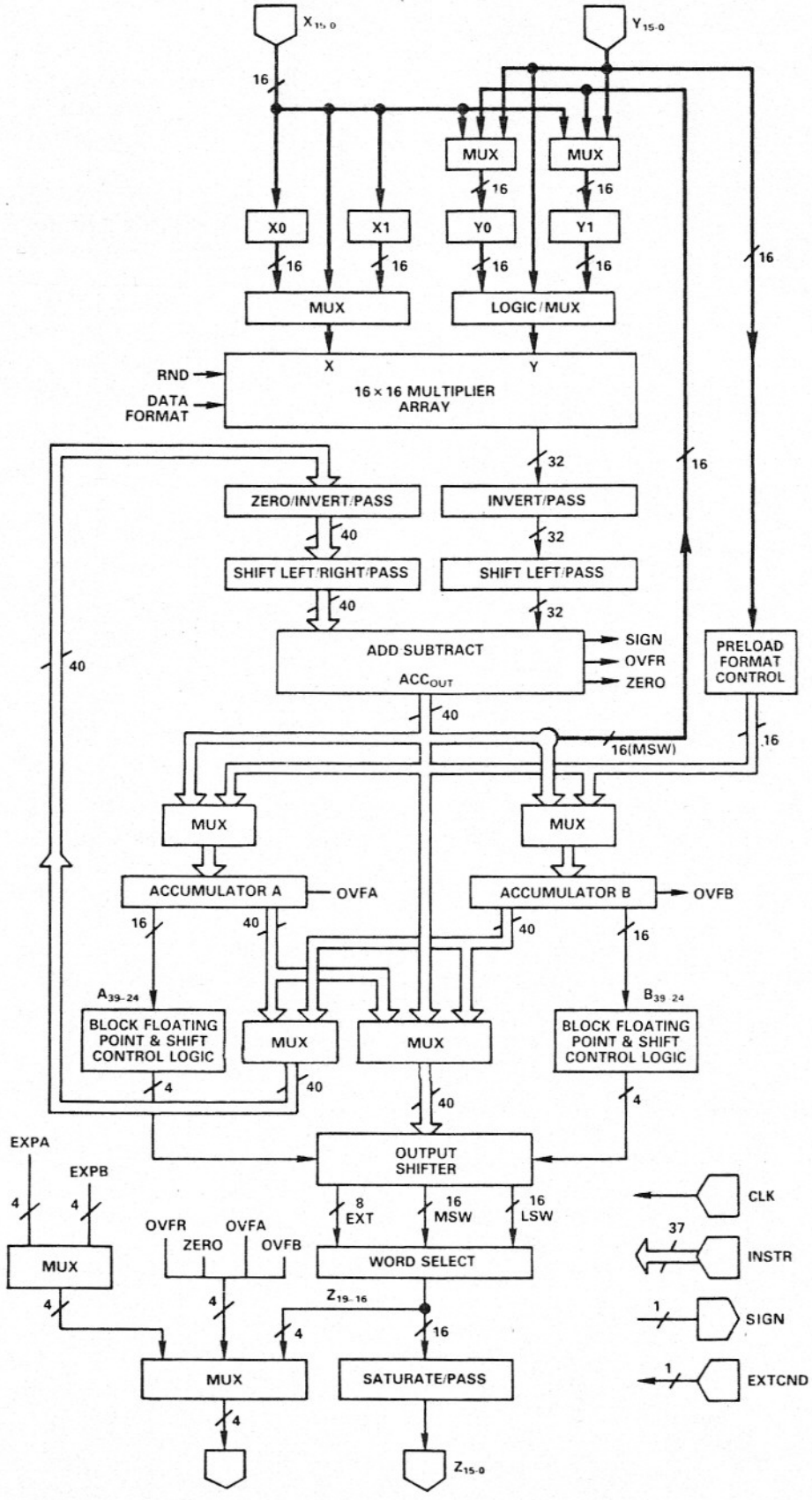
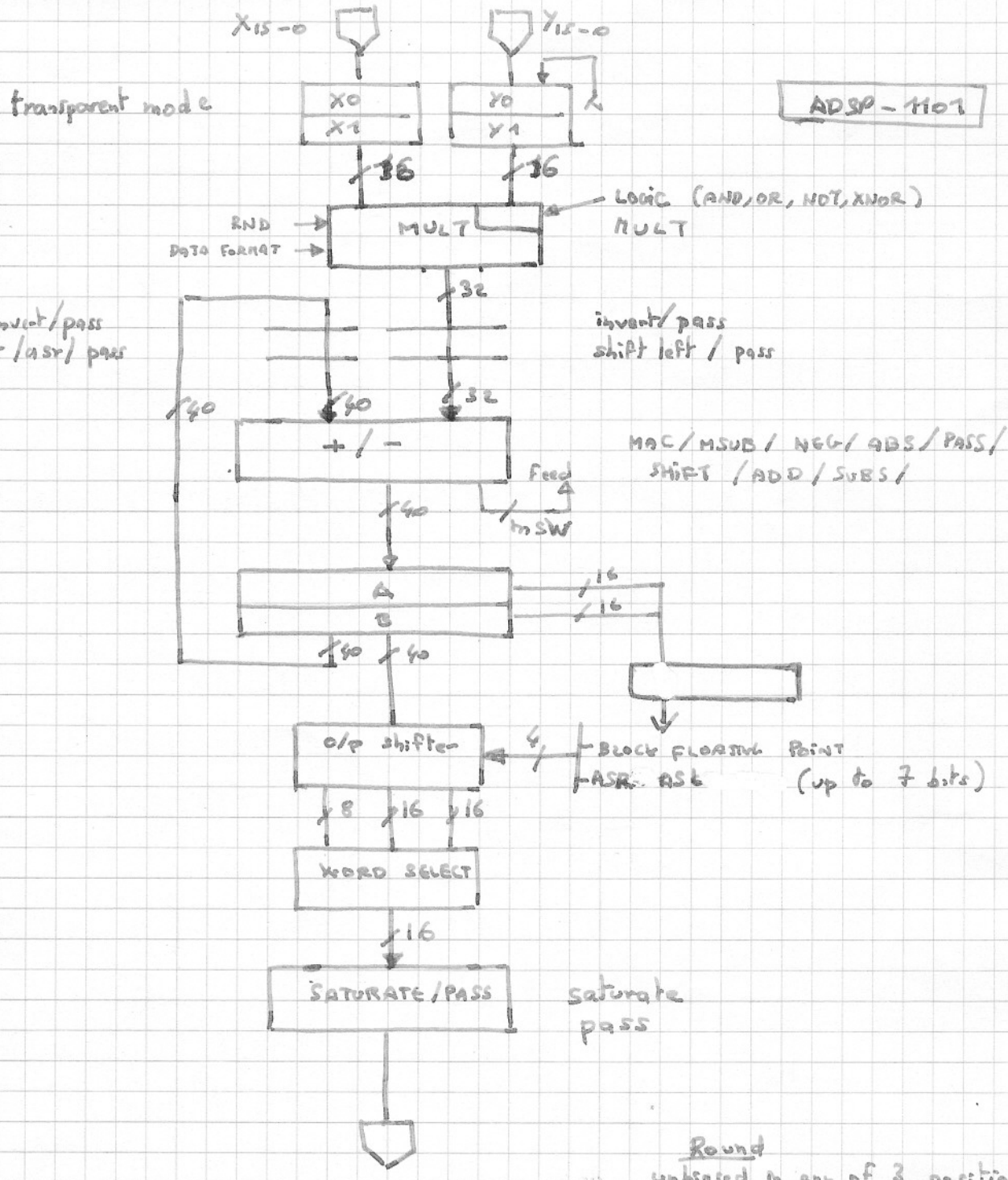

X15-0
Y15-0
16
MUX
MUX
16
16
X0
X1
Y0
Y1
16
16
16
16
16
MUX
LOGIC/MUX
X
Y
RND
DATA FORMAT
16 x 16 MULTIPLIER ARRAY
32
16
ZERO/INVERT/PASS
INVERT/PASS
40
32
SHIFT LEFT/RIGHT/PASS
SHIFT LEFT/PASS
40
32
ADD SUBTRACT
ACCOUT
SIGN
OVFR
ZERO
PRELOAD FORMAT CONTROL
40
40
16(MSW)
16
MUX
MUX
ACCUMULATOR A
OVFA
ACCUMULATOR B
OVFB
16
40
40
16
A39-24
B39-24
BLOCK FLOATING POINT & SHIFT CONTROL LOGIC
MUX
MUX
BLOCK FLOATING POINT & SHIFT CONTROL LOGIC
4
40
40
4
EXPA
EXPB
OUTPUT SHIFTER
4
4
8 EXT
16 MSW
16 LSW
CLK
37
INSTR
OVFR
ZERO
OVFA
OVFB
MUX
WORD SELECT
SIGN
4
4
Z19-16
1
4
16
MUX
SATURATE/PASS
EXTCND
1
4
Z15-0

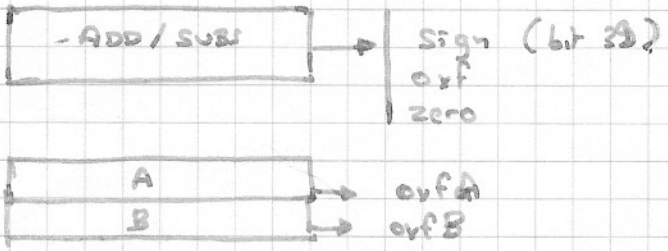


Round
unbiased in any of 3 positions
(bit 14, 15, 16) of
the accumulation.

Data format
signed, unsigned, mixed
(2's C)

Overflow
bit 39..31 must be identica
(signed); if not overflow

STATUS



expA
expB
4 bit encoded exponent of the A and B accumulators

ARCHITECTURE

As shown in the Block Diagram, the ADSP-1101 consists of the following principal architectural elements, which are interconnected by 16-bit, 32-bit, or 40-bit data paths:

* Two 16-Bit Input Ports and Four 16-Bit Input Registers
* A 16x16-Bit Multiplier Array
* 40-bit Adder/Subtractor
* A 16-Bit Logical Unit
* Dual Accumulator Registers
* Block Floating Point and Shift Control Logic
* Two Status Registers
* 16-Bit Output Port

Input Ports and Registers

The ADSP-1101 has two 16-bit input ports, each supported by a pair of input registers. The input registers are sources to the device's multiplier, adder/subtractor, and logical unit. It is also possible to by-pass the input registers -- feeding data directly from the input ports to the device's computational units.

The X input registers are loaded via the X port. The Y input registers can be loaded from either the X or Y port, or from the MSW of the current 40-bit accumulation. Also, the ADSP-1101's two accumulators can be preloaded from the Y input port.

Z Output Port

The 16-bit ZPORT is used to output the A or B accumulators, the 40-bit adder/subtractor result, or either 16-bit Status Register. For all 40-bit data, the Z Output Port allows reading either the 16 LSW bits, the 16 MSW bits, or the 8 EXT bits. When the 8-bit EXT is output onto the 8 LSB's of the 16-bit Z port, it is sign extended if appropriate.

The four status output lines can be dedicated to outputting the 4 LSB's of the 8-bit extension field when the MSW is being output -- effectively extending the Z Output Port to 20 bits.

Multiplier Array

The 16x16 array multiplier produces a 32-bit product that is fed to the 40-bit adder/subtractor. Inputs to the multiplier may be unsigned, two's complement, or mixed mode. Products may be format adjusted (left shifted by one bit) immediately after exiting the multiplier array.

GENERAL DESCRIPTION

The ADSP-1101 is a three-port multiplier/accumulator with enhancements that increase the efficiency of arithmetic calculations over standard MAC's, as well as reduce the amount of external logic required. The principal enhancements of the device include its dual accumulators, extensive shifting and rounding capabilities, flexible internal bus structure, and ALU capabilities. Packaged in a 100-pin grid array, the device supports 60ns cycles while dissipating less than 300mW.

Each of the ADSP-1101's input ports has a pair of independently-addressable registers. The X input registers are loaded via the X port; Y input registers can be loaded via either the X or Y port. In addition, the Y port can be used to preload either of the ADSP-1101's two accumulators. Two inputs can be read in a single cycle over each input port. This input structure allows single cycle computation of the term Y = M*X + B.

The ADSP-1101's 16x16 parallel array multiplier can perform two's complement, unsigned magnitude, or mixed mode multiplications. Products can be left shifted one bit before accumulation. Both the terms (product $\pm$ accumulator) or (accumulator $\pm$ product) can be obtained.

The 40-bit Adder/Subtractor supports operations such as add, subtract, pass, negate, absolute value, and pass with shift. Logical Operations are performed on the Y input registers, generating 16-bit logical results. These results can then be shifted in the multiplier array. The logical instructions also facilitate bit stuffing and bit set/clear.

The ADSP-1101 has two 40-bit accumulator registers, each segmented into a 16-bit LSW and MSW and an 8-bit Extended Precision register. Each accumulator register has an overflow flag that detects when the MSW overflows into the extension register. Saturation logic can be used in conjunction with overflow detect to conditionally saturate outputs. Block floating point and shift control registers track the magnitude of each accumulator register, for block floating point implementations and auto-normalizations.

The ADSP-1101's output port can operate at a 30ns rate, allowing the MSW and LSW of the product to be read in a single 60ns cycle. Outputs can be either left or right shifted by up to 7 bits, for maximum flexibility in data formatting. The ADSP-1101 can optionally round on either bit 14, 15, or 16 of the accumulator, and all rounding in the ADSP-1101 is done without bias. The ADSP-1101's accumulators can also be fed back to the multiplier array via an on-chip data path.

The 37-bit instruction word of the ADSP-1101 gives the designer total control over the part's internal hardware. The instruction set is fielded, however, to allow the width of the controlling microcode to be substantially reduced in many applications.

Logical Unit

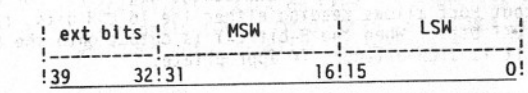
The Logical Unit operates on 16-bit data supplied from the Y input registers. The Logical block supports operations for AND, OR, NOT, and XNOR. The 16-bit logical result may be passed directly to an accumulator register; in addition, the result can first be shifted by the multiplier array prior to being written or added to an accumulator.

40-bit Adder/Subtractor and Accumulator Registers

A 40-bit adder/subtractor is used in performing multiply/accumulates and other arithmetic functions. Operations that can be executed include: ($\pm$ product $\pm$ accumulator), ($\pm$ input $\pm$ accumulator), negate, absolute value, pass, and pass with shift. The output of the 40-bit adder/subtractor is denoted ACC_{39-0}, which can be fed to one or both of the two 40-bit accumulator registers. Also, the MSW (ACC_{31-16}), LSW (ACC_{15-0}), or Extended Precision field (ACC_{39-32}) of this result can be directly output over the Z port.

The ADSP-1101's dual accumulators, denoted A and B, are valuable for complex arithmetic, with one accumulator storing the real data and the other the imaginary data. Having two accumulators is also useful for general purpose storage, or in systems that perform context switches. The 40-bit width of these accumulators protects against overflow in long multiply/accumulate chains.

Each accumulator is segmented into a 16-bit Most Significant Word (MSW) and Least Significant Word (LSW), and an 8-bit Extended Precision register (EXT). Circuitry in the accumulators tracks whether the data stored in an accumulator register is signed or unsigned (see Data Types). Also, overflow from the MSW into the EXT is monitored for each accumulator. The accumulator data format is:



Overflow Detect and Saturation

Overflow in either accumulator (OVFA and OVFB) and from the adder/subtractor result (OVFR) is detected and can be monitored off-chip. For two's complement data, the nine MSB's of the value are examined, and if not all identical, overflow has occurred. For unsigned magnitude data, only the 8 MSB's need be examined. These overflow signals are available as direct status output lines. Note that these overflow flags will not reflect cases where a value overflows solely as a consequence of output shifting.

The Saturate/Pass block allows the user to pass data to the output port, or to saturate overflowed data to positive or negative full scale. Saturation prevents serious wrap-around errors that occur when reading the MSW of an overflowed value.

Output MSW of register A or B with saturation:

Value in Accumulator A or B Or the Adder/Subtractor Result	Saturated MSW Output
Overflow and positive data	0111111111111111
Overflow and negative data	1000000000000001
Overflow and unsigned data	1111111111111111
No overflow	No Change

The MSW can be left-shifted or right-shifted by up to 7 bits on output. The ADSP-1101's saturation logic considers only whether the post-shift value of the number overflows. Saturation operates only on output values; it has no effect on the contents of either accumulator register.

When using saturation with a 20-bit output (see Z Port), saturation affects only the 16 bits output to the ZPORT (bits 16 to 31 of the unshifted output result) -- any bits output to the 4-bit output extension field are not modified.

For negative overflows, the ADSP-1101 saturates the output to a value slightly greater than negative full scale (100...001). The reason is that many applications:

1. use two's complement data,
2. format adjust (left shift) multiplier products (eliminating a redundant sign bit for two's complement results), and
3. occasionally take processed data and bring it back on chip for further processing (e.g., the data is first filtered, written to a memory, and brought back on chip for calculations such as auto- and cross-correlations).

A difficulty can be encountered when two's complement products are format adjusted -- namely, an invalid result is obtained if both multiplier inputs are negative full scale. If negative overflows were pegged to negative full scale, step 3 might frequently generate invalid products. The ADSP-1101's saturation scheme greatly reduces the frequency of such spurious results.

Block Floating Point and Shift Control Logic

Both the A and B accumulator are supported by Block Floating Point (BFP) register and a Shift Control Register (SCR). These registers can be read from the Z port and written via the Y port. In addition, these circuits can be used to control a 7-bit bi-directional output shifter -- allowing the device to implement full block floating point for DSP routines such as the FFT, or to normalize numbers in a single cycle.