2 Computation Units

Overview

The computation units of the ADSP-21160 provide the numeric processing power for performing DSP algorithms. The ADSP-21160 contains two sets of three computation units: an arithmetic/logic unit (ALU), a multiplier, and a shifter. Additionally, each set of computation units (referred to as Processing Element X, and Processing Element Y) includes a data register file. The dual processing elements provide a SIMD (Single Instruction Multiple Data) computation model capability. Both fixed-point and floating-point operations are supported by each unit. Each computation unit executes instructions in a single cycle.

A new unsigned compare instruction (COMPU) has been added to the ALU.

For SISD (Single Instruction Single Data), computations on the ADSP-21160 operate the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

IEEE Floating-Point Operations

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

Fixed Point Operations

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

Rounding

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

ALU

This feature (except for the new COMPU instruction) on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

ALU Operation

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

ALU Operating Modes

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

ALU Status Flags

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

ALU Instruction Summary

With one exception, this feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

The exception is an addition to the table in this section of the ADSP-2106x SHARC User's Manual. The addition is a new unsigned/logical, fixed point compare ALU compute operation. The COMPU allows magnitude comparison of the full 32-bit fixed point contents of the specified source operands.)

Table 2-1. New ALU Instruction

Instruction	ASTATx/y Status Flags								STKYx/y Status Flags			
Fixed-point:	AZ	AV	AN	AC	AS	AI	AF	CACC	AUS	AVS	AOS	AIS
COMPU(Rx,Ry)[1]	*[2]	0	*	0	0	0	0	*	--[3]	--	--	--

1 Rx, Ry = Any register file location; treated as unsigned fixed point.

2 * set or cleared, depending on results of instruction; For more information, see "Compute Operation Reference" on page B-1.

** may be set (but not cleared), depending on results of instruction

3 -- no effect

Multiplier

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

Shifter

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

Multifunction Computations

This feature on the ADSP-21160 operates the same as this feature on the ADSP-2106x family DSPs. For more information, see the corresponding section in the ADSP-2106x SHARC User's Manual.

Duplicate Processing Elements

The ADSP-21160 contains an additional set of computation units and register file. This additional "Processing Element" provides the additional computation resources necessary for support of "Single Instruction, Multiple Data" or SIMD operation.

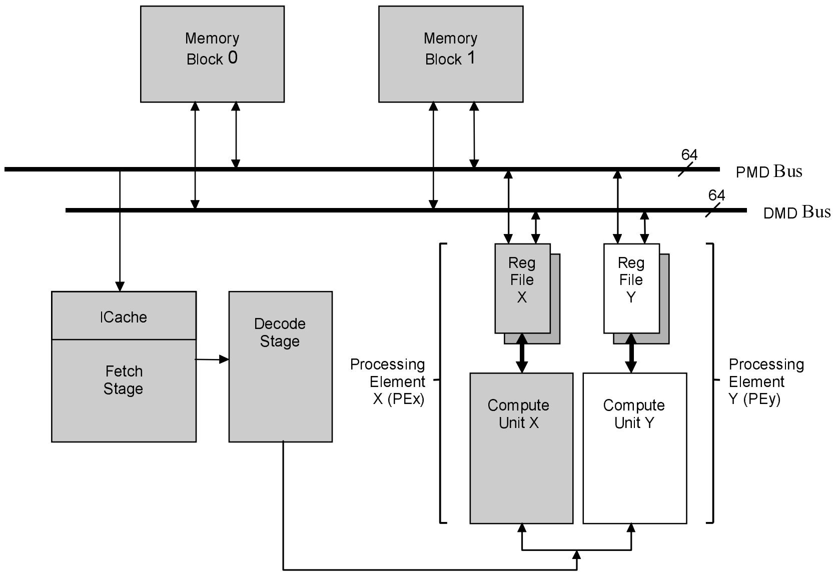

Figure 2-1. Block Diagram Showing Secondary Execution complex

Data Register Files

The register files provide the interfaces between the processor's internal data buses and the computation units. It also provides local storage for operands and results. There is one register file associated with each of the ADSP-21160's two processing elements. Each register file consists of 16 primary registers and 16 alternate (secondary) registers. All of the data registers are 40 bits wide. 32-bit data is always left-justified. If the operation specifies 32-bit data transfer, on register reads, the eight LSBs are ignored; for writes, the eight LSBs are written with zeros.

Program memory data accesses and data memory accesses to/from the register file(s) occur on the PM Data bus and DM Data bus, respectively. One PM Data bus access for each processing element and/or one DM Data bus access for each processing element can occur in one cycle. Transfers between the register files and the DM or PM data buses can move up to 64-bits of valid data on each bus.

If the same register file location is specified as both the source of an operand and the destination of a result or memory fetch, the read occurs in the first half of the cycle and the write in the second half. Thus the old data is used as the operand before the location is updated with the new result data. If writes to the same location take place in the same cycle, only the write with higher precedence actually occurs. Precedence is determined by the source of the data being written; from highest to lowest, the precedence is:

- Data memory or universal register
- Program memory
- PEx ALU
- PEy ALU
- PEx Multiplier
- PEy Multiplier
- PEx Shifter
- PEy Shifter

The individual registers of the processing element “X” (PEx) register file are prefixed with an “F” when used in floating-point computations (in assembly language source code). The PEx registers are prefixed with an “R” when used in fixed-point computations. The following instructions, for example , use the same registers:

```
F0=F1*F2;                    {floating-point multiply}
R0=R1*R2;                    {fixed-point multiply}
```

The F and R prefixes do not affect the 32-bit (or 40-bit) data transfer; they only determine how the ALU, multiplier, or shifter treat the data. The F or R may be either uppercase or lowercase; the assembler is case-insensitive.

Similarly, the individual registers of the processing element “Y” (PEy) register file maybe prefixed with an “S” (for fixed point) or “SF” (for floating point) when moving data to/from the PEy data register space. Use of this nomenclature is for programming clarity only -- PEy computation is *implicitly* specified by enabling the SIMD compute model. Use of S/SF nomenclature in computation instructions will be flagged as an error by the assembler.

Alternate (Secondary) Registers

To facilitate fast context switching, each register file has an alternate register set. Each half of each register file -- the lower halves, R0 through R7 and S0 through S7, and the upper halves, R8 through R15 and S8 through S15 -- can independently activate their alternate register sets. Two bits in the MODE1 register select the active sets. Data can be shared between contexts by placing the data to be shared in one half of either the current PE’s register file or the opposite PE’s register file and activating the alternate register set of the other halves.

Note that there is one cycle of effect latency from the instruction setting the bit in MODE1 to when the alternate registers may be accessed.