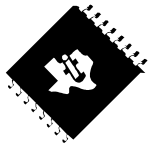


Lecture 10a: Digital Signal Processors: A TI Architectural History

Collated by: Professor Kurt Keutzer

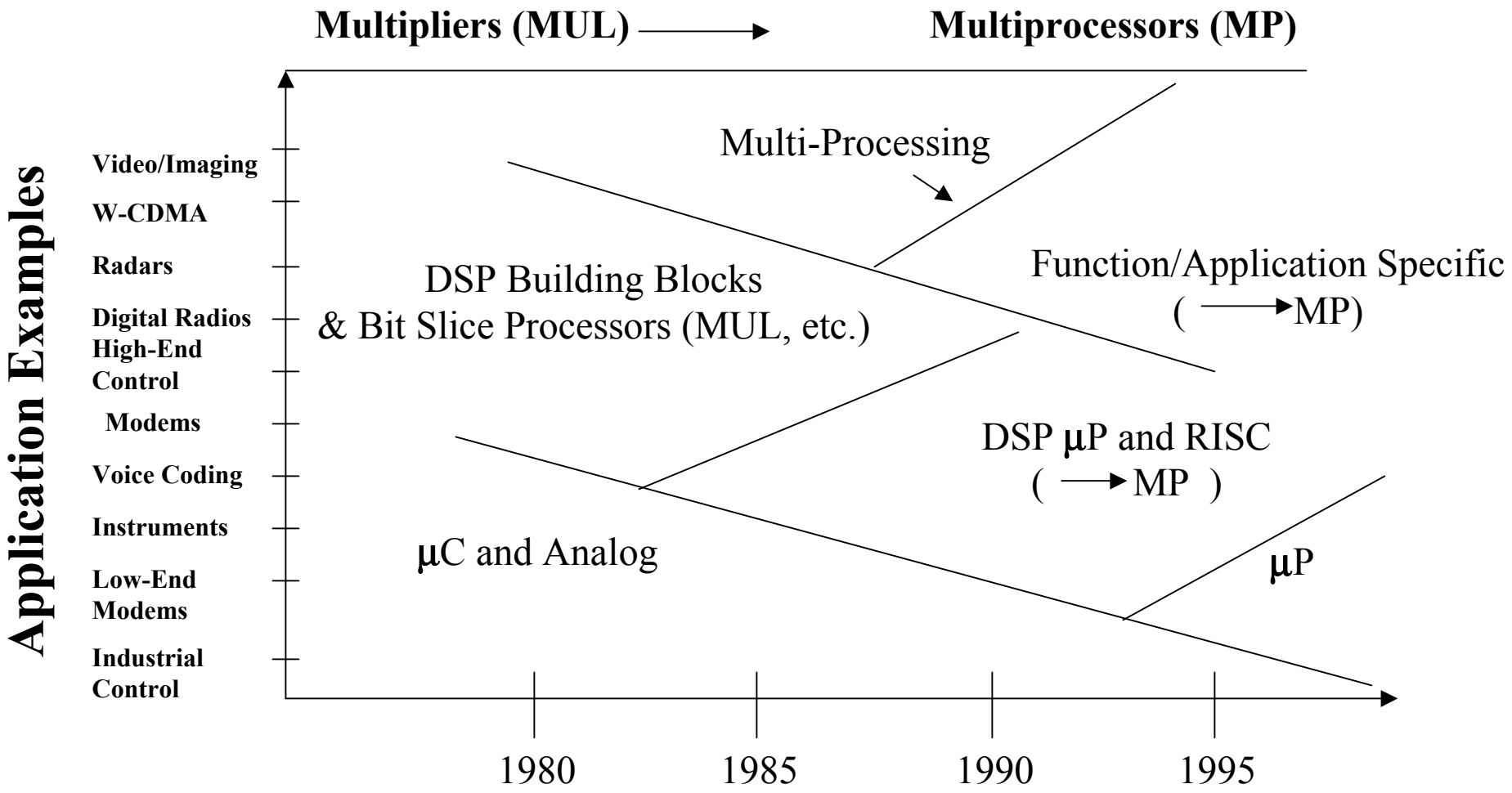
Computer Science 252, Spring 2000

With contributions from:

Dr. Brock Barton, Clark Hise TI;

Dr. Surendar S. Magar, Berkeley Concept Research Corporation

DSP ARCHITECTURE EVOLUTION



DSP ARCHITECTURE
Enabling Technologies

Time Frame	Approach	Primary Application	Enabling Technologies
Early 1970's	• Discrete logic	• Non-real time procesing • Simulation	• Bipolar SSI, MSI • FFT algorithm
Late 1970's	• Building block	• Military radars • Digital Comm.	• Single chip bipolar multiplier • Flash A/D
Early 1980's	• Single Chip DSP µP	• Telecom • Control	• µP architectures • NMOS/CMOS
Late 1980's	• Function/Application specific chips	• Computers • Communication	• Vector processing • Parallel processing
Early 1990's	• Multiprocessing	• Video/Image Processing	• Advanced multiprocessing • VLIW, MIMD, etc.
Late 1990's	• Single-chip multiprocessing	• Wireless telephony • Internet related	• Low power single-chip DSP • Multiprocessing

Texas Instruments TMS320 Family
Multiple DSP µP Generations

	First Sample	Bit Size	Clock speed (MHz)	Instruction Throughput	MAC execution (ns)	MOPS	Device density (# of transistors)
Uniprocessor Based (Harvard Architecture)							
TMS32010	1982	16 integer	20	5 MIPS	400	5	58,000 (3µ)
TMS320C25	1985	16 integer	40	10 MIPS	100	20	160,000 (2µ)
TMS320C30	1988	32 flt.pt.	33	17 MIPS	60	33	695,000 (1µ)
TMS320C50	1991	16 integer	57	29 MIPS	35	60	1,000,000 (0.5µ)
TMS320C2XXX	1995	16 integer		40 MIPS	25	80	
Multiprocessor Based							
TMS320C80	1996	32 integer/flt.				2 GOPS 120 MFLOP	MIMD
TMS320C62XX	1997	16 integer		1600 MIPS	5	20 GOPS	VLIW
TMS310C67XX	1997	32 flt. pt.			5	1 GFLOP	VLIW

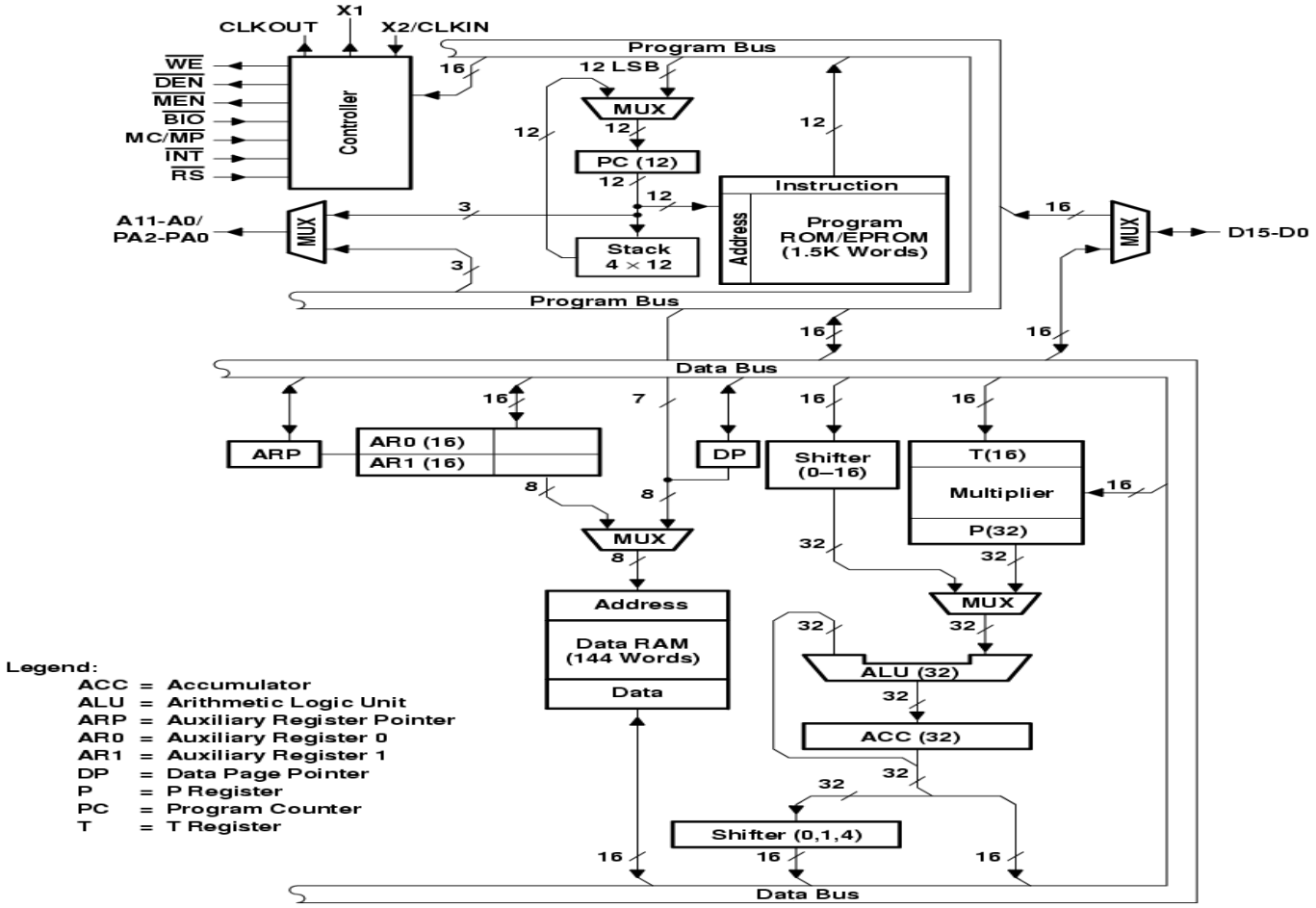
First Generation DSP µP Case Study

TMS32010 (Texas Instruments) - 1982

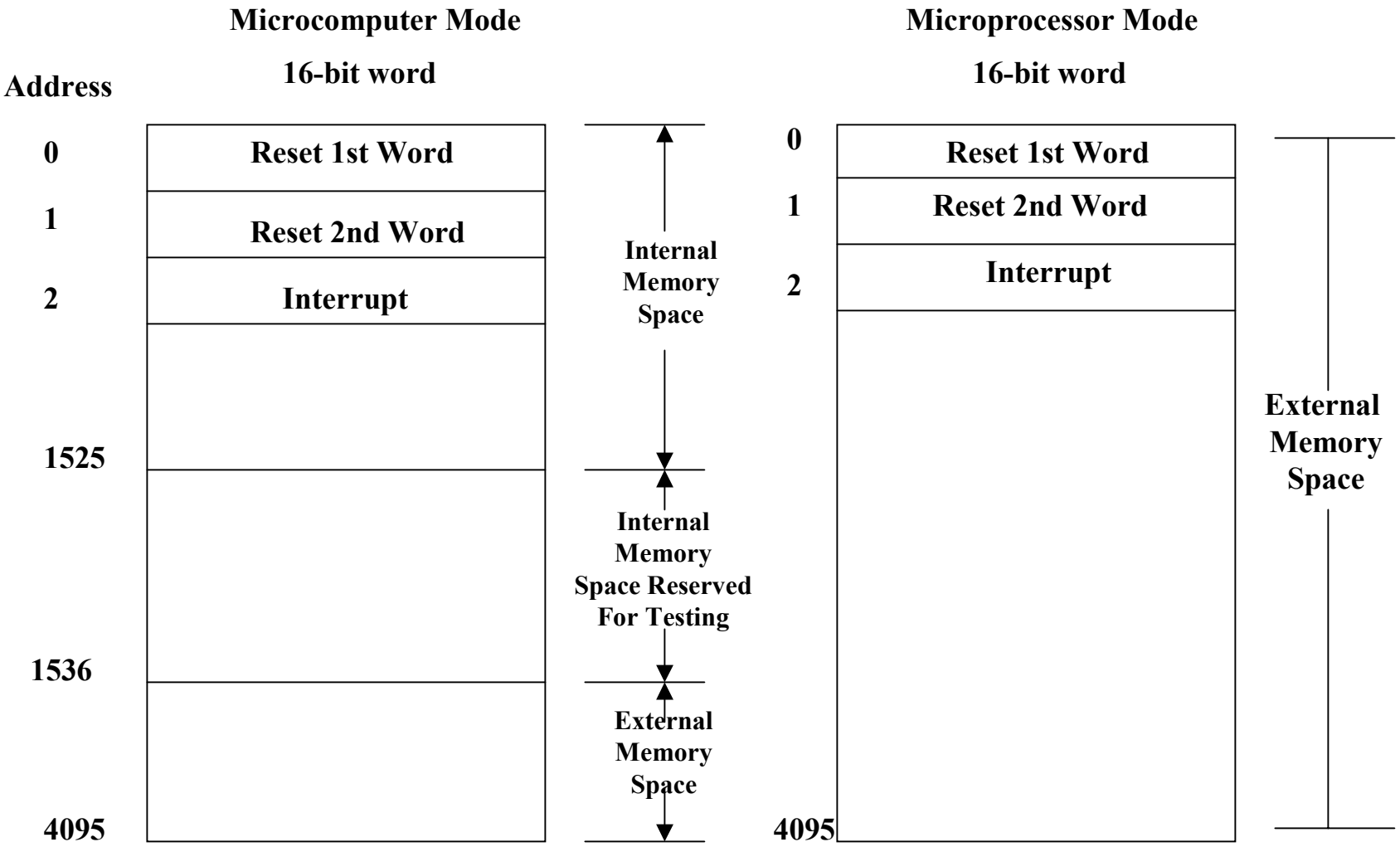
Features

- 200 ns instruction cycle (5 MIPS)
- 144 words (16 bit) on-chip data RAM
- 1.5K words (16 bit) on-chip program ROM - TMS32010
- External program memory expansion to a total of 4K words at full speed
- 16-bit instruction/data word
- single cycle 32-bit ALU/accumulator
- Single cycle 16 x 16-bit multiply in 200 ns
- Two cycle MAC (5 MOPS)
- Zero to 15-bit barrel shifter
- Eight input and eight output channels

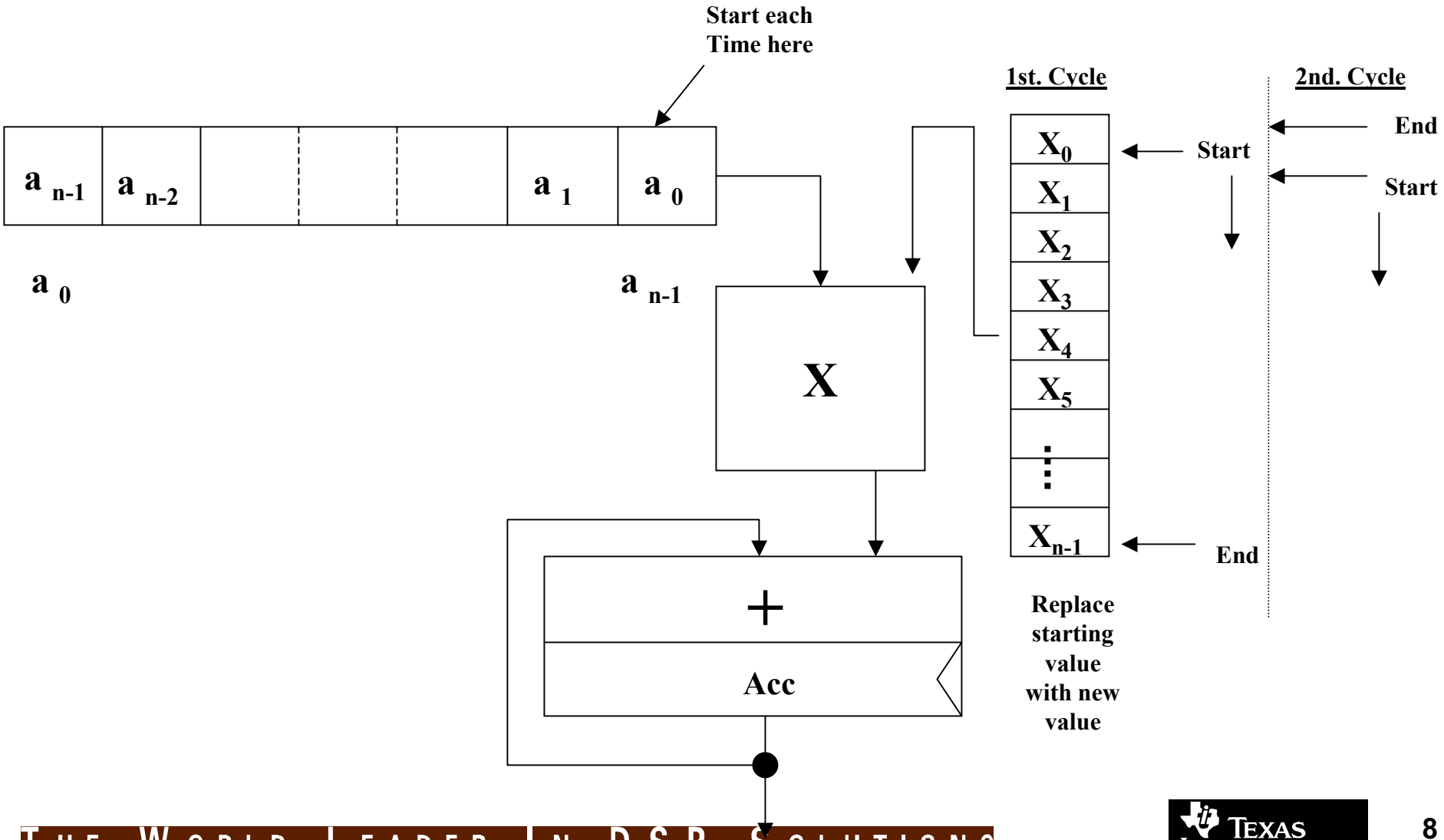
TMS32010 BLOCK DIAGRAM



TMS32010 Program Memory Maps



Digital FIR Filter Implementation (Uniprocessor-Circular Buffer)



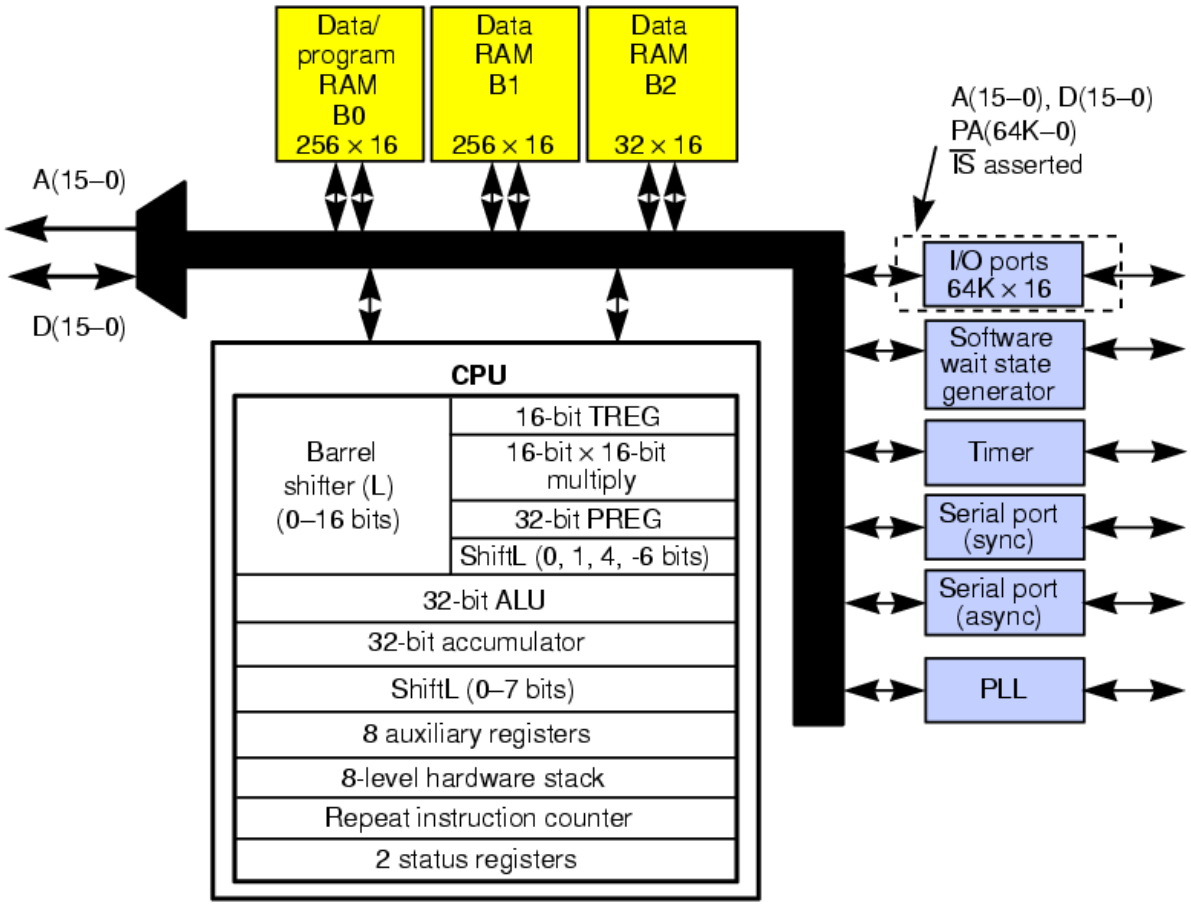
TMS32010 FIR FILTER PROGRAM
Indirect Addressing (Smaller Program Space)

$$Y(n) = x[n-(N-1)] \cdot h(N-1) + x[n-(N-2)] \cdot h(N-2) + \ldots + x(n) \cdot h(0)$$

```
* THIS SECTION OF CODE IMPLEMENTS THE EQUTION:                    *
* x(n-(N-1))h(N-1) + x(n-(N-2))h(N-2) + ... + x(n)h(0) = y(n)  *
*
        LARP AR0          * AUXILIARY REGISTER POINTER SET TO AR0 *
*
NXTPT   IN XN,PA2         * PULL IN NEW INPUT FROM PORT PA0 *
*
        LARK AR0,XNMNM1   * AR0 POINTS TO X(n-(N-1)) *
        LARK AR1,HNM1     * AR1 POINTS TO H(N-1) *
*
        ZAC               * ZERO THE ACCUMULATOR *
*
        LT *-,AR1         * x(n-(N-1))h(N-1) *
        MPY *-,AR0
*
LOOP    LTD *,AR1         * x(n-(N-1))h(N-1)+x(n-(N-2))h(N-2)+...+x(n)h(0)=y(n)*
        MPY *-,AR0
*
        BANZ LOOP         * IF AR0 DOES NOT EQUAL ZERO,                 *
*                         * THEN DECREMENT AR0 AND BRANCH TO LOOP *
*
        APAC              * ADD THE P REGISTER TO THE ACCUMULATOR *
*
        SACH YN,1         * STORE THE RESULT IN YN *
*
        OUT YN,PA2        * OUTPUT THE RESPONSE TO PORT PA1 *
*
        B NXTPT           * GO GET THE NEXT INPUT POINT *
```

For N=50, Indirect Addressing t=42 µs (23.8 KHz)
For N=50, Direct Addressing t=21.6 µs (40.2 KHz)

TMS320C203/LC203 BLOCK DIAGRAM
DSP Core Approach - 1995



Third Generation DSP µP Case Study
TMS320C30 - 1988

TMS320C30 Key Features

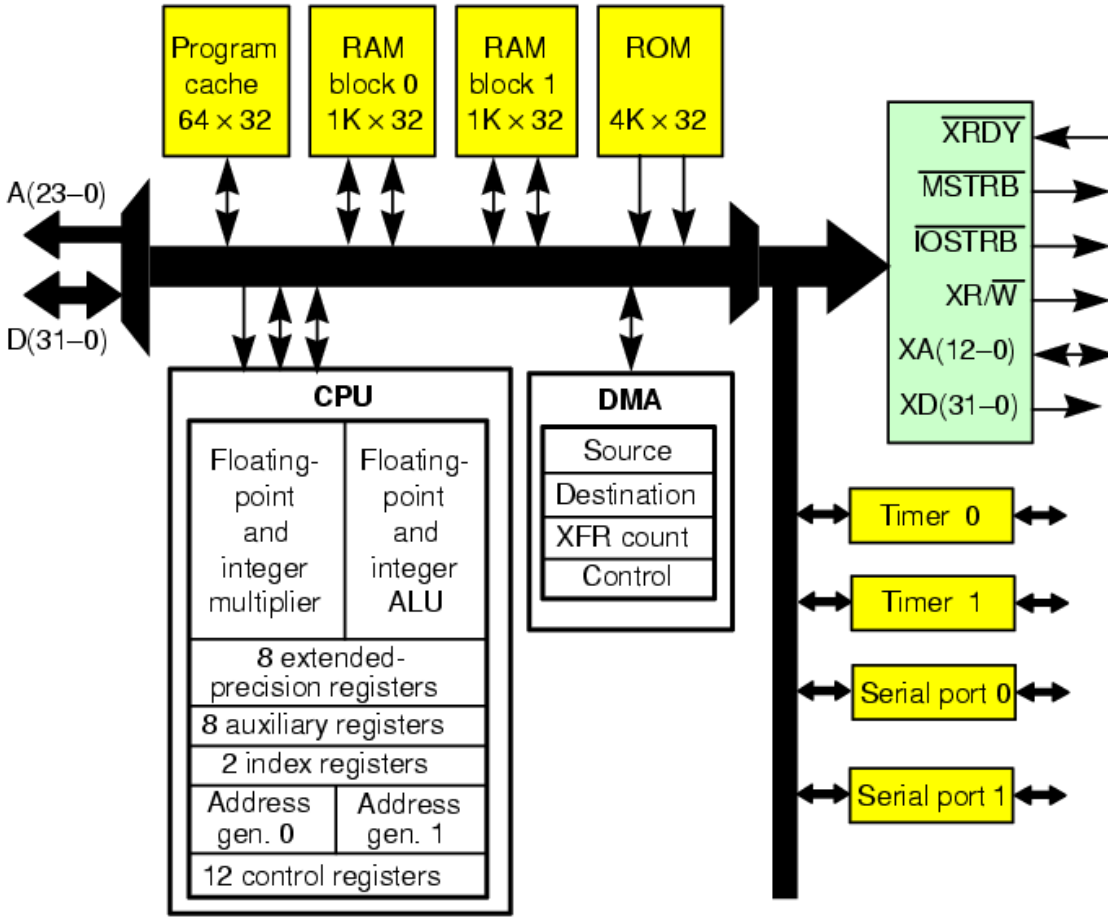
- 60 ns single-cycle instruction execution time
  - 33.3 MFLOPS (million floating-point operations per second)
  - 16.7 MIPS (million instructions per second)
- One 4K x 32-bit single-cycle dual-access on-chip ROM block
- Two 1K x 32-bit single-cycle dual-access on-chip RAM blocks
- 64 x 32-bit instruction cache
- 32-bit instruction and data words, 24-bit addresses
- 40/32-bit floating-point/integer multiplier and ALU
- 32-bit barrel shifter

Third Generation DSP µP Case Study
TMS320C30 - 1988

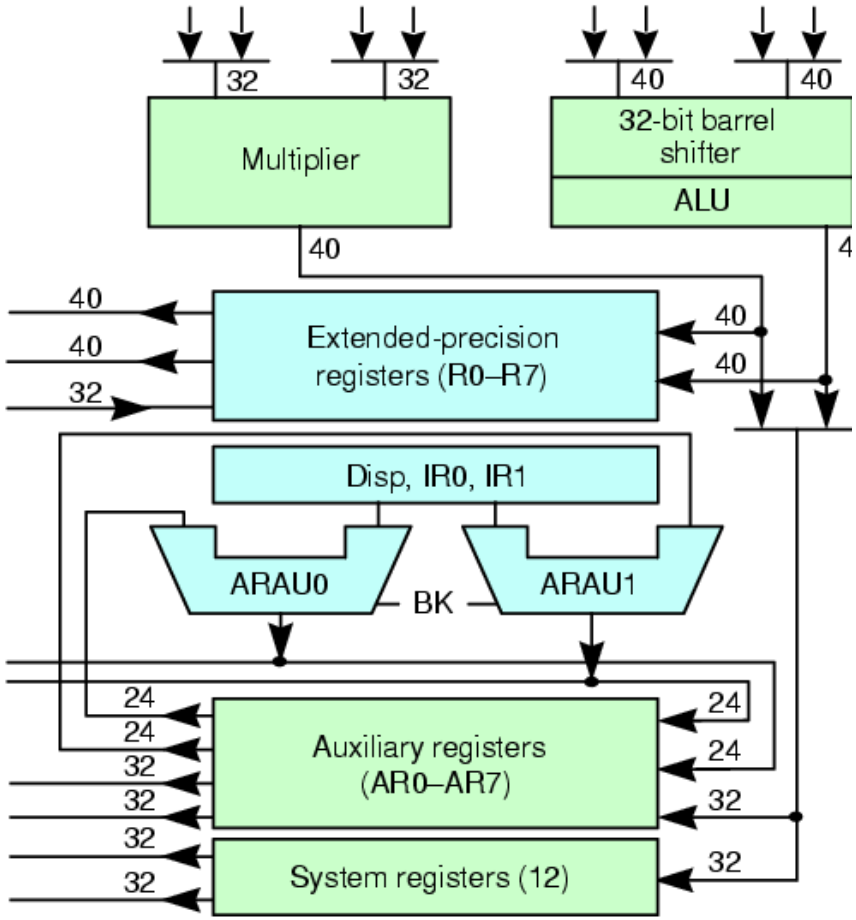
TMS320C30 Key Features *(cont.)*

- Eight extended precision registers (accumulators)
- Two address generators with eight auxiliary registers and two auxiliary register arithmetic units
- On-chip direct memory Access (DMA) controller for concurrent I/O and CPU operation
- Parallel ALU and multiplier instructions
- Block repeat capability
- Interlocked instructions for multiprocessing support
- Two serial ports to support 8/16/32-bit transfers
- Two 32-bit timers
- 1 µ CDMOS Process

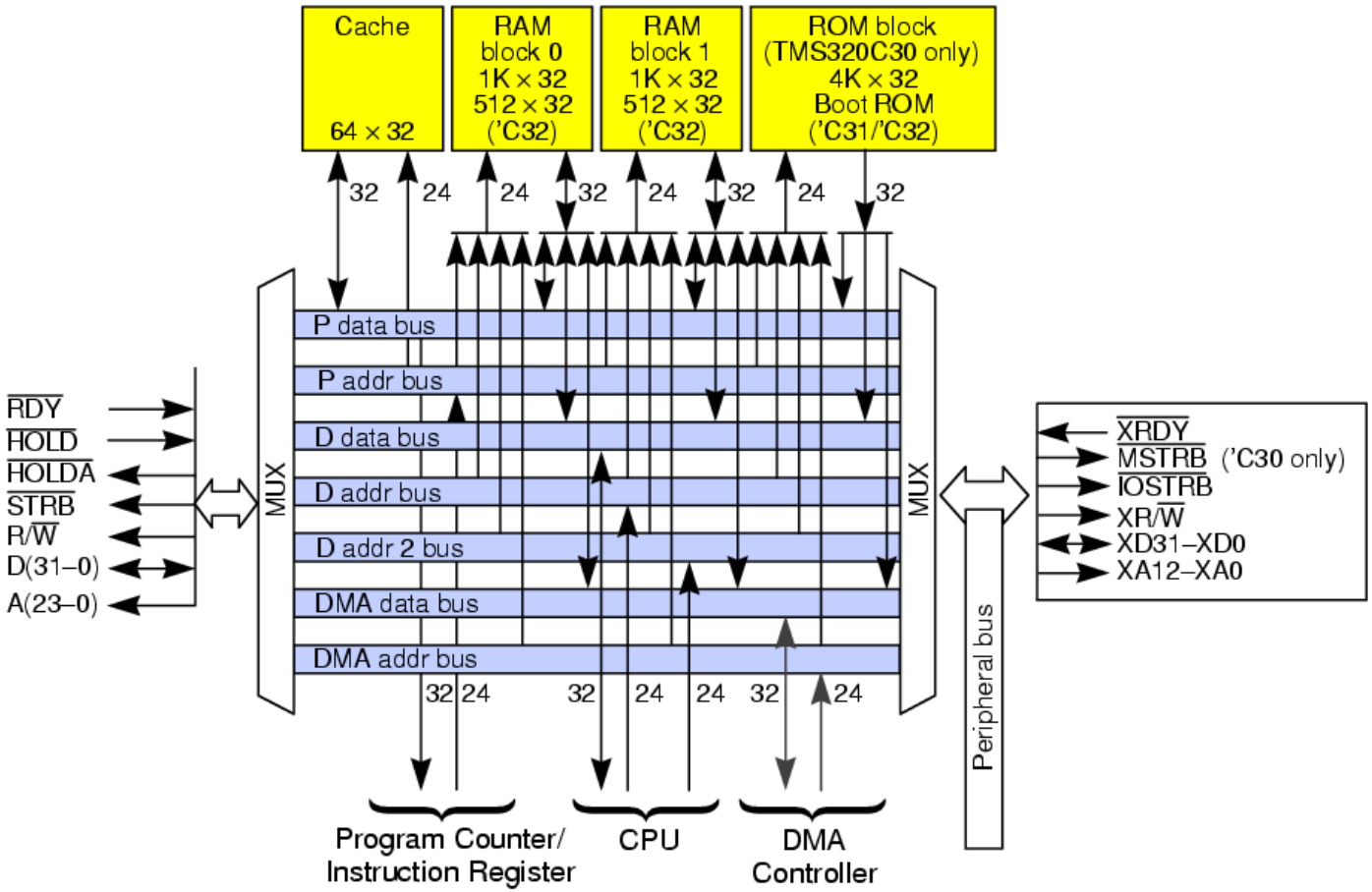
TMS320C30 BLOCK DIAGRAM



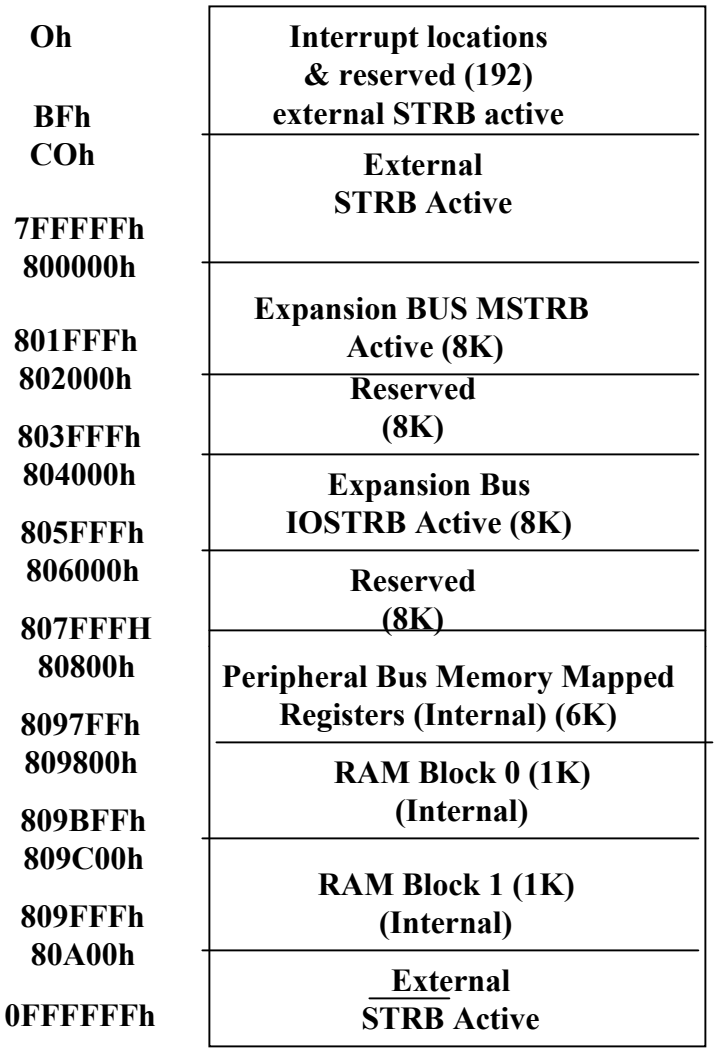
TMS320C3x CPU BLOCK DIAGRAM



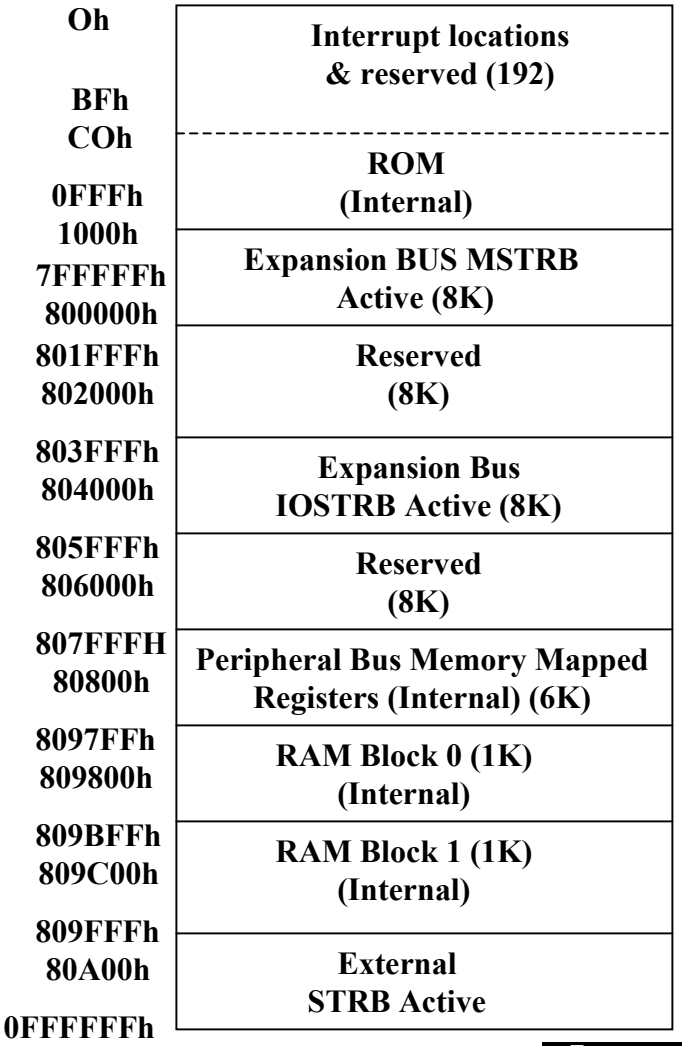
TMS320C3x MEMORY BLOCK DIAGRAM



TMS320C30 Memory Organization



Microprocessor Mode



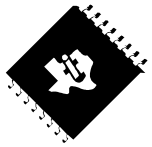
Microcomputer M

TMS320C30 FIR FILTER PROGRAM

$$Y(n) = x[n-(N-1)] \cdot h(N-1) + x[n-(N-2)] \cdot h(N-2) + \ldots + x(n) \cdot h(0)$$

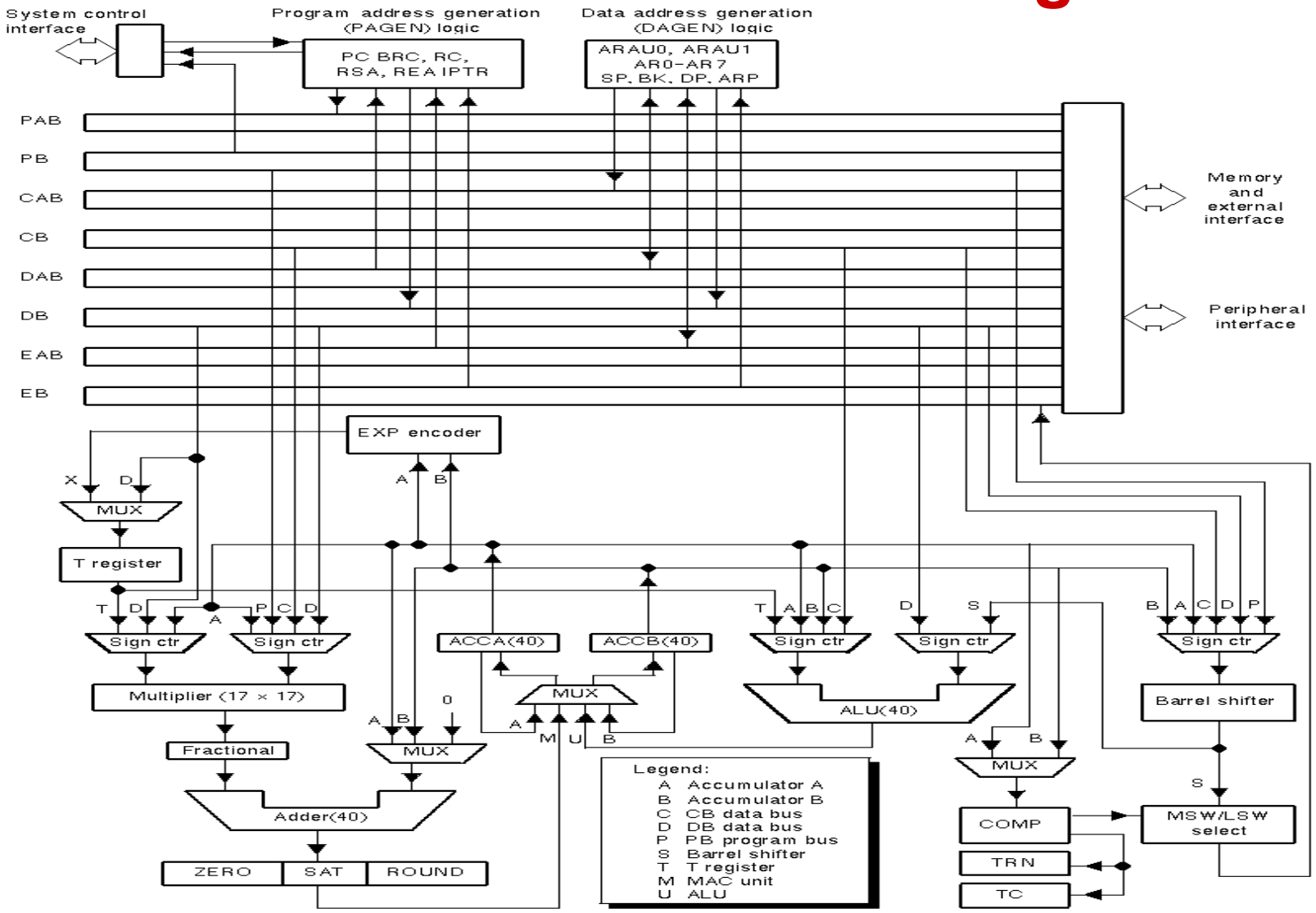
```
TOP     LDF     IN,R3               ;Read input sample.
        STF     R3,*AR1++%          ;Store with other samples,
                                    ;and point to top of buffer.
        LDF     0,R0                ;Initialize R0.
        LDF     0,R2                ;Initialize R2.
*
*       Filter
*
        RPTS    N-1                 ;Repeat next instruction.
        MPYF3   *AR0++%,*AR1++%,R0
||      ADDF3   R0,R2,R2            ;Multiply and accumulate.
        ADDF    R0,R2               ;Last product accumulated.
*
        STF     R2,Y                ;Save result.
        B       TOP                 ;Repeat.
```

For N=50, t=3.6 µs (277 KHz)



‘C54x Architecture

TMS320C54x Internal Block Diagram



Architecture optimized for DSP

#1: CPU designed for efficient DSP processing

- **MAC unit, 2 Accumulators, Additional Adder, Barrel Shifter**

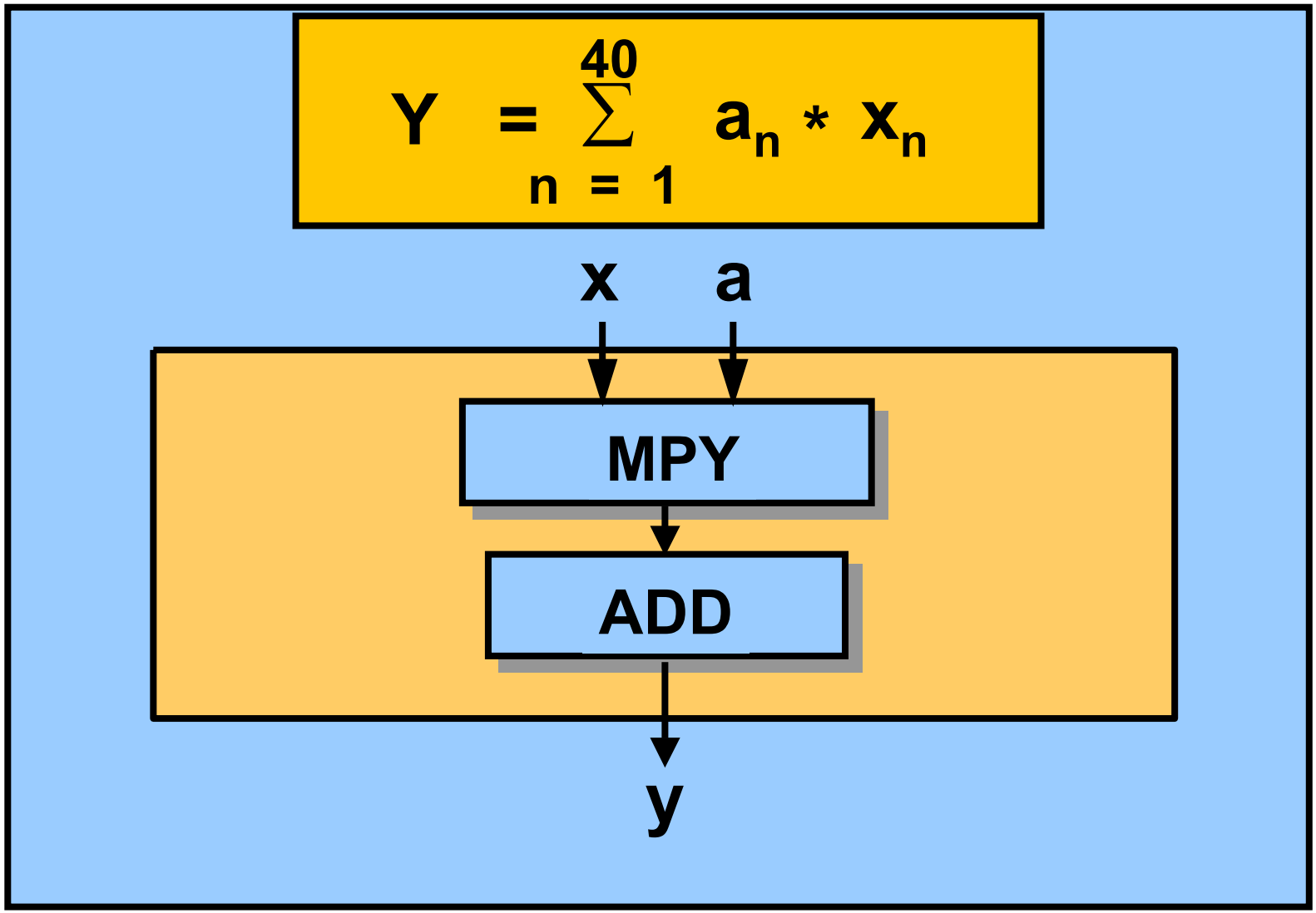
#2: Multiple busses for efficient data and program flow

- **Four busses and large on-chip memory that result in sustained performance near peak**

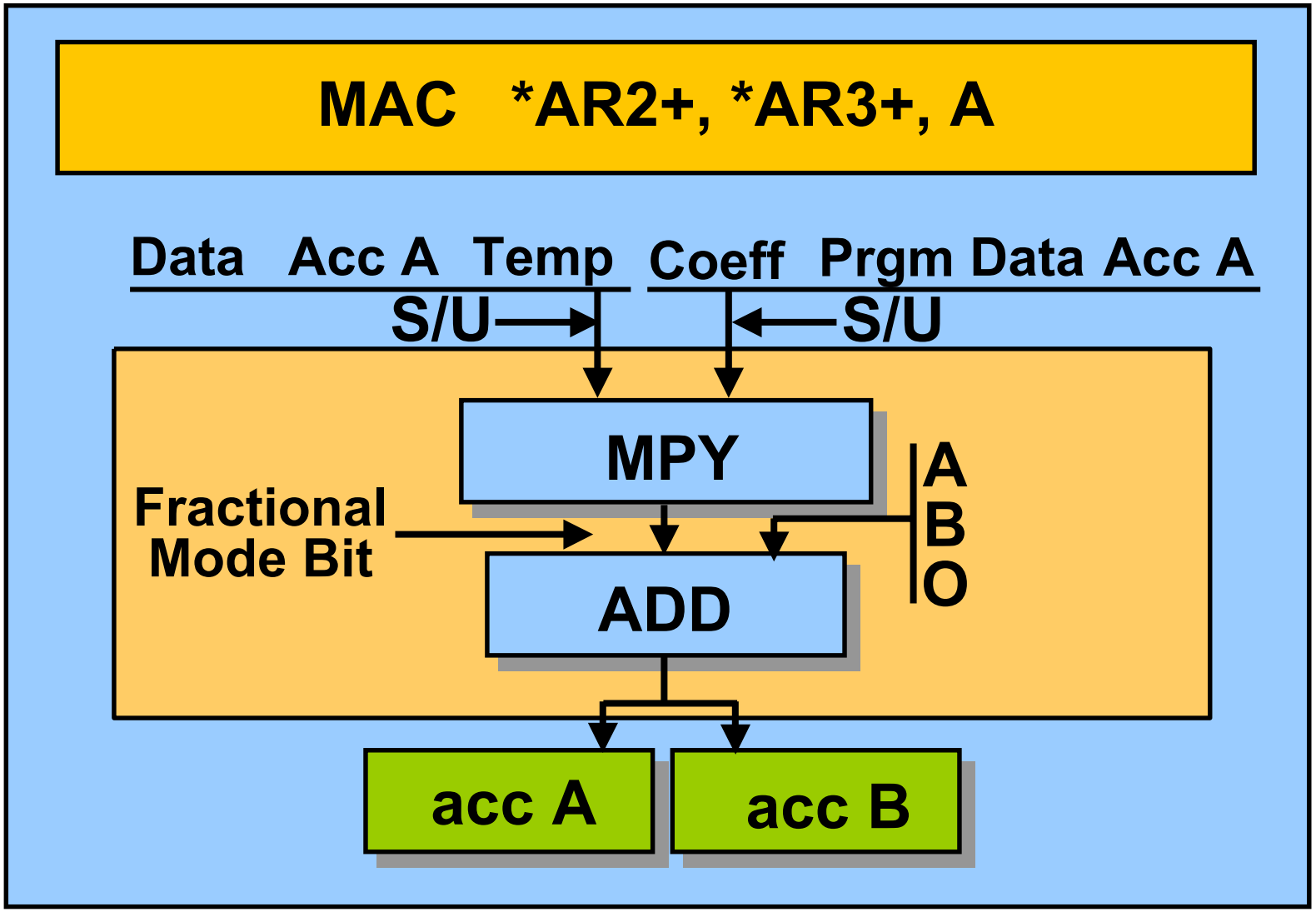
#3: Highly tuned instruction set for powerful DSP computing

- **Sophisticated instructions that execute in fewer cycles, with less code and low power demands**

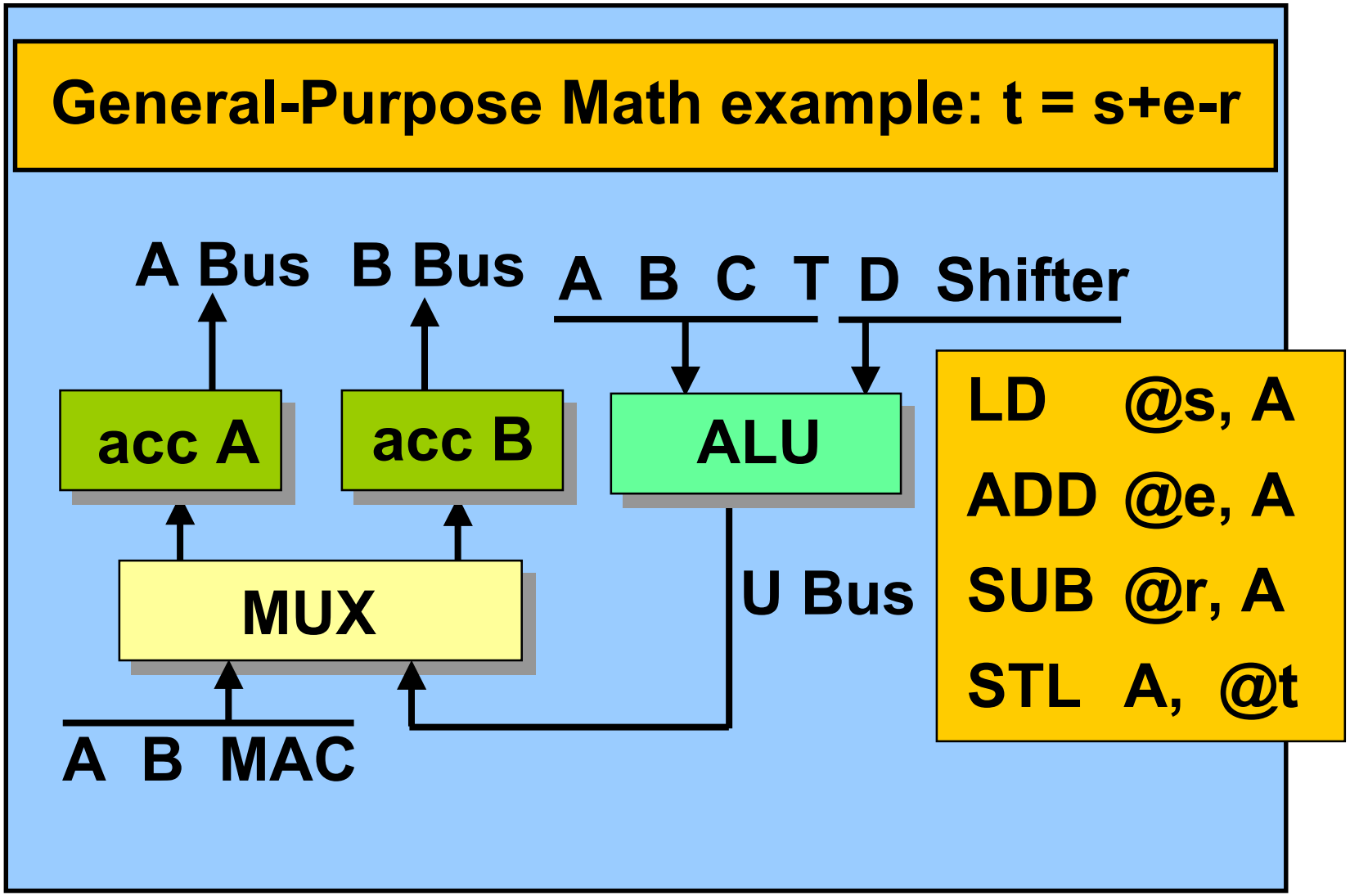
Key #1: DSP engine



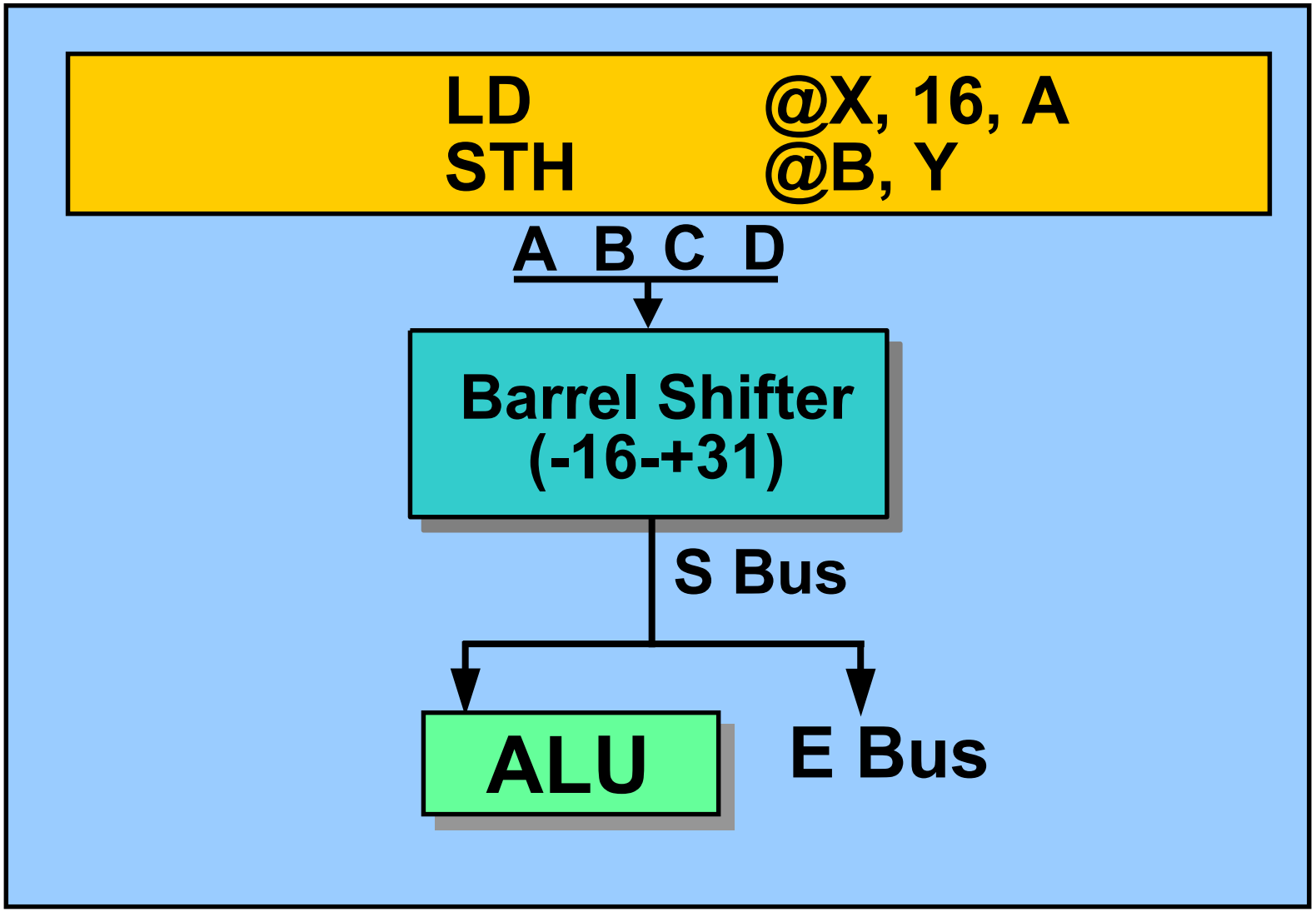
Key #1: MAC Unit



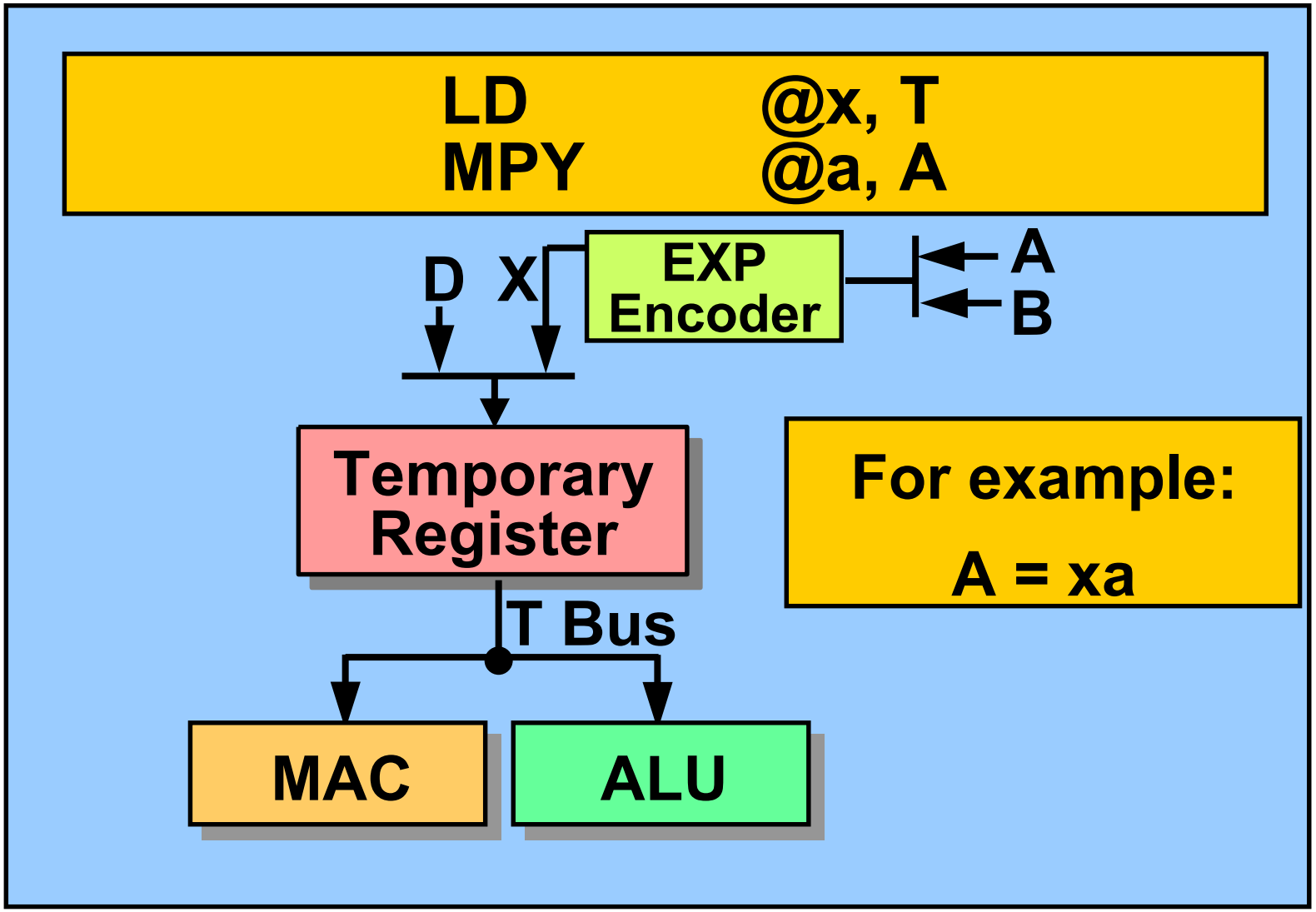
Key #1: Accumulators + Adder



Key #1: Barrel shifter



Key #1: Temporary register



Key #2: Efficient data/program flow

#1: CPU designed for efficient DSP processing

- MAC unit, 2 Accumulators, Additional Adder, Barrel Shifter

#2: Multiple busses for efficient data and program flow

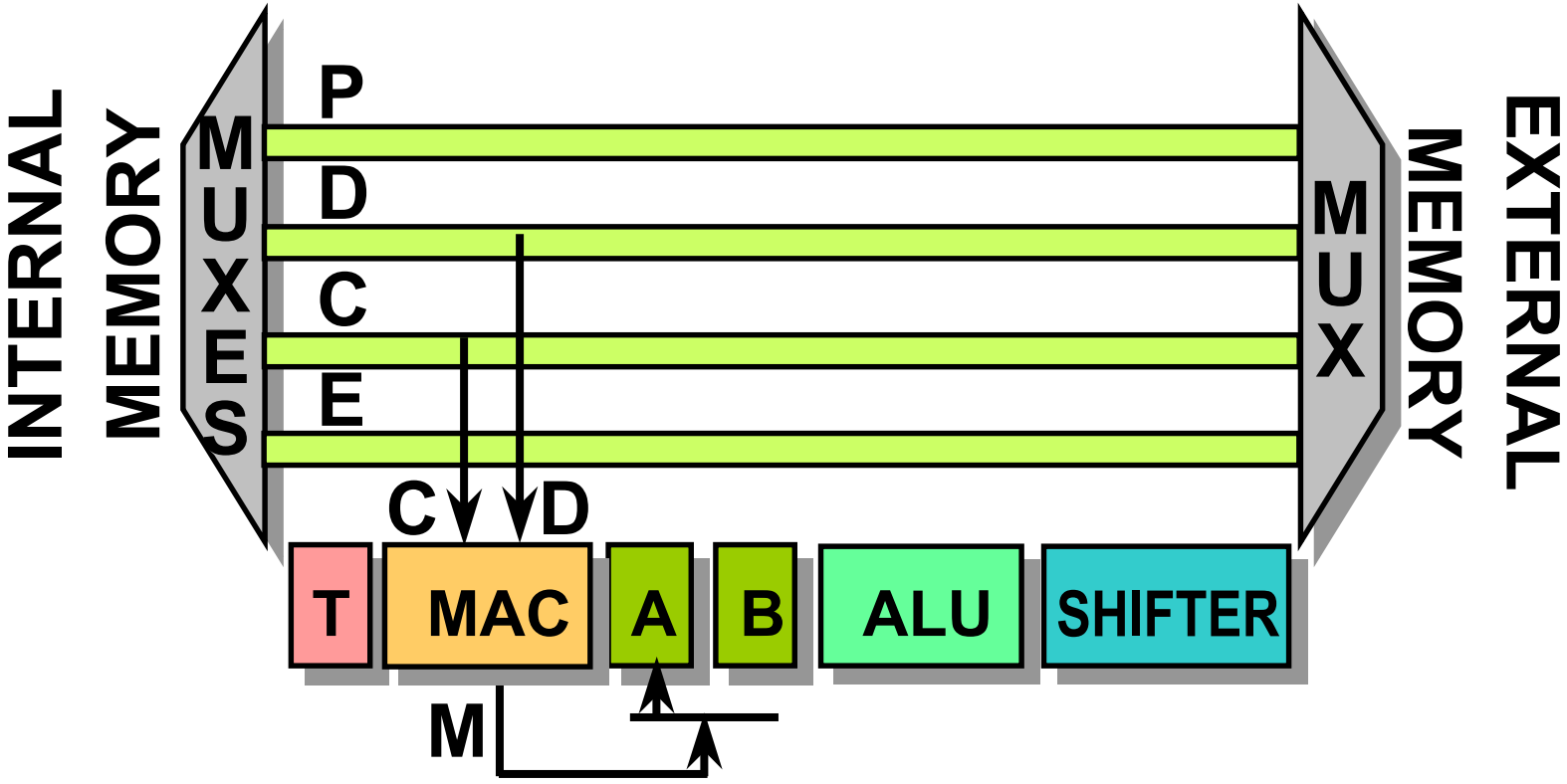
- Four busses and large on-chip memory that result in sustained performance near peak

#3: Highly tuned instruction set for powerful DSP computing

- Sophisticated instructions that execute in fewer cycles, with less code and low power demands

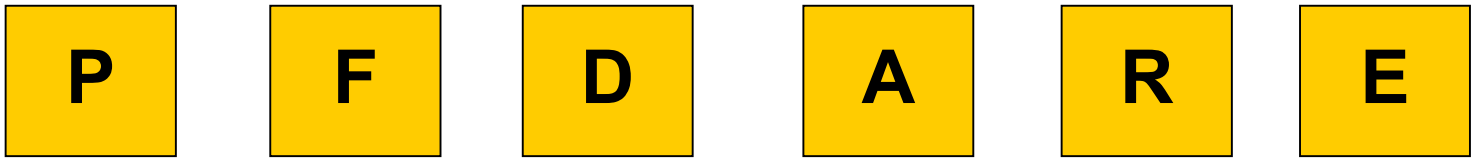
Key #2: Multiple busses

MAC *AR2+, *AR3+, A



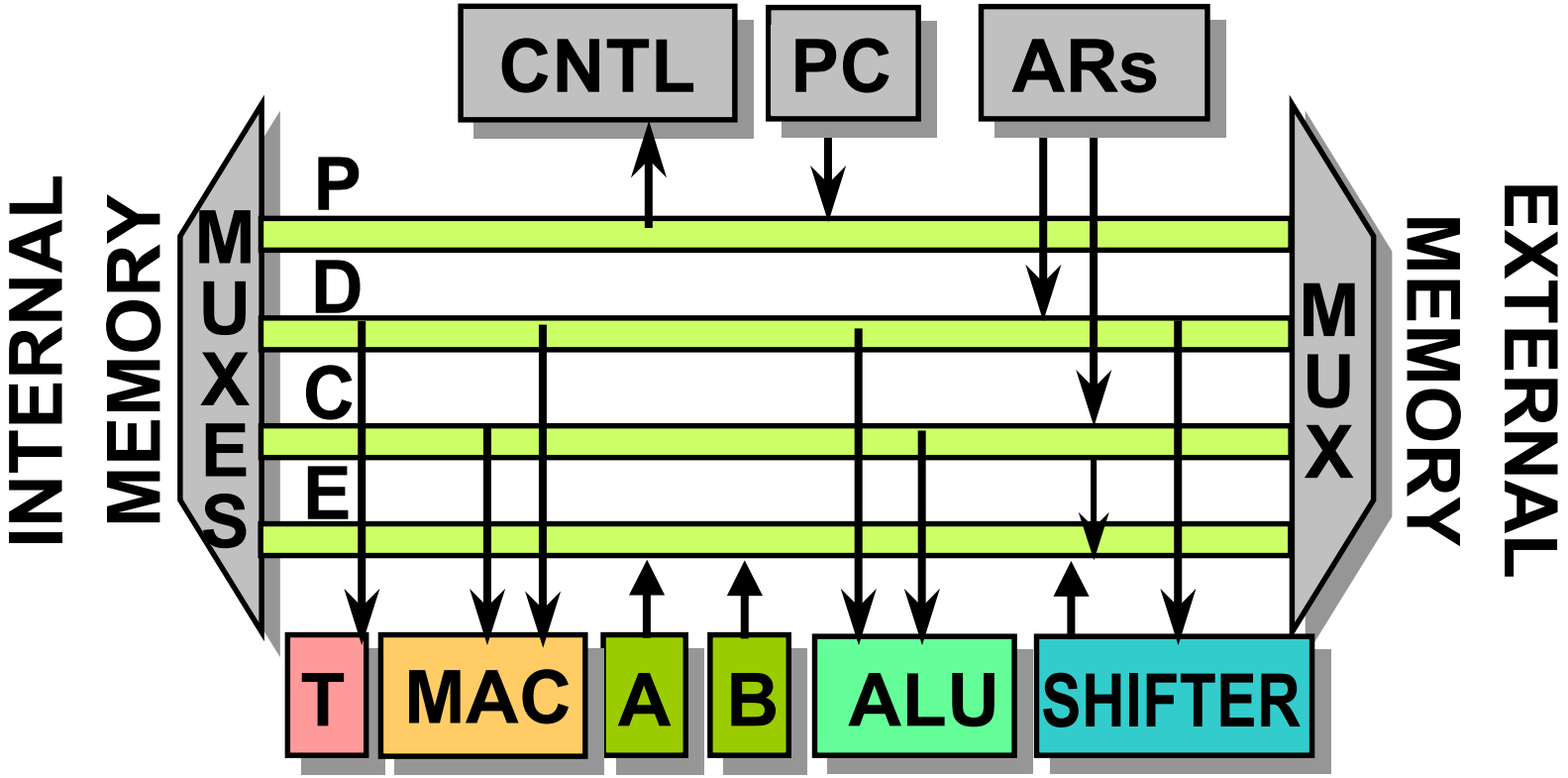
Key #2: Pipeline

Prefetch Fetch Decode Access Read Execute

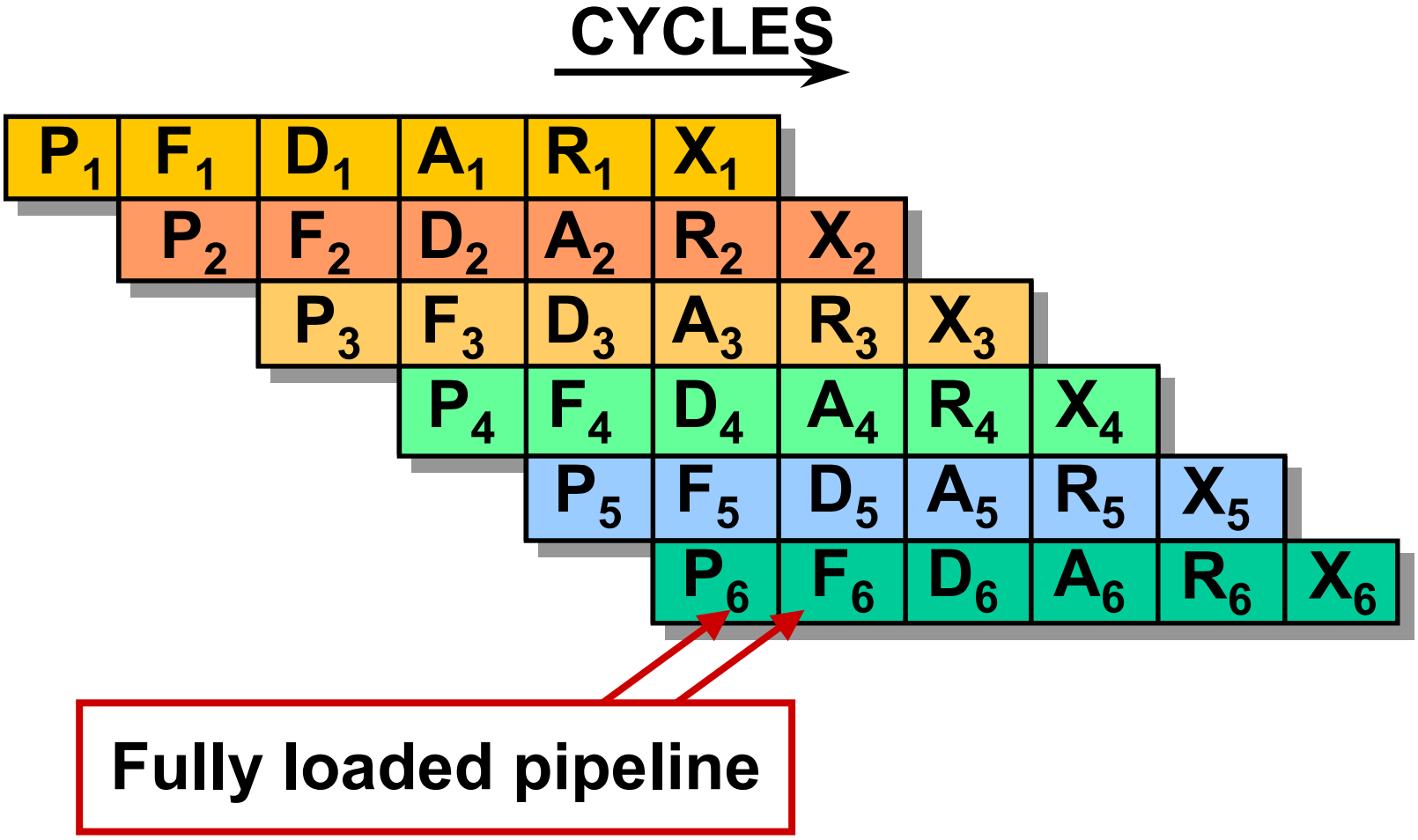


- Prefetch: Calculate address of instruction
- Fetch: Collect instruction
- Decode: Interpret instruction
- Access: Collect address of operand
- Read: Collect operand
- Execute: Perform operation

Key #2: Bus usage



Key #2: Pipeline performance



Key #3: Powerful instructions

#1: CPU designed for efficient DSP processing

- MAC Unit, 2 Accumulators, Additional Adder, Barrel Shifter

#2: Multiple busses for efficient data and program flow

- Four busses and large on-chip memory that result in sustained performance near peak

#3: Highly tuned instruction set for powerful DSP computing

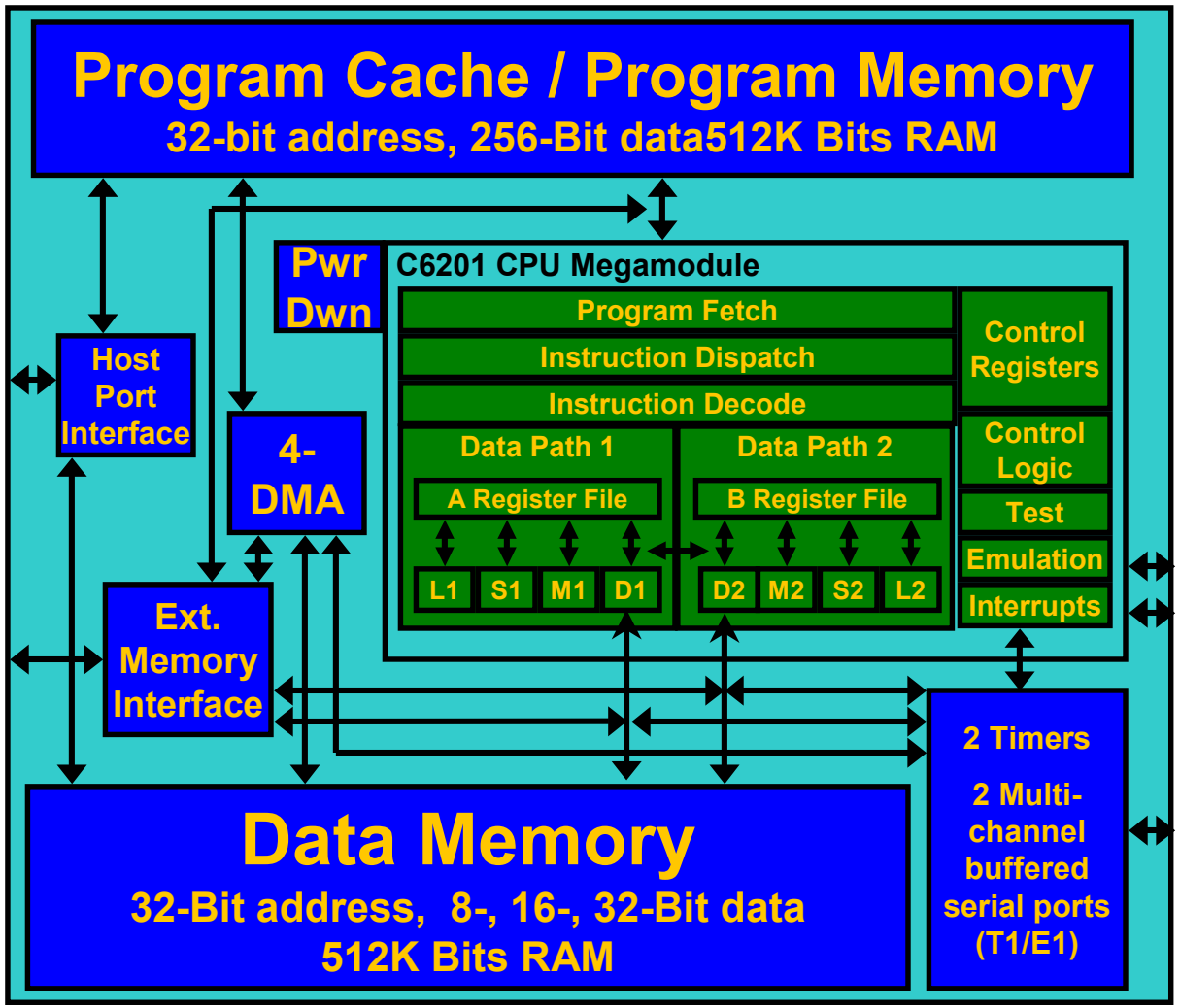
- **Sophisticated instructions that execute in fewer cycles, with less code and low power demands**

Key #3: Advanced applications

Symmetric FIR filter	**FIRS**
Adaptive filtering	**LMS**
Polynomial evaluation	**POLY**
Code book search	**STRCD**
	SACCD
	SRCCD
Viterbi	**DADST**
	DSADT
	CMPS

C62x Architecture

TMS320C6201 Revision 2



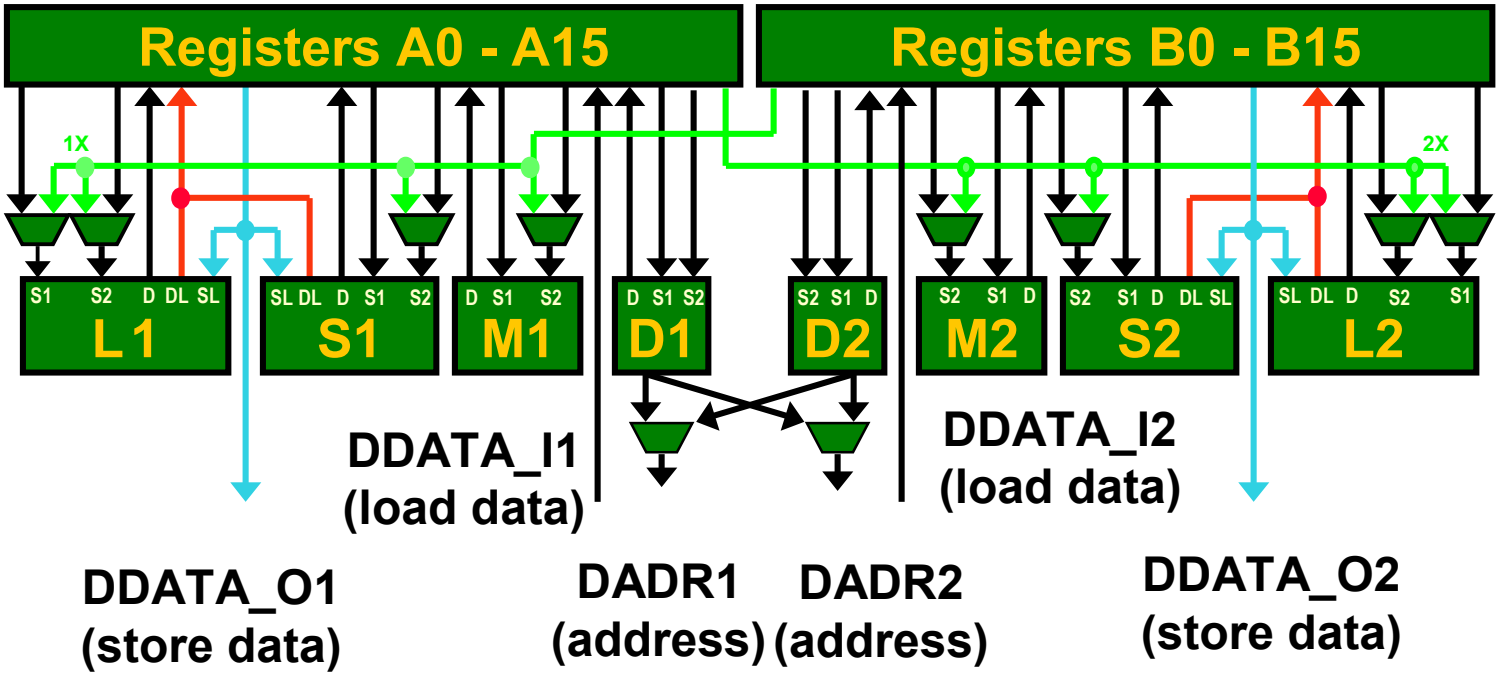
C6201 Internal Memory Architecture

- Separate Internal Program and Data Spaces
- Program
  - 16K 32-bit instructions (2K Fetch Packets)
  - 256-bit Fetch Width
  - Configurable as either
    - Direct Mapped Cache, Memory Mapped Program Memory
- Data
  - 32K x 16
  - Single Ported Accessible by Both CPU Data Buses
  - 4 x 8K 16-bit Banks
    - 2 Possible Simultaneous Memory Accesses (4 Banks)
    - 4-Way Interleave, Banks and Interleave Minimize Access Conflicts

C62x Interrupts

- 12 Maskable Interrupts , Non-Maskable Interrupt (NMI)
- Interrupt Return Pointers (IRP, NRP)
- Fast Interrupt Handing
  - Branches Directly to 8-Instruction Service Fetch Packet
  - Can Branch out with no overhead for longer service
  - 7 Cycle Overhead : Time When No Code is Running
  - 12 Cycle Latency : Interrupt Response Time
- Interrupt Acknowledge (IACK) and Number (INUM) Signals
- Branch Delay Slots Protected From Interrupts
- Edge Triggered

C62x Datapaths

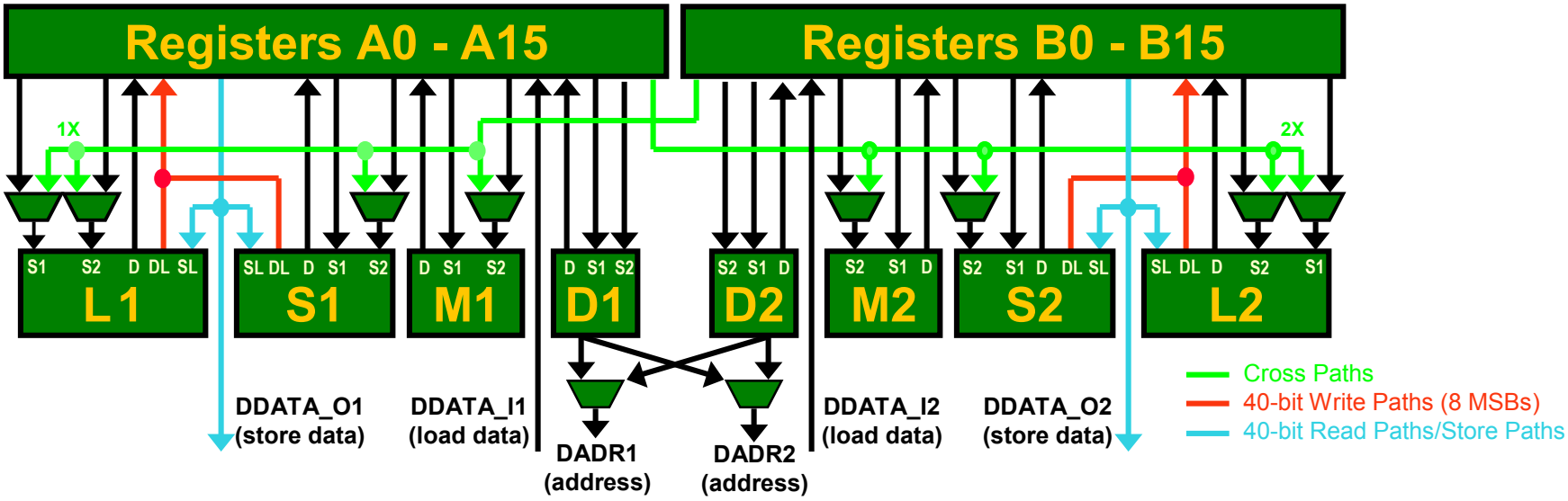


— Cross Paths
— 40-bit Write Paths (8 MSBs)
— 40-bit Read Paths/Store Paths

Functional Units

- L-Unit (L1, L2)
  - 40-bit Integer ALU, Comparisons
  - Bit Counting, Normalization
- S-Unit (S1, S2)
  - 32-bit ALU, 40-bit Shifter
  - Bitfield Operations, Branching
- M-Unit (M1, M2)
  - 16 x 16 -> 32
- D-Unit (D1, D2)
  - 32-bit Add/Subtract
  - Address Calculations

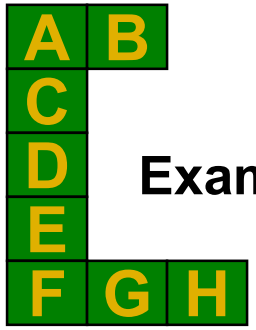
C62x Datapaths



C62x Instruction Packing
Instruction Packing Advanced VLIW

Example 1

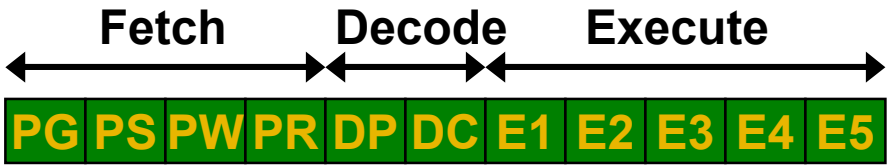
Example 2



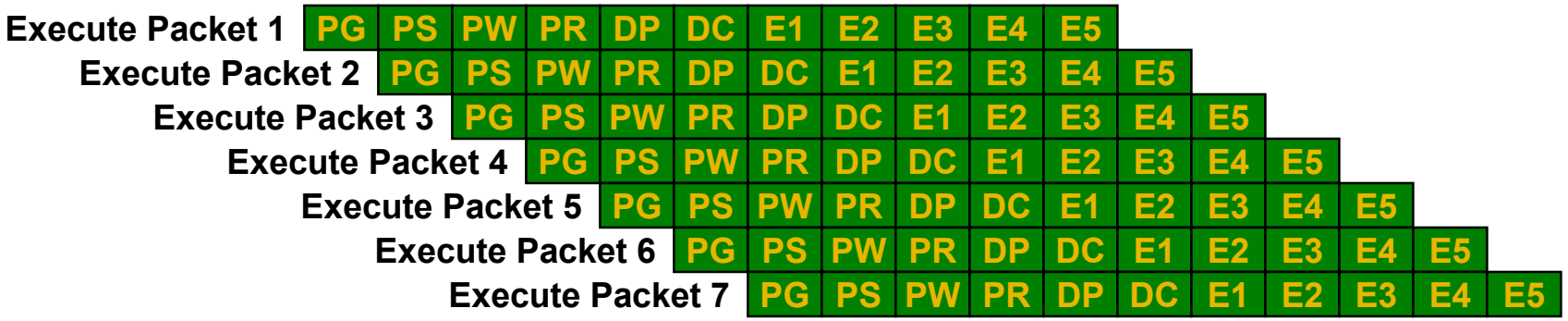
Example 3

- Fetch Packet
  - CPU fetches 8 instructions/cycle
- Execute Packet
  - CPU executes 1 to 8 instructions/cycle
  - Fetch packets can contain multiple execute packets
- Parallelism determined at compile / assembly time
- Examples
  - 1) 8 parallel instructions
  - 2) 8 serial instructions
  - 3) Mixed Serial/Parallel Groups
    - A // B
    - C
    - D
    - E // F // G // H
- Reduces Codesize, Number of Program Fetches, Power Consumption

C62x Pipeline Operation Pipeline Phases

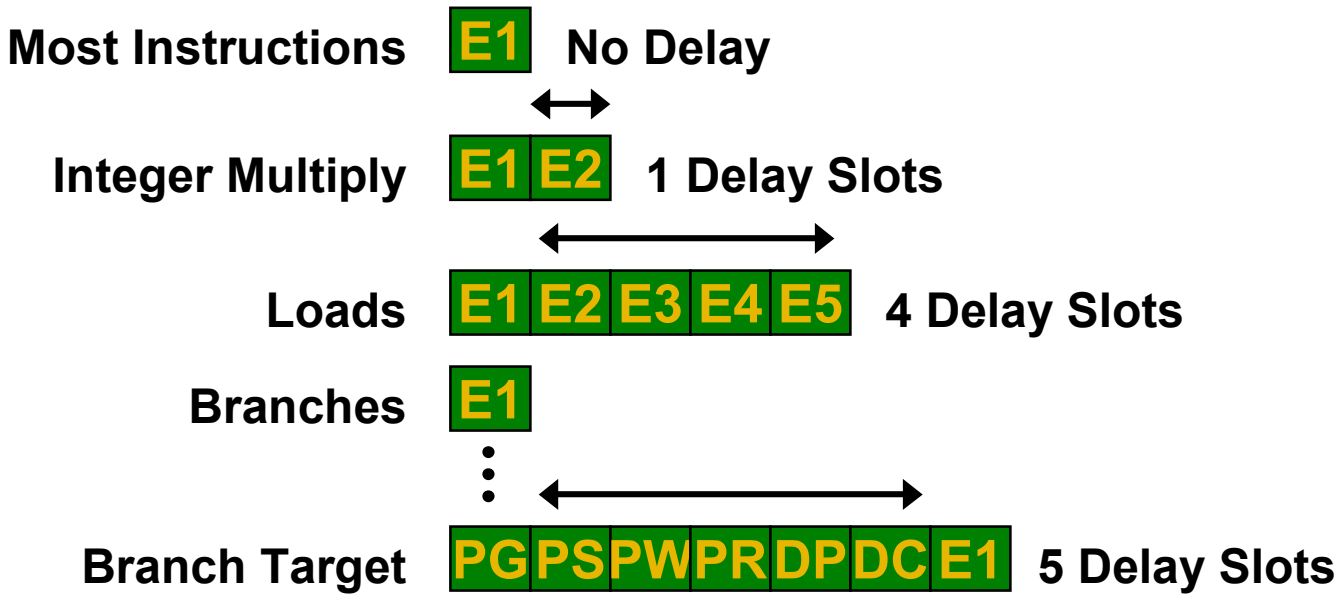


- Single-Cycle Throughput
- Operate in Lockstep
- Fetch
  - PG Program Address Generate
  - PS Program Address Send
  - PW Program Access Ready Wait
  - PR Program Fetch Packet Receive
- Decode
  - DP Instruction Dispatch
  - DC Instruction Decode
- Execute
  - E1 – E5 Execute 1 through Execute 5



C62x Pipeline Operation Delay Slots

- Delay Slots: number of extra cycles until result is:
  - written to register file
  - available for use by a subsequent instructions
  - Multi-cycle NOP instruction can fill delay slots while minimizing codesize impact



C6000 Pipeline Operation Benefits

- Cycle Time
  - Allows 6 ns cycle time on 67x
  - Allows 5 ns cycle time & single cycle execution on C62x
- Parallelism
  - 8 new instructions can always be dispatched every cycle
- High Performance Internal Memory Access
  - Pipelined Program and Data Accesses
  - Two 32-bit Data Accesses/Cycle (C62x)
  - Two 64-bit Data Accesses/Cycle (C67x)
  - 256-bit Program Access/Cycle
- Good Compiler Target
  - Visible: No Variable-Length Pipeline Flow
  - Deterministic: Order and Time of Execution
  - Orthogonal: Independent Instructions

C6000 Instruction Set Features
Conditional Instructions

- All Instructions can be Conditional
  - A1, A2, B0, B1, B2 can be used as Conditions
  - Based on Zero or Non-Zero Value
  - Compare Instructions can allow other Conditions (<, >, etc)
- Reduces Branching
- Increases Parallelism

C6000 Instruction Set Addressing Features

- Load-Store Architecture
- Two Addressing Units (D1, D2)
- Orthogonal
  - Any Register can be used for Addressing or Indexing
- Signed/Unsigned Byte, Half-Word, Word, Double-Word Addressable
  - Indexes are Scaled by Type
- Register or 5-Bit Unsigned Constant Index

C6000 Instruction Set Addressing Features

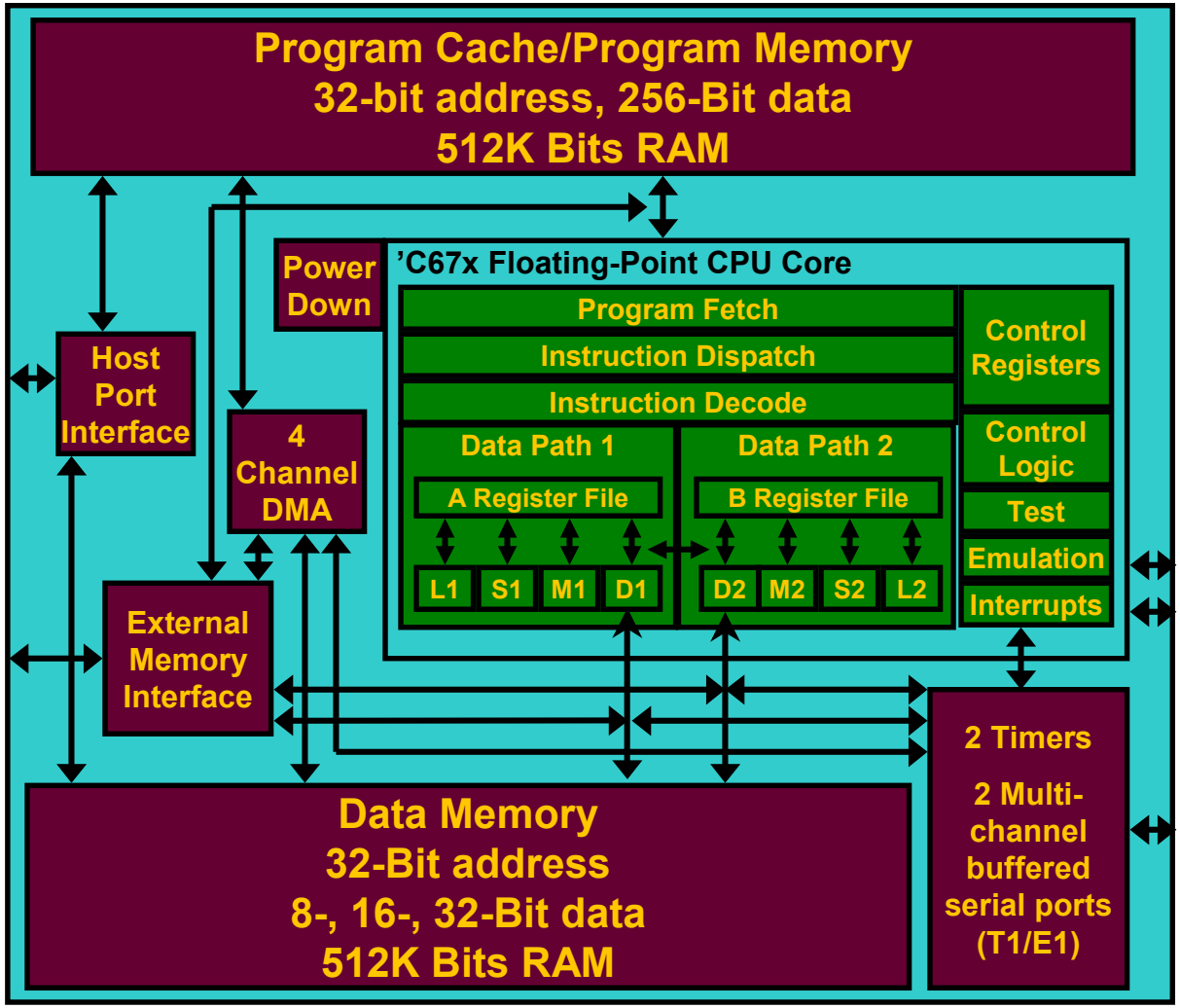
- Indirect Addressing Modes
  - Pre-Increment *++R[index]
  - Post-Increment *R++[index]
  - Pre-Decrement *--R[index]
  - Post-Decrement *R--[index]
  - Positive Offset *+R[index]
  - Negative Offset *-R[index]
- 15-bit Positive/Negative Constant Offset from Either B14 or B15

C6000 Instruction Set Addressing Features

- Circular Addressing
  - Fast and Low Cost: Power of 2 Sizes and Alignment
  - Up to 8 Different Pointers/Buffers, Up to 2 Different Buffer Sizes
- Dual Endian Support

C67x Architecture

TMS320C6701 DSP Block Diagram



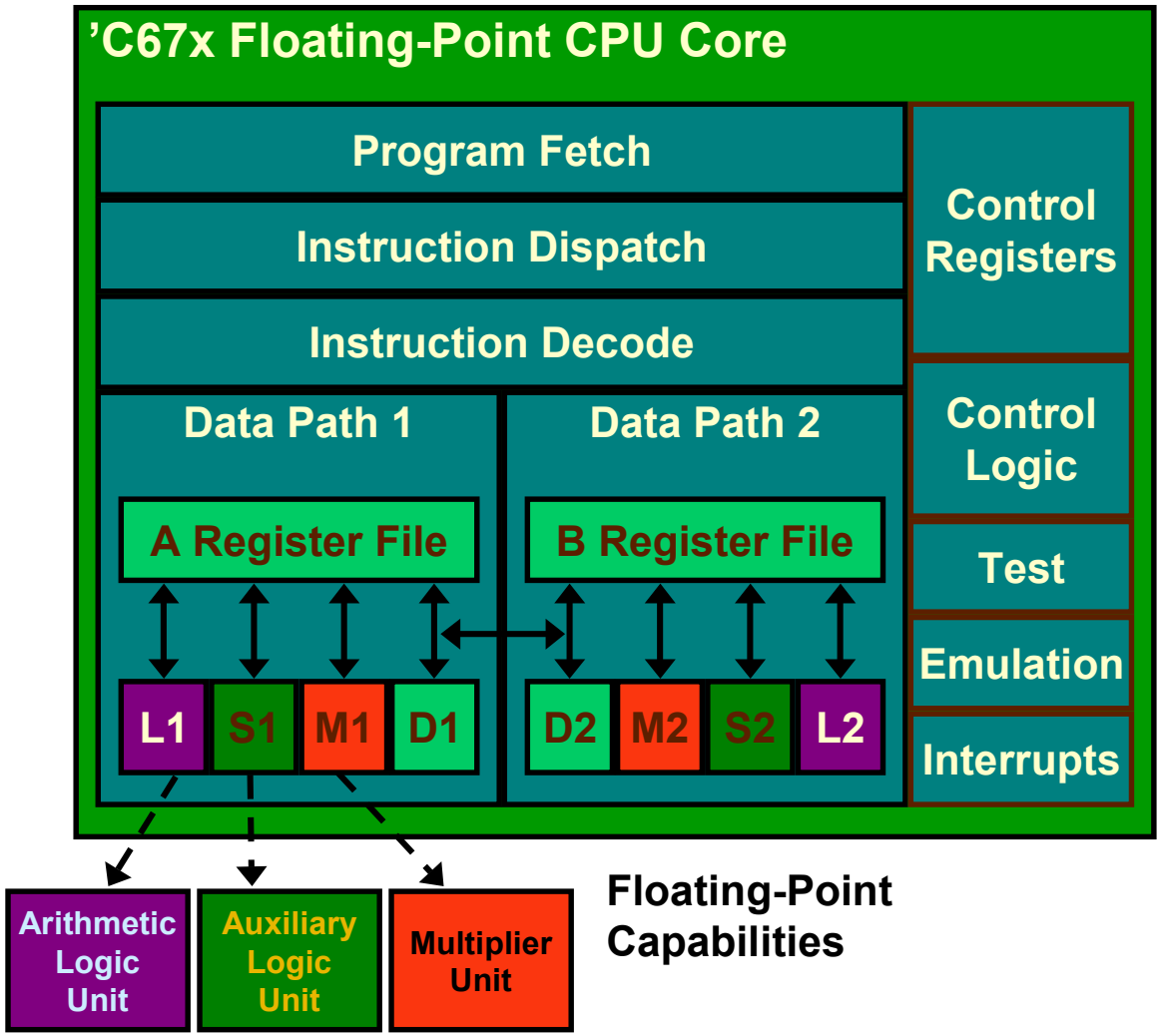
TMS320C6701
Advanced VLIW CPU (VelociTI™)

- 1 GFLOPS @ 167 MHz
  - 6-ns cycle time
  - 6 x 32-bit floating-point instructions/cycle
- Load store architecture
- 3.3-V I/Os, 1.8-V internal
- Single- and double-precision IEEE floating-point
- Dual data paths
  - 6 floating-point units / 8 x 32-bit instructions

TMS320C6701
Memory /Peripherals

- Same as ’C6201
- External interface supports
  - SDRAM, SRAM, SBSRAM
- 4-channel bootloading DMA
- 16-bit host port interface
- 1Mbit on-chip SRAM
- 2 multichannel buffered serial ports (T1/E1)
- Pin compatible with ’C6201

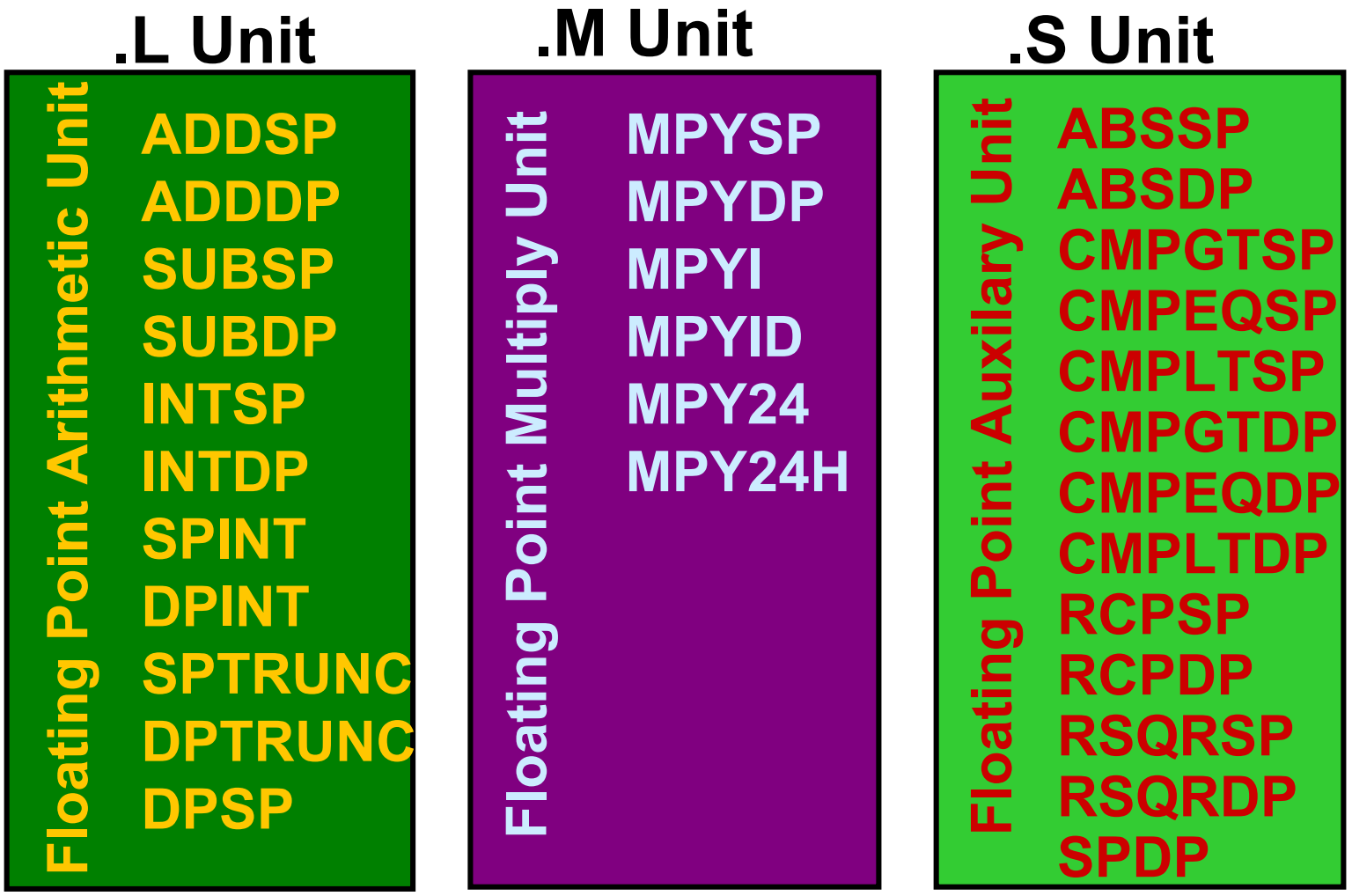
TMS320C67x CPU Core



C67x Interrupts

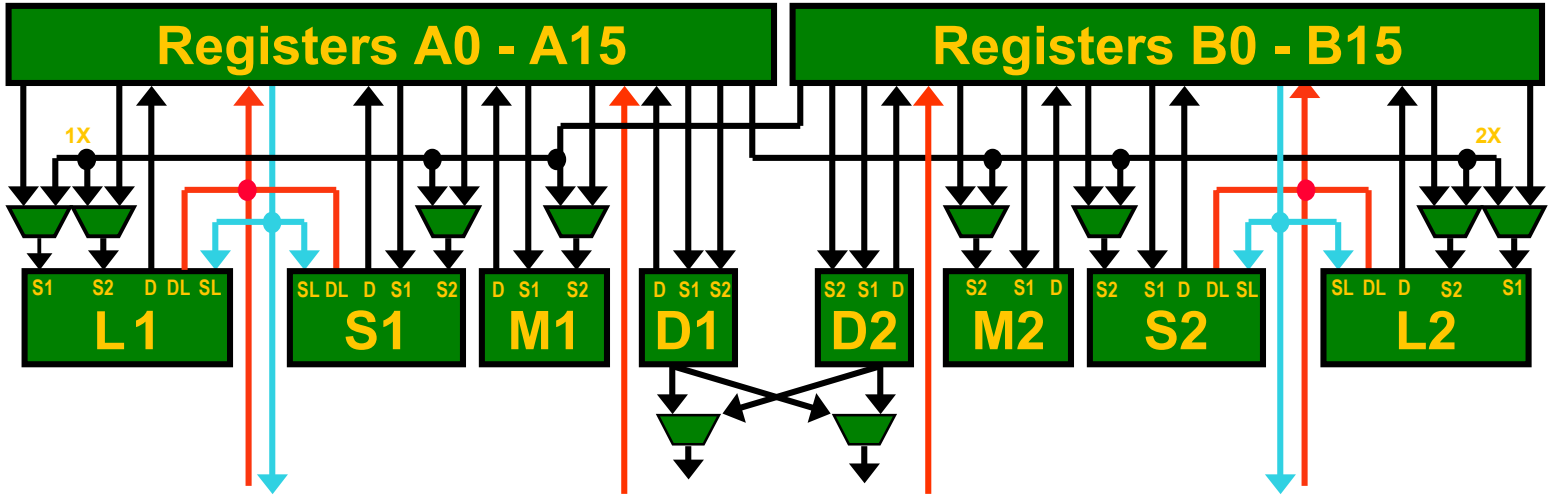
- 12 Maskable Interrupts
- Non-Maskable Interrupt (NMI)
- Interrupt Return Pointers (IRP, NRP)
- Fast Interrupt Handling
  - Branches Directly to 8-Instruction Service Fetch Packet
  - 7 Cycle Overhead: Time When No Code is Running
  - 12 Cycle Latency : Interrupt Response Time
- Interrupt Acknowledge (IACK) and Number (INUM) Signals
- Branch Delay Slots Protected From Interrupts
- Edge Triggered

C67x New Instructions



C67x Datapaths

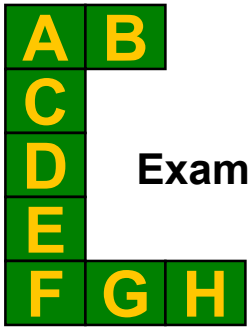
- 2 Data Paths
- 8 Functional Units
  - Orthogonal/Independent
  - 2 Floating Point Multipliers
  - 2 Floating Point Arithmetic
  - 2 Floating Point Auxiliary
- Control
  - Independent
  - Up to 8 32-bit Instructions
- Registers
  - 2 Files
  - 32, 32-bit registers total
- Cross paths (1X, 2X)
- L-Unit (L1, L2)
  - Floating-Point, 40-bit Integer ALU
  - Bit Counting, Normalization
- S-Unit (S1, S2)
  - Floating Point Auxiliary Unit
  - 32-bit ALU/40-bit shifter
  - Bitfield Operations, Branching
- M-Unit (M1, M2)
  - Multiplier: Integer & Floating-Point
- D-Unit (D1, D2)
  - 32-bit add/subtract Addr Calculations



C67x Instruction Packing
Instruction Packing Enhanced VLIW

Example 1

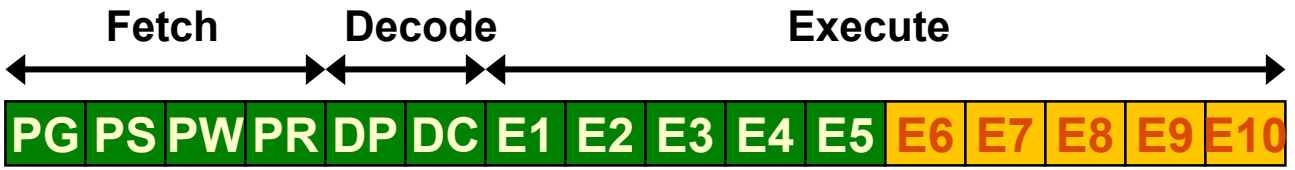
Example 2



Example 3

- Fetch Packet
  - CPU fetches 8 instructions/cycle
- Execute Packet
  - CPU executes 1 to 8 instructions/cycle
  - Fetch packets can contain multiple execute packets
- Parallelism determined at compile/assembly time
- Examples
  - 1) 8 parallel instructions
  - 2) 8 serial instructions
  - 3) Mixed Serial/Parallel Groups
    - A // B
    - C
    - D
    - E // F // G // H
- Reduces
  - Codesize
  - Number of Program Fetches
  - Power Consumption

C67x Pipeline Operation
Pipeline Phases

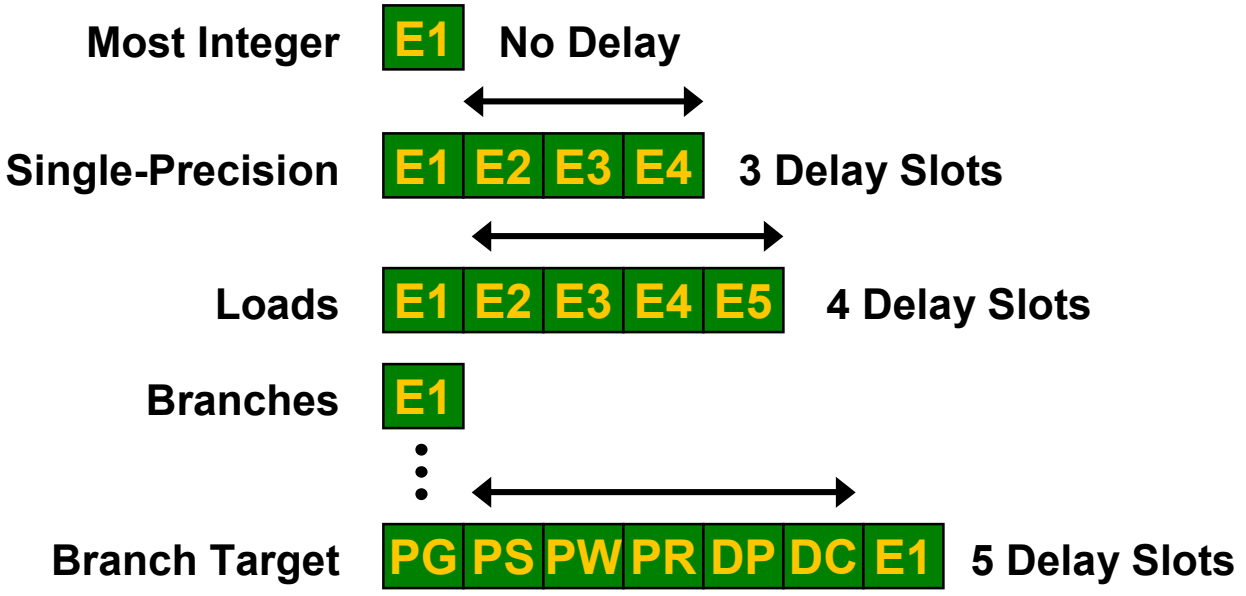


- Operate in Lock Step
- Fetch
  - PG Program Address Generate
  - PS Program Address Send
  - PW Program Access Ready Wait
  - PR Program Fetch Packet Receive
- Decode
  - DP Instruction Dispatch
  - DC Instruction Decode
- Execute
  - E1 - E5 Execute 1 through Execute 5
  - E6 - E10 Double Precision Only

C67x Pipeline Operation
Delay Slots

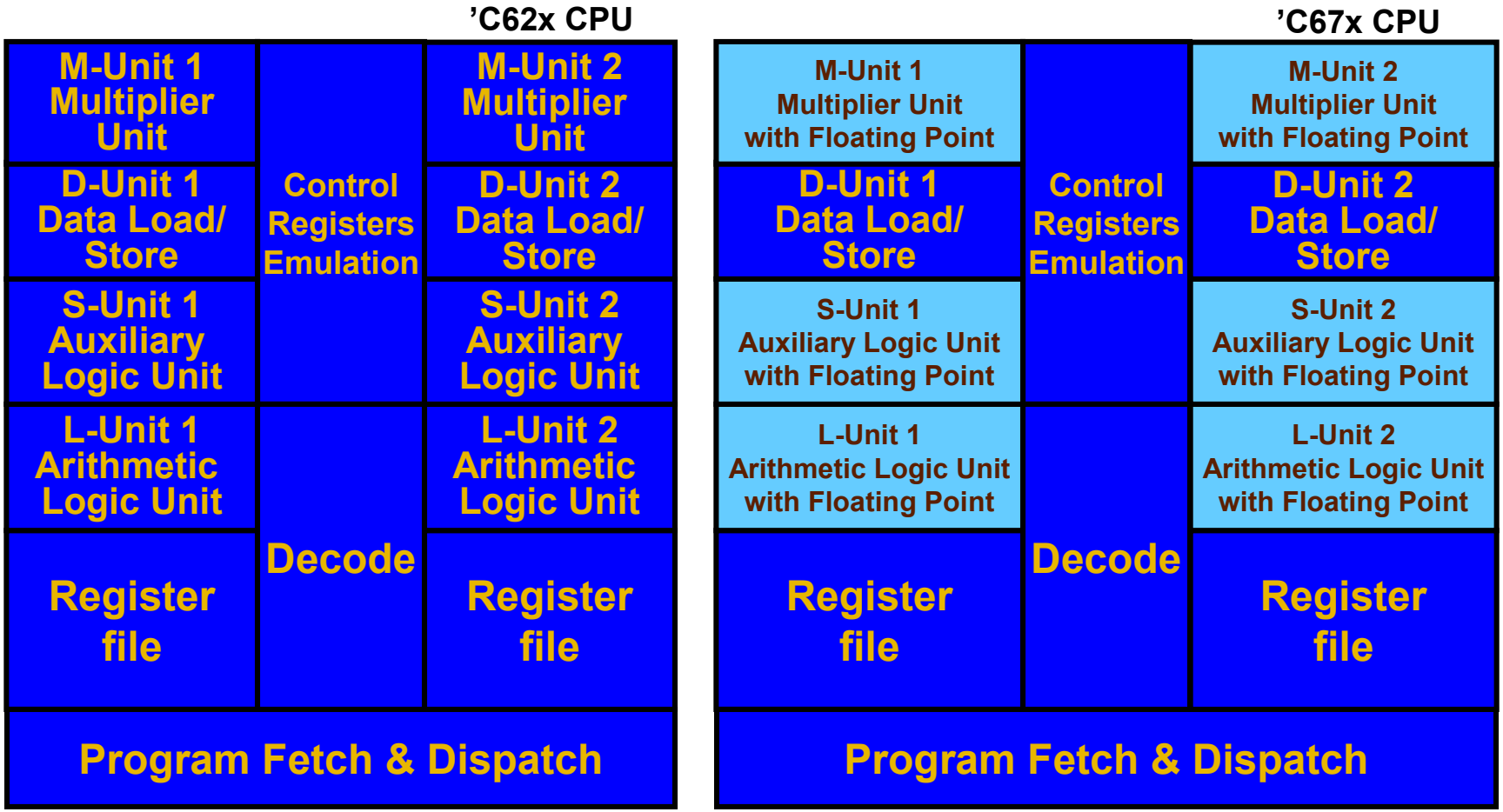
Delay Slots: number of extra cycles until result is:

- **written to register file**
- **available for use by a subsequent instructions**
- **Multi-cycle NOP instruction can fill delay slots while minimizing codesize impact**

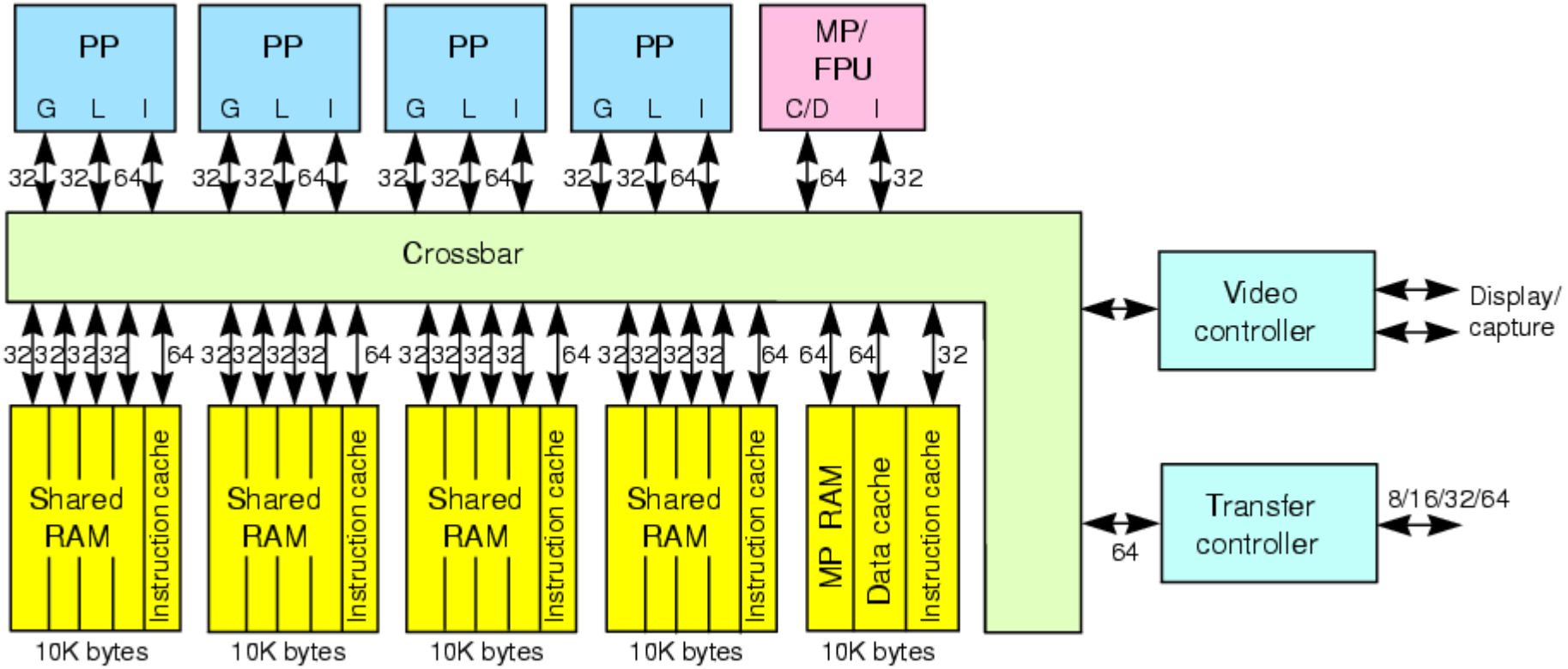


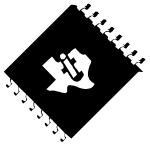
’C67x and ’C62x Commonality

- Driving commonality ( ) between ’C67x & ’C62x shortens ’C67x design time.
- Maintaining symmetry between datapaths shortens the ’C67x design time.



TMS320C80 MIMD MULTIPROCESSOR
Texas Instruments - 1996





Copyright 1999