

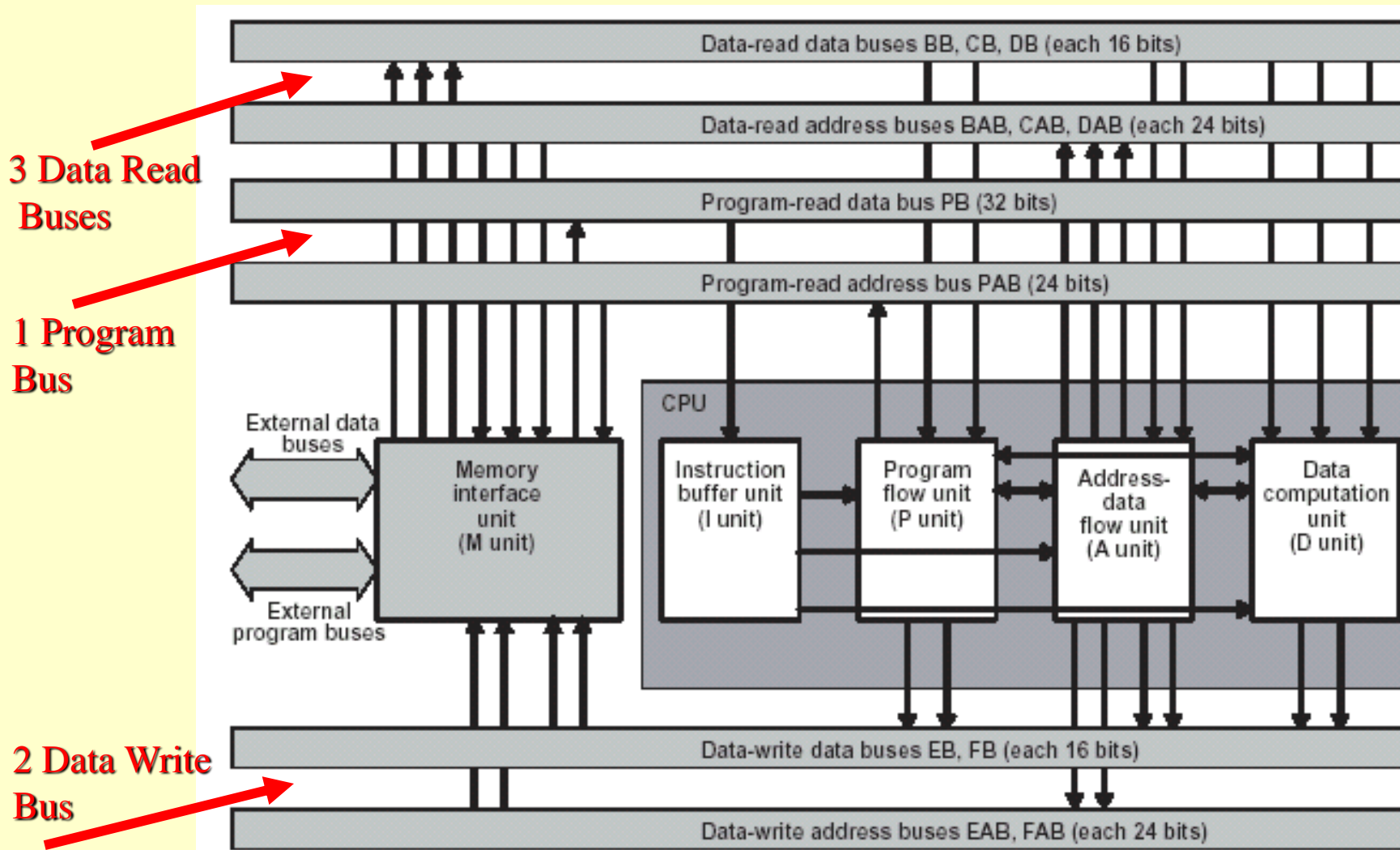
C11. TMS320C55X

Objectives:

- ✓ TMS320C55x Key Features
- ✓ Comparison Between C54x and C55x
- ✓ C55x Addressing Modes

- **Refereces: TI site ; spru371d.pdf**

TMS320C55X DSP Block Diagram








TMS320C55x Key Features

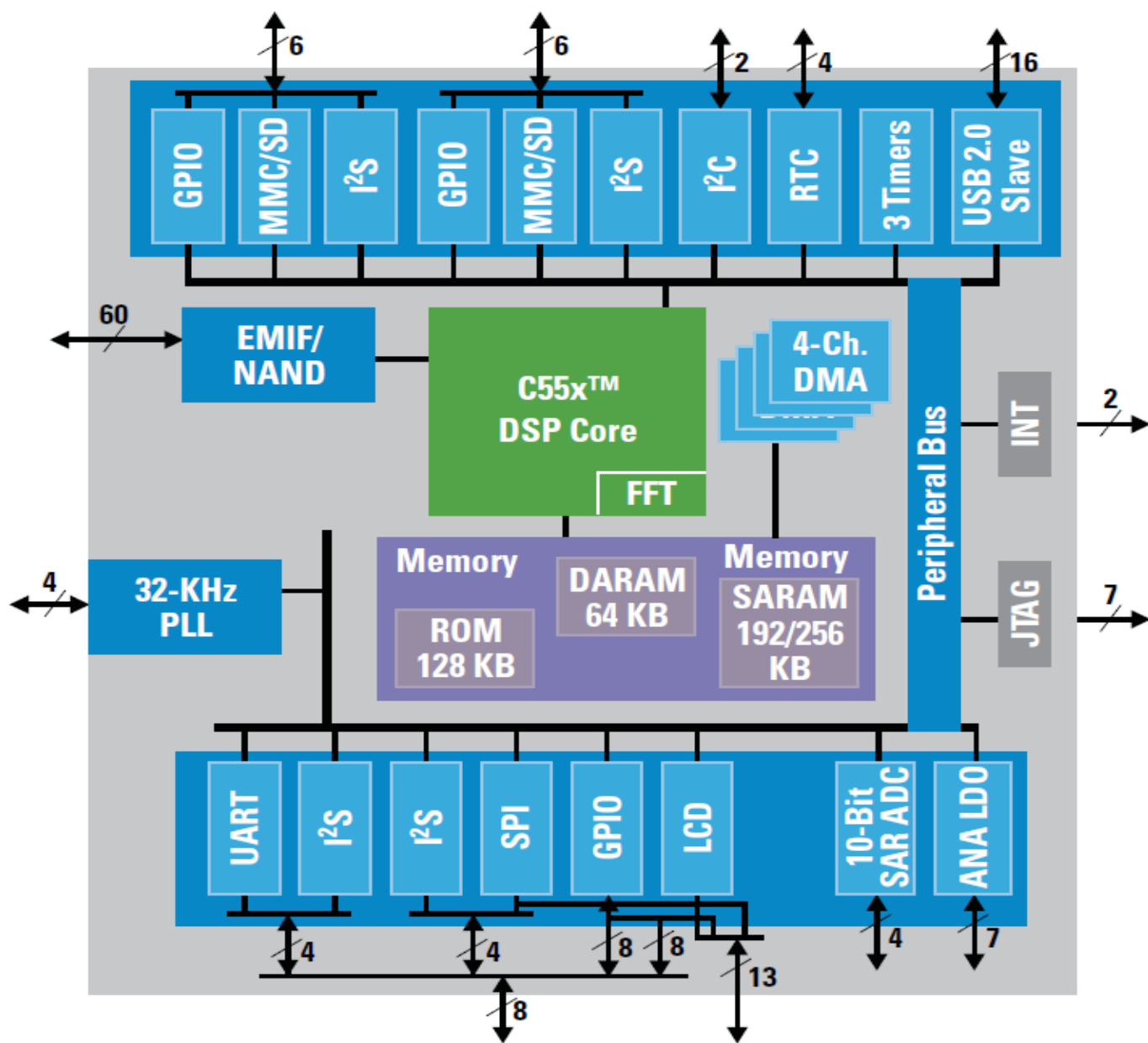
- 32 x 16-bit Instruction buffer queue (IBQ)
- Two 17-bit x17-bit MAC units
- One 40-bit ALU
- One 40-bit Barrel Shifter
- One 16-bit ALU
- Four 40-bit accumulators
- 12 independent buses:
 - Three data read buses
 - Two data write buses
 - Five data address buses
 - One program read bus
 - One program address bus

**Units work
in parallel**

C55x Product Family

- C5545 brings the ultra-low power and optimized performance of the C55xx family to the smallest package ever offered.
- Even with the 7x7mm package, 4-layer boards are still possible without the use of high density interconnects (HDI) or other expensive fabrication techniques.

	DSP TMS320C55x	Integrated Peripherals	On-Chip Memory	Co- Processor	Integrated Power Management	Package
Software Compatible Pin-to-Pin Compatible	C5545  60/100 MHz	USB LCD SAR ADC	320KB	FFT	3 LDOs	7x7mm 0.5mm pitch
	C553x  50/100 MHz	USB LCD SAR ADC	64-320KB	FFT	1-3 LDOs	12x12mm 0.8mm pitch
	C5515  60/75/100/120 MHz	USB LCD SAR ADC EMIF	256-320KB	FFT	3 LDOs	10x10mm 0.65mm pitch
	C5505  60/75/100/120/150 MHz	USB LCD SAR ADC EMIF	256-320KB	FFT	1-3 LDOs	10x10mm 0.65mm pitch
	C5517  75/175/200 MHz	USB UHPI SAR ADC McSPI EMIF McBSP	256-320KB	FFT	3 LDOs	10x10mm 0.65mm pitch



TMS320C5505 DSP Block Diagram

The C5505 DSP is one of the latest TI C5000 DSPs with the best-in-class combination of standby and active power and an integration level optimized for portable audio/voice, medical and many other applications.

More C55x Features

- User-configurable IDLE domains
- Variable length instructions and efficient block repeat operations
- 2 x MAC operations in a single cycle
- Performs high precision arithmetic and logical operations
- Shift a 40-bit result up to 31 bits to the left, or 32 bits to the right
- Performs arithmetic in a simpler ALU of 16 bits.
- Hold results of computations and reduce the required memory traffic (4 Accumulators)

**Optimised for
high performances**

Comparison Between C54x and C55x

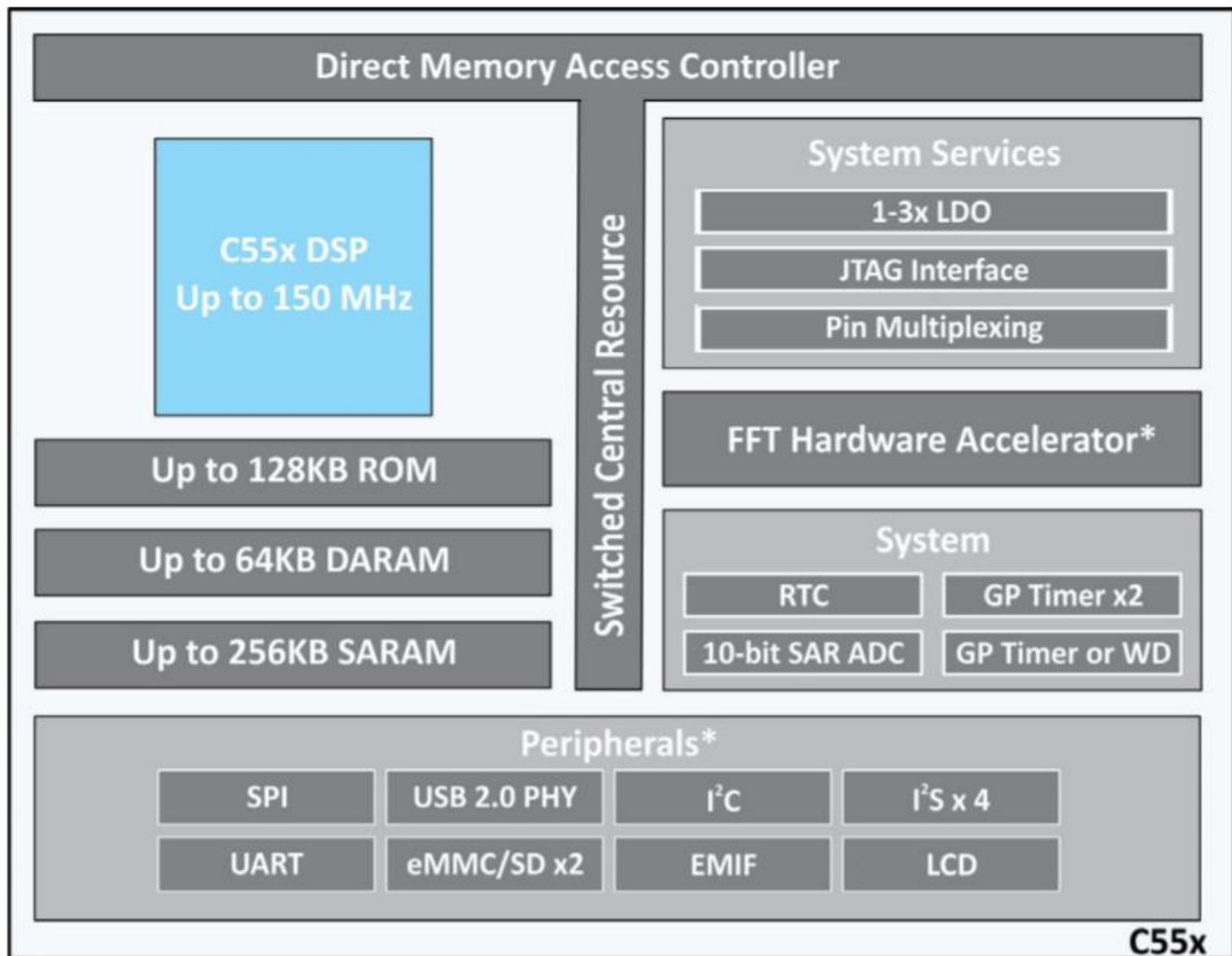
	C54x	C55x
MACs	1	2
Accumulators	2	4
Read buses	2	3
Write buses	1	2
Program fetch	1	1
Address buses	4	6
Program word size	16 bits	8/16/24/32/40/48 bits
Data word size	16 bits	16 bits
Auxiliary Register ALUs	2 (16-bit each)	3 (24-bit each)
ALU	1 (40-bit)	1 (40-bit)/1 (16-bit)
Auxiliary Registers	8	8
Data Registers	0	4
Memory	separate data/prog	Unified space

Performances Features Comparison between C54x and C55x

- 30-160 MIPS and MACs for the C54x compared to 140-800 MIPS for the C55x (5 x better)
- Core Power consumption improves by 6X from 0.32 mW/MIPS for the C54x to 0.05 mW/MIPS for the C55x.
- Variable instruction length (8 - 48 bits) for the C55x and 16 bits for the C54x gives a better code density.
- C55x has twice as many MACs (2 and 1), Accumulators (4 and 2), program fetch words (32 and 16 bits)

**Same family
but big differences!**

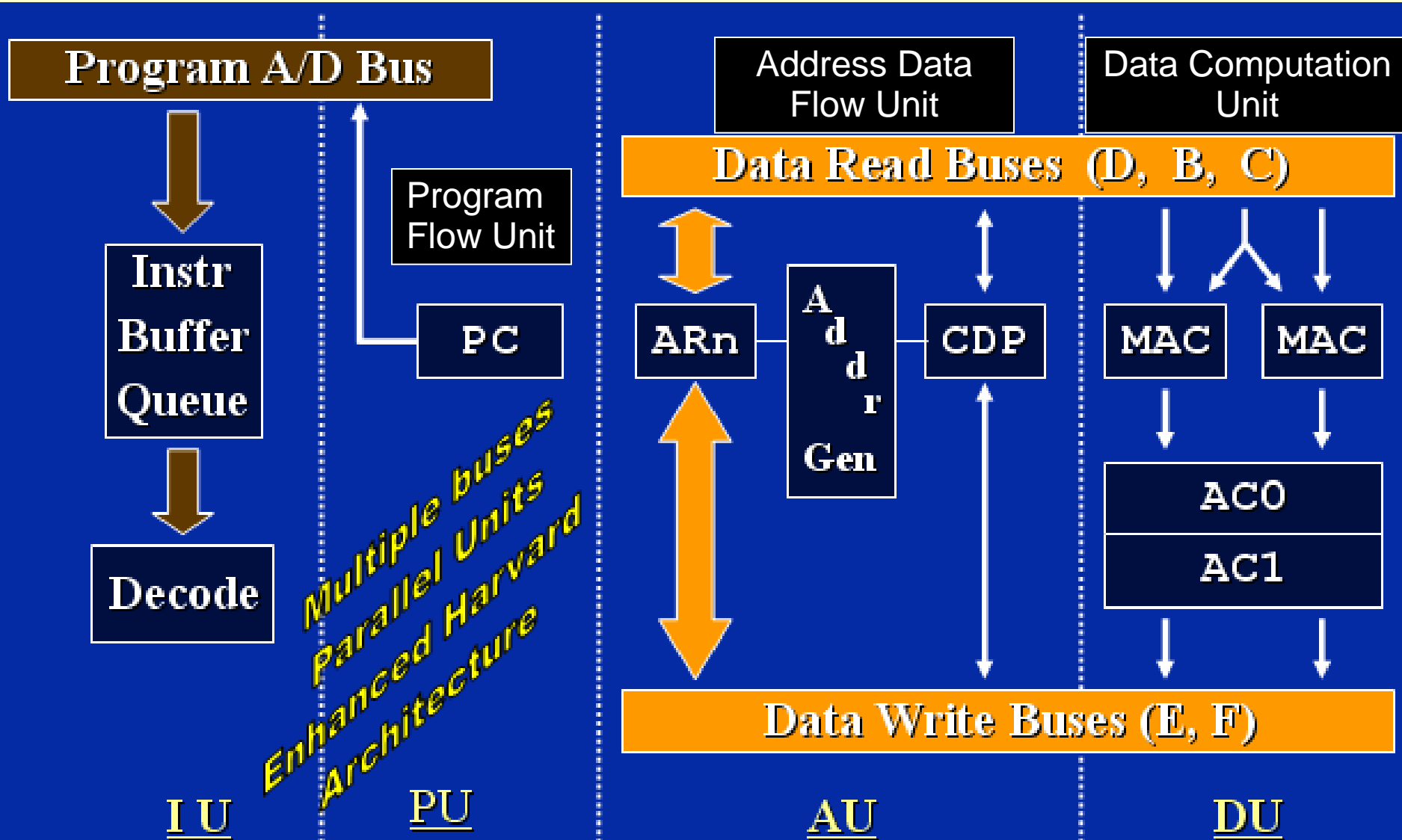
C55x Architecture



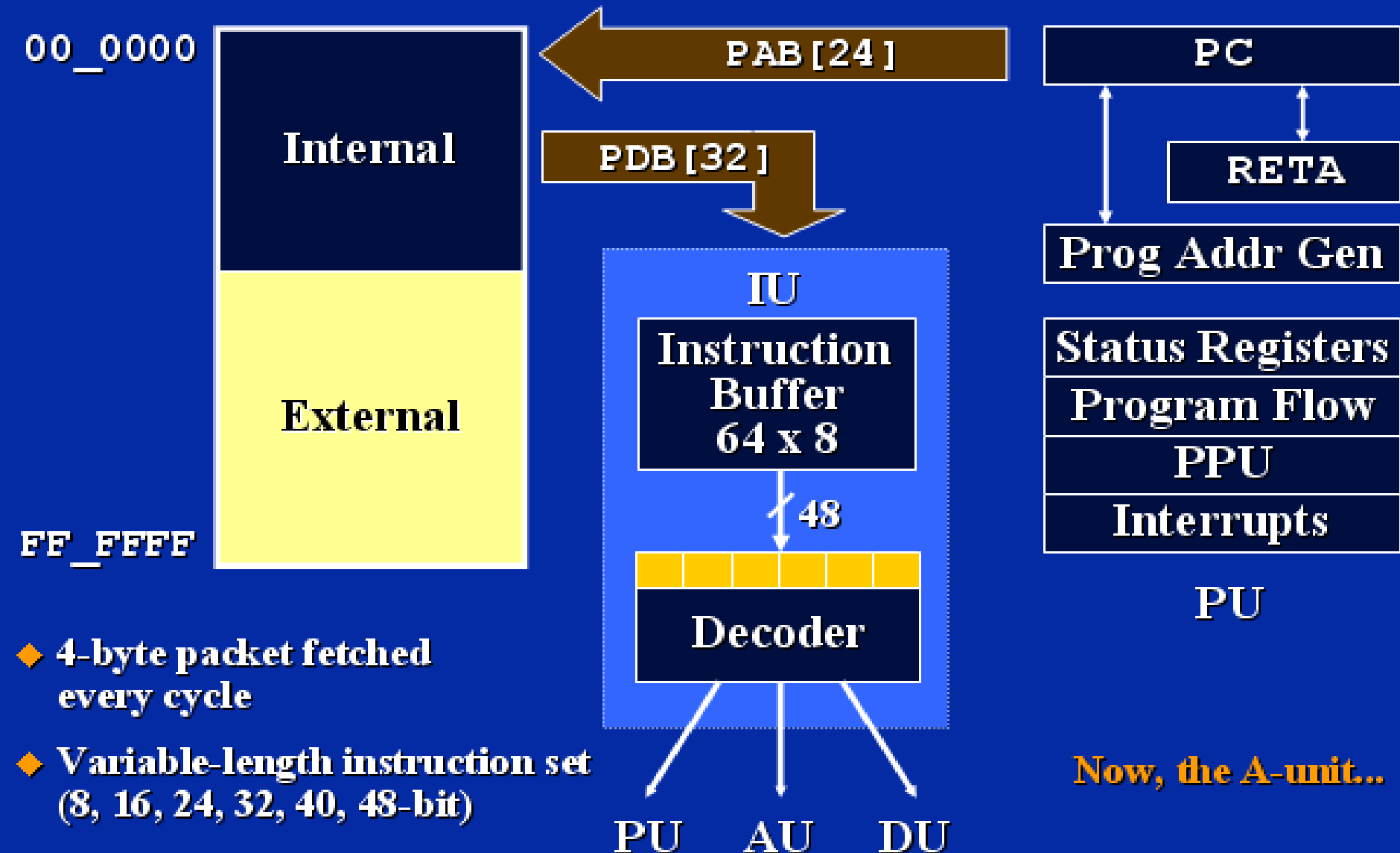
* Availability of features varies by device. Refer to the data sheet.

For more information: www.ti.com/c55x

C55x Architecture



C55x Instruction and Program Units



Now, the A-unit...

- ◆ 4-byte packet fetched every cycle
- ◆ Variable-length instruction set (8, 16, 24, 32, 40, 48-bit)

C55x Addressing Unit (AU)

00_0000

First 64KW Pg 0

BAB [24]

CAB [24]

DAB [24]

Last 64KW Pg 127

CB [16]

DB [16]

FF_FFFF

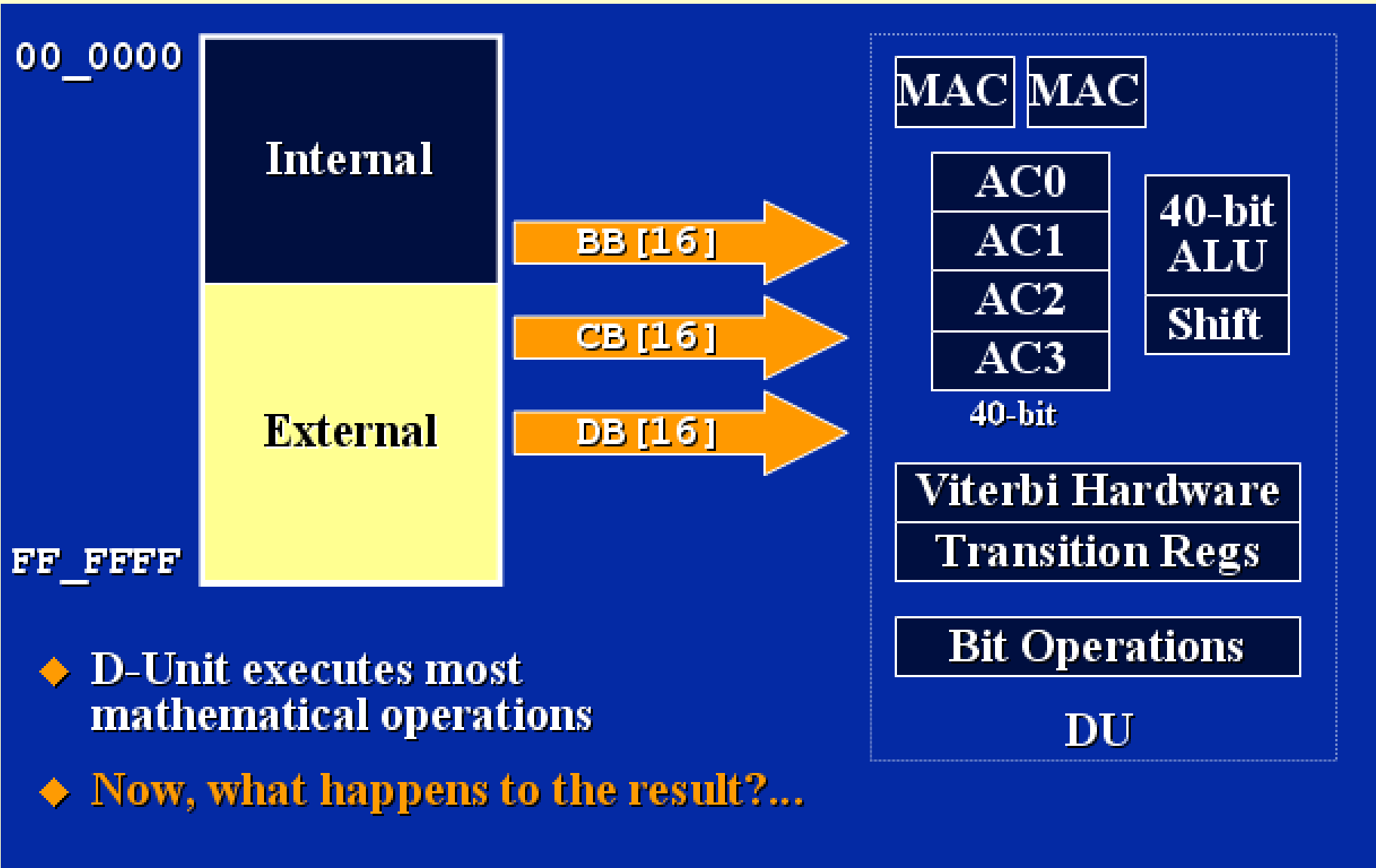
A d d r e s s i n g		ARAU
	X	ARO-7
	X	CDP
	X	DP
23/16-bit		

ALU/Shft	Stack Pointers Circular Buffers 23/16-bit
T0	
T1	
T2	
T3	
16-bit	

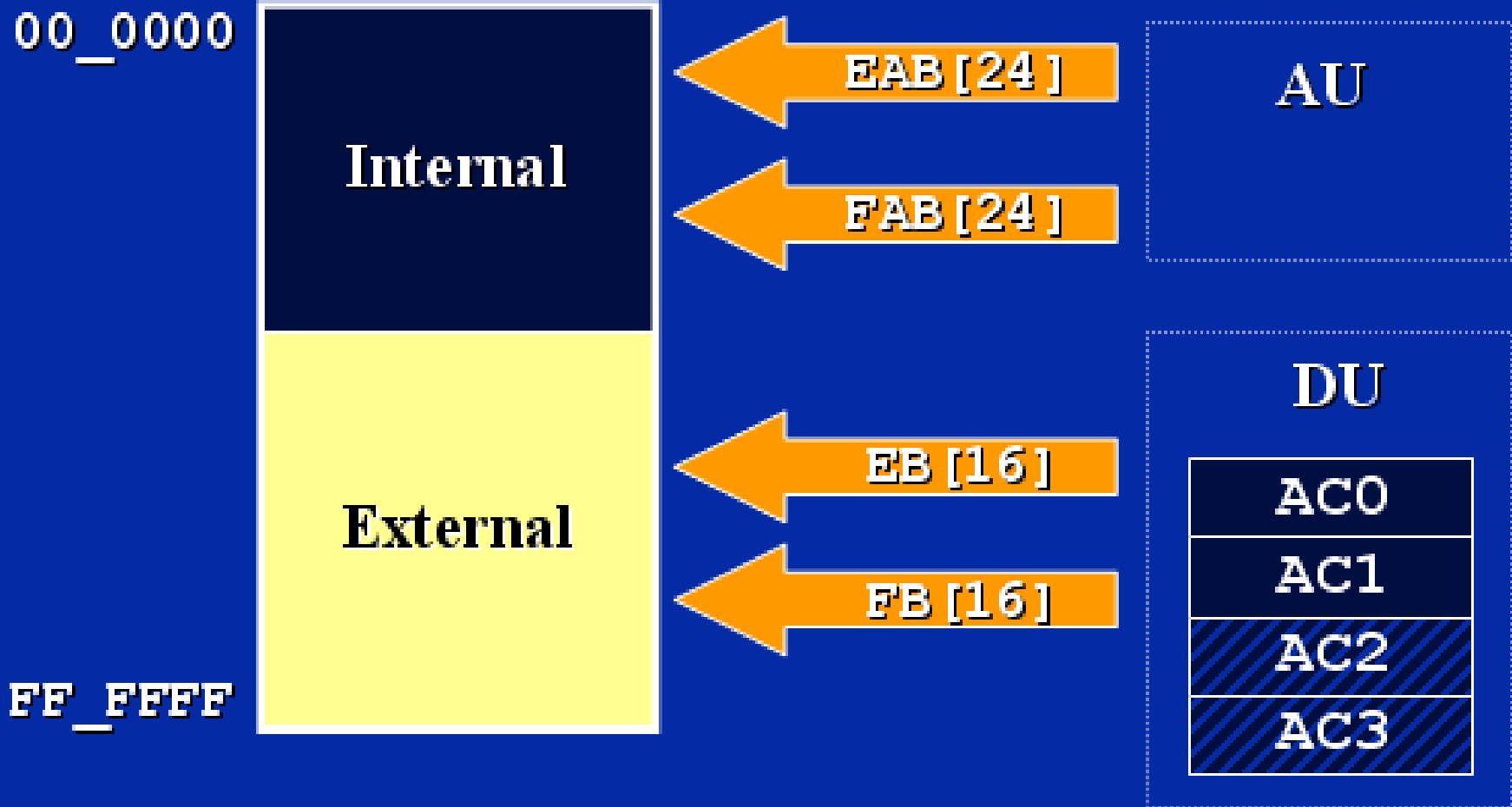
AU

- ◆ A-Unit handles all data addressing
- ◆ Xreg: 23 bits, reg: 16 bits
- ◆ Where else could the data go?...

C55x Data Computation Unit (DU)



C55x Writes (E and F buses)



◆ 32-bit write in one cycle

Pipelines of the C55x

- There are 2 independent pipelines:
 - Program fetch pipeline (3 clock cycles)
 - Program execution pipeline (7 clock cycles)

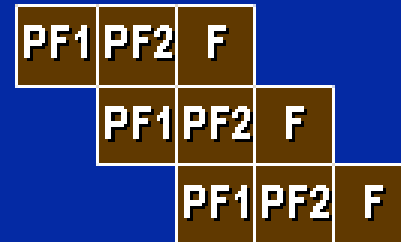
- ★ The fetch pipeline is done inside the Instruction Buffer Unit and fills IBQ
- ★ Pipeline execution breaks an operation into smaller pieces that can be executed independently
- ★ The execution pipeline fetches instructions from IBQ and executes them

C55x Fetch Packet Pipeline

PF1 - gen prog address

PF2 - memory wait

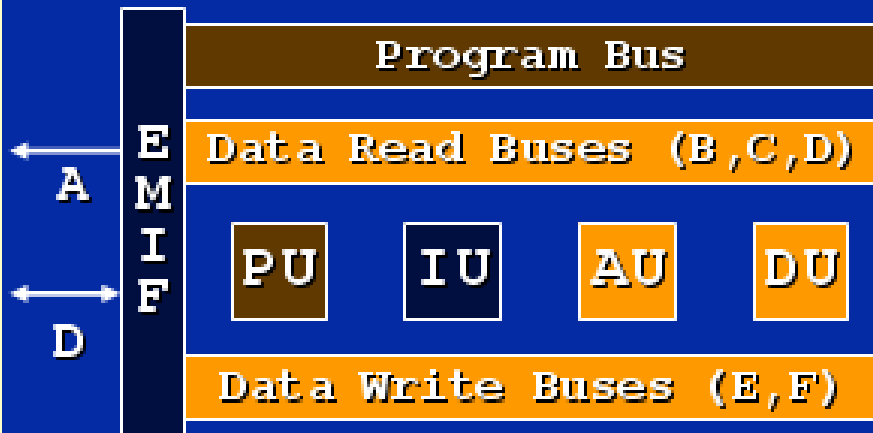
F - fetch 4-byte packet



4 bytes

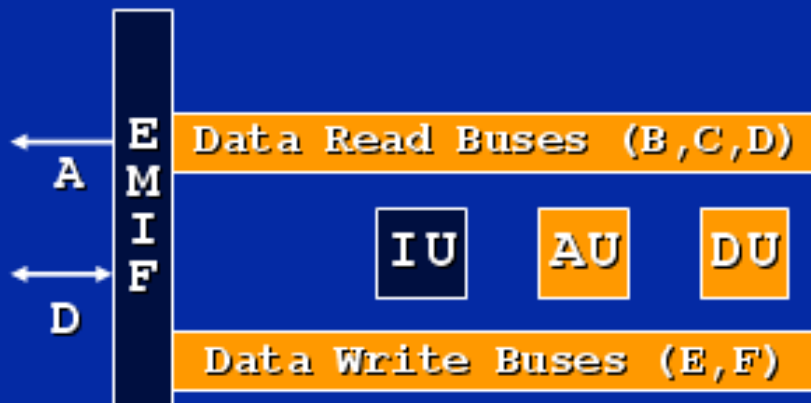
IBQ
64x8

- ◆ Fetch-packet pipeline fetches 4-byte packets from program memory INTO the IBQ every cycle (unless IBQ is full)
- ◆ Fetch packet pipeline operates independently from execute pipeline



C55x Execute Pipeline

D - decode opcode
AD - compute address
AC1 - gen read address
AC2 - memory wait
R - read operands
X - execute
W - write to memory



- ◆ *Execute pipeline* fetches instructions FROM the IBQ, then executes them
- ◆ IU performs fetch/decode from IBQ
- ◆ AU generates operand addresses
- ◆ AU/DU execute instructions
- ◆ X: result to register
W: result to memory

C5510 Unified Memory Map

Program

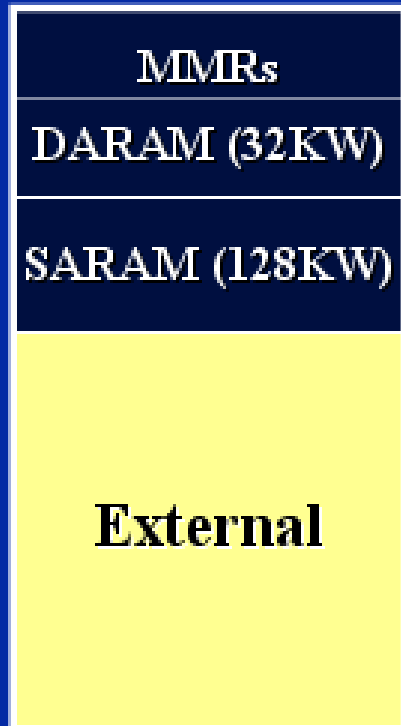
00_0000

00_00C0

01_0000

05_0000

FF_FFFF



Data

00_0000

00_0060

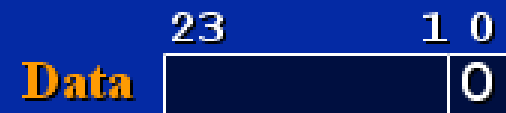
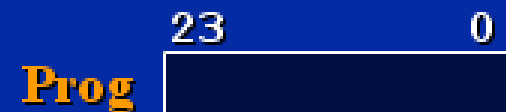
00_8000

02_8000

7F_FFFF

◆ Program and data share the same map

◆ 2 ways to view the map:



1. Program - (Bytes)

- 16M x 8-bit, linear 24-bit addresses
- Used by fetch/decode logic

2. Data (Words)

- 8M x 16-bit, segmented into 64K pages, 23-bit address
- Most code written by a user will access data

Memory Access

- 16MB of memory are addressable as program space or data space
- When the CPU uses program space to read program code from memory, it uses 24-bit addresses to reference bytes.
- When program accesses data space, it uses 23-bit addresses to reference 16-bit words.
- In both cases, the address buses carry 24-bit values, but during a data-space access, the least significant bit on the address bus is forced to 0.

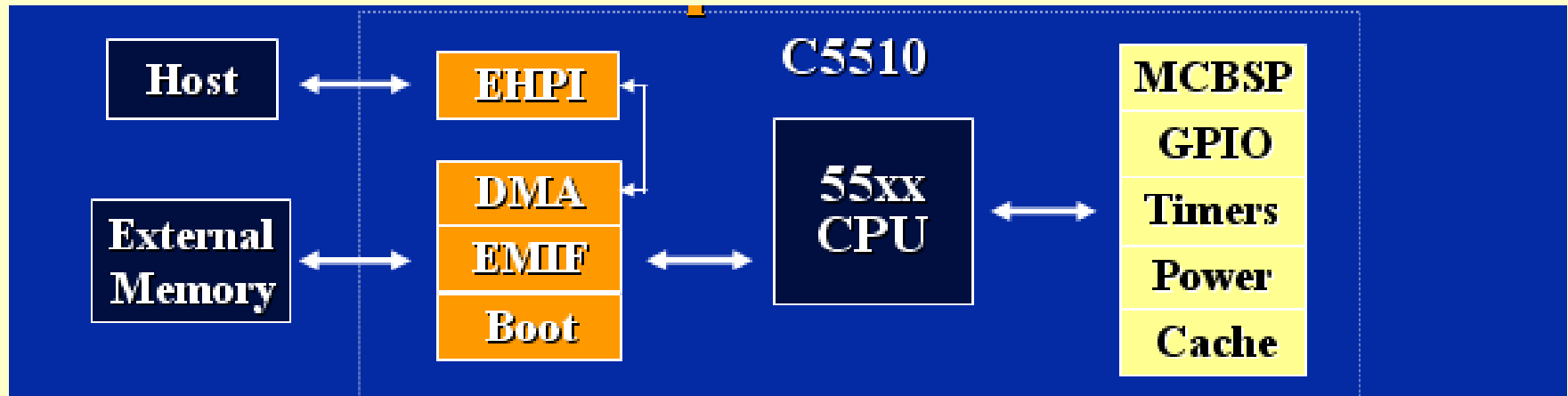
Data Memory

- Data space is divided into 128 data pages (0 through 127) of 64Kw each.
- An instruction that references a main data page concatenates a 7-bit main data page value with a 16-bit offset.
- On data page 0, the first 96 addresses (00 0000h-00 005Fh) are reserved for the memory-mapped registers (MMRs).

I/O Memory

- I/O space is separate from data/program space and is available only for accessing registers of the peripherals on the DSP. The word addresses in I/O space are 16 bits wide, enabling access to 64K locations

C5510 Peripheral Overview



EHPI

- 16-bit host access to memory

DMA

- 6 Channels (rotating priority)

EMIF

- Access to EPROM, SRAM, SBSRAM, SDRAM

BOOT Loader

- From external memory, Host, McBSP

3 Multi-Channel Buffered SPs

- High speed sync serial comm

General Purpose I/O

- 8-bit i/o port

Timer/Counters

- Two 20-bit timer/counters

Power-Down Modes

Instruction Cache (24K bytes)

CPU Registers C54x vs C55x

- The study of CPU registers gives a very good understanding on the processor architecture.
- The C54x DSP is code compatible with the C55x, therefore registers have the same functionality in both DSPs.
- Registers in the C55x are more complex, so we will see their role and give equivalents for the C54x.
- The following table summarizes the differences.

Register Files

- ◆ Following is the list of registers contained in each unit:

LOAD	STORE
REGISTER FILE	
RPTC	BRC0-1
RSA0-1	REA0-1
RETA	CSR

PU

DAGEN			LOAD	STORE	SWAP
X	Y	C	REGISTER FILE		
T0-3			XAR0-7	XSP/SSP	
Xreg			XCDP	XDP	
BKxx			BSAxx	ST0-3	
PDP					

AU

LOAD	STORE	STORE w/SHFT
REGISTER FILE		SWAP
AC0-3	TRN0-1	

DU

C55x CPU Registers and C54x Equivalents

Abbreviation	Name	Size	C54x
AC0–AC3	Accumulators 0 through 3	40 bits	A , B
AR0–AR7	Auxiliary registers 0 to 7	16 bits	same
BK03, BK47, BKC	Circular buffer size registers	16 bits	BK
BRC0, BRC1	Block-repeat counters 0 & 1	16 bits	BRC
BRS1 BRC1	Save register	16 bits	none
BSA01,BSA23, BSA45,BSA67,BSA	Circular buffer start address registers	16 bits	none
CDP	Coefficient data pointer (low part of XCDP)	16 bits	none
CDPH	High part of XCDP	7 bits	None
DP	Data page register (low XDP)	16 bits	DP(9)
DPH	High part of XDP	7 bits	none

C55x CPU Registers and C54x Equivalents

IER0, IER1	Interrupt enable registers 0& 1	16 bits	IMR
IFR0, IFR1	Interrupt flag registers 0 and 1	16 bits	IFR
IVPD, IVPH	Interrupt vector pointers	16 bits	IPTR(9)
SP	Data stack pointer	16 bits	SP
SPH	High part of XSP and XSSP	7 bits	na.
SSP	System stack pointer	16 bits	na.
ST0_55–ST3_55	Status registers 0 through 3	16 bits	ST0,ST1, PMST
T0–T3	Temporary registers 0 to 3	16 bits	T
TRN0, TRN1	Transition registers 0 and 1	16 bits	TRN
XAR0–XAR7	Extended auxiliary registers 0 through 7	23 bits	na.
XCDP	Extended coefficient data pointer	23 bits	na.
XDP	Extended data page register	23 bits	na.
XSP	Extended data stack pointer	23 bits	na.
XSSP	Extended system stack pointer	23 bits	na.

C55x CPU Registers

- **Accumulators** (AC0–AC3): four equivalent, 40-bit accumulators, used in data computation in D unit: ALU, MACs and the shifter.
- **Transition Registers** (TRN0, TRN1): - used in the compare-and-select-extremum instructions; can hold transition decisions for the path to new metrics in Viterbi algorithm implementations.
- **Temporary Registers** (T0–T3): - hold one of the memory multiplicands for multiply, MAC, MAS
- hold the shift count used in addition, subtraction.
- **Auxiliary Registers** (XAR0–XAR7 / AR0–AR7):
 - ARnH specify the 7-bit main data page to accesses data space
 - ARn can be used as:
 - a 16-bit offset to form a 23-bit address
 - a bit address (in instructions that access individual bits)
 - a general-purpose register or counter

C55x CPU Registers

- **Data Page Register** (XDP / DP):
 - DP+ DPH (extension register) → XDP extended DP
 - DPH specify the 7-bit main data page for data space access.
 - the low part specifies a 16-bit offset (local data page) that is concatenated with the main data page to form a 23-bit address.
- **Peripheral Data Page Register** (PDP) - on 9 bits, selects a 128-word page within the 64K-word I/O space.
- **Stack Pointers**: - data stack pointers - XSP / SP
 - system stack pointers XSSP / SSP
- **Interrupt Vector Pointers** (IVPD, IVPH) – point to interrupt vectors in program space
- **Interrupt Flag Registers** (IFR0, IFR1) contain flag bits for all the maskable interrupts
- **Interrupt Enable Registers** (IER0, IER1) - enable a maskable interrupt, its corresponding bit is set to 1.
- **Status Registers** (ST0_55–ST3_55) - contain control and flag bits; reflect the current status of the DSP or indicate the results of operations.

C55x Addressing Modes

- Direct
- Indirect
- Absolute
- MMR
- Immediate -Loading constants in registers (e.g.)

Format of Data and Instructions, Internal Buses for the C55x Family

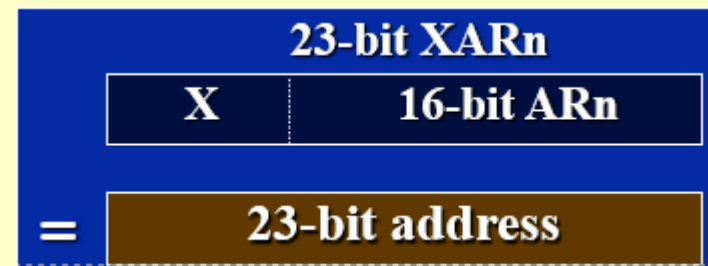
- C55x has a variable length instruction set (8-16-24-32-40-48 bits).
 - Program address bus: 24 bits, 16 Mbytes
 - 4 instructions bytes are fetched at a time
 - 6 bytes are decoded at a time
- Internal data buses: 3 data read, 2 data write
 - Data addresses: 8 Mwords of 16 bits segmented into 64K pages, **23-bit address**. A 24-bit address is automatically generated by the hardware by adding a LSB = 0.

Loading Constants in Registers

- *Immediate addressing* - used for initialization of registers.
- Addressing registers:
 - 16-bits long: ARi, DP, CDP (Coefficient Data Pointer)
 - 23-bits long: XARi, XDP, XCDP (X-Coefficient data pointer)
 - The 7 MSB of Xreg specify the 64K page.
- The ARAU auxiliary Register Arithmetic Unit is 16 bits wide: update of ARi and CDP are done modulo 64K.
- Example:
 - A in AMOV means in AD-phase.

```
x .usect "vars",4
y .usect "vars",1
.sect "init"
tbl .int 1,2,3,4
.sect "code"
```

```
indir: AMOV #x,XAR0
       AMOV #tbl,XAR6
```



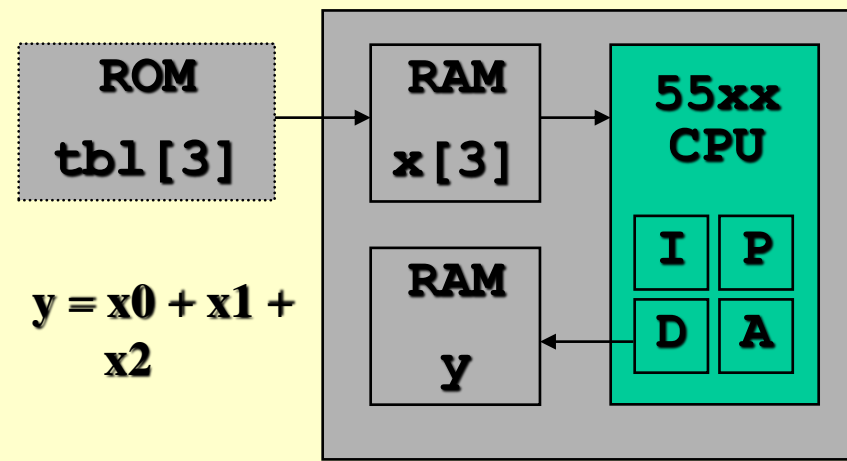
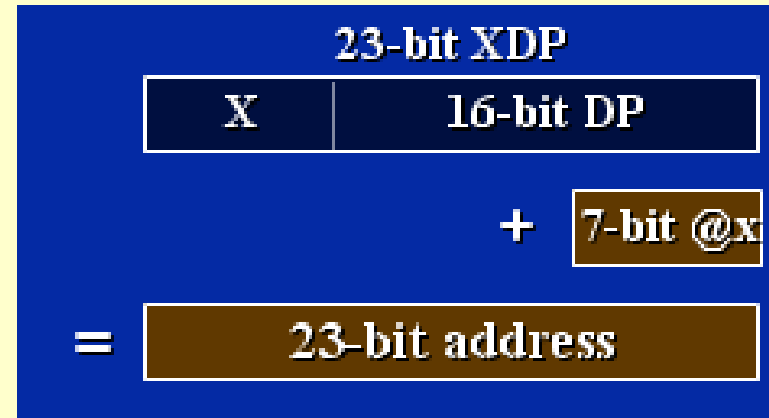
Direct Addressing Mode @

- Gives the instruction a positive 7 bit offset from DP (non-aligned).
 - In the case where the bit CPL=0 in ST55_1.
 - Calculation in the ARAU modulo 64K

Example: $y = x0 + x1 + x2$

```
x .usect "vars",4
y .usect "vars",1
.sect "init"
tbl .int 1,2,3,4
.sect "code"
```

```
ADD: MOV @ (x+0) ,AC0
      ADD @ (x+1) ,AC0
      ADD @ (x+2) ,AC0
```



Indirect Addressing Mode *ARi

- Similar to the case of the C54x, but:
 - 23-bit addresses,
 - Extended registers XARi- 23 bits specify the complete address
 - ARAU calculates on 16 bits (modulo 64K),
 - 8 ARi 16-bit pointers used in the instructions.
- Special instructions: AADD, ASUB, AMOV
 - Can be used to modify TAs registers during the address (AD) phase of the pipeline, while instructions without A operates during the execution (X) phase.
 - They only work on the TAx registers (T0-3, AR0-7)
- Example:



```
AADD #const, AR1
ASUB AR1, T0
AMOV #k23, XAR2
```


*ARn	No Modify
*ARn (T0/1)	No Modify w/offset
*ARn (#k16)	No Modify w/offset
*ARn +/-	Post Modify (+/-)
*(ARn +/- T0/1)	Post Modify (+/- by T0/1)
*+/- ARn	(+/-) Pre Modify
*+ ARn (#k16)	(+ #k16) Pre Modify
*(ARn +/- T0B)	Bit reversed using T0
*CDP	No Modify
*CDP (#k16)	No Modify w/offset
*CDP +/-	Post Modify (+/-)
*+CDP (#k16)	(+ #k16) Pre Modify

◆ Assumes $ST2_55_{ARMS}=0$ and $ST1_55_{C54CM}=0$.

◆ The reset condition is $C54CM=1$.

-Address Register Mode Select Bit ARMS

Circular Buffer Addressing Mode

Buffer Start Address =	Xeven [22:16]	BSA_{xx} [15:0]
Offset into Buffer =	+	AR_n/CDP
Calculated Address =	Xeven [22:16]	BSA_{xx} + AR_n/CDP
Buffer Length =		BK_{zz} [15:0]

Offset	Xeven	Buffer Start Address	Block size Register
AR0 AR1	XAR0[22:16]	BSA01	BK03
AR2 AR3	XAR2[22:16]	BSA01	
AR4 AR5	XAR4[22:16]	BSA01	BK03
AR6 AR7	XAR6[22:16]	BSA01	
CPD	XCDP[22:16]	BSAC	BKC

Comparison of C54x and C55x circular addressing modes

- 3 BK registers in C55X instead of 1 in C54x: allows for several simultaneous circular buffers with different size.
- In C54x, circular addressing mode is specified in indirect addressing type % in the instructions.
- In C55x, the mode is set in status register ST2_55 for each register (linear or circular).

Absolute Addressing $*(#)$

- $*(#)$ = 23 bit address
- Fast: no initialization, but long it contains the 23 bit address.

MMR Addressing using `mmap()`

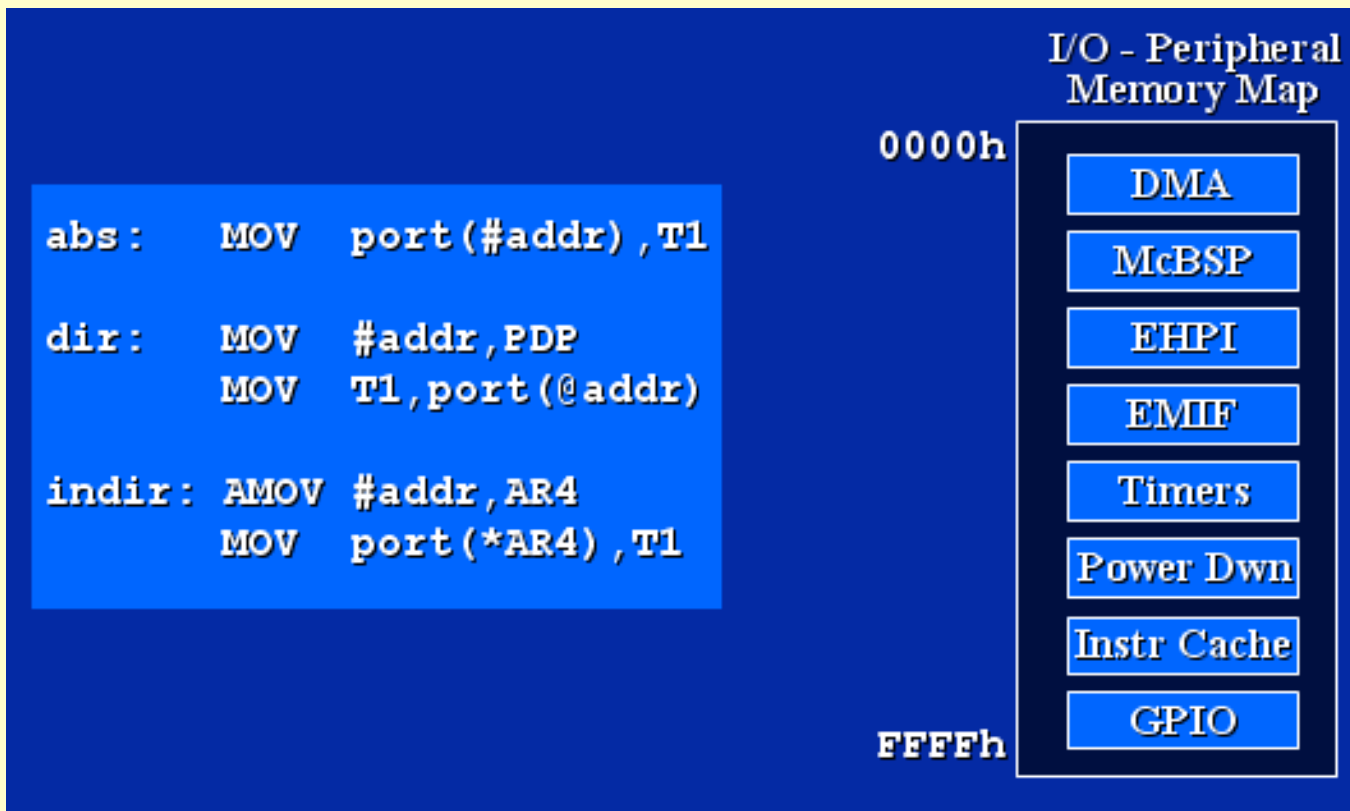
- MMRs are located between 0 and 5F.
- Scratch memory is located between 60 and 7F.

```
; write #1234h to ST0_55  
AMOV #0, XDP  
MOV #1234h, @(ST0_55 - 0)
```

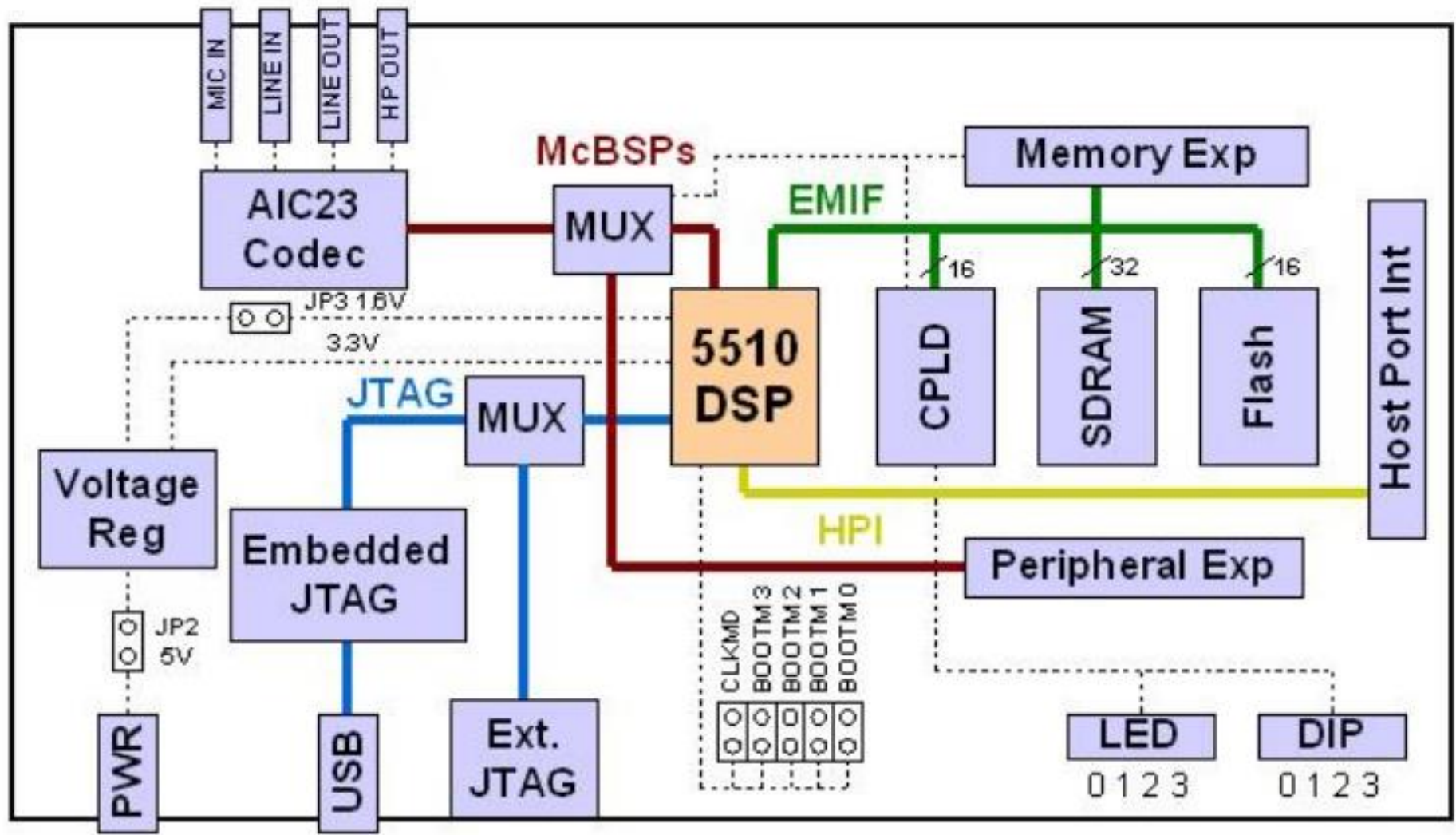
```
; write #1234h to ST0_55  
MOV #1234h, mmap(@ST0_55)
```

Access Peripheral Registers

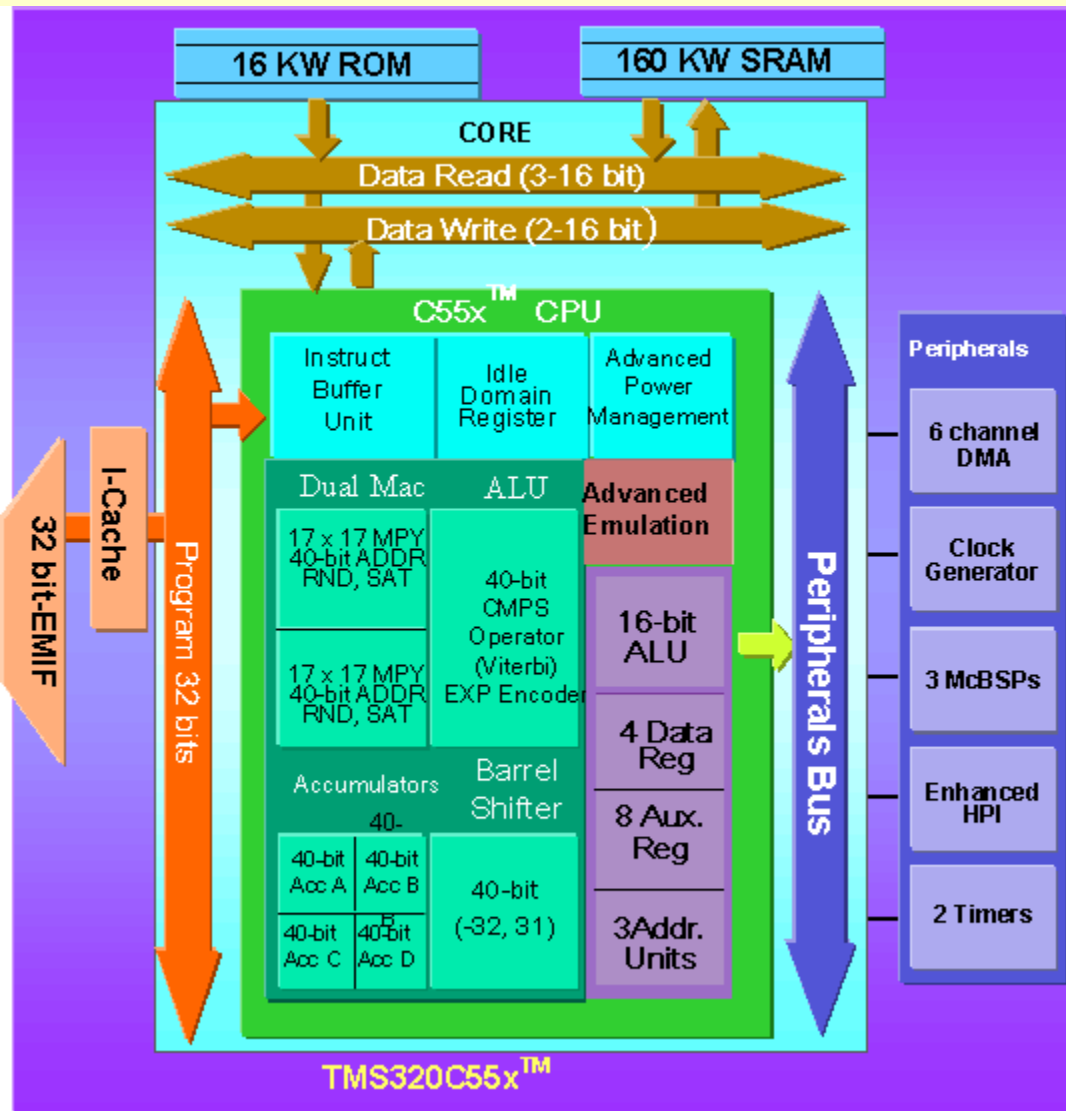
- The I/O space is internal.
- The PDP (Peripheral Data Pointer) - used to access ports using direct addressing; It is a 9-bit register concatenated with the 7 bits in the instruction to obtain a full 16-bit peripheral address.
- The port() modifier selects the peripheral map



DSK 5510 PCB



'C5510: The high runner 'C55x DSP



VC5510 Device Specific Information

On-Chip Memory	160 KW SRAM 16 KW ROM
I-Cache	24 KByte Cache
DMA	Six-channel
3 Multi-channel Buffered Serial Ports	128 Channels Each
Enhanced HPI	16-bit
External Memory Interface (EMIF)	Supports Cheaper & Faster Memories
Advanced Emulation	Easier Debug
8 GP I/O Lines	Connect Peripherals directly to the DSP
Package	240 Ball, 15x15u*BGA

