ANALOG DEVICES

Integer Arithmetic Unit

ADSP-1101

FEATURES

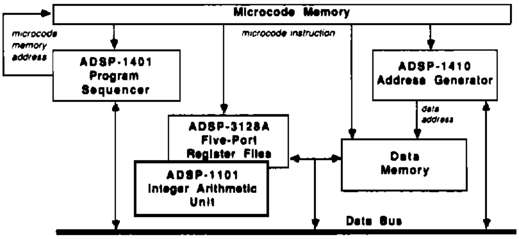
16x16-Bit Parallel Multiplication / 40-Bit Accumulation
80ns Cycle Time
Can Support 2.4ms 1024-Point Complex FFT with Block Floating-Point
40-Bit Adder/Subtracter with Status Flags
16-Bit Logic Unit
Dual 40-Bit Accumulators with Status Flags
Right/Left Shifts on Output Up to 7-Bits
Flexible Load of Six Input Registers
Flexible Preload of Both Accumulators
Feedback from Accumulators to Adder/Subtracter with Left/Right Shift Control
Feedback from Adder/Subtracter to Y Input Registers
On-Chip Block Floating-Point Control
Autonormalized Output with Exponent
Output with Saturation
32-Bits-Per-Cycle Data Transfer Rate Through Each 16-Bit Data Port (Two Input and One Output)
Twos-Complement and Unsigned-Magnitude Data Formats
Independent Microcode Control of Each Functional Unit
375mW Power Dissipation in Low-Power TTL-compatible CMOS
100-Pin Grid Array

APPLICATIONS

High-Performance Digital Signal Processing
Digital Filtering
Fourier Transformations
Correlations
General-Purpose Integer Processing
Fast Function Generation

GENERAL DESCRIPTION

The ADSP-1101 Integer Arithmetic Unit (IAU) is a versatile 16-bit integer processor which has at its core a high-speed 16x16 array multiplier, a 40-bit addition/subtraction circuit, and dual 40-bit accumulators (Figure 1). Extensive data paths and support circuitry allow its users to accomplish a broad range of integer processing tasks entirely on-chip, including complex arithmetic. The ADSP-1101 offers a full complement of arithmetic, logic, and shift functions. Block Floating-Point Control logic is also provided. Sustainable single-cycle operations of the form y=mx+b are also supported.


Microcode Memory
microcode memory address
microcode instruction
ADSP-1401 Program Sequencer
ADSP-1410 Address Generator
data address
ADSP-3128A Five-Port Register Files
ADSP-1101 Integer Arithmetic Unit
Data Memory
Data Bus

Word-Slice Microcoded System with ADSP-1101

The ADSP-1101 is ideally suited for signal processing applications such as digital filters and FFTs. Multiple ADSP-1101s can be cascaded to perform FIR filters at a single-cycle throughput rate by storing filter coefficients in input registers and passing partial sums of products to one of the Accumulators of the next IAU in the chain. The ADSP-1101 simplifies FFTs by performing six-cycle radix-2 butterfly operations entirely on-chip. Fast function generation (using Taylor/Chebyshev series, etc.) and other algorithms employing series of products can also be performed on-chip.

The ADSP-1101 has two input ports and an output port. Both of the independently controlled Y registers may be loaded from either input port. One pair of X input registers is loadable from the X-Port, the second pair, from the Y-Port. Both Accumulators may be preloaded from the Y-Port. Up to six 16-bit words can be transferred through the ADSP-1101's three data ports in a single cycle, thereby avoiding bottlenecks at the input ports and output port.

Data from the Y input registers can be passed through the Logic Unit prior to entering the Multiplier Array. The Adder/Subtracter, fed by the Multiplier Array and Accumulators, produces a result that may be routed to either one or both of the two Accumulators or to either or both Y input registers. The contents of either Accumulator can be fed back to the Adder/Subtracter or routed to the output port (via the Output Shifter). Block Floating-Point Control is implemented entirely on-chip.

The ADSP-1101's 20-bit Z-Port can output a 16-bit data word or Status Register on its lower-order 16-bits and extension data, status flags, or an exponent from an autonormalized output on its high-order 4 bits. (Adder/Subtracter flags are specified only 0-70°C.) The 16-bit data word can come from the Accumulator. Like the X and Y input ports, the Z-Port can transfer data at twice the clock rate.

The ADSP-1101's 39-bit instruction word is divided into subfields that allow independent control of the IAU's various functional elements. Instruction subfields which don't change can be hardwired, thus conserving microcode memory. A number of instructions may be conditioned on internal or external status.

The ADSP-1101 is fabricated in double-metal 1.5 µm CMOS and consumes 375mW maximum, significantly less than comparable bipolar solutions. The differential between the chip's junction temperature and the ambient temperature stays small because of this low power dissipation. Thus, unlike similar bipolar devices, the ADSP-1101 can be safely specified for operation at environmental temperatures over its extended temperature range (–55°C to +125°C ambient).

The ADSP-1101 is available for both commercial and extended temperature ranges. Extended temperature range parts are available processed fully to MIL-STD-883, Class B. The ADSP-1101 is available packaged in a ceramic 100-lead pin grid array.

TABLE OF CONTENTS

GENERAL DESCRIPTION 5-73
PIN DESCRIPTIONS 5-74
INSTRUCTION ORGANIZATION AND TIMING 5-76
INPUT BUFFERS AND TIMING 5-77
DATA FORMATS 5-79
ACCUMULATOR FEEDBACK CONTROL 5-80
ARITHMETIC AND LOGIC CONTROL 5-81
Overview 5-81
Multiplication Instructions 5-82
Non-Multiplication Instructions 5-84
ACCUMULATOR WRITE CONTROL 5-85
SHIFT CONTROL 5-85
BLOCK FLOATING-POINT 5-89
Block Floating-Point Example 5-90
AUTONORMALIZATION 5-91
SATURATION 5-91
OUTPUT CONTROL AND TIMING 5-92
STATUS FLAGS AND REGISTERS 5-92
DESIGN CONSIDERATIONS:
POWER SUPPLY DECOUPLING 5-93
OUTPUT DISABLE AND ENABLE 5-93
SPECIFICATIONS 5-94
TIMING DIAGRAMS 5-96
PINOUT 5-98
INSTRUCTION SET SUMMARY 5-99

PIN DESCRIPTIONS

PIN NAME	DESCRIPTION
DATA PORTS	
X_{15-0}	16-bit X Input Data
Y_{15-0}	16-bit Y Input Data
Y_{7-0}	Pass Magnitude Register and Shift Control Register Preload
Y_{3-0}	Shift Control Register Preload
Y_{3-0}	Bit Growth Register Preload
Z_{19-0}	20-bit Z Output Data and Status Registers
Z_{19-16}	4-bit Exponent of Autonomalized Output
Z_{19}	Accumulator A Overflow (OVFLA)
Z_{18}	Adder/Subtracter Zero (ZEROAS)
Z_{17}	Adder/Subtracter Overflow (OVFLAS)
Z_{16}	Accumulator B Overflow (OVFLB)
INSTRUCTION PORT	
IEXT	External Condition Flag
I_{38-33}	X-Buffer Input Control (XBUF)
I_{32-27}	Y-Buffer Input Control (YBUF)
I_{26-24}	Accumulator Feedback Control (FDBK)
I_{23-16}	Arithmetic/Logic Functions (ARITHL)
I_{15-12}	Accumulator A Write Control (ACCA)
I_{11-8}	Accumulator B Write Control (ACCB)
I_{7-0}	Output, BFP, and Shift Control (OUT)
STATUS FLAG	
SIGNAS	Adder/Subtracter Sign
MISCELLANEOUS	
CLK	Clock
GND	Ground (3 lines)
V_{dd}	+5V Power Supply (3 lines)

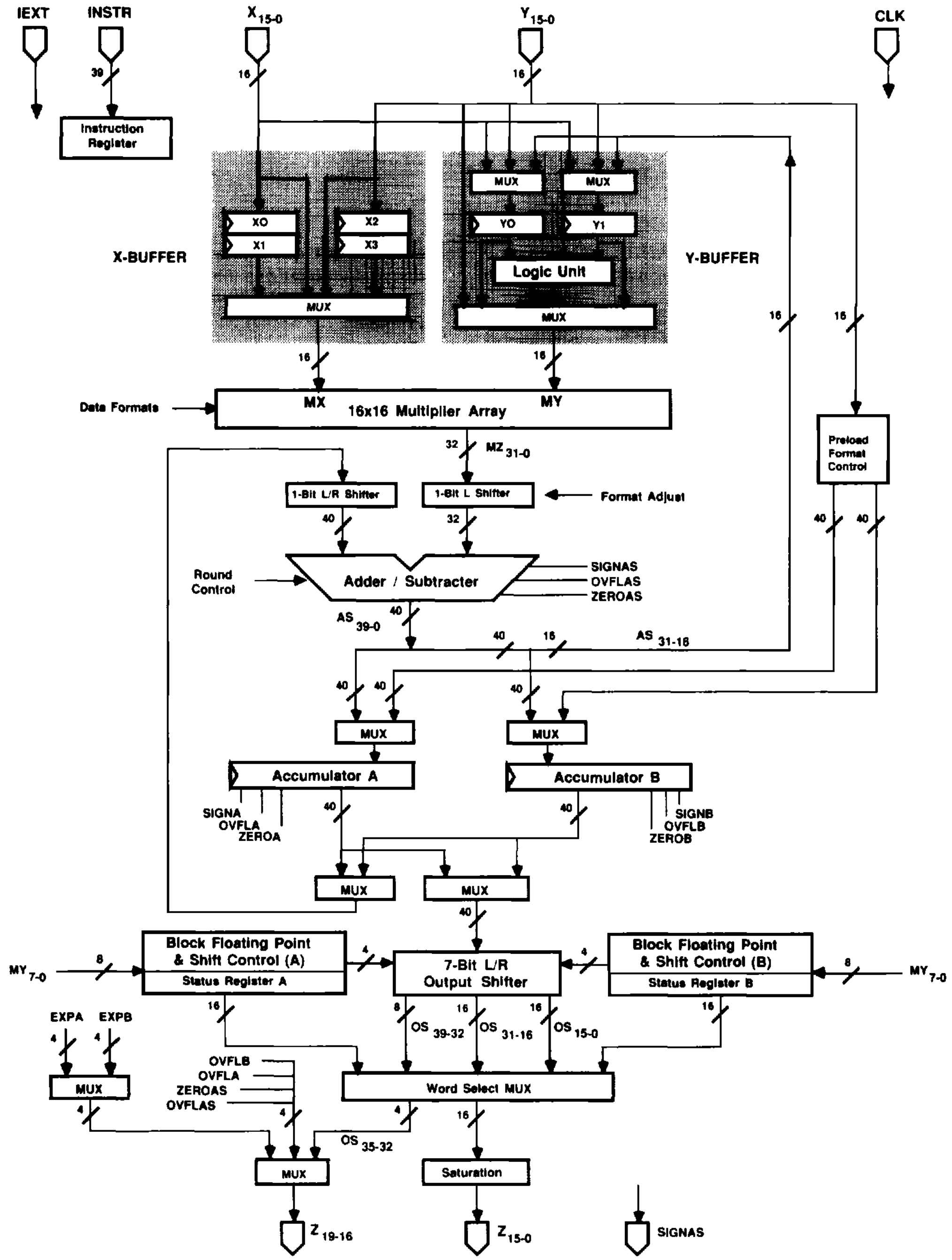


Figure 1. ADSP-1101 Functional Block Diagram

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)
External Condition	X-Buffer	Y-Buffer	Accumulator Feedback	Arithmetic / Logic Functions	Accumulator A write	Accumulator B write	Output, BFP, & Shift

Figure 2. ADSP-1101 Instruction Word Organization

INSTRUCTION ORGANIZATION AND TIMING

The ADSP-1101 features a highly orthogonal instruction set. Its 39 instruction bits are fielded to afford independent control over the key functional blocks of the Integer Arithmetic Unit (Figure 2). As a consequence, the ADSP-1101 can be treated as a single-chip integer processing subsystem. If all the flexibility of the Integer Arithmetic Unit is not required, a user can reduce the width of the instruction word coming from microcode memory by tying unchanging instruction pins to GND or +5V.

Instruction pins are numbered from I_{38} to I_0. To aid in the readability, they are referenced in this data sheet by the relevant instruction field within the instruction word. For example, instruction pins I_{23} through I_{16} are called "ARITHL23:16" since those pins control the IAU's arithmetic and logic functions. The numerical references match the "I" pin numbers.

Several arithmetic, shift, and data-path instructions can be conditioned on the state of an internal programmable flag, IFLAG. IFLAG can reflect sign, zero, or overflow status from the Adder/Subtracter. Alternatively, IFLAG can reflect the state of an external pin (IEXT), allowing for externally controlled conditional instructions. By depending on IFLAG, these conditional instructions can be made dependent on IEXT or on any one of these three Adder/Subtracter internal status conditions.

The ADSP-1101 contains an Instruction Register so that the user does not have to hold microcode instructions valid throughout the clock cycle. All instructions and IEXT share the same setup (t_{IS}) and hold-time (t_{IH}) requirements relative to the clock's rising edge (though not all are in fact registered into the internal Instruction Register). Control lines which select the data paths to registers Y0, Y1, Accumulator A, and Accumulator B at the clock's rising edge are asynchronous, allowing the muxes they control to establish data paths prior to the rising edge. Nonetheless, the user can treat all instructions as if they were registered since the asynchronous controls are no longer needed after their hold-time requirements have been met; any change after the clock edge will have no effect.

No instruction fields are internally pipelined, allowing the user complete control over the sequence of operations. When writing Adder/Subtracter results to an Accumulator and then outputting these results, for example, the instruction fields for arithmetic/logic operations would be presented on one rising edge of the clock and the instruction fields for writing this result to the Accumulators and outputting it would be presented on the next rising edge. (See Fig. 19) Instructions that put the output port into a high-impedance state take effect in the cycle after the rising edge at which they are presented.

Suggested mnemonics for the ADSP-1101's instructions have been chosen to be as short as possible while remaining descriptive. The meta-assembler-level instruction for a single cycle of IAU execution can consist of 18 or 19 independent mnemonics. Readability requires that each mnemonic express its operation in some intuitive manner. In most cases, the class of operations is denoted by the first few characters of each mnemonic. Because of the complex options available in the Accumulator write instruction sets, the mnemonics for these instructions have themselves been fielded. Some commonly used conventions include

Mnemonic	*Meaning*
XP	X-Port
YP	Y-Port
ACC	selected Accumulator
AS	Adder/Subtracter
A	absolute
P	plus
M	minus
N	negate or no change
S	sign extend
PRD	product
PASS	pass
L	shift left one bit
R	shift right one bit
U	unsigned-magnitude
2	twos-complement
Z	zero
D	default sign
LW	least significant word
MW	most significant word.

Data transfer operations have been represented in the format "[*source*]T[*destination*]" whenever anything shorter would be ambiguous, "T" meaning "to." Conditional instructions have been represented as "[*result if true*]E[*result if false*]," "E" meaning "else."

A summary of the ADSP-1101's instruction set can be found at the end of this data sheet.

INPUT BUFFERS AND TIMING

The ADSP-1101's X-Port and Y-Port are 16-bit input ports that provide data to input data buffers, the X-Buffer (Figure 3) and the Y-Buffer (Figure 4). The X-Buffer consists of four 16-bit registers, two feedthrough data paths, and muxes. Registers X0 and X1 accept data from the X-Port. Registers X2 and X3 accept data from the Y-Port. The Y-Buffer consists of two 16-bit registers, one feedthrough path, the Logic Unit, and muxes. Independently controlled registers Y0 and Y1 can both accept data either from the X-Port, from the Y-Port, or from the Adder/Subtracter.

The Y-Port can also provide up to 16-bits of data per clock phase to preload either or both Accumulators. The X-Buffer and the Y-Buffer provide inputs to the Multiplier Array and the Adder/Subtracter. The Logic Unit physically resides in the Y-Buffer, though is controlled primarily by Arithmetic and Logic Control (ARITHL23:16) instructions.

Registers in the X-Buffer can be written on either the rising or

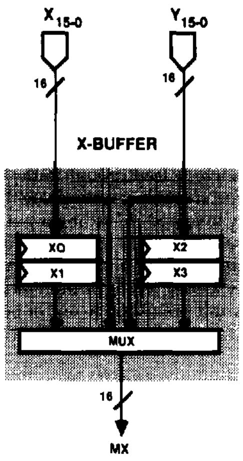
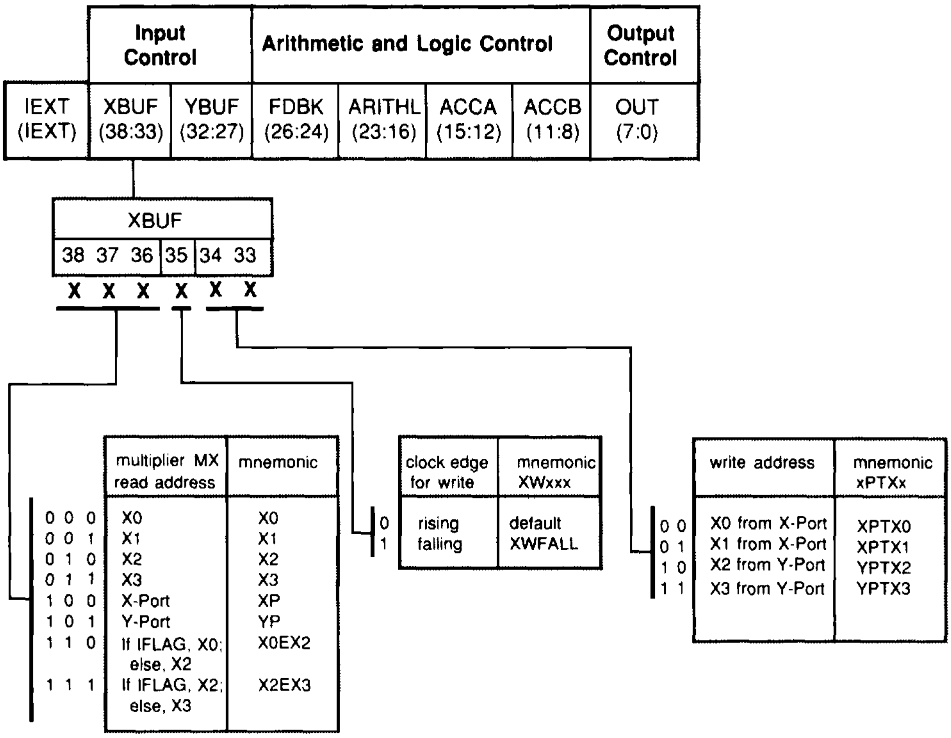


Figure 3. ADSP-1101 X-Buffer



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

XBUF					
38	37	36	35	34	33
X	X	X	X	X	X

	multiplier MX read address	mnemonic
0 0 0	X0	X0
0 0 1	X1	X1
0 1 0	X2	X2
0 1 1	X3	X3
1 0 0	X-Port	XP
1 0 1	Y-Port	YP
1 1 0	If IFLAG, X0; else, X2	X0EX2
1 1 1	If IFLAG, X2; else, X3	X2EX3

	clock edge for write	mnemonic XWxxx
0	rising	default
1	falling	XWFALL

	write address	mnemonic xPTXx
0 0	X0 from X-Port	XPTX0
0 1	X1 from X-Port	XPTX1
1 0	X2 from Y-Port	YPTX2
1 1	X3 from Y-Port	YPTX3

Table I. ADSP-1101 X-Buffer Instruction Set

falling edge of the clock, a mode that can be changed dynamically under microcode control. Y-Buffer registers, however, can be written only at the rising edge. Only one X-Buffer register can be written in a given cycle. But some X-Buffer register must be written to in every cycle. A "dummy" X -Buffer register should be designated to receive garbage X-Port data on cycles when valid data is *not* presented to the X-Port. The Y-Buffer registers, however, can be independently controlled, allowing both Y0 and Y1 to be loaded from any of three sources in the same cycle, if desired, or not loaded at all.

Both input buffers include feedthrough paths which bypass the input registers. Thus, the user can eliminate the level of pipelining normally involved in loading input data, though there is no throughput or latency advantage to doing so. Note that data loaded directly to the multiplier ports, MX or MY, can also be concurrently loaded to one or more available registers in the input buffers and preloaded to one or both Accumulators.

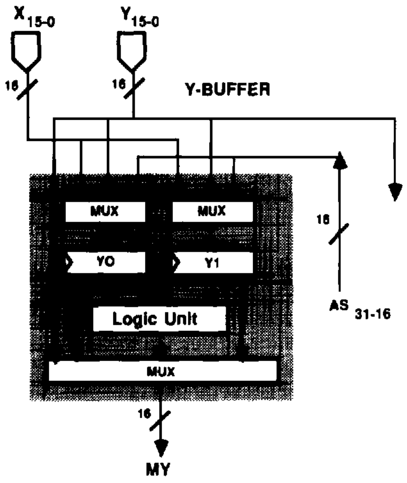


Figure 4. ADSP-1101 Y-Buffer

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

YBUF					
32	31	30	29	28	27
X	X	X	X	X	X

	multiplier MY read address	mnemonic
0 0	Y0	Y0
0 1	Y1	Y1
1 0	If IFLAG, Y0; else Y1	Y0EY1
1 1	Y-Port	YP

	Y1 write control	mnemonic xxxTY1
0 0	write from Y-Port	YPTY1
0 1	don't write	default
1 0	write from Adder/ Subtracter	ASTY1
1 1	write from X-Port	XPTY1

	Y0 write control	mnemonic xxxTY0
0 0	write from Y-Port	YPTY0
0 1	don't write	default
1 0	write from Adder/ Subtracter	ASTY0
1 1	write from X-Port	XPTY0

Table II. ADSP-1101 Y-Buffer Instruction Set

When using the feedthrough data paths, results must be clocked into one of the Accumulators before changing any input data. The data at the input ports must remain valid until t_{DHAS} before the next rising edge of the clock to insure stable results at that clock edge and also to insure stable Adder/Subtracter flags (Figure 19).

X-Buffer

The X-Buffer accepts input from either the X-Port or the Y-Port as indicated in XBUF34:33 (Table I). Note that X0 and X1 are written only from the X-Port, and X2 and X3, only from the Y-Port. XBUF35 determines whether register loading occurs on the rising edge or on the falling edge. The Multiplier Array's input source at its MX port is determined by XBUF38:36. This source can be either port or any X register. The choice of X-Buffer register can be conditional on the state of IFLAG at the beginning of the cycle.

The registers in the X-Buffer all meet the same setup (t_{DSS}) and hold time (t_{DH}) requirements regardless of whether they are clocked on the rising or falling edge. Note that when writing to X-Buffer registers on the falling edge, two additional requirements must be met. First, if an X-Buffer register is loaded mid-cycle and that register is selected as a source to the MX Multiplier Array port in that same cycle, clock LO will have to be extended to at least the minimum value of t_{CLK} as listed in "Specifications" to insure arithmetic circuits have been allowed sufficient time for propagation delays. Second, an X-Buffer register cannot be written mid-cycle on a falling edge and then re-written at the next rising edge. If attempted, the second write will not occur.

Y-Buffer

In addition to X-Port and Y-Port data sources, the Y-Buffer can also accept post-rounded bits AS_{31-16} from the Adder/Subtracter (the Adder/Subtracter's Most Significant Word) as an input (Figure 4). 16-bit results from either Accumulator can be passed through the Adder/Subtracter to load either or both of the Y-Buffer registers. Because of the feedback paths from the Adder/Subtracter to the Y-Buffer registers (and from the Accumulators to the Adder/Subtracter), these input registers can serve as auxiliary 16-bit temporary working registers. This feature is valuable in calculations involving products of sums and/or products of products.

The Y-Buffer registers accept input on the rising edge of the clock from the X-Port, the Y-Port, or the Adder/Subtracter as indicated in YBUF28:27 and YBUF30:29 (Table II). The Multiplier Array's MY port can accept data from either Y-Buffer register. Feedthrough data can come only from the Y-Port. The data source for the Multiplier Array's MY port is determined by YBUF32:31. The choice of Y-Buffer register can be conditional on the state of IFLAG at the beginning of the cycle in which the conditional instruction is executed.

For logic operations (which employ the Logic Unit), YBUF32:31 are redefined as shown in Table IV. A logic operation is defined by ARITHL18:16 = "111" (non-multiplication instruction) and ARITHL23:21 = "111" (logic instruction). The Logic Unit's operands always come from Y0 and Y1. Thus, the source for the logic operation as the Y-Buffer registers is implicit in the very fact that a logic operation is being executed. Hence, in a logic operation there would be no need to use YBUF32:31 to specify the Logic Unit's data source, and these instruction bits can be (and are) reused.

DATA FORMATS

The ADSP-1101 Integer Arithmetic Unit can process twos-complement, unsigned-magnitude, or mixed-mode fixed-point data in multiplication operations. The data formats for twos-complement and unsigned-magnitude input data are shown in Figure 5. The variable "k" determines the user's placement of the implicit binary point, which can be placed wherever desired. Integers are represented when k=0. Fractional twos-complement numbers are represented when k=–15; fractional unsigned-magnitude numbers are represented when k=–16. "Mixed-mode" operations are those with one twos-complement operand and a second unsigned-magnitude operand.

WEIGHT	Sign -2^{k+15}	2^{k+14}	2^{k+13}	...	2^{k}
VALUE	i_{15}	i_{14}	i_{13}	...	i_{0}
POSITION	15	14	13	...	0

16-Bit Twos-Complement Fixed-Point

WEIGHT	2^{k+15}	2^{k+14}	2^{k+13}	...	2^{k}
VALUE	i_{15}	i_{14}	i_{13}	...	i_{0}
POSITION	15	14	13	...	0

16-Bit Unsigned-Magnitude Fixed-Point

Figure 5. ADSP-1101 Input Data Formats

Data formats for arithmetic inputs to the Multiplier Array are specified with ARITHL23:22 pins in the Arithmetic and Logic Control / Multiplication instruction set. (See Table IV. Data formats are specified at the inputs to the Multiplier Array, *not* within the Input Buffers.) The Accumulator control instructions also support multiple formats for data preloaded from the Y-Port. Internally, the IAU tracks the data format of every result. This format tracking is essential for proper shifting, saturation at output, extension of data to wider fields, and block floating-point control. Both Accumulators and the Adder/Subtracter generate flags indicating whether their most recent contents were twos-complement or unsigned-magnitude. These flags are available in the Status Registers and can be read through the Z-Port. (See "Status Flags.")

In general, if any term in a multiplication or multiplication/accumulation operation is formatted for twos-complement, its result will be flagged as a twos-complement number. Unsigned-magnitude results are obtained only from logic operations and from Multiplication Instructions when all input and Accumulator terms are unsigned-magnitude. Mixed-mode and twos-complement operations yield twos-complement results. Mixed-mode multiplications can be useful for increasing the precision of calculations, since twos-complement multiplication results normally contain a redundant sign bit. Mixed-mode is also useful for intermediate cross-terms in double-precision multiplications.

ACCUMULATOR FEEDBACK CONTROL

Many instructions controlling the Multiplier Array and the Adder/Subtracter (ARITHL23:16) make use of feedback data from one of the two 40-bit Accumulators. These 40-bit data paths are illustrated in Figure 6. The Accumulator Feedback instructions (Table III) select the feedback path for a particular arithmetic operation. Thus, any arithmetic operation referencing an Accumulator will use the Accumulator specified in the feedback instruction. This data can be shifted right or left by one bit before entering the Adder/Subtracter. Several instructions are conditional on the state of IFLAG.

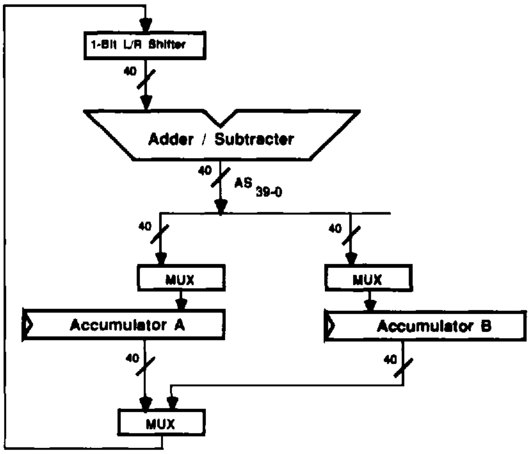


Figure 6. ADSP-1101 Accumulator Feedback Data Paths

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

FDBK		
26	25	24
X	X	X

	accumulator selection	shift selection at AS input	mnemonic FBxxx
0 0 0	AccA	no shift	FBA
0 0 1	AccB	no shift	FBB
0 1 0	If IFLAG, AccA; else AccB	no shift	FBAEB
0 1 1	If IFLAG, AccB; else AccA	no shift	FBBEA
1 0 0	AccA	If IFLAG, shift left; else no shift	FBALE
1 0 1	AccB	If IFLAG, shift left; else no shift	FBBLE
1 1 0	AccA	If IFLAG, shift right; else no shift	FBARE
1 1 1	AccB	If IFLAG, shift right; else no shift	FBBRE

Table III. ADSP-1101 Accumulator Feedback Instruction Set

Bit Position		
39 38 33 32	31 30 17 16	15 14 2 1 0
MSB EXT LSB	MSB MSW LSB	MSB LSW LSB

Figure 7. Data Fielding for Accumulators

The instruction field, FDBK26:24, determines the feedback option applicable to the current arithmetic operation. Left shifts are logical. (When Accumulator data is shifted left one bit, the Least Significant Bit (LSB) entering the Adder/Subtracter will be zero.) Right shifts are arithmetic. (When Accumulator data is shifted right one bit, the Most Significant Bit [MSB] entering the Adder/Subtracter will be sign-extended from Accumulator bit 39 in the case of twos-complement data or zero in the case of unsigned-magnitude data). These shift options are useful for effectively multiplying or dividing the contents of an Accumulator by two before adding it to or subtracting it from a product from the Multiplier Array.

ARITHMETIC AND LOGIC CONTROL

Overview

The arithmetic/logic blocks of the ADSP-1101 Integer Arithmetic Unit consist of the 16-bit Logic Unit, located in the Y-Buffer, the 16x16 Multiplier Array, and a 40-bit Adder/Subtracter. See Figure 8 for the functional arithmetic/logic blocks of the IAU. All operations using these blocks begin execution with the clock's rising edge and require t_{CLK} for completion. (When outputting twice per cycle or executing the autonormalize instruction, the clock period may have to be extended, however, to allow for the data output delay time. See "Output Control and Timing" below.) Operations are controlled primarily by the ARITHL23:16 instruction pins (Table IV). The Arithmetic and Logic Control instruction set consists of Multiplication Instructions and Non-Multiplication Instructions, as determined by ARITHL18:16.

The Logic Unit residing in the Y-Buffer performs logical operations on the contents of Y0 and Y1 and supplies the result to the MY input of the Multiplier Array.

The parallel Multiplier Array accepts 16-bit inputs through its MX port from the X-Buffer and 16-bit inputs through its MY port from either the Y-Buffer or the Logic Unit. Twos-complement, unsigned-magnitude, and mixed-mode are supported input data formats for multiplication operations. All results are internally tagged as either twos-complement or unsigned-magnitude. This format information is available in the two 16-bit Status Registers, one for each Accumulator. Unbiased rounding is supported for Multiplication Instructions

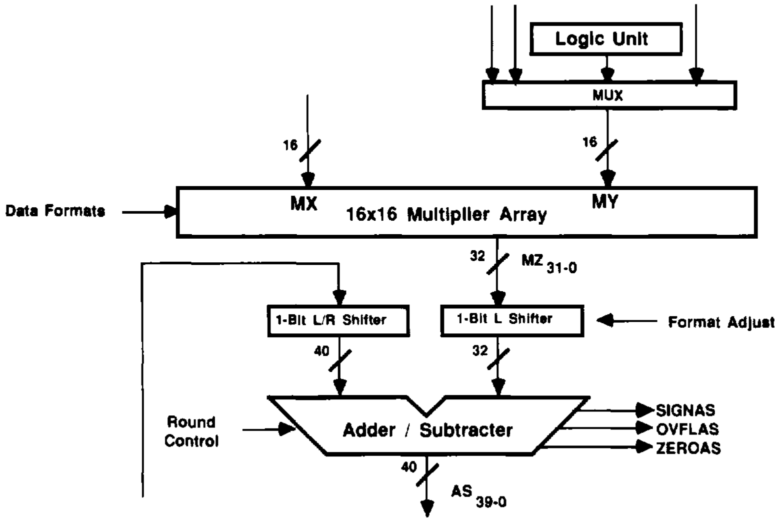


Figure 8. ADSP-1101 Logic Unit, Multiplier Array, and Adder/Subtracter

at any one of three Adder/Subtracter bit positions (AS_{16}, AS_{15}, or AS_{14}). The Multiplier Array's 32-bit product (MZ_{31-0}) can be left-shifted (logically) by one bit (Format Adjusted) to eliminate the normally redundant extra sign bit in twos-complement products before entering the Adder/Subtracter or, more generally, to scale by two.

The Adder/Subtracter accepts a 32-bit result from the Multiplier Array and a 40-bit Accumulator result from either Accumulator. The 40-bit Accumulator result can be shifted right arithmetically or left logically one bit before entering the Adder/Subtracter, as described in "Accumulator Feedback Control." The Adder/Subtracter's 40-bit output can be routed to either or both Accumulators and/or directly and asynchronously to the Output Shifter and Z-Port. The 16-bit Most Significant Word (MSW) from the Adder/Subtracter (AS_{31-16}) can also be routed to either or both registers in the Y-Buffer.

The dual 40-bit Accumulators, A and B, accept inputs from two sources: either the Adder/Subtracter or the Y-Port (via the Preload Format Control). The Accumulators are each fielded into a 16-bit Least Significant Word (LSW), a 16-bit Most Significant Word, and an 8-bit Extension (EXT) byte (Figure 7). The wide EXT byte guarantees no true data loss for at least 256 multiplication/accumulation operations. Each Accumulator can be independently written with Y-Port data. The Preload Format Control directs this data to any of the three Accumulator subfields. The other two fields in each register not receiving Y-Port data can be simultaneously cleared, sign-extended, or left unchanged. (See "Accumulator Write Control" for a complete description of the available options.)

ARITHMETIC AND LOGIC CONTROL

Multiplication Instructions

(ARITHL18:16 ≠ "111")

The Multiplication Instructions, shown in Table IV, control the IAU's multiplication and multiplication/accumulation operations. Inputs to the Multiplier Array's MX and MY ports are specified by the X-Buffer and Y-Buffer instruction fields. Input data formats for Multiplication Instructions are specified by ARITHL23:22. The Multiplier Array produces a 32-bit product (MZ_{31-0}) from these two 16-bit inputs.

For "X•Y" and "AccA/B + X•Y" instructions (ARITHL18:17="00"), the 32-bit product will be in twos-complement format if any operand is twos-complement. In Table IV, these format results are indicated by "2sC if any." If all operands are unsigned-magnitude, the product will be unsigned-magnitude. The remaining Multiplication Instructions always produce a twos-complement result. This fact is indicated by "2sC"s in Table IV for these remaining four instructions.

The 32-bit products leaving the Multiplier Array can be "format adjusted," that is, uniformly left-shifted by one bit. The Format Adjust operation is most useful for twos-complement products since they normally contain redundant sign bits in MZ_{31-30}. Unlike with first-generation array multipliers, Format Adjust on the ADSP-1101 uniformly left shifts (logical) all 32 bits one position before entering the 40-bit Adder/Subtracter. Unsigned-magnitude and mixed-mode products can also be left shifted one position using Format Adjust, an operation equivalent to multiplying by two.

Data entering the Adder/Subtracter on MZ_{31-0} is extended to 40-bits according to its data format. Twos-complement (and mixed-mode) data is signed-extended; unsigned-magnitude data is zero-extended. Because twos-complement data is sign-extended, format-adjusted full-scale negative times full-scale negative products are fully representable; there is no true data overflow in the sense of lost data. The Adder/Subtracter's overflow flag (OVFLAS) will indicate that its result has overflowed into the EXT field (AS_{39-32}) if this is the case after its operation. (See "Status Flags and Registers.")

The ADSP-1101 offers four rounding options for multiplication operations, controlled by ARITHL20:19. Rounding occurs in the Adder/Subtracter, regardless of whether it is the result of a multiplication or a multiplication/accumulation operation. The no-rounding option leaves the output from the Adder/Subtracter unaltered. The three remaining options allow unbiased rounding at one of three bit positions in the Adder/Subtracter's output field: AS_{16}, AS_{15}, or AS_{14} (Figure 9).

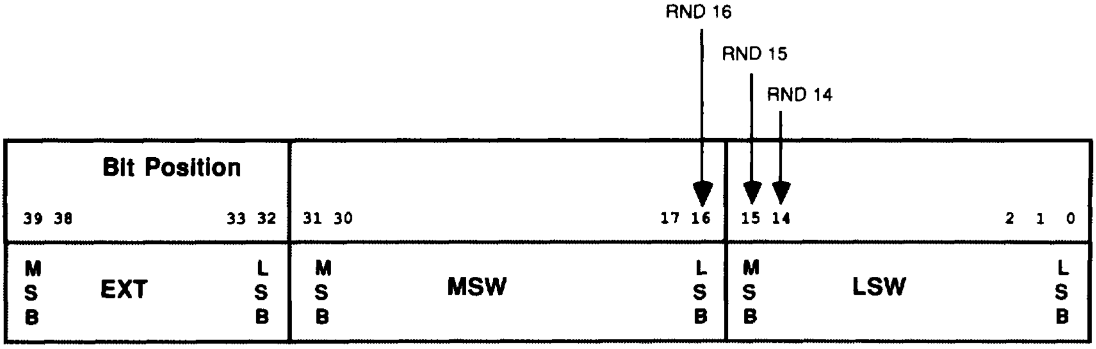
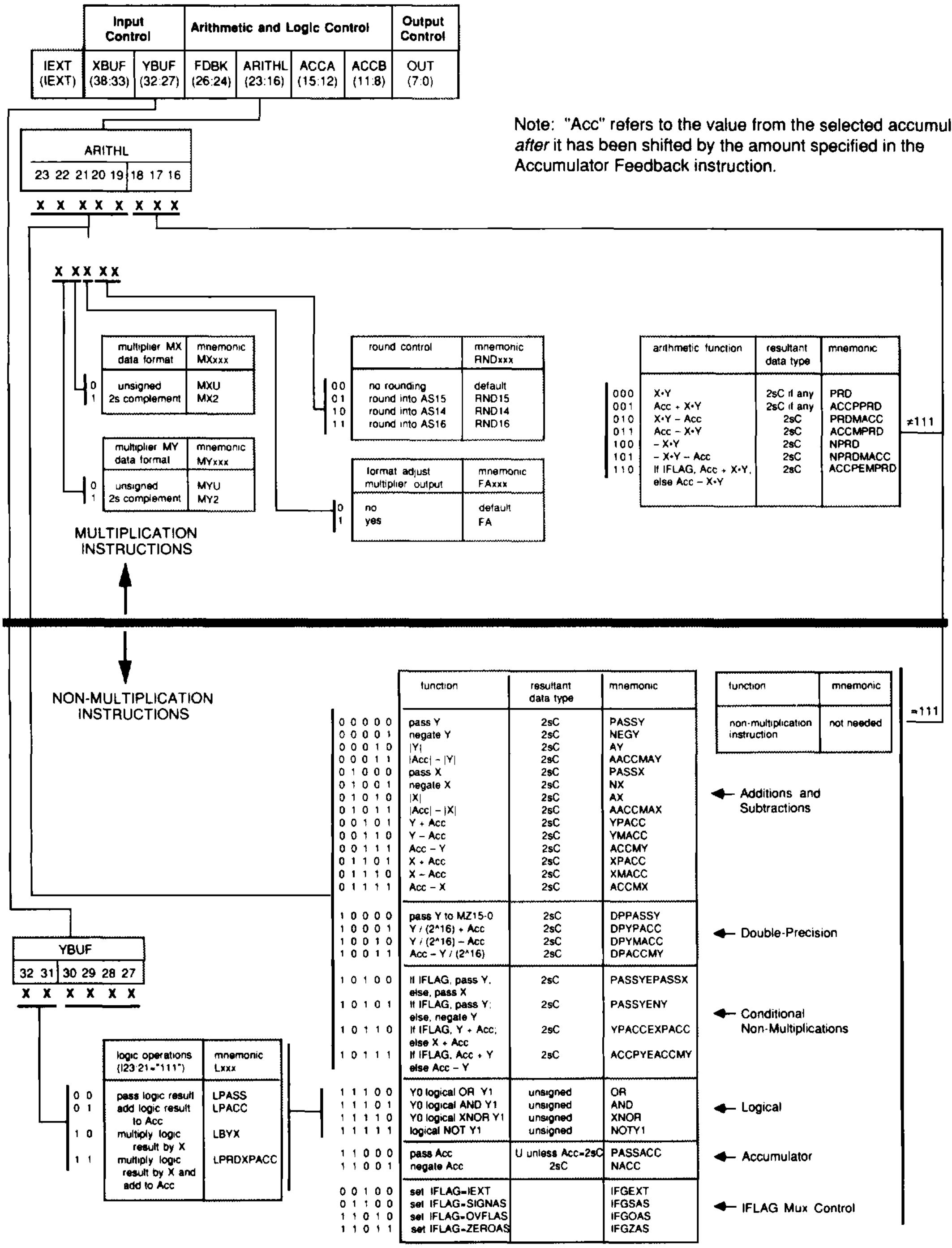


Figure 9. ADSP-1101 Rounding Positions

Note: "Acc" refers to the value from the selected accumulator *after* it has been shifted by the amount specified in the Accumulator Feedback instruction.



	multiplier MX data format	mnemonic MXxxx
0	unsigned	MXU
1	2s complement	MX2

	multiplier MY data format	mnemonic MYxxx
0	unsigned	MYU
1	2s complement	MY2

	round control	mnemonic RNDxxx
00	no rounding	default
01	round into AS15	RND15
10	round into AS14	RND14
11	round into AS16	RND16

	format adjust multiplier output	mnemonic FAxxx
0	no	default
1	yes	FA

	arithmetic function	resultant data type	mnemonic
000	X•Y	2sC if any	PRD
001	Acc + X•Y	2sC if any	ACCPPRD
010	X•Y – Acc	2sC	PRDMACC
011	Acc – X•Y	2sC	ACCMPRD
100	– X•Y	2sC	NPRD
101	– X•Y – Acc	2sC	NPRDMACC
110	If IFLAG, Acc + X•Y, else Acc – X•Y	2sC	ACCPEMPRD

	function	resultant data type	mnemonic
0 0 0 0 0	pass Y	2sC	PASSY
0 0 0 0 1	negate Y	2sC	NEGY
0 0 0 1 0	|Y|	2sC	AY
0 0 0 1 1	|Acc| – |Y|	2sC	AACCMAY
0 1 0 0 0	pass X	2sC	PASSX
0 1 0 0 1	negate X	2sC	NX
0 1 0 1 0	|X|	2sC	AX
0 1 0 1 1	|Acc| – |X|	2sC	AACCMAX
0 0 1 0 1	Y + Acc	2sC	YPACC
0 0 1 1 0	Y – Acc	2sC	YMACC
0 0 1 1 1	Acc – Y	2sC	ACCMY
0 1 1 0 1	X + Acc	2sC	XPACC
0 1 1 1 0	X – Acc	2sC	XMACC
0 1 1 1 1	Acc – X	2sC	ACCMX
1 0 0 0 0	pass Y to MZ15-0	2sC	DPPASSY
1 0 0 0 1	Y / (2^16) + Acc	2sC	DPYPACC
1 0 0 1 0	Y / (2^16) – Acc	2sC	DPYMACC
1 0 0 1 1	Acc – Y / (2^16)	2sC	DPACCMY
1 0 1 0 0	If IFLAG, pass Y, else, pass X	2sC	PASSYEPASSX
1 0 1 0 1	If IFLAG, pass Y; else, negate Y	2sC	PASSYENY
1 0 1 1 0	If IFLAG, Y + Acc; else X + Acc	2sC	YPACCEXPACC
1 0 1 1 1	If IFLAG, Acc + Y else Acc – Y	2sC	ACCPYEACCMY
1 1 1 0 0	Y0 logical OR Y1	unsigned	OR
1 1 1 0 1	Y0 logical AND Y1	unsigned	AND
1 1 1 1 0	Y0 logical XNOR Y1	unsigned	XNOR
1 1 1 1 1	logical NOT Y1	unsigned	NOTY1
1 1 0 0 0	pass Acc	U unless Acc=2sC	PASSACC
1 1 0 0 1	negate Acc	2sC	NACC
0 0 1 0 0	set IFLAG=IEXT		IFGEXT
0 1 1 0 0	set IFLAG=SIGNAS		IFGSAS
1 1 0 1 0	set IFLAG=OVFLAS		IFGOAS
1 1 0 1 1	set IFLAG=ZEROAS		IFGZAS

function	mnemonic
non-multiplication instruction	not needed

	logic operations (I23:21="111")	mnemonic Lxxx
0 0	pass logic result	LPASS
0 1	add logic result to Acc	LPACC
1 0	multiply logic result by X	LBYX
1 1	multiply logic result by X and add to Acc	LPRDXPACC

Table IV. ADSP-1101 Arithmetic and Logic Control

Unbiased rounding adds a binary one to the chosen bit position (with carry), *except* when the chosen bit position contains a "1" and all bits of lesser significance are "0." At this "mid-scale" condition, the ADSP-1101 will round to even, which may or may not imply that a "1" be added. What "round to even" means is that when the bit field bounded on the left by the chosen rounding position equals "1000...000" binary, the post-rounded, higher-order bits comprising the result field will be even, i.e. its LSB (the bit immediately to the left of the chosen rounding position) will be "0." As a consequence, in a large, statistically random sample, the IAU will round up as often as it rounds down. The processed data will not exhibit the large-sample statistical bias of +1/2 LSB of the LSW characteristic of industry-standard multipliers and multiplier/ accumulators, which *always* add a binary one to the MSB of the LSW.

Three choices of bit position make rounding consistent with 1-bit shifts left or right or with no shifting on output. If output data is not shifted, the user will normally round at AS_{15}. With a 1-bit left shift, the user will normally round at AS_{14}; with a 1-bit right shift, at AS_{16}. Note that rounding occurs subsequent to Format Adjust; the bit positions are defined in the Adder/Subtracter (AS) fields, *not* in the Multiplier Array output field (MZ).

If the user desires to round at bit positions other than AS_{16}, AS_{15}, or AS_{14}, a binary one can be added to any bit position in the MSW and LSW of the AS field using the appropriate Non-Multiplication Instruction. Of course, this rounding won't be unbiased.

ARITHMETIC AND LOGIC CONTROL

Non-multiplication Instructions

(ARITHL18:16 = "111")

The Non-Multiplication Instructions are invoked whenever ARITHL18:16 = "111" is specified (Table IV). Instruction bits ARITHL23:19 are redefined for Non-Multiplication Instructions. This class of instructions can be further subdivided into Additions-and-Subtractions, Double-Precision Instructions, Conditional Non-Multiplications, Logical Instructions, Accumulator Instructions, and IFLAG Mux Control Instructions.

Note that Additions-and-Subtractions, Double-Precision Instructions, and Conditional Non-Multiplications (Table IV) always produce twos-complement results. Logical Instructions always produce unsigned-magnitude results. Pass Accumulator produces an unsigned-magnitude result unless the last value written to the selected Accumulator was twos-complement. Negate Accumulator always produces a twos-complement result. IFLAG Mux Control Instructions produce no data results at all.

Additions-and-Subtractions and Conditional Non-Multiplications accept inputs from either Multiplier Array port (MX or MY, but not both). These 16-bit inputs are simply passed through the Multiplier Array output port to lines MZ_{31-16}. (Lines MZ_{15-0} are cleared.) They are thus scaled by 2^{16} to the MSW of the Accumulators and Adder/Subtracter. (Format Adjust is not an option in the Non-Multiplication Instruction set.) In some cases, particularly Double-Precision calculations, it is useful alternatively to scale values through the Multiplier Array to lines MZ_{15-0}, the output LSW. Four Double-Precision instructions read the value at MY into these LSW positions. Double-Precision instructions also clear lines MZ_{31-16}. The Double-Precision instructions can pass this down-scaled value from MY or add/subtractit to or from a selected Accumulator.

The four Logical Operations are indicated by ARITHL23:21 = "111" (and ARITHL18:16 = "111"). As explained above in "Input Buffers and Timing," the Y-Buffer YBUF32:31 instruction bits are not needed to specify a data source to the Logic Unit, since it always takes Y1 or both Y0 and Y1 as its sources. These YBUF instruction bits allow four permutations of the four Logical Operations: the output of the Logic Unit can be passed on to AS_{31-16} unchanged, multiplied by X, added to Accumulator A or B, or both multiplied by X and added to Accumulator A or B. (Note that these multiplications are classified as "Non-Multiplication Instructions.")

The IFLAG Mux Control Instructions determine which of four possible flags IFLAG will represent on the current and subsequent cycles (Figure 10). The four inputs to the IFLAG Mux are IEXT (the external condition flag), SIGNAS (the sign of the Adder/Subtracter), OVFLAS (the overflow flag from the Adder/Subtracter), and ZEROAS (the zero flag from the Adder/Subtracter). Once set, the path through the IFLAG Mux remains set until changed. It has no default value and must be initialized at power up.

The IFLAG multiplexer can select one of three Adder/Subtracter flags. When it does, IFLAG is updated at the end of every cycle with that cycle's Adder/Subtracter flags. The Adder/Subtracter flags can therefore be used, via IFLAG, as conditions for the next cycle's execution. For example, if IFLAG is set to SIGNAS and

X - AccA

generated a negative result, the very next instruction

If IFLAG, then AccA + Y; else AccA - Y

would yield the first result, AccA + Y, from the Adder/Subtracter. (SIGNAS is set true for negative twos-complement results.) The IFLAG multiplexer can also select the external condition, IEXT. IEXT, if selected as IFLAG, becomes the condition for the instruction with which it is set up. See "Status Flags" for a further discussion of using the ADSP-1101's various status flags.

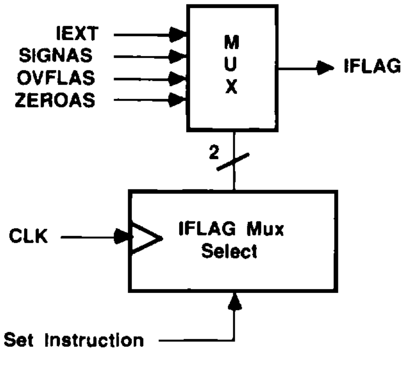


Figure 10. ADSP-1101 IFLAG Mux

ACCUMULATOR WRITE CONTROL

The ADSP-1101's dual Accumulators have independent, identical write control instruction sets. Accumulator A's Write Control Instruction set, ACCA15:12, is shown in Table V; Accumulator B's, ACCB11:8, in Table VI. They control the Accumulators and data paths as shown in Figure 11.

This instruction set controls three distinct fields within the Accumulators (EXT, MSW, and LSW), as well as an Accumulator sign flag. The two input sources to the Accumulators are the Adder/Subtracter and the Preload Format Control, which takes data from the Y-Port. Because the two instruction sets are independent, a large number of possible Accumulator Write Control combinations are possible in a single cycle. Unless otherwise indicated in Tables V and VI, the Accumulator fields are loaded on the rising edge of the clock. Fields that are loaded on the falling edge are indicated by "@falling" in these tables; to emphasize the contrast, fields in double-cycle preload instructions loaded on the rising edge are indicated by "@rising." Note that the double-cycle preload instructions are specified to match the double-cycle output instruction Table (VIII) for simple cascading of multiple IAUs. Also note that data preloaded at mid-cycle on a falling edge should not be fed back to the Adder/Subtracter input in that same cycle.

SHIFT CONTROL

A 7-Bit Left/Right Output Shifter accepts 40-bit data from the Adder/Subtracter or either Accumulator. The Output Shifter is controlled by either of two 4-bit Shift Control Registers, SCRA and SCRB (which reside in Status Registers A and B [Figure 14]). SCRA controls the shifting *on output* of values from Accumulator A. SCRB controls the shifting on output of values from Accumulator B. The Shift Control Registers are a part of the ADSP-1101's Block Floating-Point Control circuitry. However, the Shift Control Registers can be used independently to shift seven or fewer positions in either direction.

The twos-complement value in the SCR selected determines the number of positions the 40-bit field will be shifted on output. Left shifts are logical; right shifts are arithmetic. The value in Shift Control Register A (SCRA) determines the number of positions the data from Accumulator A is shifted on output. Table VII details the relationship when the Shift by SCRA Instruction is executed. Shift Control Register B (SCRB) works in exactly the same way with data output from Accumulator B. Note that even though -8 is representable, it only produces a 7-bit left shift. Additional Shift Instructions are available that can force single-bit shifts left or right, independent of the SCRs' contents.

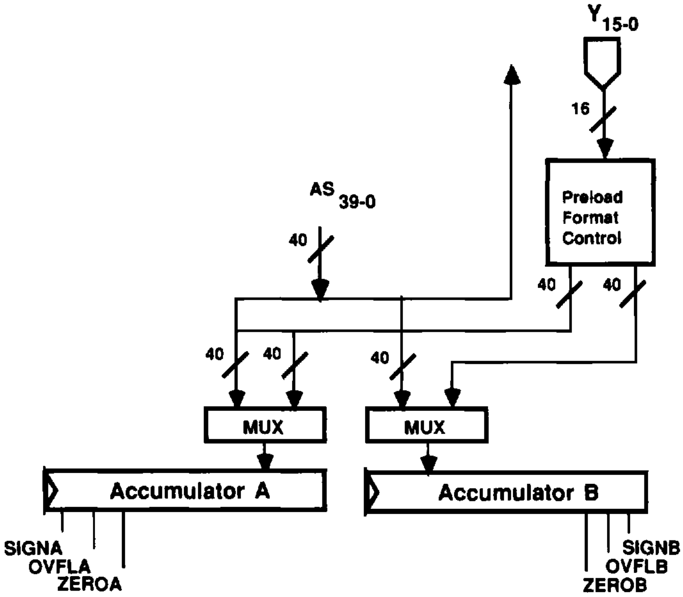


Figure 11. ADSP-1101 Accumulators

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

ACCA			
15	14	13	12
X	X	X	X

	Accumulator A 2sC flag	Accumulator A EXT load	Accumulator A MSW load	Accumulator A LSW load	mnemonic WA(2s,E,M,L)
0 0 0 0	set if AS is 2sC	AS39–32	AS31–16	AS15–0	WADASASAS
0 0 0 1	unsigned	zero	zero	Y-Port	WAUZZYP
0 0 1 0	unsigned	zero	Y-Port	no change	WAUZYPN
0 0 1 1	no change	Y-Port	no change	no change	WANYPNN
0 1 0 0	2s complement	sign extend	Y-Port	zero	WA2SYPZ
0 1 0 1	2s complement	sign extend	Y-Port	no change	WA2SYPN
0 1 1 0	unsigned	zero	Y-Port	zero	WAUZYPN
0 1 1 1	no change	no change	no change	no change	default
1 0 0 0	unsigned	zero	zero	zero	WAZERO
1 0 0 1	unsigned	zero	AS31–16	AS15–0	WAUZASAS
1 0 1 0	2s complement	sign extend	sign extend	Y-Port	WA2SSYP
1 0 1 1	no change	Y-Port @ rising	Y-Port @ falling	zero	WANYPYPZ
1 1 0 0	unsigned	zero	Y-Port @ rising	Y-Port @ falling	WAUZYPYP
1 1 0 1	2s complement	sign extend	Y-Port @ rising	Y-Port @ falling	WA2SYPYP
1 1 1 0	unsigned	zero	Y-Port @ falling	zero	WAUZYPZ
1 1 1 1	2s complement	sign extend	Y-Port @ falling	zero	WA2SYPZ

Table V. ADSP-1101 Accumulator A Write Control Instructions

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

ACCB			
11	10	9	8
X	X	X	X

	Accumulator B 2sC flag	Accumulator B EXT load	Accumulator B MSW load	Accumulator B LSW load	mnemonic WB(2s,E,M,L)
0000	set if AS is 2sC	AS39–32	AS31–16	AS15–0	WBDASASAS
0001	unsigned	zero	zero	Y-Port	WBUZZYP
0010	unsigned	zero	Y-Port	no change	WBUZYPN
0011	no change	Y-Port	no change	no change	WBNYPNN
0100	2s complement	sign extend	Y-Port	zero	WB2SYPZ
0101	2s complement	sign extend	Y-Port	no change	WB2SYPN
0110	unsigned	zero	Y-Port	zero	WBUZYPN
0111	no change	no change	no change	no change	default
1000	unsigned	zero	zero	zero	WBZERO
1001	unsigned	zero	AS31–16	AS15–0	WBUZASAS
1010	2s complement	sign extend	sign extend	Y-Port	WB2SSYP
1011	no change	Y-Port @ rising	Y-Port @ falling	zero	WBNYPYPZ
1100	unsigned	zero	Y-Port @ rising	Y-Port @ falling	WBUZYPYP
1101	2s complement	sign extend	Y-Port @ rising	Y-Port @ falling	WB2SYPYP
1110	unsigned	zero	Y-Port @ falling	zero	WBUZYPZ
1111	2s complement	sign extend	Y-Port @ falling	zero	WB2SYPZ

Table VI. ADSP-1101 Accumulator B Write Control Instructions

Independent of the SCRs, the Output Shifter can also autonormalize data from either Accumulator to a twos-complement or unsigned-magnitude MSW. (See "Autonormalization.") An Accumulator result can be autonormalized up to seven bit positions in either direction. The exponent (EXPA or EXPB) corresponding to the autonormalized MSW (from Accumulator A or B) can be simultaneously output through the Z-Port's extension field.

The user can write the two SCRs directly through the Y-Port with either the Write Status Register from Y_{7-0} Instruction or with the Write SCR from Y_{3-0}. (See Table VIII.)

SCR Value	*Accumulator Data Shift*
–8 (1000 B)	left by 7 bits (Note: not 8)
–7 (1001 B)	left by 7 bits
–6 (1010 B)	left by 6 bits
–5 (1011 B)	left by 5 bits
–4 (1100 B)	left by 4 bits
–3 (1101 B)	left by 3 bits
–2 (1110 B)	left by 2 bits
–1 (1111 B)	left by 1 bit
0 (0000 B)	no shift
1 (0001 B)	right by 1 bit
2 (0010 B)	right by 2 bits
3 (0011 B)	right by 3 bits
4 (0100 B)	right by 4 bits
5 (0101 B)	right by 5 bits
6 (0110 B)	right by 6 bits
7 (0111 B)	right by 7 bits

Table VII. Shift Control Registers' Effect on Output Shifter

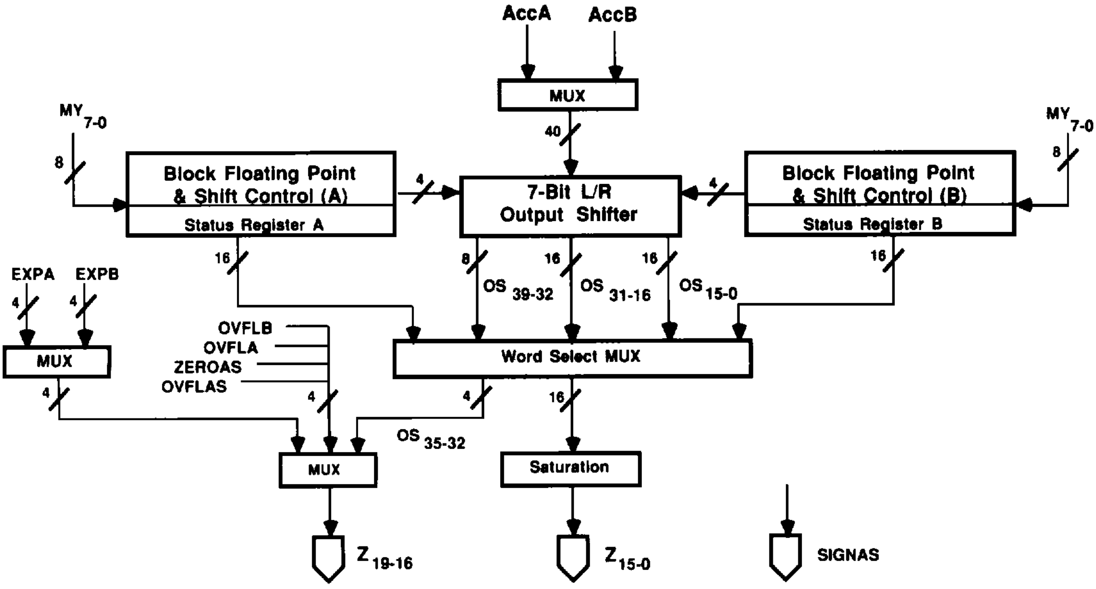
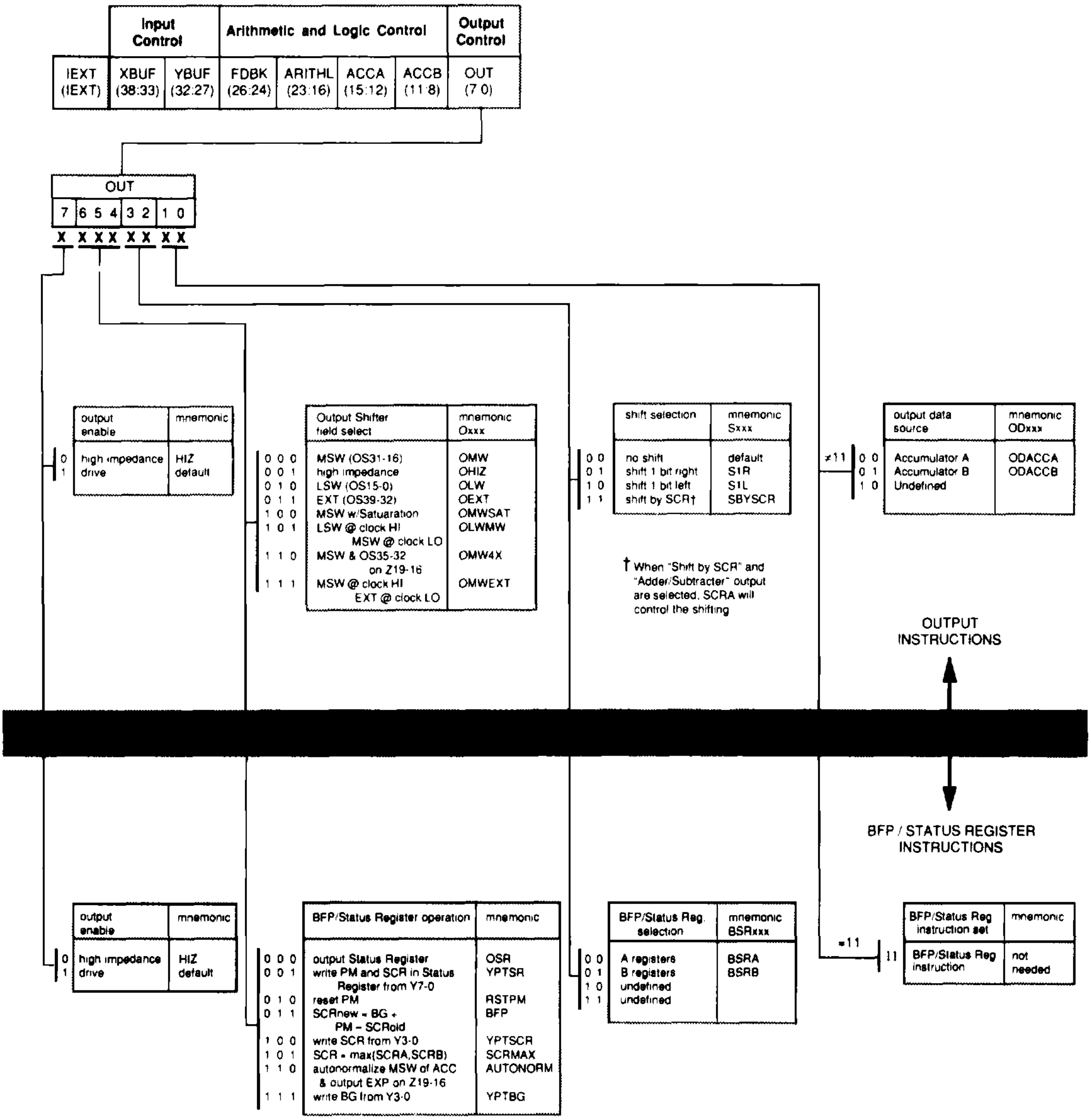


Figure 12. ADSP-1101 Shift Control, Block Floating-Point, Status Registers, Saturation, and Output Port



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

OUTPUT INSTRUCTIONS

	output enable	mnemonic
0	high impedance	HIZ
1	drive	default

	Output Shifter field select	mnemonic Oxxx
0 0 0	MSW (OS31-16)	OMW
0 0 1	high impedance	OHIZ
0 1 0	LSW (OS15-0)	OLW
0 1 1	EXT (OS39-32)	OEXT
1 0 0	MSW w/Satuaration	OMWSAT
1 0 1	LSW @ clock HI MSW @ clock LO	OLWMW
1 1 0	MSW & OS35-32 on Z19-16	OMW4X
1 1 1	MSW @ clock HI EXT @ clock LO	OMWEXT

	shift selection	mnemonic Sxxx
0 0	no shift	default
0 1	shift 1 bit right	S1R
1 0	shift 1 bit left	S1L
1 1	shift by SCR†	SBYSCR

† When "Shift by SCR" and "Adder/Subtracter" output are selected, SCRA will control the shifting

	output data source	mnemonic ODxxx
0 0	Accumulator A	ODACCA
0 1	Accumulator B	ODACCB
1 0	Undefined	

BFP / STATUS REGISTER INSTRUCTIONS

	output enable	mnemonic
0	high impedance	HIZ
1	drive	default

	BFP/Status Register operation	mnemonic
0 0 0	output Status Register	OSR
0 0 1	write PM and SCR in Status Register from Y7-0	YPTSR
0 1 0	reset PM	RSTPM
0 1 1	SCRnew = BG + PM − SCRold	BFP
1 0 0	write SCR from Y3-0	YPTSCR
1 0 1	SCR = max(SCRA,SCRB)	SCRMAX
1 1 0	autonormalize MSW of ACC & output EXP on Z19-16	AUTONORM
1 1 1	write BG from Y3-0	YPTBG

	BFP/Status Reg. selection	mnemonic BSRxxx
0 0	A registers	BSRA
0 1	B registers	BSRB
1 0	undefined	
1 1	undefined	

	BFP/Status Reg instruction set	mnemonic
11	BFP/Status Reg instruction	not needed

Table VIII. Output, Shift, Block Floating-Point, and Status Register Instructions

BLOCK FLOATING-POINT

Extensive iterative computation with fixed-point numbers can cause either an overflow of fixed-point data fields as results grow in magnitude or a loss of precision as results decrease in magnitude. Block Floating-Point (BFP) arithmetic is a method for scaling a block of fixed-point data by a common exponent to prevent either occurrence. It prevents overflow while preserving precision. It is most useful when algorithms can be structured to process data in multiple stages or "passes." Examples include the Fast Fourier Transform, Infinite Impulse Response Filters, and some matrix operations.

The ADSP-1101 contains all the circuitry necessary to implement Block Floating-Point Control on-chip. The Pass Magnitudes representable in a single pass range from 2^{-7} to 2^{7}, that is, the magnitude of data can change by up to seven binary orders of magnitude in a single pass and be handled properly. These Pass Magnitudes are consistently formatted as 4-bit twos-complement numbers. ("1000" binary ["–8" decimal] is used only to reset the Pass Magnitude registers.) The ADSP-1101's BFP circuitry works in conjunction with the 7-bit Left/Right Output Shifter (see Figure 12) to insure that changes in magnitude during a data pass cause compensating data shifts on output. These data shifts on output represent *changes* in the externally-referenced Block Floating-Point Exponent.

The IAU's Block Floating-Point circuitry includes a set of three 4-bit registers for Accumulator A: a Shift Control Register (SCRA), a Pass Magnitude (PMA) register, and a Bit Growth (BGA) register. Accumulator B has a parallel set of three 4-bit registers: SCRB, PMB, and BGB. Each set of BFP circuitry can operate entirely independently of the other set.

The SCRs control the Output Shifter described in the last section and can either be written by the user directly through the Y-Port or updated internally using the Block Floating-Point Instruction in the Block Floating-Point Instruction set (Table VIII). Like the SCRs, the Pass Magnitude (PM) registers reside as fields in the respective Status Registers (Figure 14). Note that any data written to the SCR or PM fields of the Status Registers must meed the setup time requirements for synchronous inputs, t_{DSS}. The PMs can be reset by the user to full-scale negative or written from the Y-Port, but are otherwise under the control of the IAU's internal BFP circuitry (Table VIII). (The only time a user is likely to want to write the PMs is when restoring the state of BFP processing that has been interrupted.) The BGs are written only from the Y-Port (Y_{3-0}) (Table VIII). As with the SCRs and PMs, data written to the BGs must meet setup time, t_{DSS}.

During a block of data's pass through the IAU, the Pass Magnitude register A tracks the magnitude of the largest number output from Accumulator A *as it is positioned in Accumulator A*. PMB similarly tracks the largest number output from Accumulator B. This tracking takes place on numbers *before they have been shifted by the amount specified in the SCRs*. To be tracked, however, an Accumulator value must be output.

"Magnitudes," as that word is used in this data sheet, are calculated relative to fully normalized Accumulator contents (Figure 13); a fully normalized number is defined here to have a magnitude of zero. For a twos-complement number this means both that the number has *not* overflowed into the EXT field (all EXT bits are the same) and that bit 30 differs from bit 31, the sign bit. "Full normalization" for an unsigned-magnitude number means both that it has not overflowed into the EXT field (all bits are zero) and that bit 31 is one. If an Accumulator value would be fully normalized if shifted left exactly one bit position, its magnitude is negative one. This is the exponent it would have were it fully normalized. If the Accumulator contents have overflowed the MSW into the EXT field by exactly one bit (i.e., a one-bit right shift would fully normalize the Accumulator value), its magnitude will be positive one, and so on.

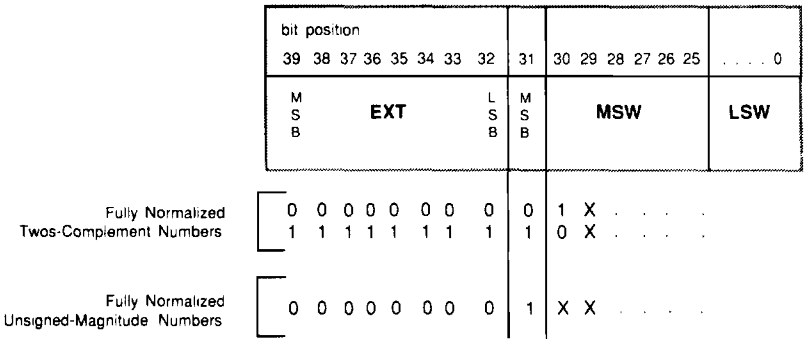

bit position
39 38 37 36 35 34 33 32
31
30 29 28 27 26 25
. . . . 0
MSB
EXT
LSB
MSB
MSW
LSW
Fully Normalized Twos-Complement Numbers
0 0 0 0 0 0 0 0 | 0 | 1 X . . . .
1 1 1 1 1 1 1 1 | 1 | 0 X . . . .
Fully Normalized Unsigned-Magnitude Numbers
0 0 0 0 0 0 0 0 | 1 | X X . . . .

Figure 13. Fully Normalized Numbers

At the beginning of a pass, the Reset PM Instruction (Table VIII) should be issued twice to set PMA and PMB both to full-scale negative ("1000" binary). Every time an Accumulator A output instruction is executed, the magnitude of the value leaving Accumulator A is compared to the current contents of the PMA register. If this magnitude is greater than that represented in PMA, then register PMA is updated to this new magnitude. Similarly, every time a value is output from Accumulator B, its magnitude is compared to the current contents of PMB and the PMB register is updated with the larger. At the end of the pass, the PM registers will contain the magnitudes of the largest values output, here called the "Pass Magnitudes." These values can then be used to calculate the output shifting required on the *next* pass. Twos-complement and unsigned-magnitude data are both handled properly in evaluating magnitude.

Output data will be shifted by the amounts specified in the Shift Control Registers. Hence, the Pass Magnitude of the output data *in external memory* will be the *differences* between the contents of the PMs and the SCRs.

To use the ADSP-1101's BFP logic, the maximum bit growth (2^{-7} to 2^{7}) possible in a given pass must have been previously calculated. This bit growth is usually apparent from the structure of the algorithm to be implemented. The value for Accumulator A's worst-case bit growth should be written to the BGA register from Y_{3-0} at the beginning of a pass. The value for Accumulator B should also be written to BGB at the beginning of a pass. (See Table VIII.) If, for example, the magnitude of data could grow by as much as 2^3 (three bit positions), the relevant BG Registers should be preloaded with "+3." These registers will retain these values until written again. Since in many algorithms these values don't change from pass to pass, the BG Registers will often only need a single initialization.

On the first pass, the SCRs should be written with the same pair of values as the BG Registers, respectively, if the input data is fully normalized. If the Pass Magnitude of the input block is other than zero (not normalized), this Pass Magnitude should be added to the Bit Growth, i.e.,

$$\text{SCR} \leftarrow \text{BG} + \text{Pass Magnitude of the input block.}$$

This insures that the data output from the first pass will not overflow.

After the first pass, the values in the Pass Magnitude registers can be used to calculate precisely the values that should be in the SCRs during the second pass to prevent overflow while retaining maximum data precision for the block. In general, the values in the PMs at the end of pass *i* can be used to calculate the SCRs for pass *i+1*. An instruction in the Block Floating-Point and Output Instruction set will automatically do this as follows for a selected set of BFP circuitry (A or B):

$$\text{SCR}_{i+1} \leftarrow \text{BG} + (\text{PM} - \text{SCR}_i).$$

The reason this update works is that ($\text{PM} - \text{SCR}_i$) represents the Pass Magnitude of the data from the ADSP-1101 *as formatted in memory* after pass *i*. If we add that Pass Magnitude to the worst-case bit growth, the data output on the *next* pass will be shifted precisely the amount required to both prevent overflow and maximize precision.

Thus, if the Pass Magnitude of data at the end of a given pass is less than the worst-case bit growth, the IAU will shift by a lesser amount on the next pass. Without this BFP circuitry, a user would have no option but to shift by the worst-case bit growth on every pass. In an algorithm with multiple passes, the ADSP-1101's Block Floating-Point logic can preserve several bits of precision in the final results, as demonstrated in the example below (Table IX). If the user desires to scale data output from both Accumulators by the same block exponent, the ADSP-1101 also provides an instruction (Table VIII) that sets SCRA or SCRB to the greater value currently in SCRA and SCRB.

The shifted output data on the final pass through the IAU will be fully normalized after that pass only if either a) both PMA=SCRA and PMB=SCRB or b) when all output results are scaled together, max(PMA,PMB)=SCRA=SCRB. If the final pass results aren't normalized, the data can be passed again through the IAU and its Output Shifter one last time to fully normalize the results (preload accumulator and output). For this post-processing normalization pass, the SCRs should be set as follows:

$$\text{SCR}_{\text{normalization pass}} \leftarrow \text{PM} - \text{SCR}_{\text{final processing pass}}$$

since there will be no bit growth on this scaling operation.

The SCRs (as well as the PMs) can be read from the Status Registers (Table VIII and Figure 14) at the end of each pass. The SCRs are useful for adjusting the externally referenced block floating-point exponent(s). Suppose that the block of fixed-point data prior to IAU processing has an exponent of X_0 and that the results of this processing will be output from Accumulator A. After the first pass, the data written back to memory will have been shifted by SCRA. Hence, the externally referenced block floating point(s) should be updated as follows:

$$X_1 \leftarrow X_0 + \text{SCRA}_0$$

or in general,

$$X_{i+1} \leftarrow X_i + \text{SCRA}_i.$$

If all data in memory are kept scaled to the same block exponent, for an N-pass algorithm,

$$X_N \leftarrow X_0 + \Sigma\,(\text{SCRA}_i).$$

Identical calculations apply to data output from Accumulator B and the SCRB register.

Block Floating-Point Example

An example may help clarify how the BFP logic in the IAU works (Table IX). Consider a radix-4 FFT with data initially scaled by block exponent, $N_0 = +2$. The basic radix-4 butterfly computes each output point by adding together 8 values. For fully normalized input data, the worst-case bit growth is therefore 3 bits. Consequently, both BGA and BGB are loaded with "three" ("0011" binary) prior to processing, in this illustration. We initialize SCRA and SCRB with the same pair of values. For simplicity here, assume that the data is all scaled by a single block exponent and remains so by setting

$$\text{SCRA}_{i+1} \leftarrow \max\,(\text{SCRA}_i, \text{SCRB}_i)$$

and

$$\text{SCRB}_{i+1} \leftarrow \max\,(\text{SCRA}_i, \text{SCRB}_i)$$

after each pass. The SCR=max(SCRA,SCRB) Instruction (Table VIII) executed selecting Accumulator A and then repeated selecting Accumulator B will do this.

Note that if only Accumulator B were used in a pass, these operations would only apply to Accumulator B's BFP circuitry. If Accumulator B were never used, they would only apply to the Accumulator A BFP circuitry.

Note that the core block floating-point instruction,

$$SCR_{i+1} \leftarrow BG + (PMB - SCR_i),$$

updates the Shift Control Registers at the end of each pass in Table IX. In this example, the last-pass block of data out was fully normalized, as indicated by a zero Pass Magnitude of data out. That means that the largest datum in the block is fully normalized. More generally, data will require an additional, final pass through the IAU if fully normalized block results are desired.

The external block exponent in this example grew from 2 to 13, i.e., by 11. Without the ADSP-1101's Block Floating-Point Control, a user would have had to shift by 3 bits on every pass to insure against overflow, for a total of 18 right shifts. This would have caused a loss of 7 bits of data precision, precision preserved by the IAU.

AUTONORMALIZATION

The ADSP-1101 Integer Arithmetic Unit will also autonormalize a single datum output from a selected Accumulator (see instruction in Table VIII) up to seven bit positions in either direction. Left shifts are logical; right shifts are arithmetic. The values from Accumulator A or B are normalized and the most significant 16 bits (the MSW) are output on Z_{15-0}. Its exponent relative to its Accumulator position is output on Z_{19-16} as shown in Table X. The ADSP-1101 handles both twos-complement and unsigned-magnitude numbers in autonormalization. Autonormalization does not affect the contents of the SCRs.

If the magnitude of the value to be autonormalized is either greater than +7 or less than –7, the value will be shifted to the seven position limit. Exponents will be as indicated in Table X. Note that an exponent of –8 (like –7) corresponds to a value that has been left-shifted by seven positions.

Autonormalization on output causes a longer output delay (t_{ACOD}) than any other operation and is specified separately (Figure 19). Depending on system requirements, the IAU's clock may need to be slowed down to accommodate this extended data delay for the autonormalization operation.

SATURATION

The ADSP-1101 Integer Arithmetic Unit can optionally saturate on output an overflowed, post-shifted MSW (OS_{31-16}) from either Accumulator or from the Adder/Subtracter to full scale. Saturation circuitry can prevent wrap-around errors. (See Saturation Instruction in Table VIII.) If the sign bit or any significant bits have overflowed to Output Shifter lines OS_{39-32}, the output is considered to have overflowed. Note that the Saturation logic operates on data leaving the Output Shifter.

Pass	*External Block Exponent Data In* [assumed in example]	*Pass Magnitude of Results In Acc (PM)* [assumed in example]	*SCR During Pass* [new SCR]	*Pass Magnitude of Data Out* [PM – SCR]	*External Block Exponent Data Out* [Extnl In + SCR]	*New SCR After Pass* [BG+(PM–old SCR)]
1	2	2	3=(BG)	–1	5 ← 2+3	2 ← 3+(2–3)
2	5	1	2	–1	7 ← 5+2	2 ← 3+(1–2)
3	7	2	2	0	9 ← 7+2	3 ← 3+(2–2)
4	9	1	3	–2	12 ← 9+3	1 ← 3+(1–3)
5	12	–2	1	–3	13 ← 12+1	0 ← 3+(–2–1)
6	13	0	0	0	13 ← 13+0	3 ← 3+(0–0)

Table IX. An Example of the IAU's BFP Control

Accumulator Contents	*Location of Most Significant Accumulator Data Bit Prior to Autonormalization*		*Bits of Shift*	*Exponent (EXP)*
	unsigned	*2sC*		
Accumulator overflowed bit position 31 by at least 7 bits	38	37	+7	7 (0111 B)
Accumulator overflowed bit position 31 by 6 bits	37	36	+6	6 (0110 B)
Accumulator overflowed bit position 31 by 5 bits	36	35	+5	5 (0101 B)
Accumulator overflowed bit position 31 by 4 bits	35	34	+4	4 (0100 B)
Accumulator overflowed bit position 31 by 3 bits	34	33	+3	3 (0011 B)
Accumulator overflowed bit position 31 by 2 bits	33	32	+2	2 (0010 B)
Accumulator overflowed bit position 31 by 1 bit	32	31	+1	1 (0001 B)
Accumulator contents are fully normalized	31	30	0	0 (0000 B)
Accumulator underflowed bit position 31 by 1 bit	30	29	–1	–1 (1111 B)
Accumulator underflowed bit position 31 by 2 bits	29	28	–2	–2 (1110 B)
Accumulator underflowed bit position 31 by 3 bits	28	27	–3	–3 (1101 B)
Accumulator underflowed bit position 31 by 4 bits	27	26	–4	–4 (1100 B)
Accumulator underflowed bit position 31 by 5 bits	26	25	–5	–5 (1011 B)
Accumulator underflowed bit position 31 by 6 bits	25	24	–6	–6 (1010 B)
Accumulator underflowed bit position 31 by 7 bits	24	23	–7	–7 (1001 B)
Accumulator underflowed bit position 31 by at least 8 bits	23	22	–7	–8 (1000 B)

Table X. Exponents (EXPA and EXPB) from Autonormalization

Thus, whether an Accumulator or Adder/ Subtracter value gets saturated will depend in part on how it is shifted. If right shifting brings the most significant 16 bits of the fully normalized value back into OS_{31-16}, the output will not be saturated; similarly, left shifts can cause a value to overflow into OS_{39-32} and thereby saturate. Since Saturation operates on post-shifted data, it affects no register contents.

The IAU handles both twos-complement and unsigned-magnitude shifted outputs. The Saturation values for conditions with both formats are shown in Table XI. Saturation on negative numbers is defined as full-scale negative plus one. This guarantees that saturated values re-entering the Multiplier Array never cause overflow (by themselves) in the Adder/Subtracter even when Format Adjusted prior to entering the Adder/Subtracter.

OUTPUT CONTROL AND TIMING

The 20-bit output Z-Port is fielded into a 16-bit result field and a 4-bit extension field, which can contain a) data bits 35 through 31 from the Output Shifter, b) the 4-bit exponent from an autonormalized output, or c) 4 status flags. The general-purpose output instructions (OUT1:0 ≠ "11") are shown in Table VIII.

The low-order 16 bits of the Z-Port (Z_{15-0}) can be put in a high-impedance state either by setting OUT7 to LO or by selecting "high impedance" with OUT6:4 (only when OUT1:0 ≠ "11"). The high-order 4 bits of the Z-Port (Z_{19-16}) will reflect the four status flags described in "Status Flags and Registers" except when executing OMW4X and AUTONORM, which use these four bits. When the 8-bit extension register is output on Z_{15-0}, it will be sign-extended according to its data type to a 16-bit field. Single-cycle double-output instructions have been chosen to coordinate with the single-cycle double-input instructions, allowing for efficient cascading of multiple IAUs. Note their timing requirements in Figure 19. Cycle times may need to be extended to accommodate the data delays of double-output operations. Shifting options are specified by OUT3:2. Output data source is selected by OUT1:0.

An output from an Accumulator will become available t_{ACOD} after the rising edge of the clock. This value had to have been computed during the previous cycle. The output instruction is presented to the IAU at the end of the processing cycle. The minimum clock cycle time is t_{CLK}. Note that is Adder/ Subtracter flags are needed off chip in the same cycle as the Adder/Subtracter operation that generates them, then the cycle time of the ADSP-1101 will have to be extended to accommodate that flag delay (t_{FDAS}).

STATUS FLAGS AND REGISTERS

Five status flags are updated every cycle and can be continuously asserted off chip. In addition to these five status flags, each Accumulator has associated with it a 16-bit Status Register.

The five status flags are:

Name	*Description*	*Output Pin*
OVFLAS	Adder/Subtracter result has overflowed into AS_{39-32}	Z_{17}
ZEROAS	Adder/Subtracter result AS_{39-0} is zero	Z_{18}
OVFLA	Accumulator A has overflowed into its EXT byte	Z_{19}
OVFLB	Accumulator B has overflowed into its EXT byte	Z_{16}
SIGNAS	Sign (AS39) of Adder/Subtracter result	SIGNAS

SIGNAS is always available at a dedicated status pin of the same name. The other four flags are generally available on Z_{19-16}. However, for two instructions the Z_{19-16} pins serve other functions: the MSW&OS35-32 Output Instruction will output the Output Shifter fields OS_{35-32} on Z_{19-16}, and the Autonormalization Instruction will output the autonormalized value's exponent on Z_{19-16}. (See instructions in Table VIII.)

OVFLAS, ZEROAS, and SIGNAS are transparent in the second half of the clock cycle (clock LO) and are latched internally in clock HI of the subsequent cycle. They become valid externally t_{FDAS} into the cycle. They therefore can be used externally for control of the operation of the next cycle. (Note that Adder/Subtracter flags are specified only over the commercial temperature range, 0-70°C. They should not be used in systems with extended temperature range requirements.) Note, however, that t_{FDAS} is greater than the minimum specified cycle times. If you use the Adder/ Subtracter flags for external control of the next cycle, you must extend the clock period to accommodate this flag delay. OVFLA and OVFLB become valid early (t_{FDAC}) in the cycle in which their respective Accumulators are written, i.e., the cycle after processing.

For unsigned-magnitude Adder/Subtracter results, SIGNAS is always cleared (LO). For twos-complement Adder/ Subtracter results, SIGNAS is set (HI) if the sign bit (AS_{39}) is one. For unsigned-magnitude Accumulator results, OVFLA and OVFLB will be cleared (LO) if the respective extension byte (bits 39-32) is all zeros. For twos-complement Accumulator results,

Overflow Condition	*OS_{31-16} after Saturation Instruction*
positive twos-complement overflow	0111111111111111 B
negative twos-complement overflow	1000000000000001 B
positive unsigned-magnitude overflow	1111111111111111 B

Table XI. ADSP-1101 Saturation Values

OVFLA and OVFLB will be set (HI) if any respective Accumulator bits 38-31 (EXT byte) differ from the sign bit, bit 39. OVFLAS is defined analogously for Adder/ Subtracter results. ZEROAS is set (HI) only when all $AS_{39\text{-}0}$ are zero.

The two Status Registers are fielded as shown in Figure 14. All 16 bits of a given Status Register can be read out $Z_{15\text{-}0}$ using the Output Status Register Instruction (Table VIII). Only the low-order 8 bits are writable. The Write Status Register Instruction (Table VIII) writes $Y_{7\text{-}0}$ to the PM and SCR fields, allowing restoration of the IAU if interrupted. The Write Shift Control Register Instruction (Table VIII) transfers data from the Y-Port ($Y_{3\text{-}0}$) to SCRA or SCRB without affecting the PM registers. Note that any data written to the Status Registers must meed the setup time requirements for synchronous inputs, t_{DSS}. Reset Pass Magnitude Register Instructions (Table VIII) force PMA or PMB to full-scale negative.

Bits 15 in the respective Status Registers are HI if the datum most recently written to Accumulator A or B, respectively, was twos-complement. Bits 14 are identical to OVFLA and OVFLB described above. Bits 13 are HI if the result most recently written to Accumulator A or B was both twos-complement and negative. Bits 12 are HI if the contents of Accumulator A or B, respectively, are zero. Bits 11-8 are defined analogously for the previous results from the Adder/Subtracter. Note that bits 11-8 in Status Register A and Status Register B are identical. Also bits 15-12 will match bits 11-8 when the Adder/Subtracter results are written to the Accumulator associated with the Status Register in question.

DESIGN CONSIDERATIONS: POWER SUPPLY DECOUPLING

The ADSP-1101 is designed with high-speed drivers on all output pins. This means that large peak currents may pass through the ground and V_{DD} pins, particularly when all output port lines are simultaneously charging their load capacitance in transition, whether from LO to HI or vice versa. These peak currents can cause a large disturbance in the ground and supply lines. For printed circuit boards, the ADSP-1101's GND and V_{DD} pins must be tied directly to solid ground and V_{DD} planes, respectively, with 0.1µf ceramic and 20µf tantalum bypass capacitors as close as possible to the tie points. Lead lengths and trace lengths should be as short as possible. The ground plane should tie to driver GND in particular with a very low inductance path.

For breadboarding with wirewrap construction, V_{DD} should be bypassed to GND with 0.1µf ceramic and 20µf tantalum capacitors. Both sets of capacitors should then be common at a point with a low impedance path to the power supply. Lead lengths should be as short as possible. This will reduce coupling of output driver current spikes into the logic supply.

OUTPUT DISABLE AND ENABLE INFORMATION

Output disable time, t_{DIS}, is measured from the time OEN reaches 1.5V to the time when all outputs have ceased driving. This is calculated by measuring the time, $t_{measured}$, from the same starting point to when the output voltages have changed by 0.5V toward +1.5V. From the tester capacitive loading, C_L, and the measured current, i_L, the decay time, t_{DECAY}, can be approximated to first order by:

$$t_{DECAY} = \frac{C_L \bullet 0.5V}{i_L}$$

from which

$$t_{DIS} = t_{measured} - t_{DECAY}$$

is calculated. Disable times are longest at the highest specified temperature.

The minimum output enable time, minimum t_{ENA}, is the earliest that outputs begin to drive. It is measured from the control signal OEN reaching 1.5V to the point at which the fastest outputs have changed by 0.1V from $V_{tristate}$ toward their final output voltages. Minimum enable times are shortest at the lowest specified temperature.

The maximum output enable time, maximum t_{ENA}, is also measured from OEN at 1.5V to the time when all outputs have reached TTL input levels (V_{OH} or V_{OL}). This could also be considered as "data valid." Maximum enable times are longest at the highest specified temperature.

bit	15	14	13	12	11	10	9	8	7-4	3-0
contents	Acc A/B 2sC?	Acc A/B OVFL?	Acc A/B sign	Acc A/B zero?	A/S 2sC?	A/S OVFL?	A/S sign	A/S zero?	PMA/B register	SCRA/B register

Figure 14. ADSP-1101 Status Registers A and B

SPECIFICATIONS [1]

RECOMMENDED OPERATING CONDITIONS

	ADSP-1101				
	J & K Grades		S & T Grades		
Parameter	Min	Max	Min	Max	Unit
V_{DD} Supply Voltage	4.75	5.25	4.5	5.5	V
T_{AMB} Operating Temperature (ambient)	0	+70	-55	+125	°C

ELECTRICAL CHARACTERISTICS

		ADSP-1101				
		J & K Grades		S & T Grades		
Parameter	Test Conditions	Min	Max	Min	Max	Unit
V_{IH} High-Level Input Voltage	@V_{DD} =max	2.0		2.2		V
V_{IHC} High-Level Input Voltage, CLK	@V_{DD} =max	2.4		2.6		V
V_{IL} Low-Level Input Voltage	@V_{DD} =min		0.8		0.8	V
V_{ILC} Low-Level Input Voltage, CLK	@V_{DD}=min		0.7		0.7	V
V_{OH} High-Level Output Voltage	@V_{DD}=min&I_{OH}=–1.0mA	2.4		2.4		V
V_{OL} Low-Level Output Voltage	@V_{DD}=min&I_{OL}=4.0mA		0.4		0.6	V
I_{IH} High-Level Input Current, All Inputs	@V_{DD}=max&V_{IN}=5.0V		10		10	μA
I_{IL} Low-Level Input Current, All Inputs	@V_{DD}=max&V_{IN}=0.0V		10		10	μA
I_{OZ} Three-State Leakage Current	@V_{DD}=max;High Z;V_{IN}=0V or max		50		50	μA
I_{DD} Supply Current	@max clock rate;TTL inputs		75		75	mA
I_{DD} Supply Current-Quiescent	All V_{IN}=2.4V		40		60	mA

SWITCHING CHARACTERISTICS[3]

	ADSP-1101								
	J Grades		K Grades		S Grades		T Grades		
	0 to 70°C		0 to 70°C		–55°C to 125°C		–55°C to 125°C		
Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Unit
t_{CLK} Clock Period	90		80		105		95		ns
t_{LO} Clock LO Period	38		35		47		44		ns
t_{HI} Clock HI Period	38		35		47		44		ns
t_{IS} Instruction Setup	19		17		23		21		ns
t_{IH} Instruction Hold	3		3		3		3		ns
t_{DSS} Synchronous Data Setup	22		20		26		24		ns
t_{DH} Synchronous Data Hold	7		7		7		7		ns
t_{DSAS} Asynchronous Data Setup	90		80		105		95		ns
t_{DHAS} Asynchronous Data Hold		5		5		5		5	ns

Parameter		J Grades		K Grades		S Grades		T Grades		
		0 to 70°C		0 to 70°C		-55°C to 125°C		-55°C to 125°C		
		Min	Max	Min	Max	Min	Max	Min	Max	Unit
t_{ACOD}	ACC&SR&EXP Output Delay	58		53		69		64		ns
t_{ACOD}	ACC&EXP Output Delay Autonormalize Instruction	65		60		73		68		ns
t_{ACH}	ACC&SR&EXP Output Hold	10		10		10		10		ns
t_{ENA}	Three-State Enable Delay	5	45	5	40	5	49	5	44	ns
t_{DIS}	Three-State Disable Delay	35		30		37		32		ns
t_{FDAC}	Accumulator Flag Delay	45		40		47		42		ns
t_{FHAC}	Accumulator Flag Hold	10		10		10		10		ns

NOTES

[1] All min and max specifications are over power-supply and temperature range indicated.

[2] S and T grade parts are available processed and tested in accordance with MIL-STD-883B. The processing and test methods used for S/883B and T/883B versions of the ADSP-1101 can be found in Analog Devices' Military Data Book.

[3] Input levels are GND and +3.0V. Rise times are 5ns. Input timing reference levels and output reference levels are 1.5V, except for 1) t_{ENA} and t_{DIS} which are as indicated in Figures 18 and 19, and 2) t_{DS} and t_{DH} which are measured from clock V_{IHA} to data input V_{IH} or V_{IL} crossing points.

Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage...........	–0.3V to 7V	Operating Temperatute Range (ambient)......	–55°C to +125°C
Input Voltage...............	–0.3V to V_{DD}	Storage Temperature Range.....................	–65°C to +150°C
Output Voltage Swing...	–0.3V to V_{DD}	Lead Temperature (10 seconds)..............................	300°C

ORDERING INFORMATION

Part Number	Temperature Range	Package	Package Outline
ADSP-1101JG	0 to +70°C	100-Pin Grid Array	G-100A
ADSP-1101KG	0 to +70°C	100-Pin Grid Array	G-100A
ADSP-1101SG	-55 to +125°C	100-Pin Grid Array	G-100A
ADSP-1101TG	-55 to +125°C	100-Pin Grid Array	G-100A
ADSP-1101SG/883B	-55 to +125°C	100-Pin Grid Array	G-100A
ADSP-1101TG/883B	-55 to +125°C	100-Pin Grid Array	G-100A

Contact DSP Marketing in Norwood concerning the availability of other package types.

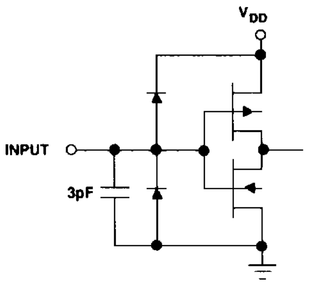


Figure 15. Equivalent Input Circuits

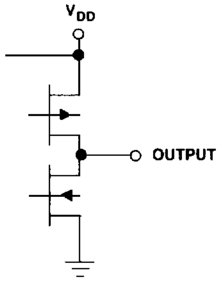


Figure 16. Equivalent Output Circuits

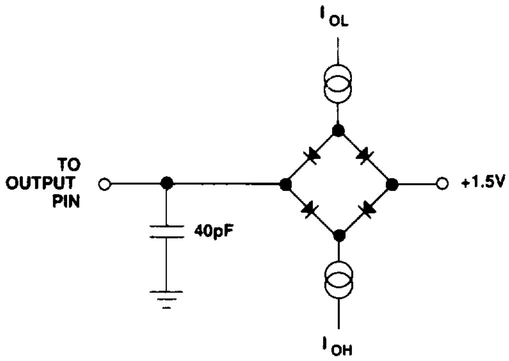


Figure 17. Normal Load for ac Measurements

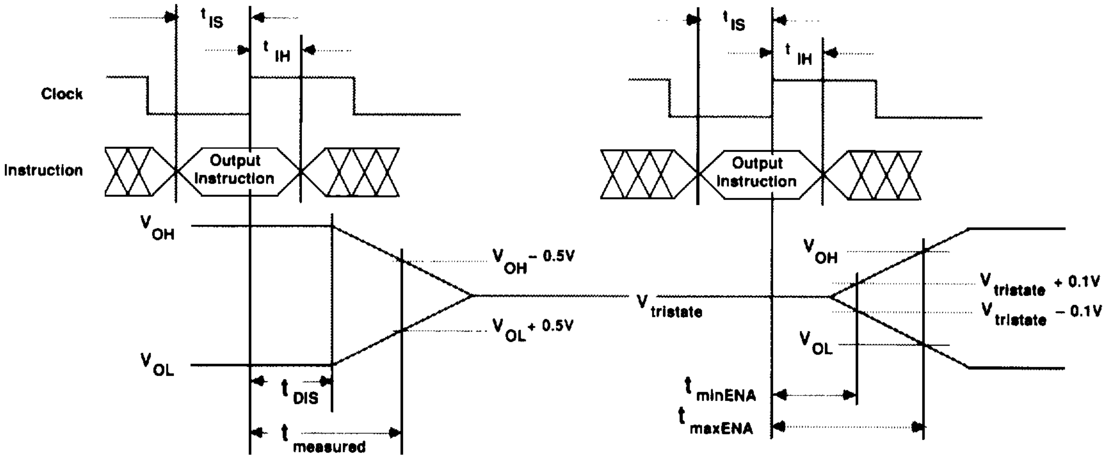


Figure 18. Output Disable and Enable Time Measurement

ESD SENSITIVITY
The ADSP-1101 features proprietary input protection circuitry to dissipate high-energy discharges (Human Body Model). Per Method 3015 of MIL-STD-883C, the ADSP-1101 has been classified as a Category A device.

Proper ESD precautions are strongly recommended to avoid functional damage or performance degradation. Charges as high as 4000 volts readily accumulate on the human body and test equipment and discharge without detection. Unused devices must be stored in conductive foam or shunts, and the foam should be discharged to the destination socket before devices are removed. For further information on ESD precautions, refer to Analog Devices' ESD Prevention Manual.

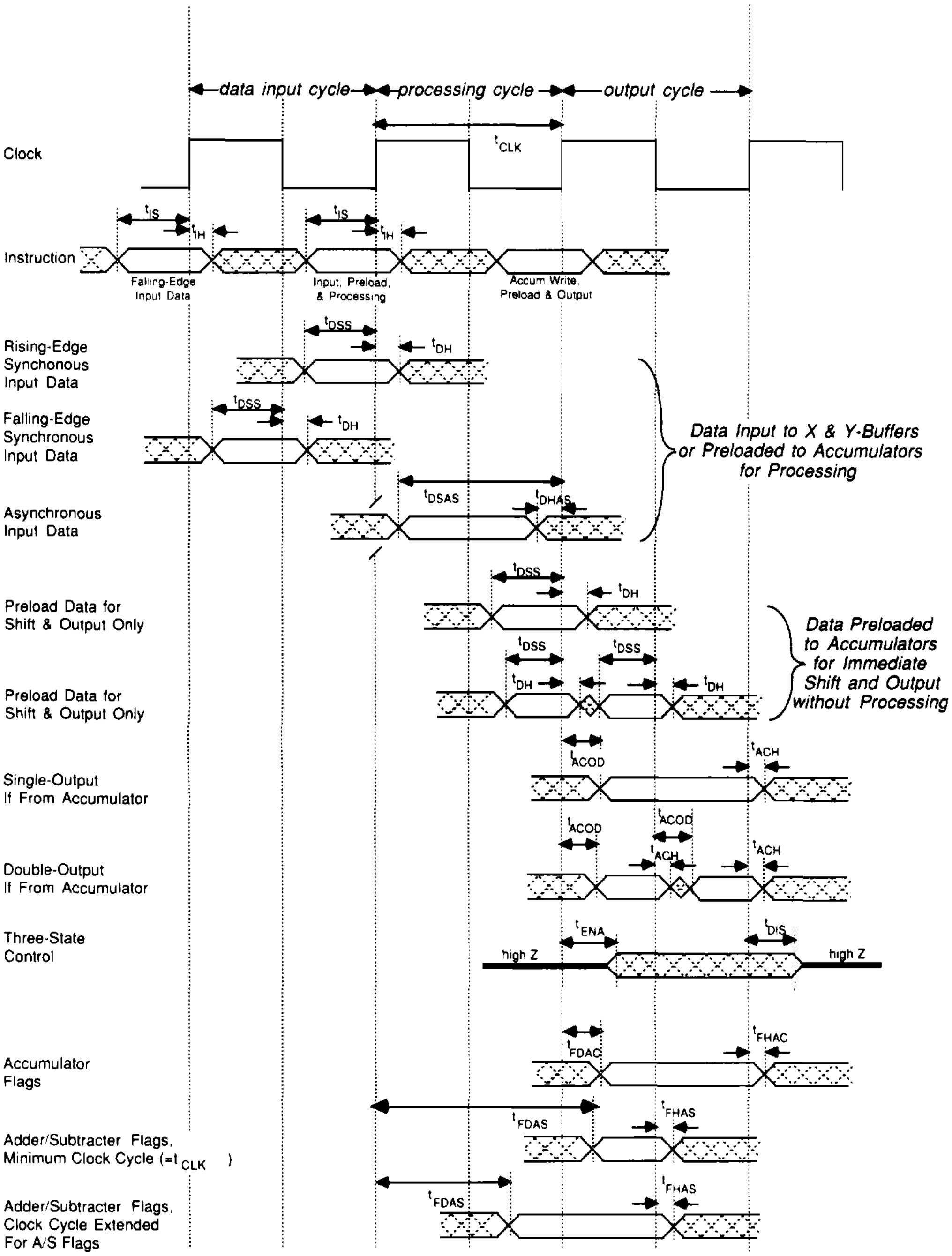
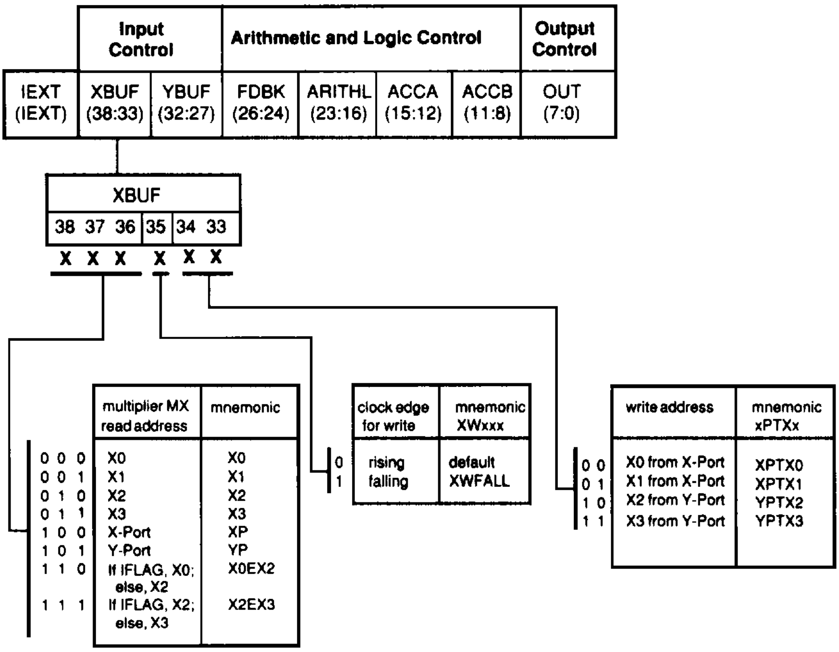

data input cycle
processing cycle
output cycle
Clock
tCLK
Instruction
tIS
tIH
Falling-Edge Input Data
Input, Preload, & Processing
Accum Write, Preload & Output
Rising-Edge Synchonous Input Data
tDSS
tDH
Falling-Edge Synchronous Input Data
Asynchronous Input Data
tDSAS
tDHAS
Data Input to X & Y-Buffers or Preloaded to Accumulators for Processing
Preload Data for Shift & Output Only
Preload Data for Shift & Output Only
Data Preloaded to Accumulators for Immediate Shift and Output without Processing
Single-Output If From Accumulator
tACOD
tACH
Double-Output If From Accumulator
Three-State Control
tENA
tDIS
high Z
high Z
Accumulator Flags
tFDAC
tFHAC
Adder/Subtracter Flags, Minimum Clock Cycle (=tCLK)
tFDAS
tFHAS
Adder/Subtracter Flags, Clock Cycle Extended For A/S Flags

Figure 19. ADSP-1101 Timing For Synchronous Accumulator Outputs

BOTTOM VIEW

	1	2	3	4	5	6	7	8	9	10	11	12	13	
N	I24	I26	I7	I5	I3	I0	Vdd	Z15	Z12	Z10	Z8	Z7	Z5	N
M	I10	I25	IEXT	I6	I4	I1	CLK	Z14	Z11	Z9	Z6	Z4	Z3	M
L	I9	I11				I2	Vdd	Z13				Z2	Z1	L
K	I15	I8										Z0	GND	K
J	I13	I14										Z17	Z18	J
H	I17	I16	I12								Z19	Z16	SIGNAS	H
G	I18	I20	I19								GND	GND	I32	G
F	I21	I22	I23								I29	I30	I31	F
E	Vdd	Y15										I27	I28	E
D	Y14	Y13										X1	X0	D
C	Y12	Y11	INDEX PIN			Y0	I35	X14				X4	X2	C
B	Y10	Y9	Y7	Y4	Y2	I38	I34	X15	X12	X10	X8	X6	X3	B
A	Y8	Y6	Y5	Y3	Y1	I37	I36	I33	X13	X11	X9	X7	X5	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	

ADSP-1101 Pin Grid Array Pinout



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

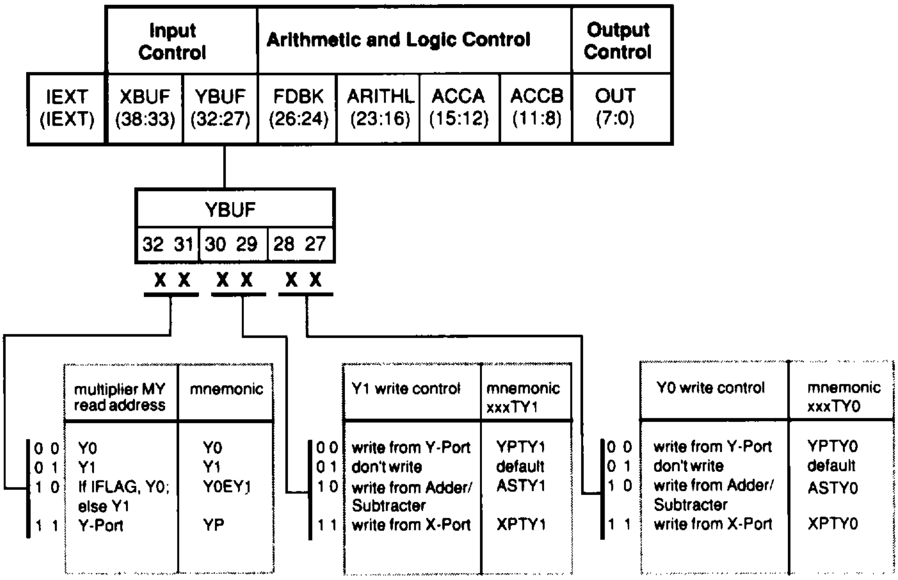
XBUF		
38 37 36	35	34 33
X X X	X	X X

	multiplier MX read address	mnemonic
0 0 0	X0	X0
0 0 1	X1	X1
0 1 0	X2	X2
0 1 1	X3	X3
1 0 0	X-Port	XP
1 0 1	Y-Port	YP
1 1 0	If IFLAG, X0; else, X2	X0EX2
1 1 1	If IFLAG, X2; else, X3	X2EX3

	clock edge for write	mnemonic XWxxx
0	rising	default
1	falling	XWFALL

	write address	mnemonic xPTXx
0 0	X0 from X-Port	XPTX0
0 1	X1 from X-Port	XPTX1
1 0	X2 from Y-Port	YPTX2
1 1	X3 from Y-Port	YPTX3

Table I. ADSP-1101 X-Buffer Instruction Set



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

YBUF		
32 31	30 29	28 27
X X	X X	X X

	multiplier MY read address	mnemonic
0 0	Y0	Y0
0 1	Y1	Y1
1 0	If IFLAG, Y0; else Y1	Y0EY1
1 1	Y-Port	YP

	Y1 write control	mnemonic xxxTY1
0 0	write from Y-Port	YPTY1
0 1	don't write	default
1 0	write from Adder/ Subtracter	ASTY1
1 1	write from X-Port	XPTY1

	Y0 write control	mnemonic xxxTY0
0 0	write from Y-Port	YPTY0
0 1	don't write	default
1 0	write from Adder/ Subtracter	ASTY0
1 1	write from X-Port	XPTY0

Table II. ADSP-1101 Y-Buffer Instruction Set

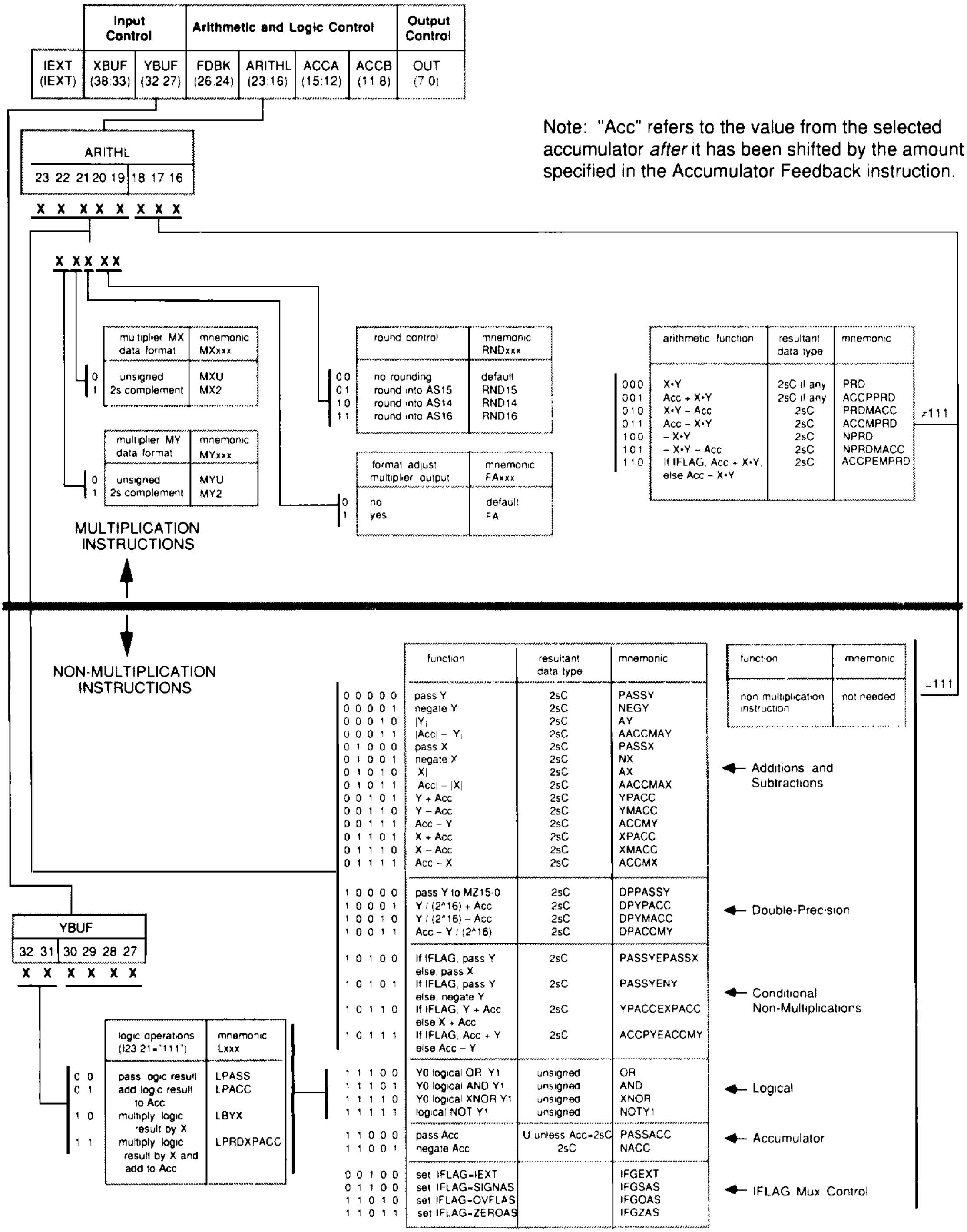
	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

FDBK
26 25 24
X X X

	accumulator selection	shift selection at AS input	mnemonic FBxxx
0 0 0	AccA	no shift	FBA
0 0 1	AccB	no shift	FBB
0 1 0	If IFLAG, AccA; else AccB	no shift	FBAEB
0 1 1	If IFLAG, AccB; else AccA	no shift	FBBEA
1 0 0	AccA	If IFLAG, shift left; else no shift	FBALE
1 0 1	AccB	If IFLAG, shift left; else no shift	FBBLE
1 1 0	AccA	If IFLAG, shift right; else no shift	FBARE
1 1 1	AccB	If IFLAG, shift right; else no shift	FBBRE

Table III. ADSP-1101 Accumulator Feedback Instruction Set

Note: "Acc" refers to the value from the selected accumulator *after* it has been shifted by the amount specified in the Accumulator Feedback instruction.



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32 27)	FDBK (26 24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11 8)	OUT (7 0)

ARITHL	
23 22 21 20 19	18 17 16
X X X X X	X X X

X XX XX

MULTIPLICATION INSTRUCTIONS

	multiplier MX data format	mnemonic MXxxx
0	unsigned	MXU
1	2s complement	MX2

	multiplier MY data format	mnemonic MYxxx
0	unsigned	MYU
1	2s complement	MY2

	round control	mnemonic RNDxxx
00	no rounding	default
01	round into AS15	RND15
10	round into AS14	RND14
11	round into AS16	RND16

	format adjust multiplier output	mnemonic FAxxx
0	no	default
1	yes	FA

	arithmetic function	resultant data type	mnemonic
000	X•Y	2sC if any	PRD
001	Acc + X•Y	2sC if any	ACCPPRD
010	X•Y – Acc	2sC	PRDMACC
011	Acc – X•Y	2sC	ACCMPRD
100	– X•Y	2sC	NPRD
101	– X•Y – Acc	2sC	NPRDMACC
110	If IFLAG, Acc + X•Y, else Acc – X•Y	2sC	ACCPEMPRD

≠111

NON-MULTIPLICATION INSTRUCTIONS

function	mnemonic
non multiplication instruction	not needed

=111

	function	resultant data type	mnemonic	
0 0 0 0 0	pass Y	2sC	PASSY	
0 0 0 0 1	negate Y	2sC	NEGY	
0 0 0 1 0	|Y|	2sC	AY	
0 0 0 1 1	|Acc| – |Y|	2sC	AACCMAY	
0 1 0 0 0	pass X	2sC	PASSX	
0 1 0 0 1	negate X	2sC	NX	
0 1 0 1 0	|X|	2sC	AX	← Additions and Subtractions
0 1 0 1 1	|Acc| – |X|	2sC	AACCMAX	
0 0 1 0 1	Y + Acc	2sC	YPACC	
0 0 1 1 0	Y – Acc	2sC	YMACC	
0 0 1 1 1	Acc – Y	2sC	ACCMY	
0 1 1 0 1	X + Acc	2sC	XPACC	
0 1 1 1 0	X – Acc	2sC	XMACC	
0 1 1 1 1	Acc – X	2sC	ACCMX	
1 0 0 0 0	pass Y to MZ15-0	2sC	DPPASSY	
1 0 0 0 1	Y / (2^16) + Acc	2sC	DPYPACC	← Double-Precision
1 0 0 1 0	Y / (2^16) – Acc	2sC	DPYMACC	
1 0 0 1 1	Acc – Y / (2^16)	2sC	DPACCMY	
1 0 1 0 0	If IFLAG, pass Y else, pass X	2sC	PASSYEPASSX	
1 0 1 0 1	If IFLAG, pass Y else, negate Y	2sC	PASSYENY	← Conditional Non-Multiplications
1 0 1 1 0	If IFLAG, Y + Acc, else X + Acc	2sC	YPACCEXPACC	
1 0 1 1 1	If IFLAG, Acc + Y else Acc – Y	2sC	ACCPYEACCMY	
1 1 1 0 0	Y0 logical OR Y1	unsigned	OR	
1 1 1 0 1	Y0 logical AND Y1	unsigned	AND	← Logical
1 1 1 1 0	Y0 logical XNOR Y1	unsigned	XNOR	
1 1 1 1 1	logical NOT Y1	unsigned	NOTY1	
1 1 0 0 0	pass Acc	U unless Acc=2sC	PASSACC	← Accumulator
1 1 0 0 1	negate Acc	2sC	NACC	
0 0 1 0 0	set IFLAG=IEXT		IFGEXT	
0 1 1 0 0	set IFLAG=SIGNAS		IFGSAS	← IFLAG Mux Control
1 1 0 1 0	set IFLAG=OVFLAS		IFGOAS	
1 1 0 1 1	set IFLAG=ZEROAS		IFGZAS	

YBUF	
32 31	30 29 28 27
X X	X X X X

	logic operations (I23 21="111")	mnemonic Lxxx
0 0	pass logic result	LPASS
0 1	add logic result to Acc	LPACC
1 0	multiply logic result by X	LBYX
1 1	multiply logic result by X and add to Acc	LPRDXPACC

Table IV. ADSP-1101 Arithmetic and Logic Control

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38.33)	YBUF (32 27)	FDBK (26 24)	ARITHL (23:16)	ACCA (15 12)	ACCB (11 8)	OUT (7 0)

ACCA			
15	14	13	12
X	X	X	X

	Accumulator A 2sC flag	Accumulator A EXT load	Accumulator A MSW load	Accumulator A LSW load	mnemonic WA(2s,E M,L)
0 0 0 0	set if AS is 2sC	AS39 32	AS31-16	AS15–0	WADASASAS
0 0 0 1	unsigned	zero	zero	Y Port	WAUZZYP
0 0 1 0	unsigned	zero	Y Port	no change	WAUZYPN
0 0 1 1	no change	Y Port	no change	no change	WANYPNN
0 1 0 0	2s complement	sign extend	Y Port	zero	WA2SYPZ
0 1 0 1	2s complement	sign extend	Y Port	no change	WA2SYPN
0 1 1 0	unsigned	zero	Y Port	zero	WAUZYPN
0 1 1 1	no change	no change	no change	no change	default
1 0 0 0	unsigned	zero	zero	zero	WAZERO
1 0 0 1	unsigned	zero	AS31-16	AS15–0	WAUZASAS
1 0 1 0	2s complement	sign extend	sign extend	Y-Port	WA2SSYP
1 0 1 1	no change	Y Port @ rising	Y Port @ falling	zero	WANYPYPZ
1 1 0 0	unsigned	zero	Y Port @ rising	Y-Port @ falling	WAUZYPYP
1 1 0 1	2s complement	sign extend	Y Port @ rising	Y-Port @ falling	WA2SYPYP
1 1 1 0	unsigned	zero	Y Port @ falling	zero	WAUZYPZ
1 1 1 1	2s complement	sign extend	Y Port @ falling	zero	WA2SYPZ

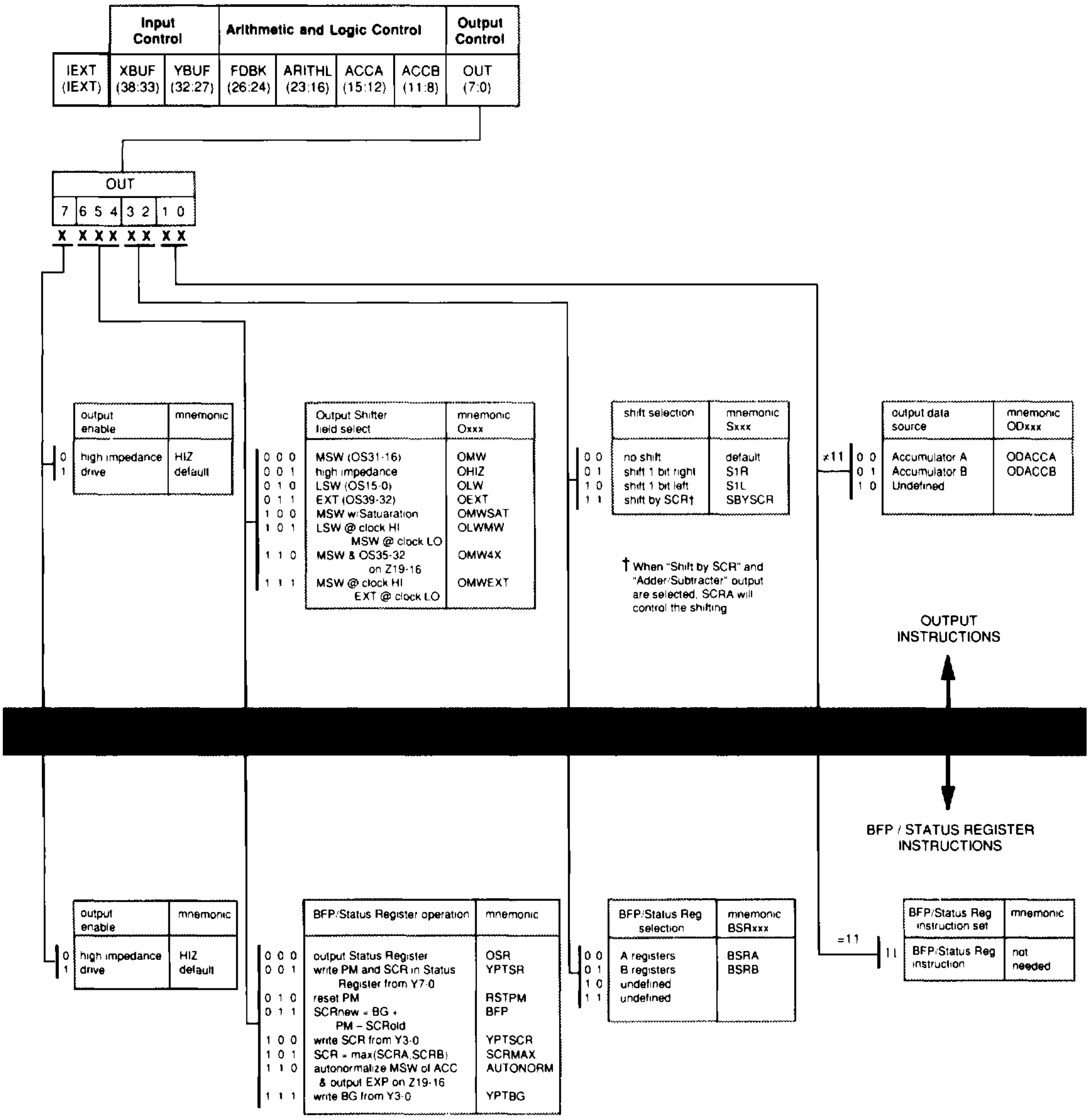
Table V. ADSP-1101 Accumulator A Write Control Instructions

	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38 33)	YBUF (32.27)	FDBK (26 24)	ARITHL (23 16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

ACCB			
11	10	9	8
X	X	X	X

	Accumulator B 2sC flag	Accumulator B EXT load	Accumulator B MSW load	Accumulator B LSW load	mnemonic WB(2s,E,M L)
0000	set if AS is 2sC	AS39–32	AS31-16	AS15–0	WBDASASAS
0001	unsigned	zero	zero	Y Port	WBUZZYP
0010	unsigned	zero	Y-Port	no change	WBUZYPN
0011	no change	Y Port	no change	no change	WBNYPNN
0100	2s complement	sign extend	Y-Port	zero	WB2SYPZ
0101	2s complement	sign extend	Y-Port	no change	WB2SYPN
0110	unsigned	zero	Y-Port	zero	WBUZYPN
0111	no change	no change	no change	no change	default
1000	unsigned	zero	zero	zero	WBZERO
1001	unsigned	zero	AS31-16	AS15-0	WBUZASAS
1010	2s complement	sign extend	sign extend	Y Port	WB2SSYP
1011	no change	Y-Port @ rising	Y Port @ falling	zero	WBNYPYPZ
1100	unsigned	zero	Y Port @ rising	Y Port @ falling	WBUZYPYP
1101	2s complement	sign extend	Y Port @ rising	Y-Port @ falling	WB2SYPYP
1110	unsigned	zero	Y-Port @ falling	zero	WBUZYPZ
1111	2s complement	sign extend	Y-Port @ falling	zero	WB2SYPZ

Table VI. ADSP-1101 Accumulator B Write Control Instructions



	Input Control		Arithmetic and Logic Control				Output Control
IEXT (IEXT)	XBUF (38:33)	YBUF (32:27)	FDBK (26:24)	ARITHL (23:16)	ACCA (15:12)	ACCB (11:8)	OUT (7:0)

OUTPUT INSTRUCTIONS

	output enable	mnemonic
0	high impedance	HIZ
1	drive	default

	Output Shifter field select	mnemonic Oxxx
0 0 0	MSW (OS31-16)	OMW
0 0 1	high impedance	OHIZ
0 1 0	LSW (OS15-0)	OLW
0 1 1	EXT (OS39-32)	OEXT
1 0 0	MSW w/Satuaration	OMWSAT
1 0 1	LSW @ clock HI MSW @ clock LO	OLWMW
1 1 0	MSW & OS35-32 on Z19-16	OMW4X
1 1 1	MSW @ clock HI EXT @ clock LO	OMWEXT

	shift selection	mnemonic Sxxx
0 0	no shift	default
0 1	shift 1 bit right	S1R
1 0	shift 1 bit left	S1L
1 1	shift by SCR†	SBYSCR

† When "Shift by SCR" and "Adder/Subtracter" output are selected, SCRA will control the shifting

≠11

	output data source	mnemonic ODxxx
0 0	Accumulator A	ODACCA
0 1	Accumulator B	ODACCB
1 0	Undefined	

BFP / STATUS REGISTER INSTRUCTIONS

	output enable	mnemonic
0	high impedance	HIZ
1	drive	default

	BFP/Status Register operation	mnemonic
0 0 0	output Status Register	OSR
0 0 1	write PM and SCR in Status Register from Y7-0	YPTSR
0 1 0	reset PM	RSTPM
0 1 1	SCRnew = BG + PM – SCRold	BFP
1 0 0	write SCR from Y3-0	YPTSCR
1 0 1	SCR = max(SCRA,SCRB)	SCRMAX
1 1 0	autonormalize MSW of ACC & output EXP on Z19-16	AUTONORM
1 1 1	write BG from Y3-0	YPTBG

	BFP/Status Reg selection	mnemonic BSRxxx
0 0	A registers	BSRA
0 1	B registers	BSRB
1 0	undefined	
1 1	undefined	

=11

	BFP/Status Reg instruction set	mnemonic
11	BFP/Status Reg instruction	not needed

Table VIII. ADSP-1101 Output, Shift, Block Floating-Point, and Status Register Instructions