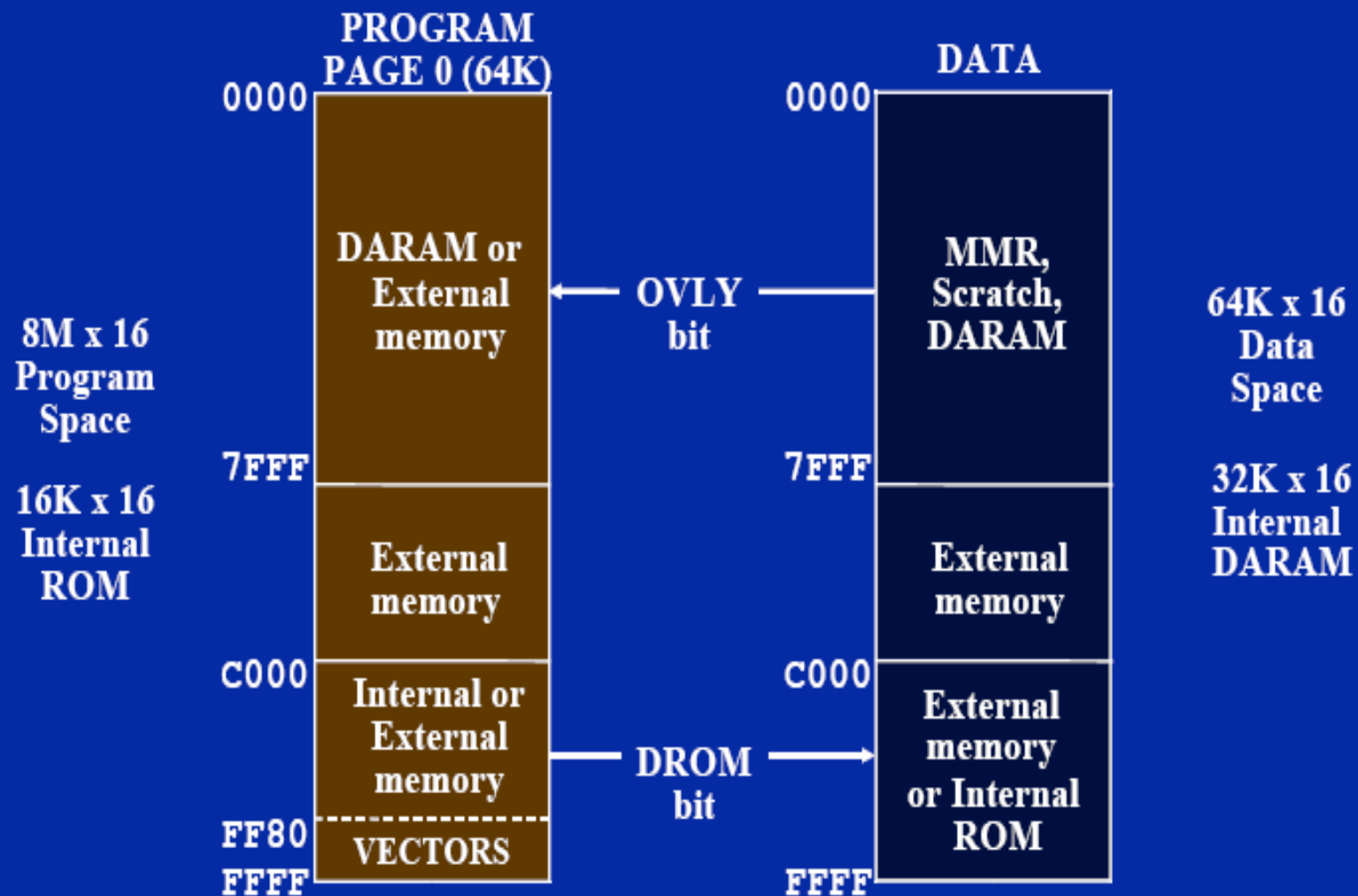


C5416 example



Focus on C54x

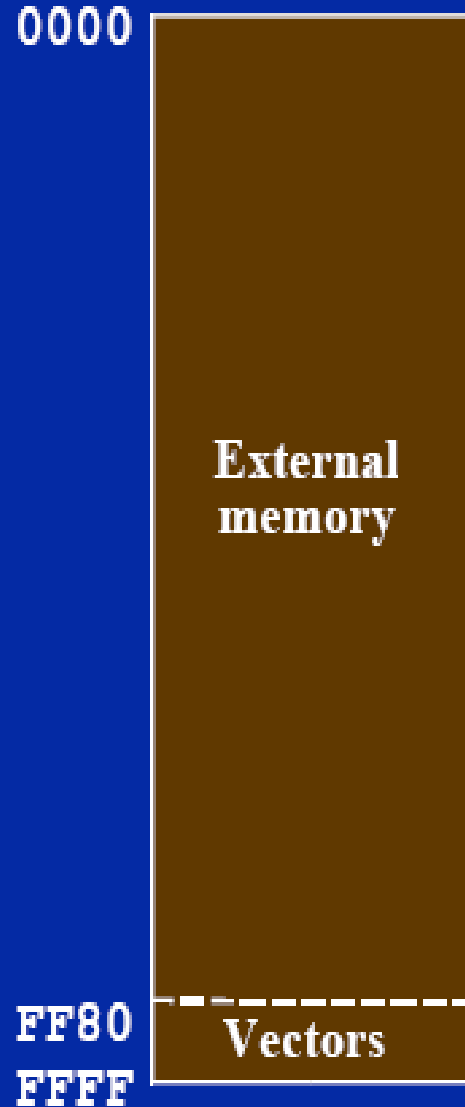
C5409 Memory Maps



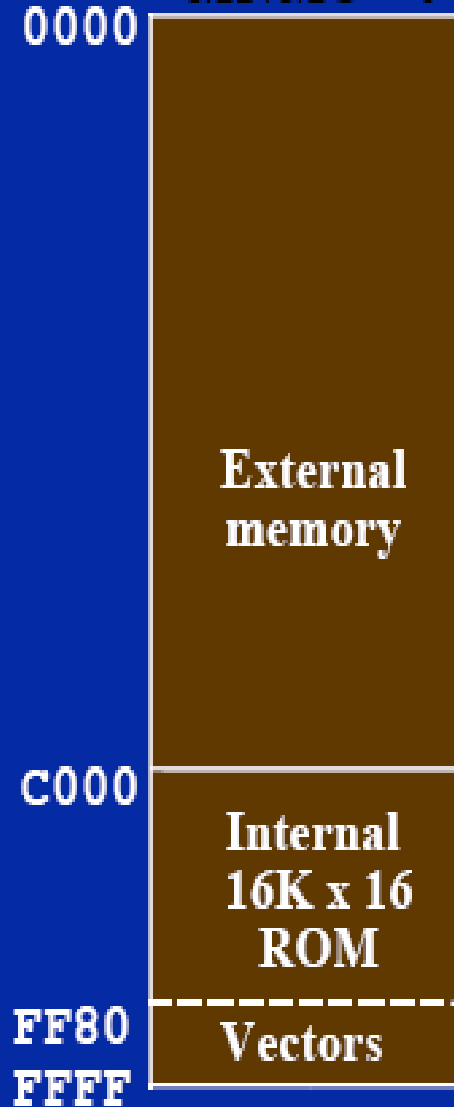
- OVLY** Maps most of on-chip DARAM into Program space
- DROM** Maps most of on-chip Program ROM into Data space

C5409 Program Memory Options (Page 0)

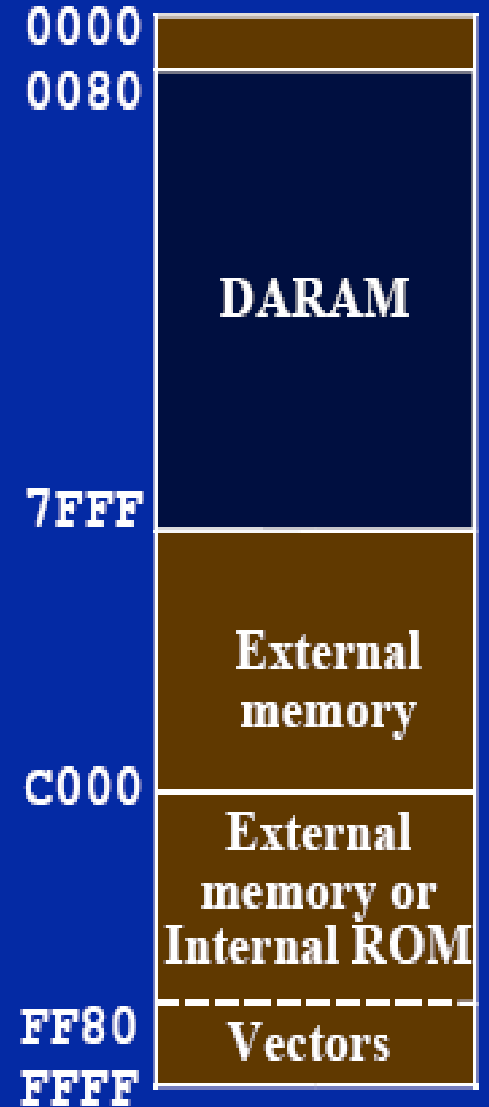
All External
MP/MC = 1



Internal/External
MP/MC = 0

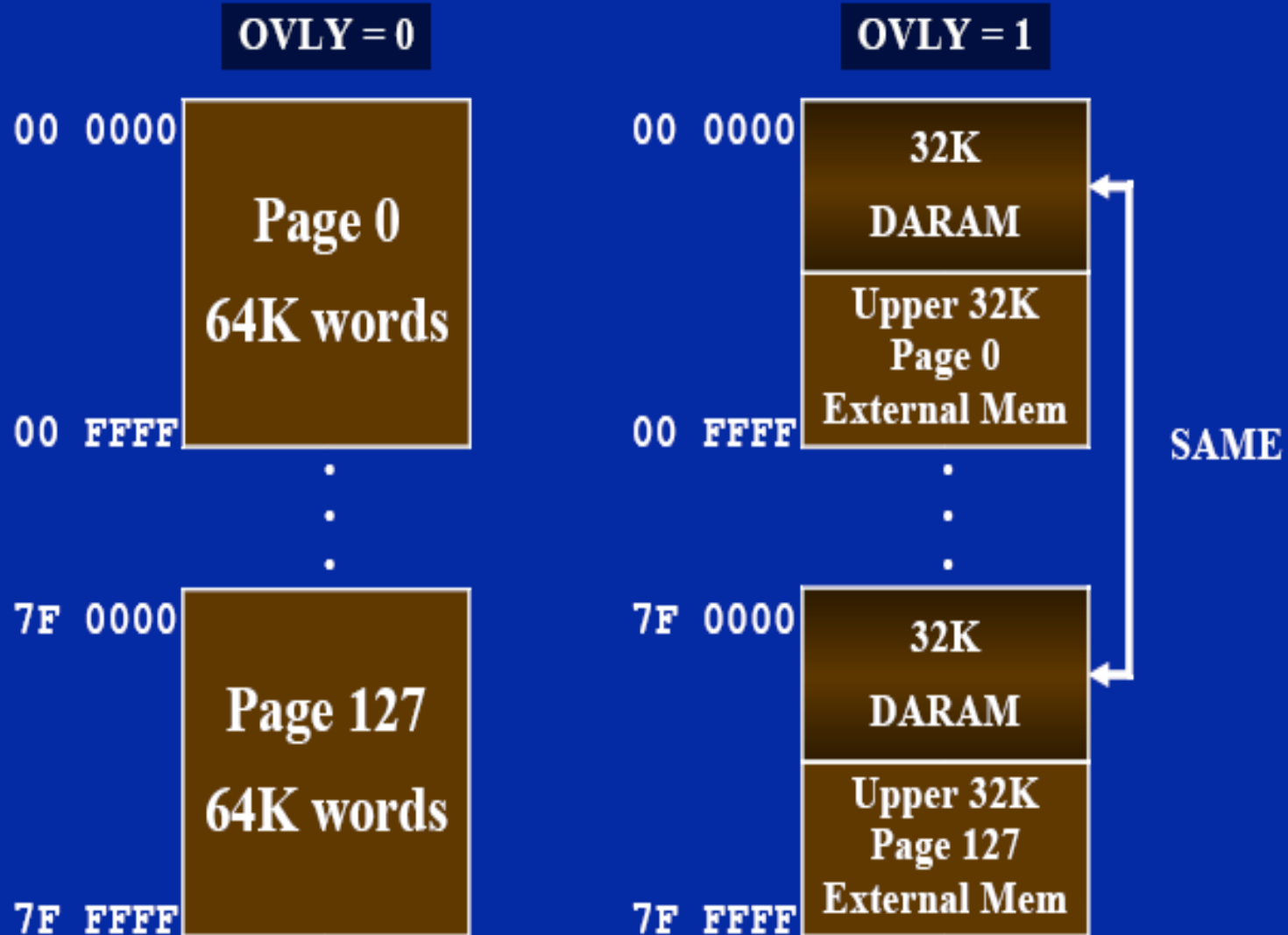


'RAM' Option
OVLV = 1

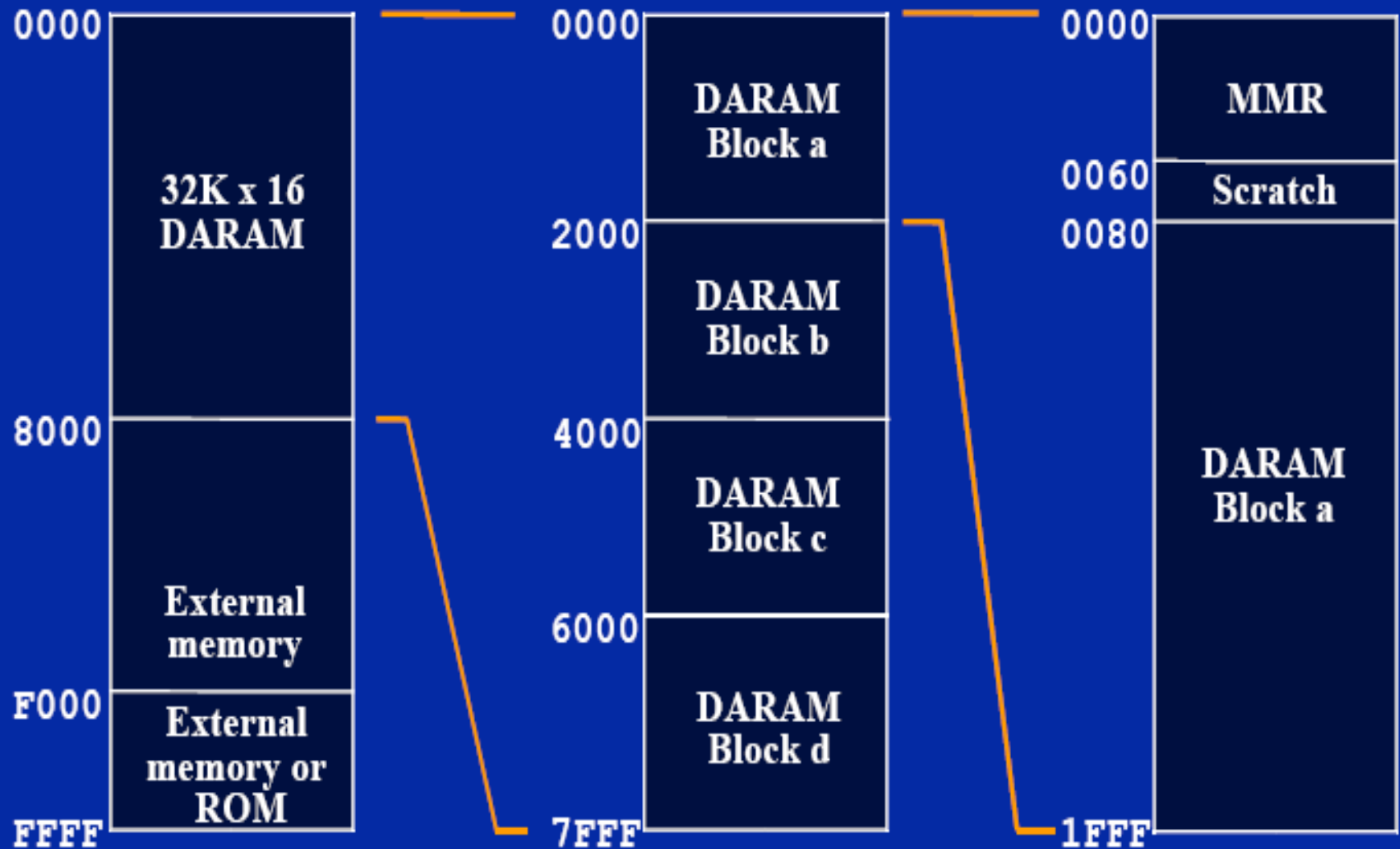


C5409 8M x 16 Program Space

- ◆ The OVLY bit selects between two different program memory maps:



C5409 Data Memory



Memory-Mapped Registers (MMR)

Name	Addr. (Hex)	Description
IMR	0000	Interrupt Mask Register
IFR	0001	Interrupt Flag Register
----	2 - 5	Reserved
ST0	0006	Status 0 Register
ST1	0007	Status 1 Register
AL	0008	A accumulator low (A[15:00])
AH	0009	A accumulator high (A[31:16])
AG	000A	A accumulator guard (A[39:32])
BL	000B	B accumulator low (B[15:00])
BH	000C	B accumulator high (B[31:16])
BG	000D	B accumulator guard (B[39:32])
T	000E	Temporary Register
TRN	000F	Transition Register

Name	Addr. (Hex)	Description
AR0	0010	Auxiliary Register 0
AR1	0011	Auxiliary Register 1
AR2	0012	Auxiliary Register 2
AR3	0013	Auxiliary Register 3
AR4	0014	Auxiliary Register 4
AR5	0015	Auxiliary Register 5
AR6	0016	Auxiliary Register 6
AR7	0017	Auxiliary Register 7
SP	0018	Stack Pointer Register
BK	0019	Circular Size Register
BRC	001A	Block Repeat Counter
RSA	001B	Block Repeat Start Address
REA	001C	Block Repeat End Address
PMST	001D	PMST Register
-----	01E-01F	Reserved

Table 7. Core Processor Memory-Mapped Registers

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
IMR	0	0	Interrupt mask register
IFR	1	1	Interrupt flag register
—	2–5	2–5	Reserved for testing
STD	6	6	Status register 0
ST1	7	7	Status register 1
AL	8	8	Accumulator A low word (15–0)
AH	9	9	Accumulator A high word (31–16)
AG	10	A	Accumulator A guard bits (39–32)
BL	11	B	Accumulator B low word (15–0)
BH	12	C	Accumulator B high word (31–16)
BG	13	D	Accumulator B guard bits (39–32)
TREG	14	E	Temporary register
TRN	15	F	Transition register
AR0	16	10	Auxiliary register 0
AR1	17	11	Auxiliary register 1
AR2	18	12	Auxiliary register 2
AR3	19	13	Auxiliary register 3
AR4	20	14	Auxiliary register 4
AR5	21	15	Auxiliary register 5
AR6	22	16	Auxiliary register 6
AR7	23	17	Auxiliary register 7
SP	24	18	Stack pointer register
BK	25	19	Circular buffer size register
BRC	26	1A	Block-repeat counter
RSA	27	1B	Block-repeat start address
REA	28	1C	Block-repeat end address
PMST	29	1D	Processor mode status (PMST) register
XPC	30	1E	Extended program counter ('548 and '549 only)
—	31	1F	Reserved

Table 8. Peripheral Memory-Mapped Registers ('541 Only)

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
DRR0	32	20	Serial port 0 data-receive register
DXR0	33	21	Serial port 0 data-transmit register
SPC0	34	22	Serial port 0 control register
—	35	23	Reserved
TIM	36	24	Timer register
PRD	37	25	Timer period register
TCR	38	26	Timer control register
—	39	27	Reserved
SWWSR	40	28	S/W wait-state register
BSCR	41	29	Bank-switching control register
—	42–47	2A–2F	Reserved
DRR1	48	30	Serial port 1 data-receive register
DXR1	49	31	Serial port 1 data-transmit register
SPC1	50	32	Serial port 1 control register
—	51	33	Reserved
—	52–95	34–5F	Reserved

DSK5416 Default Memory Map

◆ Status bit values:

- ◆ *MEMTYPE_DS=1, MEMTYPE_PS=0, MP/MC=0, DROM=1, OVLY=1.*

Program memory

Hex	Page 0 Program
0x0000	Reserved (OVLY=1)
0x007F	
0x0080	On-Chip DARAM0-3 (OVLY=1)
0x7FFF	
0x8000	External
0xBFFF	
0xC000	On-Chip ROM (4K x 16 bit)
0xFEFF	
0xFF00	Reserved
0xFF7F	
0xFF80	Interrupts (On-Chip)
0xFFFF	

MP/MC=0
(Microcomputer Mode)

Hex	Program	Hex	Program	Hex	Program
010000	On-Chip DARAM0-3 (OVLY=1)	020000	On-Chip DARAM0-3 (OVLY=1)	030000	On-Chip DARAM0-3 (OVLY=1)
017FFF		027FFF		037FFF	
018000	On-Chip (DARAM4-7 (MP/MC=0))	028000	On-Chip (SARAM0-3 (MP/MC=0))	038000	On-Chip (SARAM4-7 (MP/MC=0))
01FFFF		02FFFF		03FFFF	
	Page 1 XPC=1		Page 2 XPC=2		Page 3 XPC=3

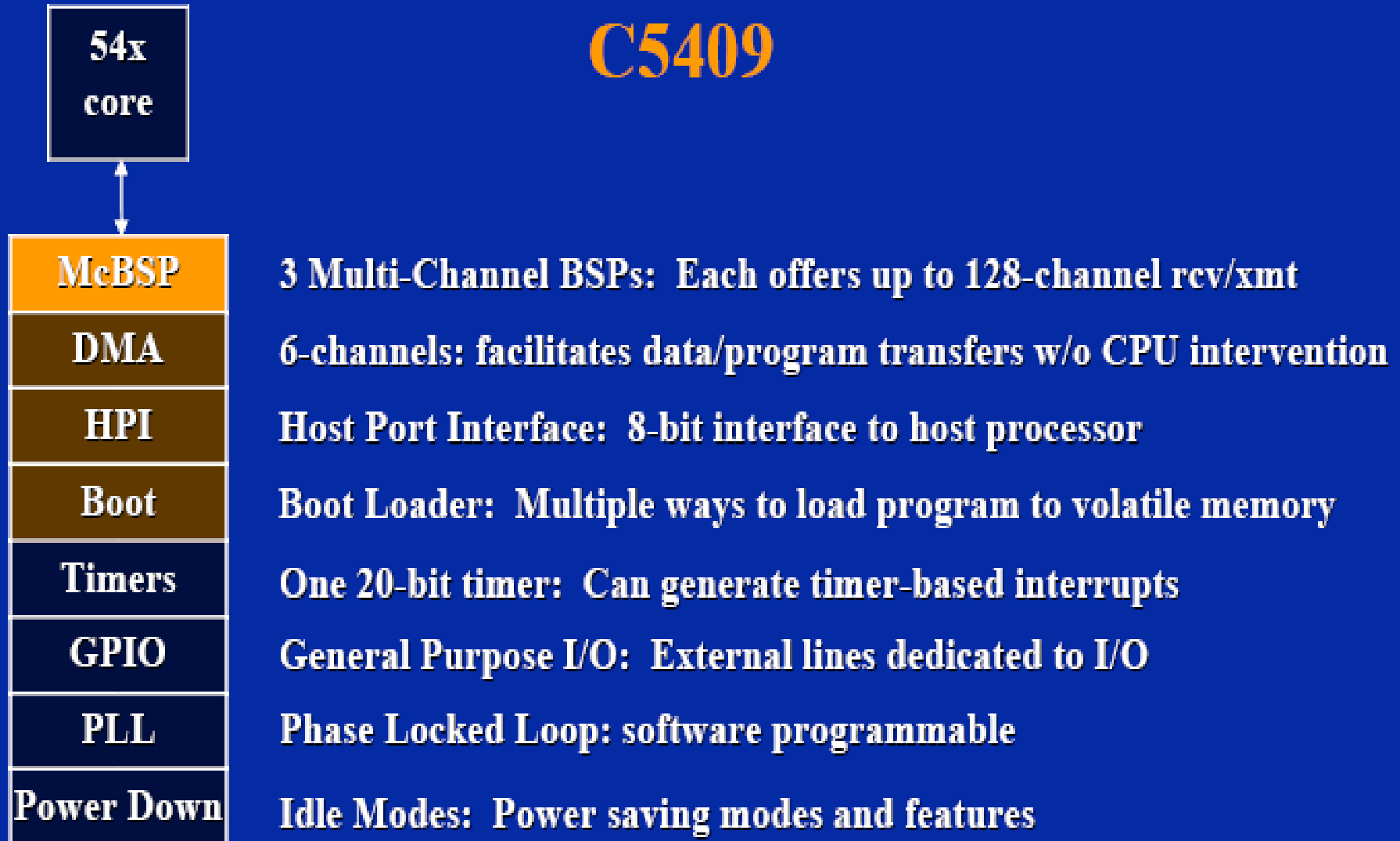
Data memory

Hex	Data
0x0000	Memory-Mapped Registers
0x005F	
0x0060	Scratch-Pad RAM
0x007F	
0x0080	On-Chip DARAM0-3 (32K x 16 bit)
0x7FFF	
0x8000	On-Chip DARAM4-7 (DROM=1)
0xFFFF	

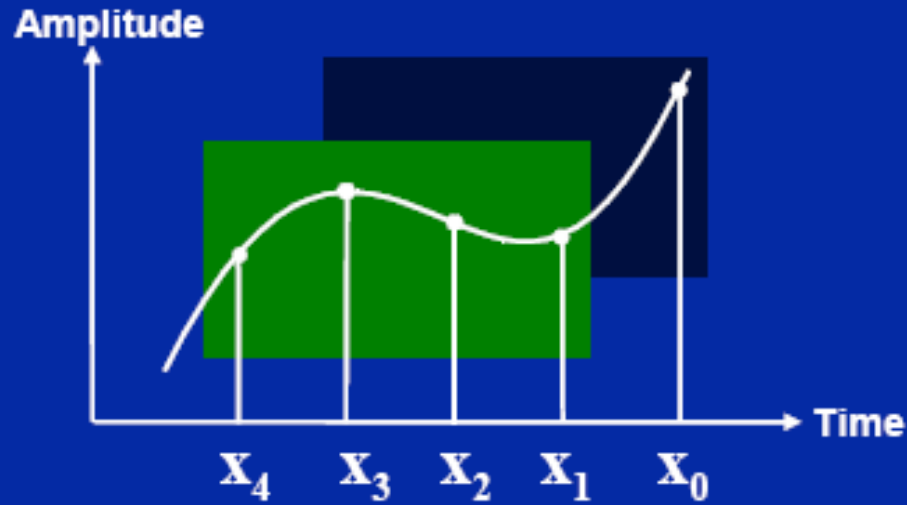


C5409 Peripheral Overview

C5409



Convolution with the C54x



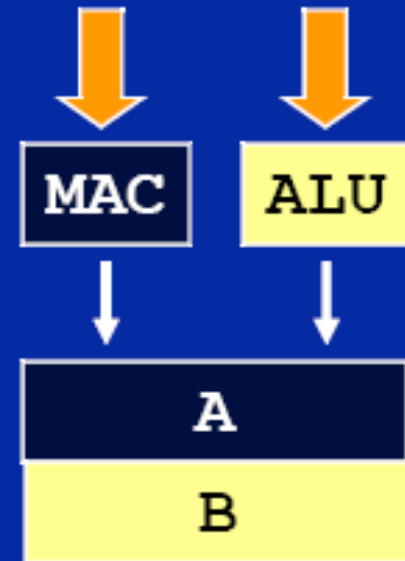
$$y_0 = \sum_{n=0}^3 a_n x_n$$

$$z = x_2 + x_4 + x_3 + x_1$$

MAC *AR2+, *AR3+, A

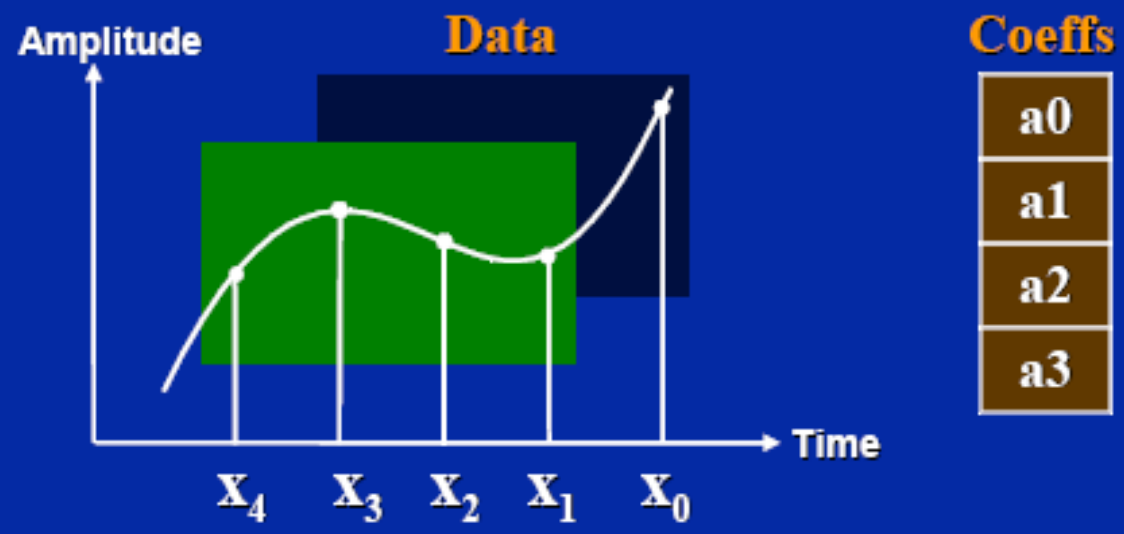
ADD @x2, B ...

Data Read Buses

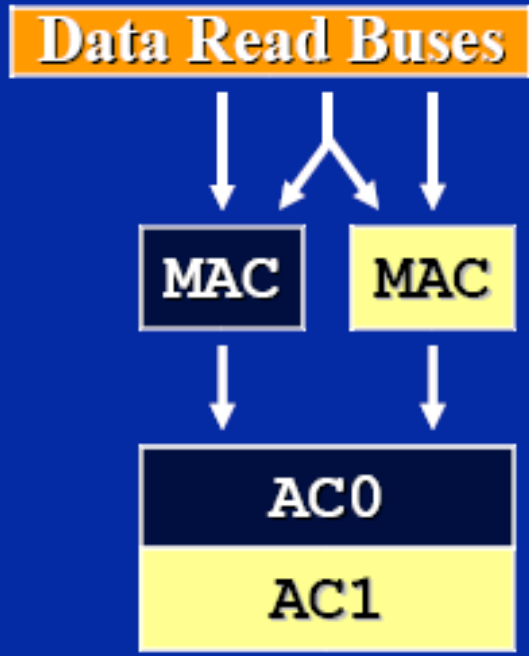


- ◆ Single-cycle MAC
- ◆ Single-cycle ADD

With the C55x it can be done faster!



◆ 2 taps/cycle



Results

$$y_0 = a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3$$

$$y_1 = a_0x_1 + a_1x_2 + a_2x_3 + a_3x_4$$

C55x: MAC *AR2+, *CDP+, AC0 :: MAC *AR3+, *CDP+, AC1

C54x: MAC *AR2+, *AR3+, A

C54x Review

- ◆ Name the buses on the C54x
- ◆ How large are the accumulators?
- ◆ How many adders are on the part?
- ◆ Where are the Memory Mapped Registers located?
- ◆ Where is the Reset Vector located?

C54x Review - Answers

◆ Name the buses on the C54x

PA,PD CA,CD DA,DD EA,ED

◆ How large are the accumulators?

40 bits

◆ How many adders are on the part?

2, one in the MAC and the other in the ALU

◆ Where are the Memory Mapped Registers located?

From 0x00 to 0x5F in Data Memory

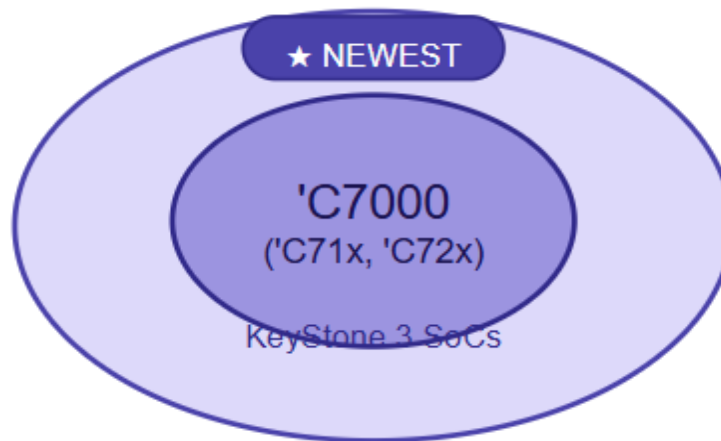
◆ Where is the Reset Vector located?

0xFF80 in Program Memory

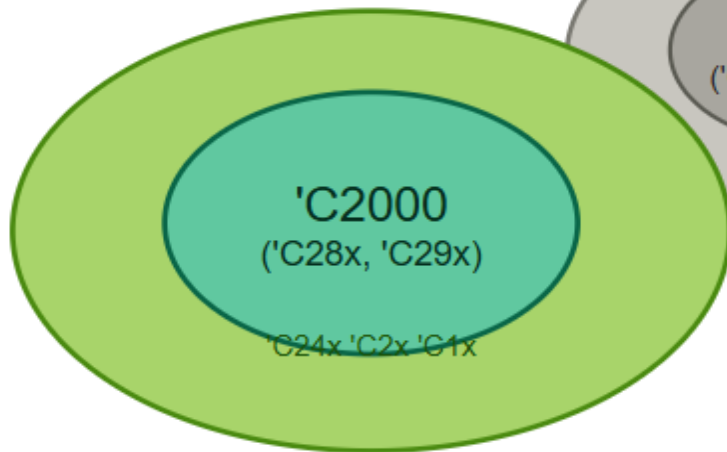
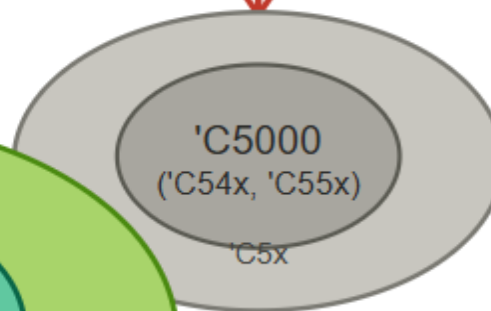
Different needs? Multiple families!

Best perf. / lowest power

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



△ Legacy / EOL



Lowest cost

Real-time control

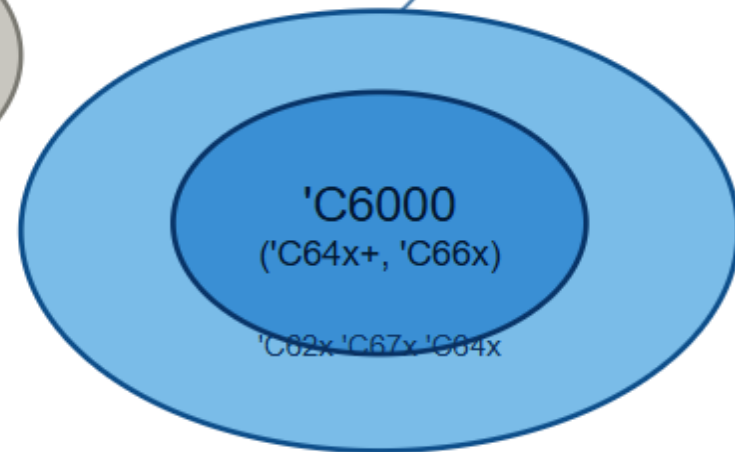
- Motor & power conversion
- Industrial automation
- Edge AI / NPU (2024)

Legacy — no new designs

Was: wireless phones, VoIP

Was: modems / telephony

→ Migrate to Sitara AM57x

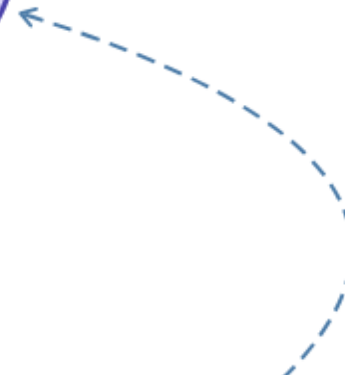


Maximum performance

Multi-channel / function

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

code
compat.



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; generally 5x 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.