TMS AND ADSP PROCESSORS Architecture and Features

Dr pooja Sahni

Professor,ECE Department

What is DSP ?

Digital Signal Processing (DSP) is used in a wide variety of applications, and it is hard to find a good definition that is general.

changing or analyzing information which is measured as discrete sequences of numbers

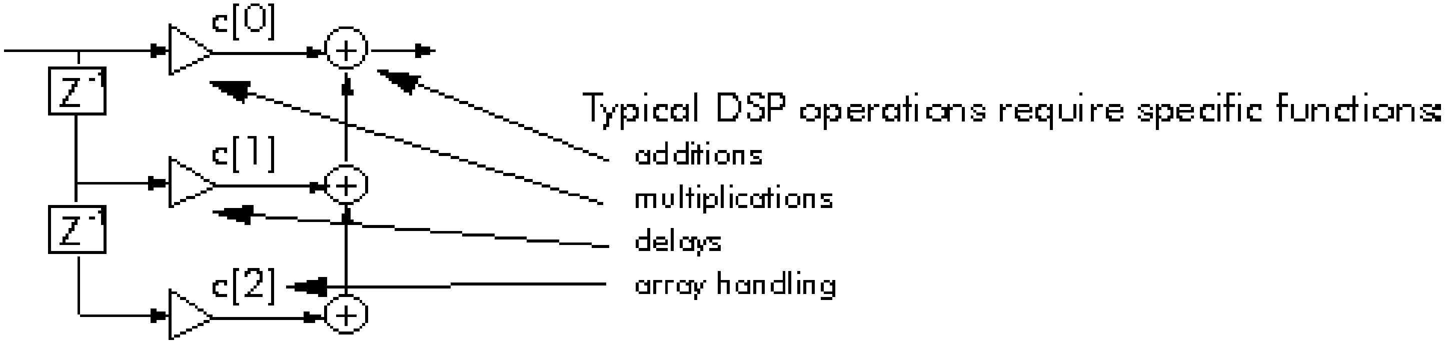
DSP vs. Microcontroller

- DSP
  - Harvard Architecture
  - VLIW(Very Long Instruction Word) (parallel execution units)
  - No bit level operations
  - Hardware MACs
  - DSP applications

- Microcontroller
  - Mostly von Neumann Architecture
  - Single execution unit
  - Flexible bit-level operations
  - No hardware MACs
  - Control applications

DSP Algorithm

Example : Digital Filters (e.g. Digital FIR Filters ,...)



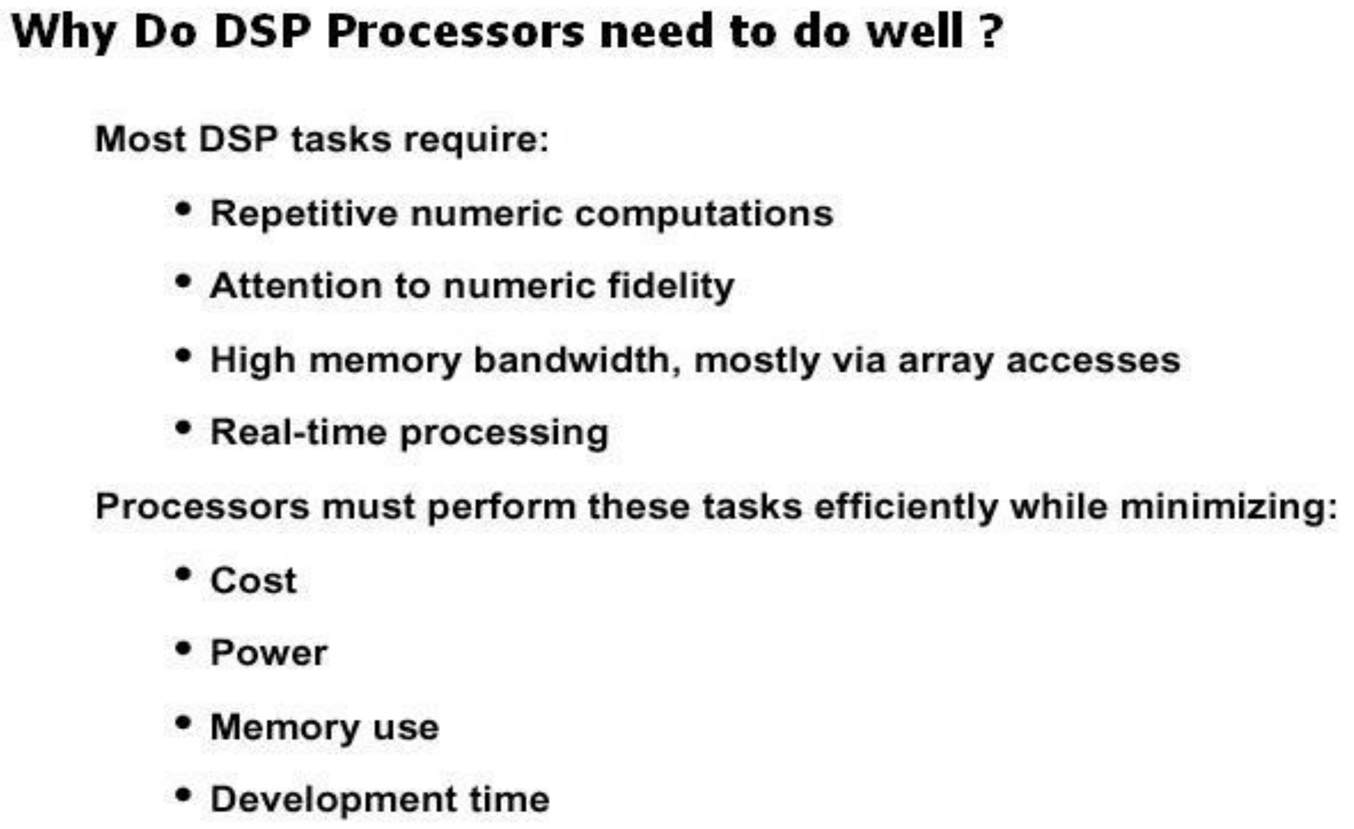
Typical DSP operations require specific functions:

additions
multiplications
delays
array handling

Most share common features:

- They use a lot of maths (multiplying and adding signals)
- They deal with signals that come from the real world
- They require a response in a certain time

Why DSP Processors? (Contd.)



Why Do DSP Processors need to do well ?

Most DSP tasks require:

- Repetitive numeric computations
- Attention to numeric fidelity
- High memory bandwidth, mostly via array accesses
- Real-time processing

Processors must perform these tasks efficiently while minimizing:

- Cost
- Power
- Memory use
- Development time

Some Typical Applications

- **General-Purpose**
  - Adaptive filtering
  - Digital filtering
  - Fast Fourier transforms
- **Control**
  - Disk drive control
  - Laser printer control
  - Robotics control
- **Military**
  - Missile guidance
  - Radar processing
  - Secure communication
- **Telecommunications**
  - 1200- to 19200-bps modems
  - Adaptive equalizers
  - Cellular telephones

Some key features

CPU

- Advanced multi bus architecture with three separate 16-bit data buses and one program bus

- 40-bit arithmetic logic unit (ALU), including a 40-bit barrel shifter and two independent 40-bit accumulators

- 17-bit × 17-bit parallel multiplier coupled to a 40-bit dedicated adder for non-pipelined single-cycle multiply/accumulate (MAC) operation

Memory

▪192K words × 16-bit maximum addressable memory space (64K words program, 64K words data, and 64K words I/O)

▪ 28K words × 16-bit single-access on-chip ROM with 8K words configurable as program or data memory (’C541 only)

On-chip peripherals

- On-chip phase-locked loop (PLL) clock generator with internal oscillator or external clock source
- Two full-duplexed serial ports to support 8- and 16-bit transfers (’C541only)
- Time-division multiplexed (TDM) serial port (’C542/’C543 only)
- One 16-bit timer

Speed: 25/20-ns execution time for a single-cycle fixed-point instruction (40 MIPS/50 MIPS) with 5-V power supply

History of the TMS320 family

Texas Instruments,the company

This family currently includes five generations of DSPs.

TMS320C1x, TMS320C2x, TMS320C3x, TMS320C4x, and TMS320C5x

TMS320C25, a CMOS 40-MHz digital signal processor capable of twice the performance of the TMS320C1x devices

is capable of executing 10 million instructions per second.

24 additional instructions (133 total)

eight auxiliary registers

an eight-level hardware stack

4K words of on-chip program ROM

low power dissipation inherent to CMOS

Architectural overview

Harvard architecture

On-chip memory

ALU

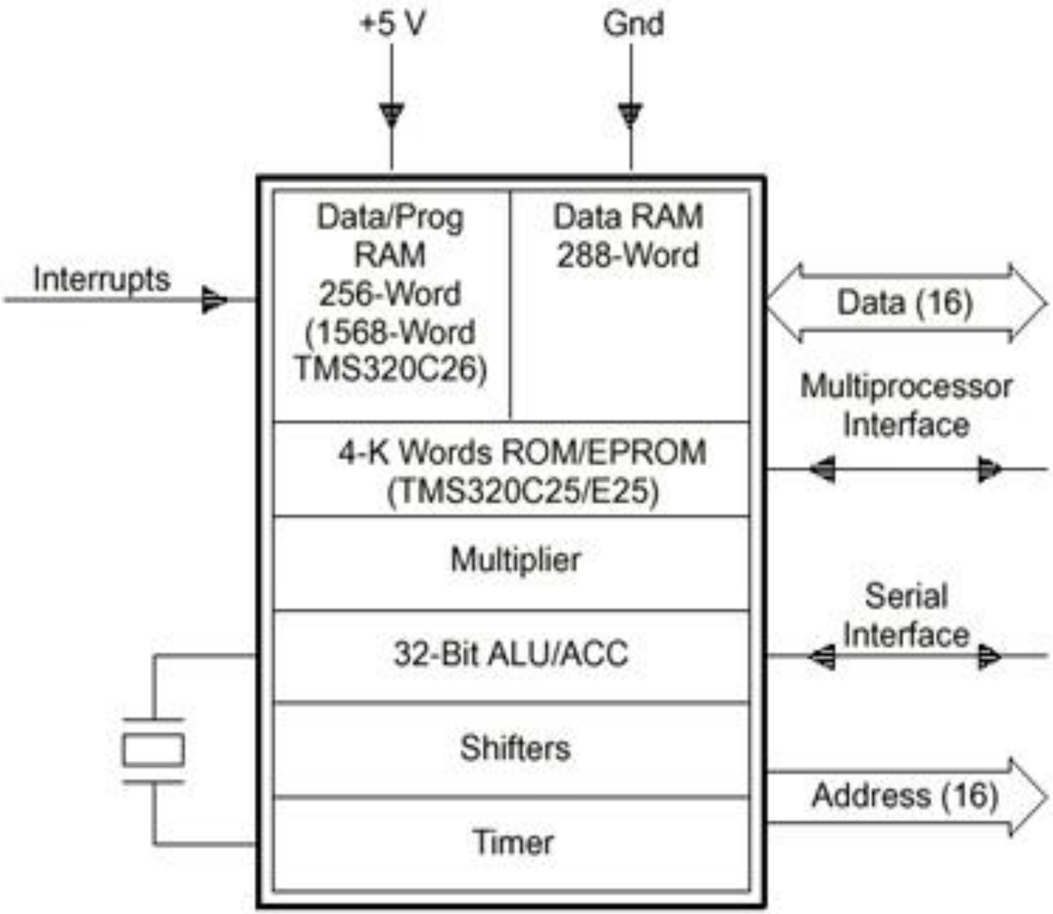
Multiplier

Memory interface

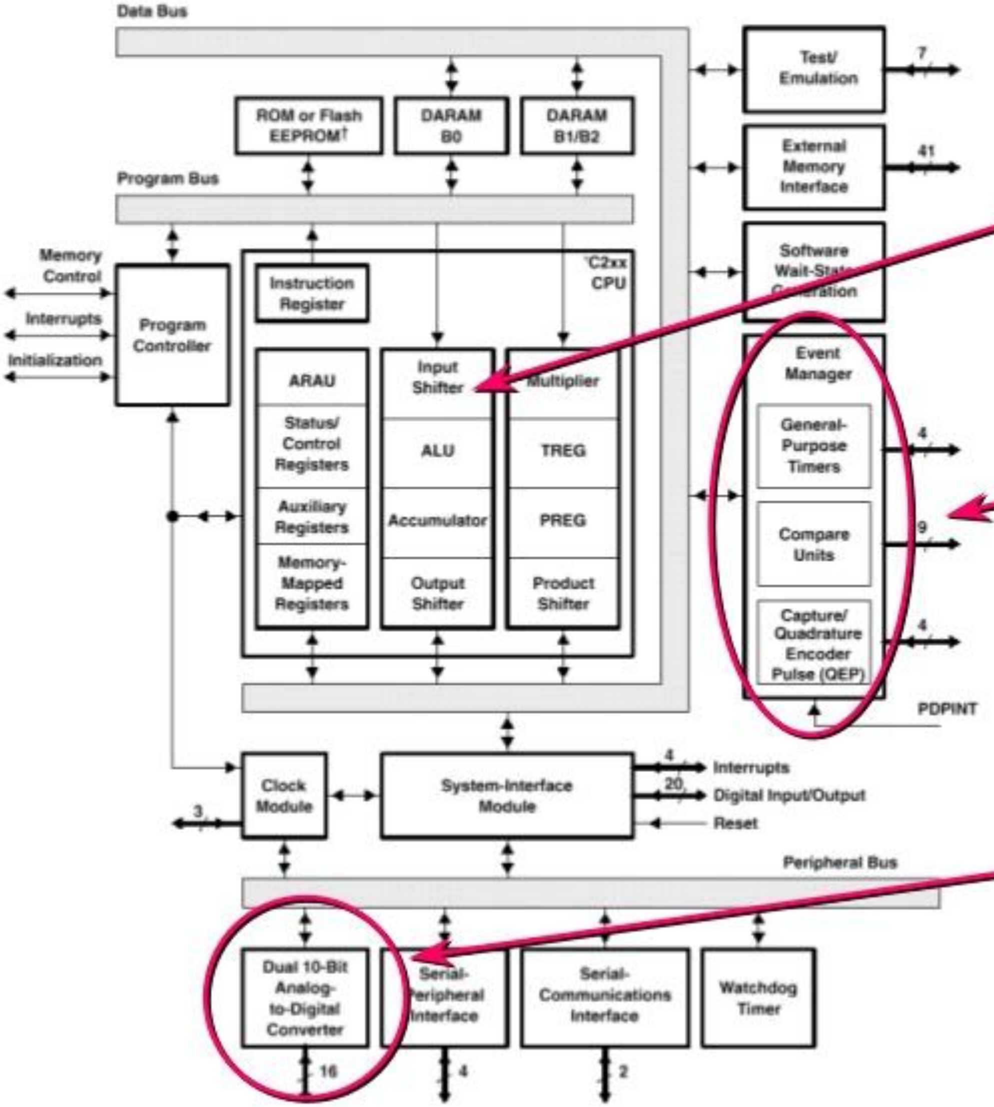
Serial ports

Multiprocessing application

Direct Memory Access



TMS320F240 Functional Block Diagram



Central Arithmetic Logic Unit (CALU)

Event Manager:
Timers
Compare Units ...

Dual 10 bit ADC

Basic Architectural Features of DSPs

- **Data path configured for DSP**
  - Fixed-point arithmetic
  - MAC- Multiply-accumulate
- **Multiple memory banks and buses**
  - Harvard Architecture
  - Multiple data memories
- **Specialized addressing modes**
  - Bit-reversed addressing
  - Circular buffers
- **Specialized instruction set and execution control – Zero-overhead loops**
  - Support for fast MAC
  - Fast Interrupt Handling
- **Specialized peripherals for DSP**

Memory Organization

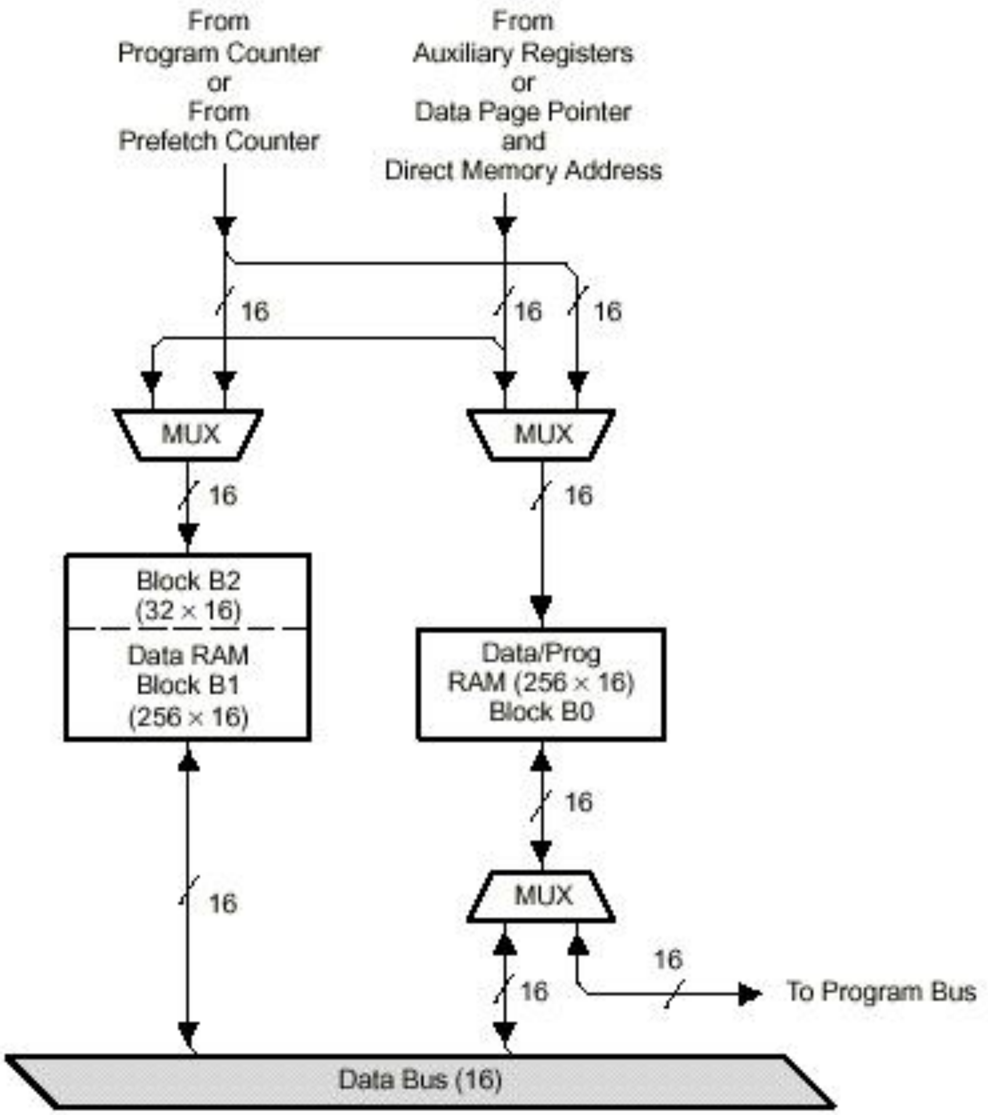
Total of 544 16-bit words of on-chip data RAM,

Program and Data Memory

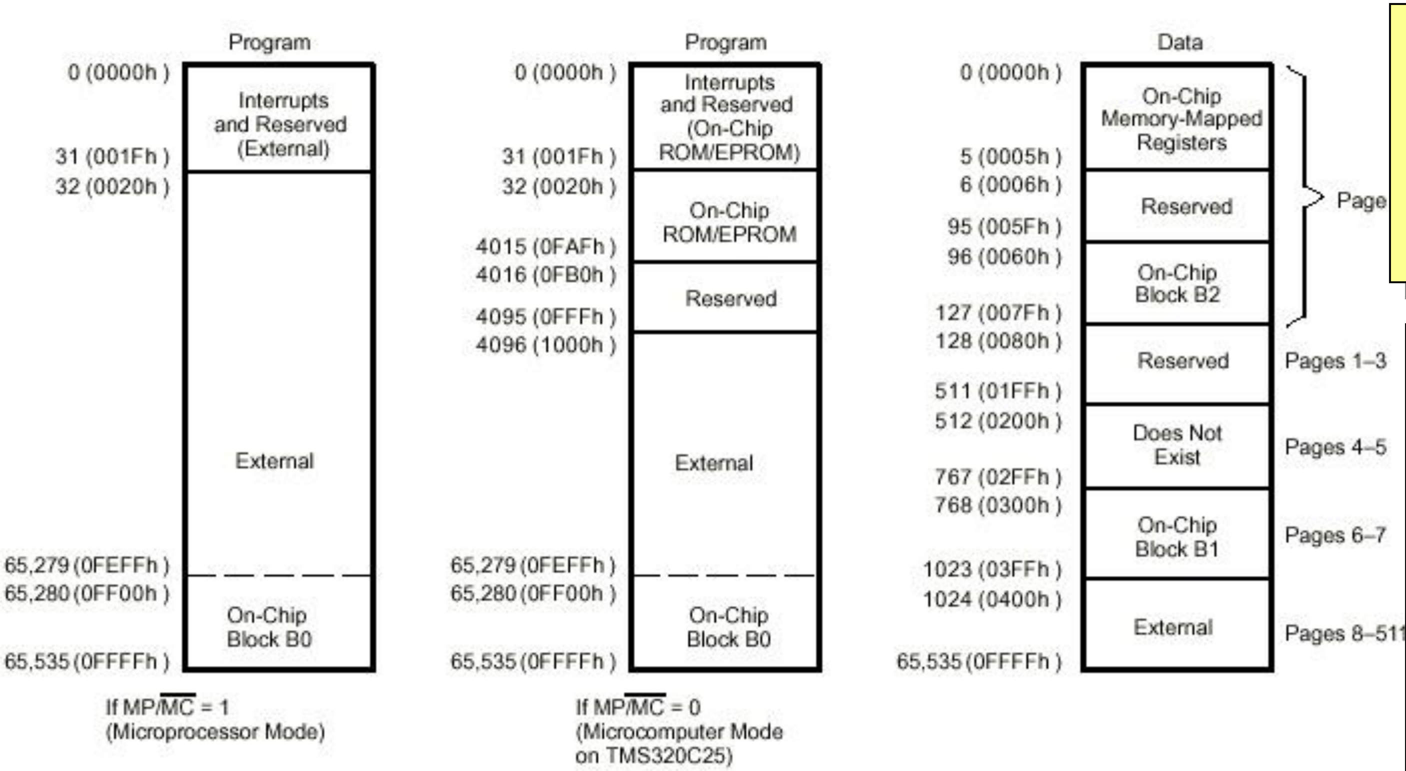
288 words are always data memory and the remaining 256 words may
be configured as either program or data memory.

The TMS320C2x can address a total of 64K words of data memory.

TMS320C2x On-Chip Data Memory



Memory Organization (Cntd.)



spaces are distinguished externally by means of the PS, DS, and IS

The on-chip program ROM can be mapped into the lower 4K words of program

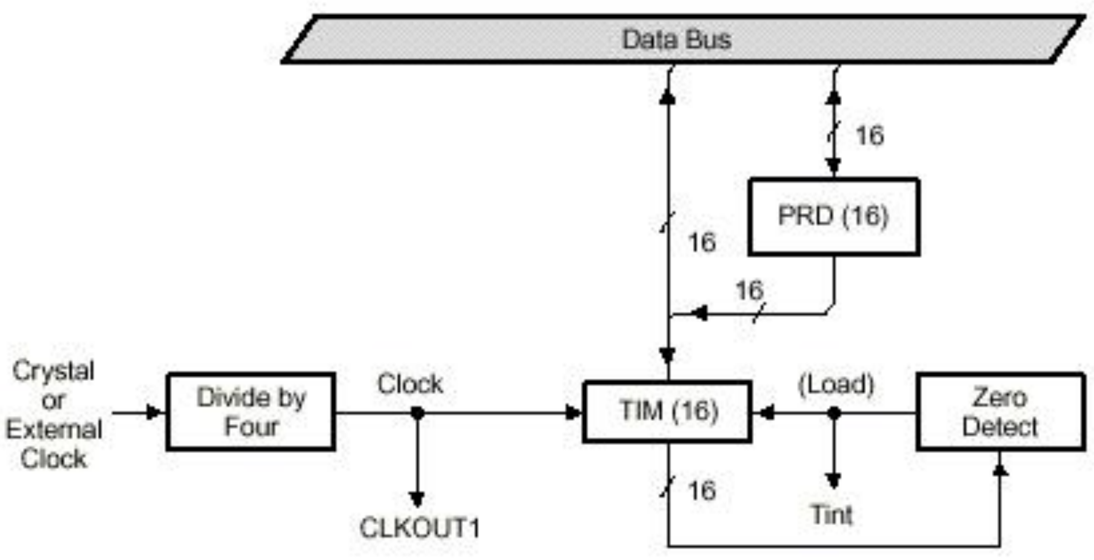
memory. This ROM is enabled when MP/MC is set to a logic low.

Three separate address spaces for program memory, data memory, and I/O

System Control (Timer Operation+Repeat Counter)

The TMS320C2x provides a memory-mapped 16-bit timer (TIM) register and a 16-bit period (PRD) register.

The on-chip timer is a down counter that is continuously clocked by CLKOUT1.



The repeat counter (RPTC) is an 8-bit counter.It can be loaded with a number from 0 to 255 .

RPTC is cleared by reset.

External Memory and IO Interface

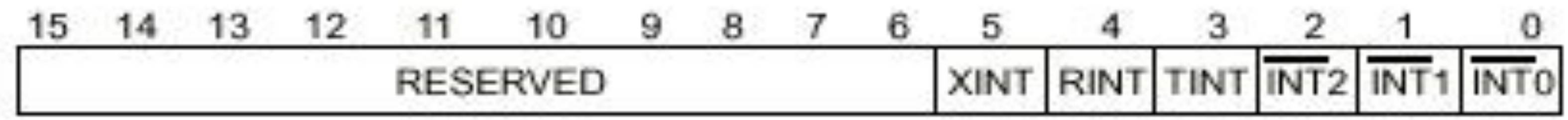
A 16-bit parallel data bus (D15–D0),
A 16-bit address bus (A15–A0),
Data, program, and I/O space select (DS, PS, and IS) signals, and
Various system control signals.

1) Program Internal RAM/Data Internal (PI/DI)
2) Program Internal RAM/Data External (PI/DE)
3) Program External/Data Internal (PE/DI)
4) Program External/Data External (PE/DE)
5) Program Internal ROM/Data Internal (PR/DI)
6) Program Internal ROM/Data External (PR/DE)

Interrupts

three external maskable user interrupts (INT2–INT0),

Internal interrupts are generated by the serial port (RINT and XINT), by the timer (TINT), and by the software interrupt (TRAP) instruction.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										XINT	RINT	TINT	$\overline{\text{INT2}}$	$\overline{\text{INT1}}$	$\overline{\text{INT0}}$

The TMS320C2x has a built-in mechanism for protecting multicycle instructions from interrupts.

ADSP – 21060 SHARC Digital Signal Processor

General Information

- SHARC stands for Super Harvard Architecture Computer
- The ADSP-21060 SHARC chip is made by Analog Devices, Inc.
- It is a 32-bit signal processor made mainly for sound, speech,graphics, and imaging applications.
- It is a high-end digital signal processor designed with RISC techniques.

Memory Structure

- Memory is arranged in a unified, word-addressable address space containing both instructions and data.

- Separate address generators, address buses, and data buses allow both on-chip memory blocks to be accessed by the core processor in a single instruction cycle.

- The total on-chip memory size of the ADSP-21060 is 4 Mbits. The block size is 2 MBits.

- The on-chip memory can be configured as 16, 32 or 48 bit words, and is organized into two independent halves. Each can be used for instructions or data.

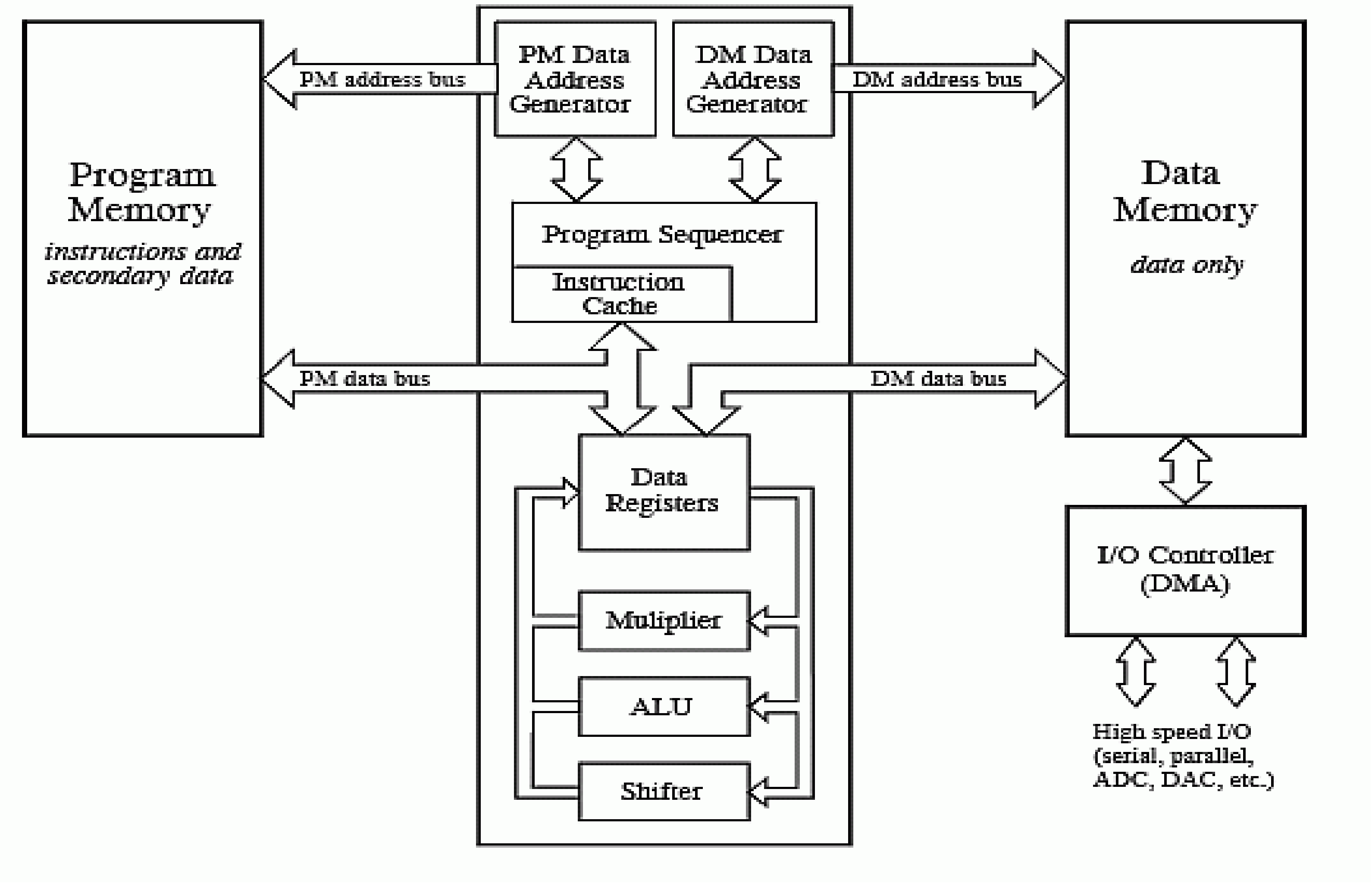
Endian Format

- SHARC uses big-endian format
  - Most significant byte is at the lowest address
- EXCEPT
  - Bit order for data transfer through the serial port.
  - Word order for packing through the external port.
- For compatibility with little-endian (least-significant-first) peripherals, the DSP supports both big- and little-endian bit order data transfers. Also for compatibility little endian hosts, the DSP supports both big- and little endian word order data transfers.

Number Formats

- 32-bit Fixed Format
  - Fractional/Integer
  - Unsigned/Signed
- Floating Point
  - 32-bit single-precision IEEE floating-point data format
  - 40-bit version of the IEEE floating-point data format.
  - 16-bit shortened version of the IEEE floating-point data format.

Architecture ADSP

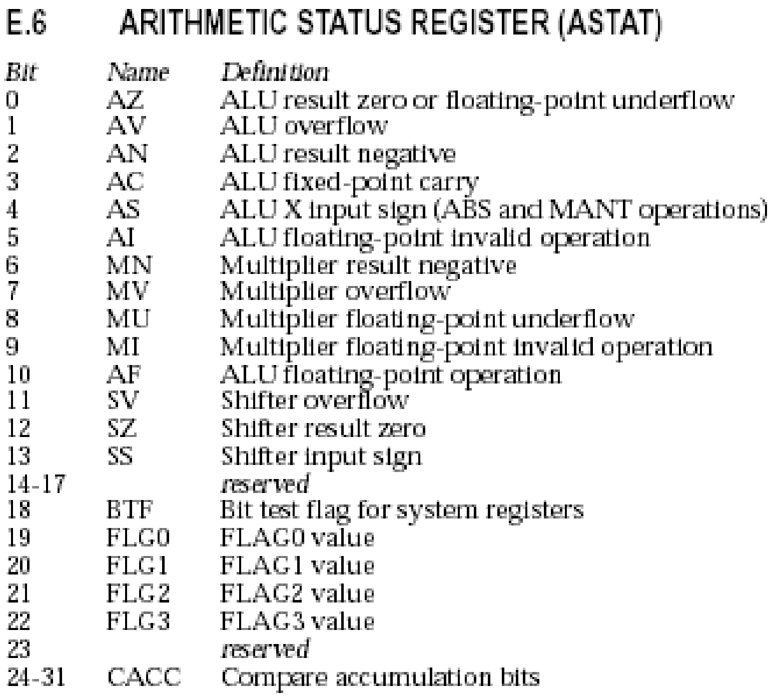


General Registers

- **16 Primary Registers**
- **16 Alternate Registers**
- **Each Register holds 40-bits**
- **Registers are references by the type of numbers they are holding**
    - **R0 – R15 are for Fixed-Point Numbers**
    - **F0 – F15 are for Floating-Point Numbers**

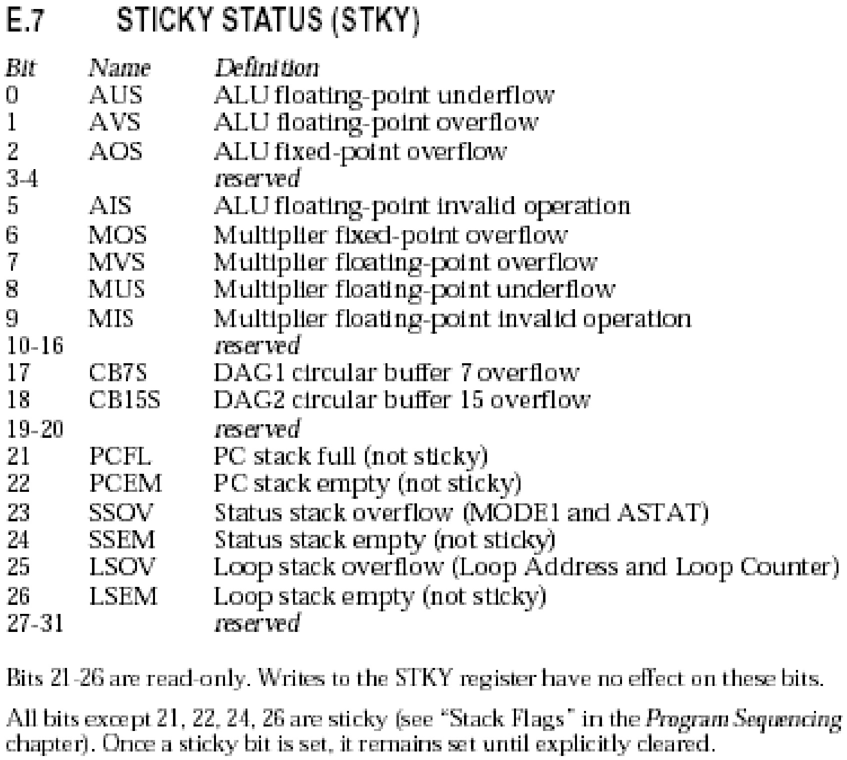
Specialized Registers

A few examples of some of the many registers and their components



E.6 ARITHMETIC STATUS REGISTER (ASTAT)

Bit	*Name*	*Definition*
0	AZ	ALU result zero or floating-point underflow
1	AV	ALU overflow
2	AN	ALU result negative
3	AC	ALU fixed-point carry
4	AS	ALU X input sign (ABS and MANT operations)
5	AI	ALU floating-point invalid operation
6	MN	Multiplier result negative
7	MV	Multiplier overflow
8	MU	Multiplier floating-point underflow
9	MI	Multiplier floating-point invalid operation
10	AF	ALU floating-point operation
11	SV	Shifter overflow
12	SZ	Shifter result zero
13	SS	Shifter input sign
14-17		*reserved*
18	BTF	Bit test flag for system registers
19	FLG0	FLAG0 value
20	FLG1	FLAG1 value
21	FLG2	FLAG2 value
22	FLG3	FLAG3 value
23		*reserved*
24-31	CACC	Compare accumulation bits



E.7 STICKY STATUS (STKY)

Bit	*Name*	*Definition*
0	AUS	ALU floating-point underflow
1	AVS	ALU floating-point overflow
2	AOS	ALU fixed-point overflow
3-4		*reserved*
5	AIS	ALU floating-point invalid operation
6	MOS	Multiplier fixed-point overflow
7	MVS	Multiplier floating-point overflow
8	MUS	Multiplier floating-point underflow
9	MIS	Multiplier floating-point invalid operation
10-16		*reserved*
17	CB7S	DAG1 circular buffer 7 overflow
18	CB15S	DAG2 circular buffer 15 overflow
19-20		*reserved*
21	PCFL	PC stack full (not sticky)
22	PCEM	PC stack empty (not sticky)
23	SSOV	Status stack overflow (MODE1 and ASTAT)
24	SSEM	Status stack empty (not sticky)
25	LSOV	Loop stack overflow (Loop Address and Loop Counter)
26	LSEM	Loop stack empty (not sticky)
27-31		*reserved*

Bits 21-26 are read-only. Writes to the STKY register have no effect on these bits.

All bits except 21, 22, 24, 26 are sticky (see "Stack Flags" in the *Program Sequencing* chapter). Once a sticky bit is set, it remains set until explicitly cleared.

Pipelining

- Instructions are processed in three cycles:
  - Fetch instruction from memory
  - Decode the opcode and operand
  - Execute the instruction

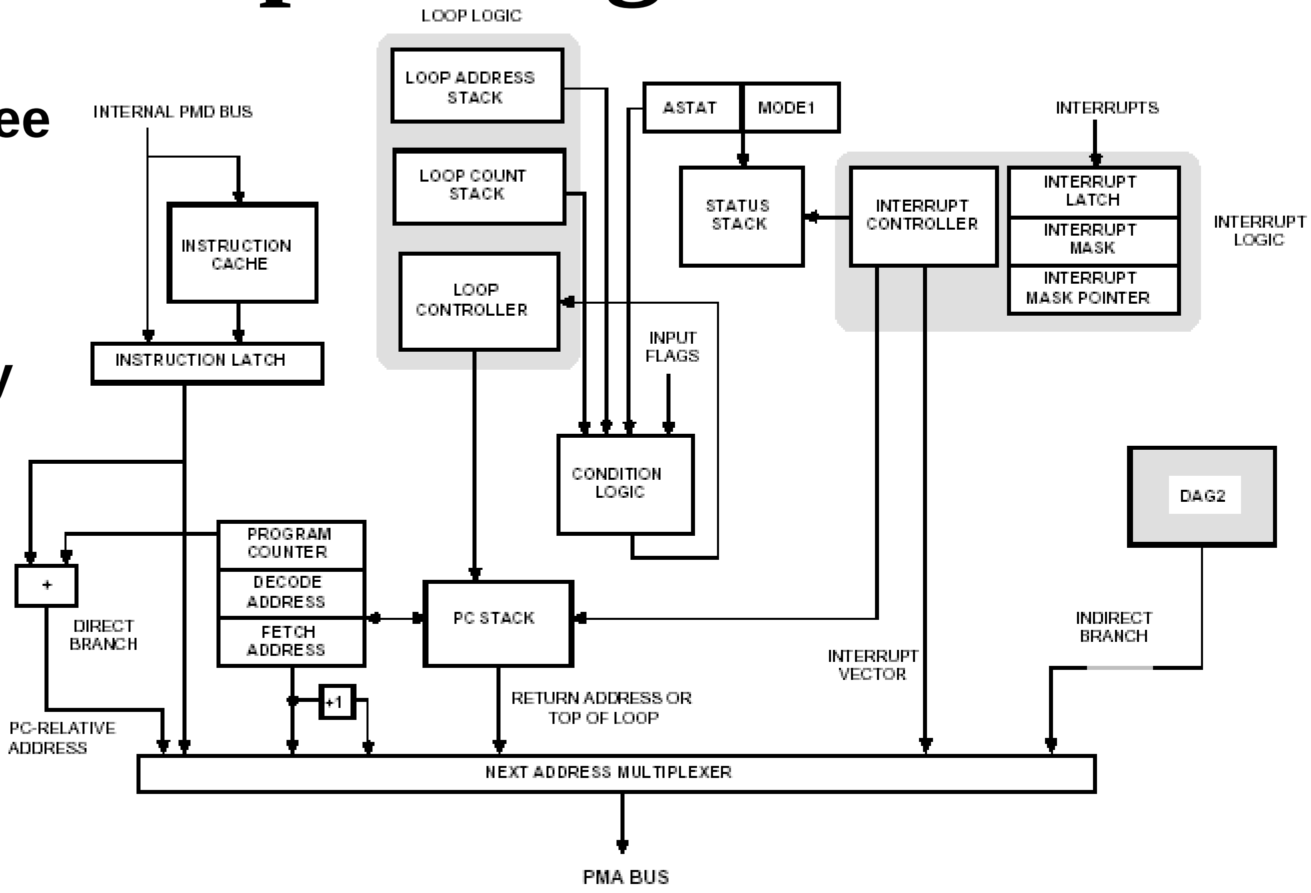
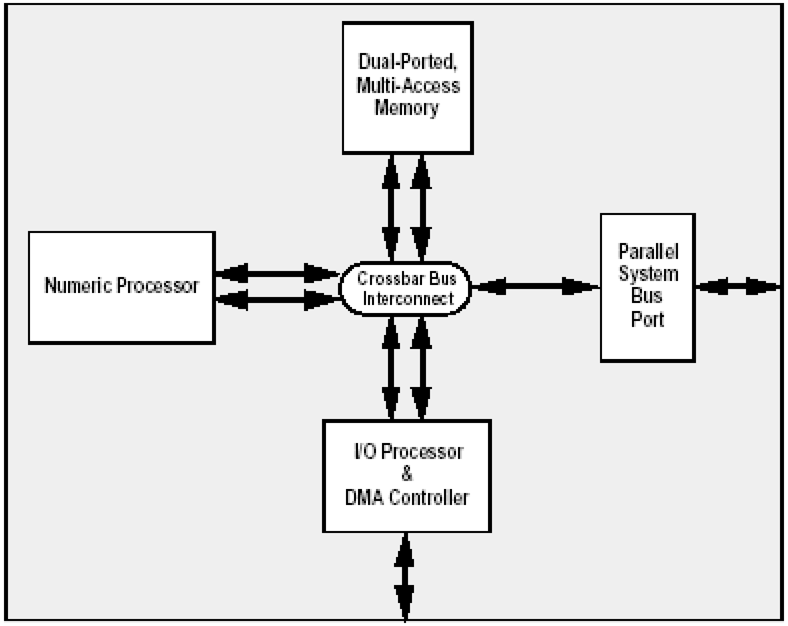


Figure 3.3 Program Sequencer Block Diagram

Bus Architecture

- Twin Bus Architecture:
  - 1 bus for Fetching Instructions
  - 1 bus for Fetching Data
- Helps avoid instruction/data conflicts
- Improves multiprocessing by allowing more steps to occur during each clock


Dual-Ported, Multi-Access Memory
Numeric Processor
Crossbar Bus Interconnect
Parallel System Bus Port
I/O Processor & DMA Controller

Data Address Generators

- There are two data address generators (DAG1 & DAG2) for addressing memory indirectly (with pre-modify or post-modify).

- Data address generator 1 (DAG1) generates 32-bit addresses on the Data Memory Address Bus.

- Data address generator 2 (DAG2) generates 24-bit addresses on the Program Memory Address Bus.

- Each DAG has four types of registers:
  - The Index (I) register acts as a pointer to memory.
  - The Modify (M) register contains the increment value for advancing the pointer.
  - Base and Limit Registers

•The base registers indicate where the page table starts in memory (this can be either a physical or logical addresses)

•limit register indicates the side of the table

Circular Buffer

- The DAGs allow circular buffer addressing.
- A circular buffer is a set of memory locations that stores data.
- An index pointer steps through the buffer.
- If the modified address pointer falls outside the buffer, the length of the buffer is subtracted from or added to the value, as required to wrap the index pointer back to the start of the buffer.
- Circular buffer addressing must use M registers for post-modify of I registers, not pre-modify.
- The Length(L) register sets the size (address range) of the circular buffer that the I register is allowed to circulate through. L must be positive or 0 (for disabled).
- The Base(B) register holds the address of the start of the circular buffer.

Bit Reversal Addressing

- Bit Reversal can be performed 2 ways:
  - Using the DAGS
  - Using the BITREV instruction
- DAG Bit Reversal
  - DAG1 reverses a 32-bit address value from register I0. This mode is enabled by the BR0 bit in the MODE1 register.
  - DAG2 reverses a 24-bit address value from register I8. This mode is enabled by the BR8 bit in the MODE1 register.
  - Bit Reversal affects both pre-modify and post-modify operations.
  - Bit Reversal affects only the outputted value not the value in the I register.

BITREV Instruction

- BITREV instruction bit reverses addresses in any I registers (I0 – I15) in either DAG.
- It performs the modification without accessing memory.
- It is independent of the DAG bit reversing mode.
- When using BITREV with DAG1, it adds a 32-bit immediate value to a DAG1 index register, reverses the result, and puts it into the DAG1 register.
- When using BITREV with DAG2, it adds a 24-bit immediate value to a DAG2 index register, reverses the result, and puts it into the DAG2 register.

Example:

`BITREV(I1,4);` ***I1 = Bit-reverse of (I1+4)***

Program Counter Stack

- The Program Counter(PC) Stack has 30 locations.
- Each location is 24 bits wide.
- Used for interrupt returns, subroutine returns, and loop terminations
- There is are Full and Empty Stack Flags in the STKY register. The Full Flag causes a maskable interrupt when TRUE.
- When the PC Stack is almost full (29 locations full) it causes an interrupt which causes a push onto the stack, filling the stack and issuing a Stack Full interrupt.
- PCSTKP is the PC Stack Pointer which contains the address to the top of the stack.
- There are other stacks: loop address stack, loop counter stack, status stack all of which have the same interrupt procedure.

Instruction Cache

- There is a 32-word instruction cache.
- It enables three-bus operation for fetching an instruction and two data values.
- Only instructions whose fetches conflict with program memory data are caches.
- More efficient than a cache that loads every instruction. Only a few instructions access data from program memory blocks.
- If instruction needed is in cache, a “cache hit” happens and the cache provides the instruction while the program memory data access is performed.
- If instruction is not in cache, the instruction fetch taken place in the next cycle and the instruction is put into the cache for next time.

Other SHARC Facts

- There are over 25 million transistors in the SHARC chip

- Power Consumption is 3.5 Watts for 5 Volts

- Software tools include a c compiler, assembler, linker, debugger, libraries, in-circuit emulator, evaluation board, and a simulator.

- Up to six SHARCs can easily be combined in a shared memory multi-processor configuration. The multi-processor interface allows for zero wait-state operation across the system bus when a SHARC is accessing the memory of another SHARC.

Resources

http://www.signal.uu.se/Staff/pd/DSP/Doc/SHARC/

http://www.phys.uu.nl/~wwwigf/sharc.htm

http://www.cs.nthu.edu.tw/~mr894363/files/lecture_2-3.pdf

http://www.bdti.com/procsum/adi060.htm

http://www.ece.utexas.edu/~bevans/courses/realtime/lectures/01_Architecture/lecture1.ppt

http://mes.loyola.edu/faculty/phs_eg769/SHARC.html

http://www.struck.de/shproc.htm

http://www-ese.fnal.gov/eseproj/trigger/prototype/sharc.pdf