

The PDSP1601 is a high performance 16-bit arithmetic logic unit with an independent on-chip 16-bit barrel shifter. The PDSP1601A has two operating modes giving 20MHz or 10MHz register-to-register transfer rates.

The PDSP1601 supports Multicycle multiprecision operation. This allows a single device to operate at 20MHz for 16-bit fields, 10MHz for 32-bit fields and 5MHz for 64-bit fields. The PDSP1601 can also be cascaded to produce wider words at the 20MHz rate using the Carry Out and Carry In pins. The Barrel Shifter is also capable of extension, for example the PDSP1601 can be used to select a 16-bit field from a 32-bit input in 100ns.

FEATURES

- 16-bit, 32 instruction 20MHz ALU
- 16-bit, 20MHz Logical, Arithmetic or Barrel Shifter
- Independent ALU and Shifter Operation
- 4 x 16-bit On Chip Scratchpad Registers
- Multiprecision Operation; e.g. 200ns 64-bit Accumulate
- Three Port Structure with Three Internal Feedback Paths Eliminates I/O Bottlenecks
- Block Floating Point Support
- 300mW Maximum Power Dissipation
- 84-pin Pin Grid Array or 84 Contact LCC Packages or 100 pin Ceramic Quad Flat Pack

APPLICATIONS

- Digital Signal Processing
- Array Processing
- Graphics
- Database Addressing
- High Speed Arithmetic Processors

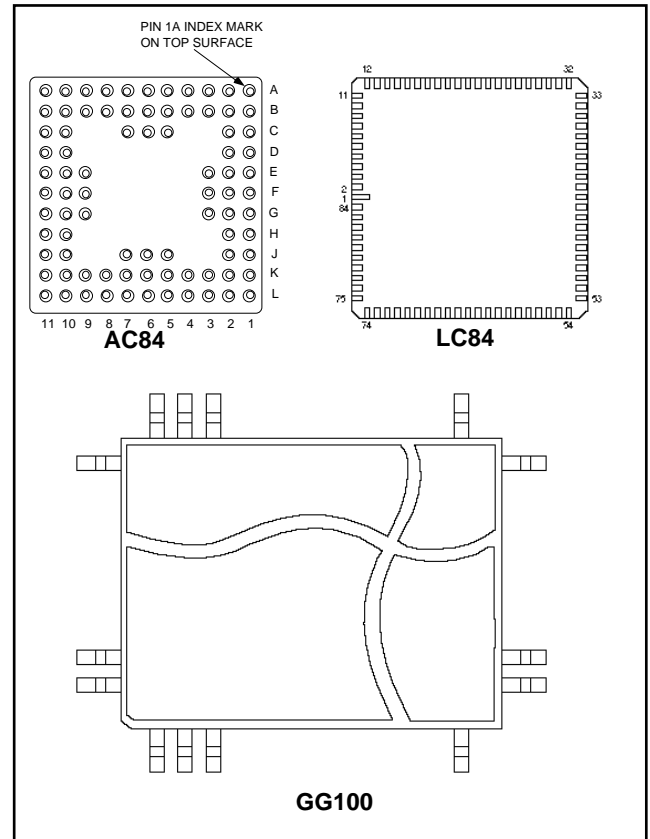


Fig.1 Pin connections - bottom view

ASSOCIATED PRODUCTS

- PDSP16112** Complex Multiplier
- PDSP16116** 16 x 16 Complex Multiplier
- PDSP16318** Complex Accumulator
- PDSP16330** Pythagoras Processor

PDSP1601/PDSP1601A

PIN DESCRIPTION

| LC pin | AC pin | Function | LC pin | AC pin | Function | LC pin | AC pin | Function | LC pin | AC pin | Function |
|--------|--------|----------|--------|--------|----------|--------|--------|----------|--------|--------|----------|
| 1 | C6 | IA4 | 22 | F3 | GND | 43 | J6 | IS0 | 64 | F9 | GND |
| 2 | A6 | MSB | 23 | G3 | MSA0 | 44 | J7 | IS1 | 65 | F11 | C8 |
| 3 | A5 | MSS | 24 | G1 | MSA1 | 45 | L7 | IS2 | 66 | E11 | C9 |
| 4 | B5 | B15 | 25 | G2 | A15 | 46 | K7 | IS3 | 67 | E10 | C10 |
| 5 | C5 | B14 | 26 | F1 | A14 | 47 | L6 | SV0 | 68 | E9 | C11 |
| 6 | A4 | B13 | 27 | H1 | A13 | 48 | L8 | SV1 | 69 | D11 | C12 |
| 7 | B4 | B12 | 28 | H2 | A12 | 49 | K8 | SV2 | 70 | D10 | C13 |
| 8 | A3 | B11 | 29 | J1 | A11 | 50 | L9 | SV3 | 71 | C11 | C14 |
| 9 | A2 | B10 | 30 | K1 | A10 | 51 | L10 | SVOE | 72 | B11 | C15 |
| 10 | B3 | B9 | 31 | J2 | A9 | 52 | K9 | RS0 | 73 | C10 | OE |
| 11 | A1 | B8 | 32 | L1 | A8 | 53 | L11 | RS1 | 74 | A11 | BFP |
| 12 | B2 | B7 | 33 | K2 | A7 | 54 | K10 | VCC | 75 | B10 | VCC |
| 13 | C2 | B6 | 34 | K3 | A6 | 55 | J10 | RS2 | 76 | B9 | CO |
| 14 | B1 | B5 | 35 | L2 | A5 | 56 | K11 | C0 | 77 | A10 | RA0 |
| 15 | C1 | B4 | 36 | L3 | A4 | 57 | J11 | C1 | 78 | A9 | RA1 |
| 16 | D2 | B3 | 37 | K4 | A3 | 58 | H10 | C2 | 79 | B8 | RA2 |
| 17 | D1 | B2 | 38 | L4 | A2 | 59 | H11 | C3 | 80 | A8 | CI |
| 18 | E3 | B1 | 39 | J5 | A1 | 60 | F10 | C4 | 81 | B6 | IA0 |
| 19 | E2 | B0 | 40 | K5 | A0 | 61 | G10 | C5 | 82 | B7 | IA1 |
| 20 | E1 | CEB | 41 | L5 | CEA | 62 | G11 | C6 | 83 | A7 | IA2 |
| 21 | F1 | CLK | 42 | K6 | MSC | 63 | G9 | C7 | 84 | C7 | IA3 |

| GG | SIG | GG | SIG | GG | SIG | GG | SIG |
|----|-----|----|------|----|------|-----|-----|
| 1 | N/C | 26 | N/C | 51 | N/C | 76 | N/C |
| 2 | N/C | 27 | N/C | 52 | N/C | 77 | N/C |
| 3 | N/C | 28 | N/C | 53 | N/C | 78 | N/C |
| 4 | N/C | 29 | N/C | 54 | N/C | 79 | N/C |
| 5 | VCC | 30 | B7 | 55 | A7 | 80 | VCC |
| 6 | C0 | 31 | B6 | 56 | A6 | 81 | RS2 |
| 7 | RA0 | 32 | B5 | 57 | A5 | 82 | C0 |
| 8 | RA1 | 33 | B4 | 58 | A4 | 83 | C1 |
| 9 | RA2 | 34 | B3 | 59 | A3 | 84 | C2 |
| 10 | CI | 35 | B2 | 60 | A2 | 85 | C3 |
| 11 | IA0 | 36 | B1 | 61 | A1 | 86 | C4 |
| 12 | IA1 | 37 | B0 | 62 | A0 | 87 | C5 |
| 13 | IA2 | 38 | CEB | 63 | CEA | 88 | C6 |
| 14 | IA3 | 39 | CLK | 64 | MSC | 89 | C7 |
| 15 | IA4 | 40 | GND | 65 | IS0 | 90 | GND |
| 16 | MSB | 41 | MSA0 | 66 | IS1 | 91 | C8 |
| 17 | MSS | 42 | MSA1 | 67 | IS2 | 92 | C9 |
| 18 | B15 | 43 | A15 | 68 | IS3 | 93 | C10 |
| 19 | B14 | 44 | A14 | 69 | SV0 | 94 | C11 |
| 20 | B13 | 45 | A13 | 70 | SV1 | 95 | C12 |
| 21 | B12 | 46 | A12 | 71 | SV2 | 96 | C13 |
| 22 | B11 | 47 | A11 | 72 | SV3 | 97 | C14 |
| 23 | B10 | 48 | A10 | 73 | SVOE | 98 | C15 |
| 24 | B9 | 49 | A9 | 74 | RS0 | 99 | OE |
| 25 | B8 | 50 | A8 | 75 | RS1 | 100 | BFP |

N/C = not connected - leave open circuit

All GND and VDD pin must be used

PIN DESCRIPTIONS

| Symbol | Pin No. (LC84 Package) | Description |
|------------------|------------------------------|---|
| MSB | 2 | ALU B-input multiplexer select control. ¹ This input is latched internally on the rising edge of CLK. |
| MSS | 3 | Shifter Input multiplexer select control. ¹ This input is latched internally on the rising edge of CLK. |
| B15 - B0 | 4 - 19 | B Port data input. Data presented to this port is latched into the input register on the rising edge of CLK. B15 is the MSB. |
| <u>CEB</u> | 20 | Clock enable, B Port input register. When low the clock to this register is enabled. |
| CLK | 21 | Common clock to all internal registered elements. All registers are loaded, and outputs change on the rising edge of CLK. |
| MSA0 - MSA1 | 23 - 24 | ALU A-input multiplexer select control. ¹ These inputs are latched internally on the rising edge of CLK. |
| A15 - A0 | 25 - 40 | A Port data input. Data presented to this port is latched into the input register on the rising edge of CLK. A15 is the MSB. |
| <u>CEA</u> | 41 | Clock enable, A Port input register. When low the clock to this register is enabled. |
| MSC | 42 | C-Port multiplexer select control. ¹ This input is latched internally on the rising edge of CLK. |
| IS0 - IS3 | 43 - 46 | Instruction inputs to Barrel Shifter, IS3 = MSB. ¹ These inputs are latched internally on the rising edge of CLK. |
| SV0 - SV3 | 47 - 50 | Shift Value I/O Port. This port is used as an input when shift values are supplied from external sources, and as an output when Normalise operations are invoked. The I/O functions are determined by the IS0 - IS3 instruction inputs, and by the <u>SVOE</u> control. The shift value is latched internally on the rising edge of CLK. |
| <u>SVOE</u> | 51 | SV Output enable. When high the SV port can only operate as an input. When low the SV port can act as an input or as an output, according to the IS0 - IS3 instruction. This pin should be tied high or low, depending upon the application. |
| RS0, RS1 RS2 | 52 - 53 55 | Instruction inputs to Barrel Shifter registers. ¹ These inputs are latched internally on the rising edge of CLK. |
| C0 - C15 | 56 - 63 65 - 72 | C Port data output. Data output on this port is selected by the C output multiplexer. C15 is the MSB. |
| <u>OE</u> | 73 | Output enable. The C Port outputs are in high impedance condition when this control is high. |
| BFP | 74 | Block Floating Point Flag from ALU, active high. |
| CO | 76 | Carry out from MSB of ALU. |
| RA0 - RA2 | 77 - 79 | Instruction inputs to ALU registers. ¹ These inputs are latched internally on the rising edge of CLK. |
| CI | 80 | Carry in to LSB of ALU. |
| IA0 - IA3 IA4 | 81 - 84 1 | Instruction inputs to ALU. ¹ IA4 = MSB. These inputs are latched internally on the rising edge of CLK. |
| Vcc | 54 & 75 | +5V supply: Both Vcc pins must be connected. |
| GND | 22 & 64 | 0V supply: Both GND pins must be connected. |

NOTES

1. All instructions are executed in the cycle commencing with the rising edge of the CLK which latches the inputs.

PDSP1601/PDSP1601A

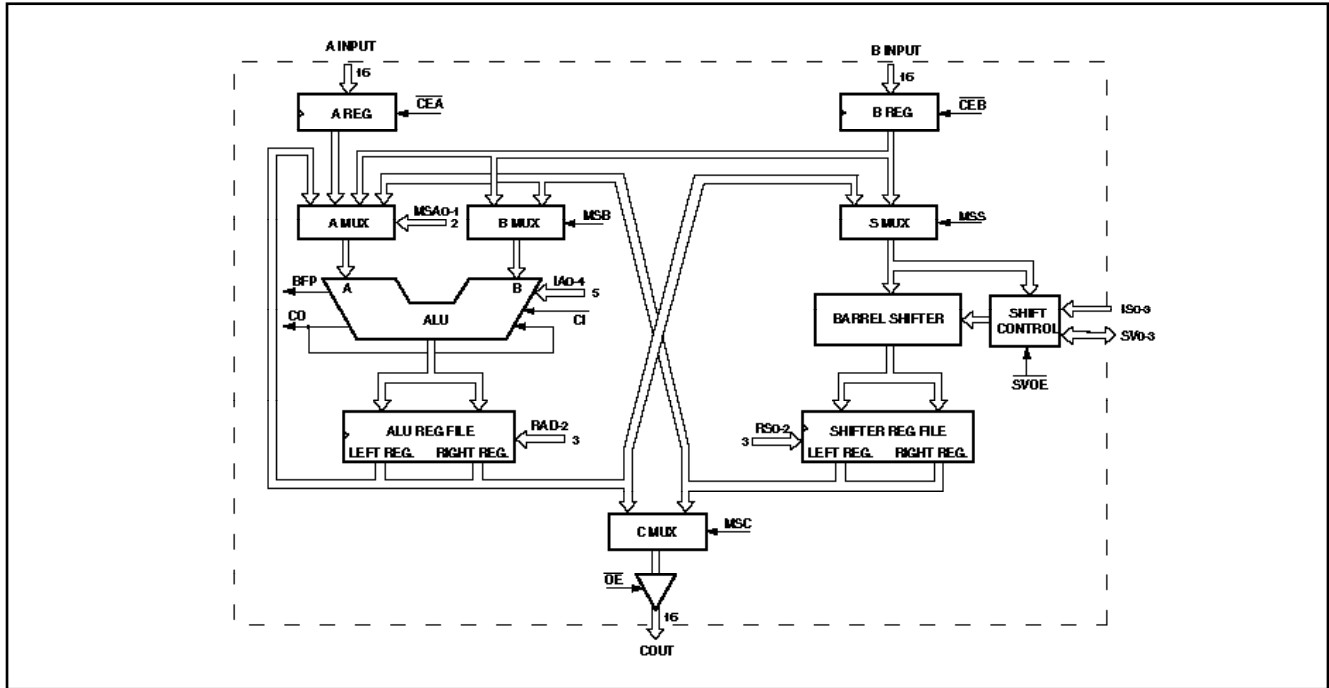


Fig.2 PDSP1601 block diagram

FUNCTIONAL DESCRIPTION

The PDSP1601 contains four main blocks: the ALU, the Barrel Shifter and the two Register Files.

The ALU

The ALU supports 32 instructions as detailed in Table 1.

The inputs to the ALU are selected by the A and B MUXs. Data will fall through from the selected register through the A or B input MUXs and the ALU to the ALU output register file in 50ns for the PDSP1601A (100ns for the PDSP1601).

The ALU instructions are latched, such that the instruction will not start executing until the rising edge of CLK latches the instruction into the device.

The ALU accepts a carry in from the CI input and supplies a carry out to the CO output. Additionally, at the end of each cycle, the carry out from the ALU is loaded into an internal 1 bit register, so that it is available as an input to the ALU on the next cycle. In the manner, multicycle, multiprecision operations are supported. (See MULTICYCLE CASCADE OPERATIONS).

BFP Flag

The ALU has a user programmable BFP flag. This flag may be programmed to become active at any one of four conditions. Two of these conditions are intended to support Block Floating Point operations, in that they provide flags indicating that the ALU result is within a factor of two or four of overflowing the 16 bit number range. For multiprecision operations the flag is only valid whilst the most significant 16 bit byte is being processed. In this manner the BFP flag may be used over any extended word width.

The remaining two conditions detect either an overflow condition or a zero result. For the overflow condition to be

active the ALU result must have overflowed into the 16th (sign) bit, (this flag is only valid whilst the most significant 16 bit byte is being processed). The zero condition is active if the result from the ALU is equal to zero. For multiprecision operations the zero flag must be active for all of the 16 bit bytes of an extended word.

The BFP flag is programmed by executing one of the four SBFXX instructions (see Table 1). During the execution of any of these four instructions, the output of the ALU is forced to zero.

Multicycle/Cascade Operation

The ALU arithmetic instructions contain two or three options for each arithmetic operation.

The ALU is designed to operate with two's complement arithmetic, requiring a one to be added to the LSB for all subtract operations. The instructions set includes instructions that will force a one into the LSB, e.g. MIAX1, AMBX1, BMAX1 (see Table 1).

These instructions are used for the least significant 16 bit byte of any subtract operation.

The user has an option of cascading multiple devices, or multicycling a single device to extend the arithmetic precision. Should the user cascade multiple devices, then the cascade arithmetic instructions using the external CI input should be employed for all but the least significant 16 bit byte, e.g. MIACI, APBCI, BMACI (see Table 1).

Should the user multicycle a single device, then the Multicycle Arithmetic instructions, using the internally registered CO bit should be employed for all but the least significant 16 bit byte, e.g. MIACO, APBCO, AMBCO, BMACO (see Table 1).

Table 1 ALU instructions

1a. ARITHMETIC INSTRUCTIONS

| Inst | IA4-AI0 | Mnemonic | Operation | Function | Mode |
|------|---------|----------|-----------|---------------------|------------|
| 00 | 00000 | CLRXX | RESET | CLEAR ALL REGISTERS | ----- |
| 01 | 00001 | MIAX1 | MINUS A | NA Plus 1 | LSBYTE |
| 02 | 00010 | MIACI | MINUS A | NA Plus CI | CASCADE |
| 03 | 00011 | MIACO | MINUS A | NA Plus CO | MULTICYCLE |
| 04 | 00100 | A2SGN | A/2 | A/2 Sign Extend | MSBYTE |
| 05 | 00101 | A2RAL | A/2 | A/2 with RAL LSB | MULTICYCLE |
| 06 | 00110 | A2RAR | A/2 | A/2 with RAR LSB | MULTICYCLE |
| 07 | 00111 | A2RSX | A/2 | A/2 with RSX LSB | MULTICYCLE |
| 08 | 01000 | APBCI | A PLUS B | A Plus B Plus CI | CASCADE |
| 09 | 01001 | APBCO | A PLUS B | A Plus B Plus CO | MULTICYCLE |
| 0A | 01010 | AMBX1 | A MINUS B | A Plus NB Plus 1 | LSBYTE |
| 0B | 01011 | AMBCI | A MINUS B | A Plus NB Plus CI | CASCADE |
| 0C | 01100 | AMBCO | A MINUS B | A Plus NB Plus CO | MULTICYCLE |
| 0D | 01101 | BMAX1 | B MINUS A | NA Plus B Plus 1 | LSBYTE |
| 0E | 01110 | BMACI | B MINUS A | NA Plus B Plus CI | CASCADE |
| 0F | 01111 | BMACO | B MINUS A | NA Plus B Plus CO | MULTICYCLE |

1b. LOGICAL INSTRUCTIONS

| Inst | IA4-AI0 | Mnemonic | Operation | Function |
|------|---------|----------|-----------|----------|
| 10 | 10000 | ANXAB | A AND B | A. B |
| 11 | 10001 | ANANB | A AND NB | A. NB |
| 12 | 10010 | ANNAB | NA AND B | NA. B |
| 13 | 10011 | ORXAB | A OR B | A + B |
| 14 | 10100 | ORNAB | NA OR B | NA + B |
| 15 | 10101 | XORAB | A XOR B | A XOR B |
| 16 | 10110 | PASXA | PASS A | A |
| 17 | 10111 | PASNA | INVERT A | NA |

1c. CONTROL INSTRUCTIONS

| Inst | IA4-AI0 | Mnemonic | Operation |
|------|---------|----------|--|
| 18 | 11000 | SBFOV | Set BFP Flag to OVR, Force ALU output to zero |
| 19 | 11001 | SBFU1 | Set BFP Flag to UND 1 Force ALU output to zero |
| 1A | 11010 | SBFU2 | Set BFP Flag to UND 2 Force ALU output to zero |
| 1B | 11011 | SBFZE | Set BFP Flag to ZERO Force ALU output to zero |
| 1C | 11100 | OPONE | Output 0001 Hex |
| 1D | 11101 | OPBYT | Output 00FF Hex |
| 1E | 11110 | OPNIB | Output 000F Hex |
| 1F | 11111 | OPALT | Output 5555 Hex |

KEY

- A = A input to ALU
- B = B input to ALU
- CI = External Carry in to ALU
- CO = Internally Registered Carry out from ALU
- RAL = ALU Register (Left)
- RAR = ALU Register (Right)
- RSX = Shifter Register (Left or Right)

MNEMONICS

- CLRXX Clear All Registers to zero
- MIAXX Minus A, XX = Carry in to LSB
- A2XXX A Divided by 2, XXX = Source of MSB
- APBXX A Plus B, XX = Carry in to LSB
- AMBXX A Minus B, XX = Carry in to LSB
- BMAXX B Minus A, XX = Carry in to LSB
- ANX-Y AND X = Operand 1, Y = Operand 2
- ORX-Y OR X = Operand 1, Y = Operand 2
- XORXY Exclusive OR X = Operand 1, Y = Operand 2
- PASXX Pass XX = Operand
- SBFXX Set BFP Flag XX = Function
- OPXXX Output Constant XXX

PDSP1601/PDSP1601A

Divide by Two

The ALU has four (A2SGN, A2RAL, A2RAR, A2RSX) instructions used for right shifting (dividing by two) extended precision words. These words, (up to 64 bits) may be stored in the two on-chip register files. When the least significant 16 bit word is shifted, the vacant MSB must be filled with the LSB from the next most significant 16 bit byte. This is achieved via the A2RAL, A2RAR or A2RSX instructions which indicate the source of the new MSB (see ALU INSTRUCTION SET).

When the most significant 16 bit byte is right shifted, the MSB must be filled with a duplicate of the original MSB so as to maintain the correct sign (Sign Extension). This operation is achieved via the A2SGN instruction (see Table 1).

Constants

The ALU has four instructions (OPONE, OPBYT, OPNIB, OPALT) that force a constant value onto the ALU output. These values are primarily intended to be used as masks, or the seeds for mask generation, for example, the OPONE instruction will set a single bit in the least significant position. This bit may be rotated any where in the 16 bit field by the Barrel Shifter, allowing the AND function of the ALU to perform bit-pick operations on input data.

CLR

The ALU instruction CLRXX is used as a Master Reset for the entire device. This instruction has the effect of:

1. Clearing ALU and Barrel Shifter register files to zero.
2. Clearing A and B port input registers to zero.
3. Clearing the R1 and R2 shift control registers to zero.
4. Clearing the internally registered CO bit to zero.
5. Programming the BFP flag to detect *overflow* conditions.

The Barrel Shifter

The Barrel Shifter supports 16 instructions as detailed in Table 2. The input to the Barrel Shifter is selected by the S MUX. Data will fall through from the selected register, through the S MUX and the Barrel Shifter to the shifter output register file in 50ns for the PDSP1601A (100ns for the PDSP1601).

The Barrel Shifter instructions are latched, such that the instructions will not start executing until the rising edge of CLK latches the instruction into the device.

The Barrel Shifter is capable of Logical Arithmetic or Barrel Shifts in either direction.

- A. Logical shifts discard bits that exit the 16 bit field and fill spaces with zeros.
- B. Arithmetic shifts discard bits that exit the 16 bit field and fill spaces with duplicates of the original MSB.
- C. Barrel Shifts rotate the 16 bit fields such that bits that exit the 16 bit field to the left or right reappear in the vacant spaces on the right or left.

The amount of shift applied is encoded onto the 4 bit Barrel Shifter input as illustrated in Table 3. The type of shift and the amount are determined by the shift control block. The shift control block (see Fig.3) accepts and decodes the four bit ISO-3 instruction. The shift control block contains a priority encoder and two user programmable 4 bit registers R1 and R2.

There are four possible sources of shift value that can be passed onto the Barrel Shifter, there are:

1. The Priority Encoder
2. The SV input
3. The R1 register
4. The R2 register

| Inst | IS3-IS0 | Mnemonic | Operation | I/O |
|------|---------|----------|------------------------------|-----|
| 0 | 0000 | LSRSV | Logical Shift Right by SV | I |
| 1 | 0001 | LSLSV | Logical Shift Left by SV | I |
| 2 | 0010 | BSRSV | Barrel Shift Right by SV | I |
| 3 | 0011 | BSLSV | Barrel Shift Left by SV | I |
| 4 | 0100 | LSRR1 | Logical Shift Right by R1 | X |
| 5 | 0101 | LSLR1 | Logical Shift Left by R1 | X |
| 6 | 0110 | LSRR2 | Logical Shift Right by R2 | X |
| 7 | 0111 | LSLR2 | Logical Shift Left by R2 | X |
| 8 | 1000 | LR1SV | Load Register 1 From SV | I |
| 9 | 1001 | LR2SV | Load Register 2 From SV | I |
| A | 1010 | ASRSV | Arithmetic Shift Right by SV | I |
| B | 1011 | ASRR1 | Arithmetic Shift Right by R1 | X |
| C | 1100 | ASRR2 | Arithmetic Shift Right by R2 | X |
| D | 1101 | NRMXX | Normalise Output PE | O |
| E | 1110 | NRMR1 | Normalise Output PE, Load R1 | O |
| F | 1111 | NRMR2 | Normalise Output PE, Load R2 | O |

Table 2 Barrel shifter instructions

KEY

- SV = Shift Value
R1 = Register 1
R2 = Register 2
PE = Priority Encoder Output
I => SV Port operates as an Input
O => SV Port operates as an Output
X => SV Port in a High Impedance State

MNEMONICS

- LSXYY Logical Shift, X = Direction YY = Source of Shift Value
BSXYY Barrel Shift, X = Direction YY = Source of Shift Value
ASXYY Arithmetic Shift, X = Direction YY = Source of Shift Value
LXXYY Load XX = Target YY = Source
NRMY Y Normalise by PE, Output PE value on SV Port, Load YY Reg

| SV3 | SV2 | SV1 | SV0 | Shift |
|-----|-----|-----|-----|-----------|
| 0 | 0 | 0 | 0 | No shift |
| 0 | 0 | 0 | 1 | 1 place |
| 0 | 0 | 1 | 0 | 2 places |
| 0 | 0 | 1 | 1 | 3 places |
| 0 | 1 | 0 | 0 | 4 places |
| 0 | 1 | 0 | 1 | 5 places |
| 0 | 1 | 1 | 0 | 6 places |
| 0 | 1 | 1 | 1 | 7 places |
| 1 | 0 | 0 | 0 | 8 places |
| 1 | 0 | 0 | 1 | 9 places |
| 1 | 0 | 1 | 0 | 10 places |
| 1 | 0 | 1 | 1 | 11 places |
| 1 | 1 | 0 | 0 | 12 places |
| 1 | 1 | 0 | 1 | 13 places |
| 1 | 1 | 1 | 0 | 14 places |
| 1 | 1 | 1 | 1 | 15 places |

Table 3 Barrel shifter codes

Priority Encoder

If the priority encoder is selected as the source of the shift value (instructions:- NRMXX, NRMR1, MRMRZ), then within one 100ns cycle or two 50ns cycles for the PDSP1601A (one 200ns or two 100ns cycles for the PDSP1601), the shift circuitry will:

(1) Priority encode the 16 bit input to the Barrel Shifter and place the 4 bit value in either of the R1 or R2 registers and output the value on the SV port (if enabled by SVOE).

(2) Shift the 16 bit input by the amount indicated by the Priority Encoder such that the output from the Barrel Shifter is a normalised value.

SV Input

If the SV port is selected as the source of the shift value, then the input to the Barrel Shifter is shifted by the value stored in the internal SV register.

SVOE

The SV port acts as an input or an output depending upon the ISO-3 instruction. If the user does not wish to use the normalise instructions, then the SV port may be forced to be input only by typing SVOE control high. In this mode the SV port may be considered an extension of the instruction inputs.

R1 and R2 Registers

The R1 and R2 registers may be loaded from the Priority Encoder (NRMR1 and NRMR2) or from the SV input (LR1SV, LR2SV).

Whilst the latter two instructions are executing, the Barrel Shifter will pass its input to the output unshifted.

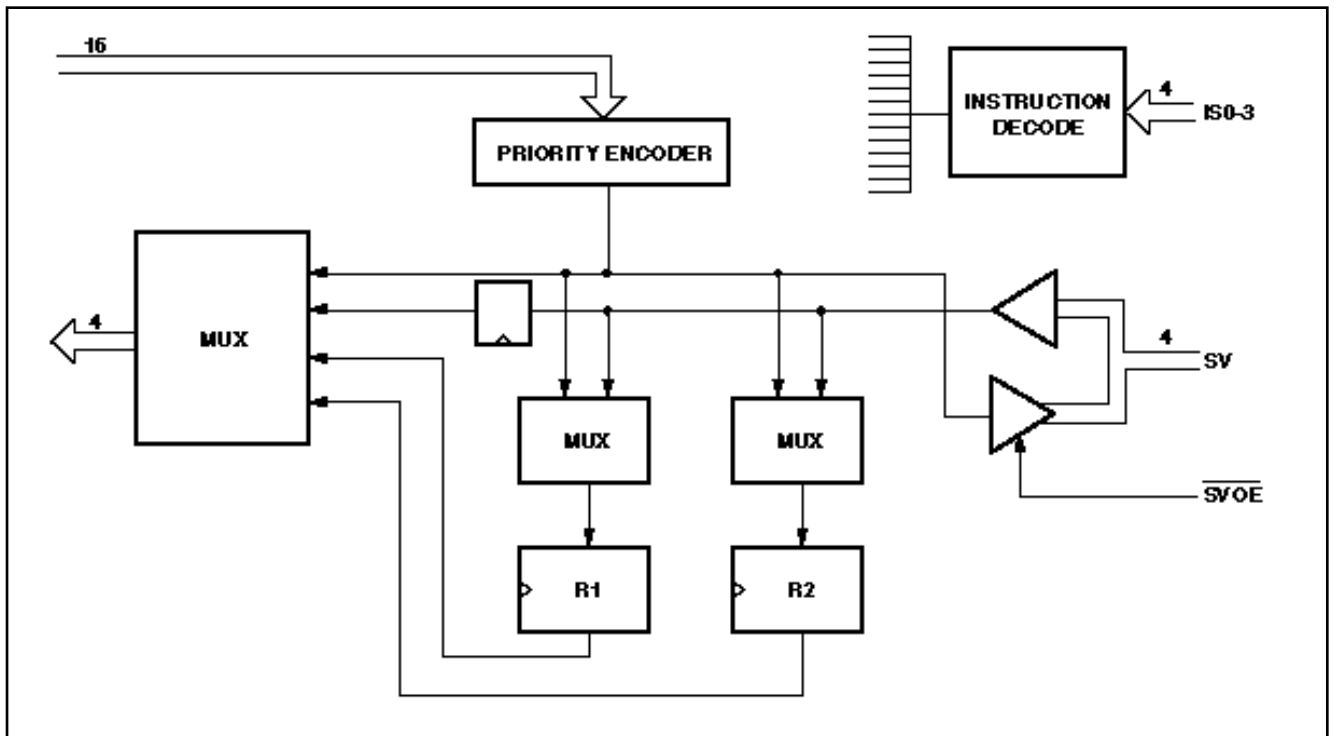


Fig.3 Shift control block

PDSP1601/PDSP1601A

The Register Files

There are two on-chip register files (ALU and Shifter), each containing two 16 bit registers and each supporting 8 instructions (see Table 4). The instructions for the ALU register file and the Barrel Shifter Register file are the same.

The Inputs to the register files come from either the ALU or the Barrel Shifter, and are loaded into the Register files on the rising edge of CLK.

The register file instructions are latched such that the instruction will not start executing until the rising edge of the

CLK latches the instruction into the device.

The register file instructions (see Table 4) allow input data to be loaded into either, neither or both of the registers. Data is loaded at the end of the cycle in which the instruction is executing.

The register file instructions allow the output to be sourced from either of the two registers, the selected output will be valid during the cycle in which the instruction is executing.

| ALU REGISTER INSTRUCTIONS | | | |
|-------------------------------|---------|----------|---|
| Inst | RA2-RA0 | Mnemonic | Operation |
| 0 | 000 | LLRRR | Load Left Reg Output Right Reg |
| 1 | 001 | LRRLR | Load Right Reg Output Left Reg |
| 2 | 010 | LLRLR | Load Left Register, Output Left Reg |
| 3 | 011 | LRRRR | Load Right Register, Output Right Reg |
| 4 | 100 | LBRLR | Load Both Registers, Output Left Reg |
| 5 | 101 | NOPRR | No Load Operation, Output Right Reg |
| 6 | 110 | NOPLR | No Load Operation, Output Left Reg |
| 7 | 111 | NOPPS | No Load Operation, Pass ALU Result |
| SHIFTER REGISTER INSTRUCTIONS | | | |
| Inst | RA2-RA0 | Mnemonic | Operation |
| 0 | 000 | LLRRR | Load Left Reg Output Right Reg |
| 1 | 001 | LRRLR | Load Right Reg Output Left Reg |
| 2 | 010 | LLRLR | Load Left Register, Output Left Reg |
| 3 | 011 | LRRRR | Load Right Register, Output Right Reg |
| 4 | 100 | LBRLR | Load Both Registers, Output Left Reg |
| 5 | 101 | NOPRR | No Load Operation, Output Right Reg |
| 6 | 110 | NOPLR | No Load Operation, Output Left Reg |
| 7 | 111 | NOPPS | No Load Operation, Pass Barrel Shifter Result |

Table 4 ALU and shift register instructions mnemonics

MNEMONICS

LXXYY Load XX = Target, YY = Source of Output

LBOXX Load Both Registers, XX = Source of Output

NOPXX No Load Operation, XX = Source of Output

Multiplexers

There are four user selectable on-chip multiplexers (A-MUX, B-MUX, S-MUX and C-MUX).

These four multiplexers support instructions as tabulated in Table 5.

The MUX instructions are latched such that the instruction will not start executing until the rising edge of CLK latches the instruction onto the device.

| | | MSA1 | MSA0 | Output |
|--------------|------------|------|------------------------------|------------------------------|
| A-MUX | MARAX | 0 | 0 | ALU REGISTER FILE OUPUT |
| | MAAPR | 0 | 1 | A-PORT INPUT |
| | MABPR | 1 | 0 | B-PORT INPUT |
| | MARSX | 1 | 1 | SHIFTER REGISTER FILE OUTPUT |
| B-MUX | MSB | | Output | |
| | 0 | | B-PORT INPUT | |
| | 1 | | SHIFTER REGISTER FILE OUTPUT | |
| S-MUX | MSS | | Output | |
| | 0 | | B-PORT INPUT | |
| | 1 | | SHIFTER REGISTER FILE OUTPUT | |
| C-MUX | MSC | | Output | |
| | 0 | | ALU REGISTER FILE OUTPUT | |
| | 1 | | SHIFTER REGISTER FILE OUTPUT | |

Table 5

PDSP1601/PDSP1601A

INSTRUCTION SET

ALU Arithmetic Instructions

| Mnemonic | Op Code | Function |
|----------|---------|---|
| CLRXX | <00> | On the rising edge of CLK at the end of the cycle in which this instruction is executing, the A Port, B Port, ALU, Barrel Shifter, and Shift Control Registers will be loaded with zeros. The internal registered CO will also be set to zero, and the BFP flag will be set to activate on overflow conditions. |
| MIAX1 | <01> | The A input to the ALU is inverted and a one is added to the LSB. |
| MIAC1 | <02> | The A input to the ALU is inverted and the CI input is added to the LSB. |
| MIACO | <03> | The A input to the ALU is inverted and the CO output from the ALU on the previous cycle is added to the LSB. |
| A2SGN | <04> | The A input to the ALU is right shifted one bit position. The LSB is discarded, and the vacant MSB is filled by duplicating the original MSB (Sign Extension). |
| A2RAL | <05> | The A input to the ALU is right shifted one bit position. The LSB is discarded, and the vacant MSB is filled with the LSB from the ALU register. |
| A2RAR | <06> | The A input to the ALU is right shifted one bit position. The LSB is discarded, and the vacant MSB is filled with the LSB from the ALU register. |
| A2RSX | <07> | The A input to the ALU is right shifted one bit position. The LSB is discarded, and the vacant MSB is filled with the LSB from the B input to the ALU. |
| APBCI | <08> | The A input to the ALU is added to the B input, and the CI input is added to the LSB. |
| APBCO | <09> | The A input to the ALU is added to the B input, and the CO out from the ALU on the previous cycle is added to the LSB. |
| AMBX1 | <0A> | The A input to the ALU is added to the inverted B input, and a one is added to the LSB. |
| AMBCI | <0B> | The A input to the ALU is added to the inverted B input, and the CI input is added to the LSB. |
| AMBCO | <0C> | The A input to the ALU is added to the inverted B input, and the CO out from the ALU on the previous cycle is added to the LSB. |
| BMAX1 | <0D> | The inverted A input to the ALU is added to the B input, and a one is added to the LSB. |
| BMAC1 | <0E> | The inverted A input to the ALU is added to the B input, and the CI input is added to the LSB. |
| BMACO | <0F> | The inverted A input to the ALU is added to the B input, and the CO out from the ALU on the previous cycle is added to the LSB. |

ALU Logical Instructions

| Mnemonic | Op Code | Function |
|----------|---------|--|
| ANXAB | <10> | The A input to the ALU is logically 'ANDed' with the B input. |
| ANANB | <11> | The A input to the ALU is logically 'ANDed' with the inverse of the B input. |
| ANNAB | <12> | The inverse of the A input to the ALU is logically 'ANDed' with the B input. |
| ORXAB | <13> | The A input to the ALU is logically 'ORed' with the B input. |
| ORNAB | <14> | The inverse A input to the ALU is logically 'ORed' with the B input. |
| XORAB | <15> | The A input to the ALU is logically Exclusive-ORed with the B input. |
| PASXA | <16> | The A input to the ALU is passed to the output. |
| PASNA | <17> | The inverse of the A input to the ALU is passed to the output. |

ALU Control Instructions

| Mnemonic | Op Code | Function |
|----------|---------|--|
| SBOV | <18> | The BFP flag is programmed to activate when an ALU operation causes an overflow of the 16 bit number range. This flag is logically the exclusive-or of the carry into and out of the MSB of the ALU. For the most significant Byte this flag indicates that the result of an arithmetic two's complement operation has overflowed into the sign bit. The output of the ALU is forced to zero for the duration of this instruction. |
| SBOU1 | <19> | The BFP flag is programmed to activate when an ALU operation comes within a factor of two of causing an overflow of the 16 bit number range. For the most significant Byte this flag indicates that the result of an arithmetic two's complement operation is within a factor of two of overflowing into the sign bit. The output of the ALU is forced to zero for the duration of this instruction. |
| SBOU2 | <1A> | The BFP flag is programmed to activate when an ALU operation comes within a factor of four of causing an overflow of the 16 bit number range. For the most significant Byte this flag indicates that the result of an arithmetic two's complement operation is within a factor of four of overflowing into the sign bit. The output of the ALU is forced to zero for the duration of this instruction. |
| SBOZE | <1B> | The BFP flag is programmed to activate when an ALU operation causes a result of zero. The output of the ALU is forced to zero for the duration of this instruction. During the execution of this instruction the BFP flag will become active. |
| OPONE | <1C> | The ALU will output the binary value 0000000000000001, the MSB on the left. |
| OPBYT | <1D> | The ALU will output the binary value 0000000011111111, the MSB on the left. |
| OPNIB | <1E> | The ALU will output the binary value 0000000000001111, the MSB on the left. |
| OPALT | <1F> | The ALU will output the binary value 0101010101010101, the MSB on the left. |

Barrel Shifter Instructions

| Mnemonic | Op Code | Function |
|----------|---------|---|
| LSRSV | <0> | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number present in the SV register. The LSBs are discarded, and the vacant MSBs are filled with zeros. |
| LSLSV | <1> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number present in the SV register. The LSBs are discarded, and the vacant MSBs are filled with zeros. |
| BSRSV | <2> | The 16 bit input to the Barrel Shifter is rotated to the right by the number of places indicated by the magnitude of the four bit number present in the SV register. The LSBs that exit the 16 bit field to the right, reappear in the vacant MSBs on the left. |
| BSLSV | <3> | The 16 bit input to the Barrel Shifter is rotated to the left by the number of places indicated by the magnitude of the four bit number present in the SV register. The LSBs that exit the 16 bit field to the right, reappear in the vacant MSBs on the right. |
| LSRR1 | <4> | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number resident within the R1 register. The LSBs are discarded, and the vacant MSBs are filled with zeros. |
| LSLR1 | <5> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number resident within the R1 register. The LSBs are discarded, and the vacant LSBs are filled with zeros. |
| LSRR2 | <6> | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number resident within the R2 register. The LSBs are discarded, and the vacant MSBs are filled with zeros. |
| LSLR2 | <7> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number resident within the R2 register. The LSBs are discarded, and the vacant LSBs are filled with zeros. |

PDSP1601/PDSP1601A

| Mnemonic | Op Code | Function |
|-----------------|----------------|--|
| LR1SV | <8> | On the rising edge of CLK at the end of the cycle in which this instruction is executing, the R1 register will be loaded with the data present on the SV port. The input to the Barrel Shifter will be passed onto the output unshifted. |
| LR2SV | <9> | On the rising edge of CLK at the end of the cycle in which this instruction is executing, the R2 register will be loaded with the data present on the SV port. The input to the Barrel Shifter will be passed onto the output unshifted. |
| ASRSV | <A> | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number present in the SV register. The LSBs are discarded, and the vacant MSBs are filled with duplicates of the original MSB. (Sign Extension). |
| ASRR1 | | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number resident within the R1 register. The LSBs are discarded, and the vacant MSBs are filled with duplicates of the original MSB. (Sign Extension). |
| ASRR2 | <C> | The 16 bit input to the Barrel Shifter is right shifted by the number of places indicated by the magnitude of the four bit number resident within the R2 register. The LSBs are discarded, and the vacant MSBs are filled with duplicates of the original MSB. (Sign Extension). |
| NRMXX | <D> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number output from the Priority Encoder. This value is also output on the SV port (provided SVOE is low). The effect of this operation is to left shift the input by the necessary amount (max 15 places) to result in the MSB and the next most significant bit being different. This has the effect of eliminating unnecessary Sign Bits, and hence Normalising the input data. The MSBs shifted out to the left are discarded, and the vacant LSBs on the right are filled with zeros. |
| NRMR1 | <E> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number output from the Priority Encoder. This value is also loaded into the R1 register at the end of the cycle, and is output on the SV port (provided SVOE is low). The effect of this operation is to left shift the input by the necessary amount (max 15 places) to result in the MSB and the next most significant bit being different. This has the effect of eliminating unnecessary Sign Bits, and hence Normalising the input data. The MSBs shifted out to the left are discarded, and the vacant LSBs on the right are filled with zeros. |
| NRMR2 | <F> | The 16 bit input to the Barrel Shifter is left shifted by the number of places indicated by the magnitude of the four bit number output from the Priority Encoder. This value is also loaded into the R2 register at the end of the cycle, and is output on the SV port (provided SVOE is low). The effect of this operation is to left shift the input by the necessary amount (max 15 places) to result in the MSB and the next most significant bit being different. This has the effect of eliminating unnecessary Sign Bits, and hence Normalising the input data. The MSBs shifted out to the left are discarded, and the vacant LSBs on the right are filled with zeros. |

Barrel Shifter or ALU Register Instructions

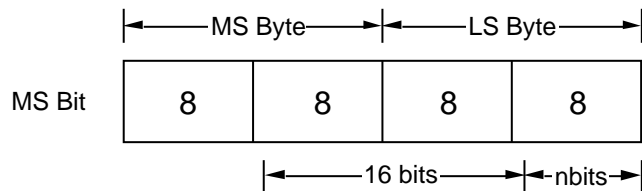
| Mnemonic | Op Code | Function |
|----------|---------|---|
| LLRRR | <0> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Right register will appear on the output. On the rising edge of CLK at the end of the cycle, and the data on the register inputs will be loaded into the Left Register. |
| LRRLR | <1> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Left register will appear on the output. On the rising edge of CLK at the end of the cycle, the data on the register inputs will be loaded into the Right Register. |
| LLRLR | <2> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Left register will appear on the output. On the rising edge of CLK at the end of the cycle, the data on the register inputs will be loaded into the Left Register. |
| LRRRR | <3> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Right register will appear on the output. On the rising edge of CLK at the end of the cycle, the data on the register inputs will be loaded into the Right Register. |
| LBRLR | <4> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Left register will appear on the output. On the rising edge of CLK at the end of the cycle, and the data on the register inputs will be loaded into both Left and Right Register. |
| NOPRR | <5> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Right register will appear on the output. On the rising edge of CLK at the end of the cycle no load operation will occur, the register contents will remain unchanged. |
| NOPLR | <6> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the contents of the Left register will appear on the output. On the rising edge of CLK at the end of the cycle no load operation will occur, the register contents will remain unchanged. |
| NOPPS | <7> | After the rising edge of CLK at the beginning of the cycle in which this instruction is executed, the input to the registers will appear on the output. On the rising edge of CLK at the end of the cycle no load operation will occur, the register contents will remain unchanged. |

PDSP1601/PDSP1601A

TYPICAL APPLICATION

Select a 16 bit field from each word in a block of 32 bit words with a 10MHz throughput.

The 16 bit field indicated is to be selected from each 32 bit word.



The 32 bit words are fed into the B port of the PDSP1601 in two cycles, MS byte first.

The PDSP1601 shift control is initiated by programming the R1 and R2 registers with n and 16-n respectively.

The shift operation is implemented in three steps:-

(1) The MS byte is logically left shifted (16-n) places, the MSBs being discarded and the LSB spaces being filled with zeros. This shifted data is loaded into the shifter register file left register.

(2) The LS byte is logically right shifted, n-places, the LSBs being discarded and the MSBs being filled with zeros. This shifted data is loaded into the shifter register file left register.

During this cycle the previous contents of this register are passed through the ALU to the ALU register file left register.

(3) While the MS byte of the next 32 bit word is shifted in the Barrel Shifter, the two previous results, resident within the left registers of the ALU and Shifter Register files are 'ORed' by the ALU, the result being the desired 16 bit field is loaded into the ALU register file right register ready to be output on the next cycle.

The instructions from initialisation are given in Table 6.

| CLK | CEB | MSA | MSB | MSS | MSC | IA | IS | SV | RA | RS | Comment |
|-----|-----|-------|-----|-----|-----|-------|-------|--------|-------|-------|--|
| 1/ | 1 | MARSX | 1 | 0 | 0 | CLRXX | X | X | NOPLR | NOPLR | Clear |
| 2/ | 1 | MARSX | 1 | 0 | 0 | PASXA | LR1SV | n | NOPLR | NOPLR | Load R1 with n |
| 3/ | 0 | MARSX | 1 | 0 | 0 | PASXA | LR2SV | (16-n) | NOPLR | NOPLR | Load R2 with (16-n) |
| 4/ | 0 | MARSX | 1 | 0 | 0 | PASXA | LSLR2 | X | NOPLR | LLRLR | Shift 1st MS byte |
| 5/ | 0 | MARSX | 1 | 0 | 0 | PASXA | LSRR1 | X | LLRRR | LLRLR | Shift 1st LS byte |
| 6/ | 0 | MARAX | 1 | 0 | 0 | ORXAB | LSLR2 | X | LRRLR | LLRLR | OR 1st bytes and shift 2nd MS byte |
| 7/ | 0 | MARSX | 1 | 0 | 0 | PASXA | LSRR1 | X | LLRRR | LLRLR | Shift 2nd LS byte |
| 8/ | 0 | MARAX | 1 | 0 | 0 | ORXAB | LSLR2 | X | LRRLR | LLRLR | and output first result Shift 3rd LS byte |

Repeat instruction pair 5/ and 6/ until all 16 bit fields have been selected.

Table 6

ABSOLUTE MAXIMUM RATINGS (Note 1)

| | |
|--|-------------------------|
| Supply voltage V_{CC} | -0.5V to 7.0V |
| Input voltage V_{IN} | -0.9 to $V_{CC} + 0.9V$ |
| Output voltage V_{OUT} | -0.9 to $V_{CC} + 0.9V$ |
| Clamp diode current per pin I_k (see note 2) | $\pm 18mA$ |
| Static discharge voltage (HMB) | 500V |
| Storage temperature T_s | -65°C to +150°C |
| Ambient temperature with power applied T_{amb} | |
| Military | -40°C to +125°C |
| Industrial | -40°C to +85°C |
| Package power dissipation P_{TOT} | |
| AC | 1000mw |
| LC | 1000mw |

NOTES

- Exceeding these ratings may cause permanent damage. Functional operation under these conditions is not implied.
- Maximum dissipation or 1 second should not be exceeded, only one output to be tested at any one time.

THERMAL CHARACTERISTICS

| Package type | J_C °C/W | J_A °C/W |
|--------------|------------|------------|
| AC | 12 | 36 |
| LC | 12 | 35 |

ELECTRICAL CHARACTERISTICS

Operating Conditions (unless otherwise stated)

T_{AMB} (Commercial) = 0°C to +70°C, V_{CC} = 5.0V±5%, Ground = 0V
 T_{AMB} (Industrial) = -40°C to +85°C, V_{CC} = 5.0V±10%, Ground = 0V
 T_{AMB} (Military) = -55°C to +125°C, V_{CC} = 5.0V±10%, Ground = 0V

Static Characteristics

| Characteristic | Symbol | Value | | | Units | Conditions |
|------------------------|-----------------|-------|------|------|-------|---|
| | | Min. | Typ. | Max. | | |
| Output high voltage | V _{OH} | 2.4 | | | V | I _{OH} = 8mA I _{OL} = -8mA |
| Output low voltage | V _{OL} | | | 0.4 | V | |
| Input high voltage | V _{IH} | 3.5 | | | V | GND < V _{IN} < V _{CC} T _{amb} = -40°C to +85°C GND < V _{OUT} < V _{CC} V _{CC} = Max |
| Input low voltage | V _{IL} | | | 0.5 | V | |
| Input leakage current | I _{IL} | -10 | | +10 | µA | |
| Vcc current | I _{CC} | | | 60 | mA | |
| Output leakage current | I _{OZ} | -50 | | +50 | µA | |
| Output S/C current | I _{SC} | 12 | | 80 | mA | |
| Input capacitance | C _{IN} | | 5 | | pF | |

Switching Characteristics

| Characteristic | Value | | | | Units | Conditions |
|---|----------|------|------------|------|-------|--|
| | PDSP1601 | | PDSP1601/A | | | |
| | Min. | Max. | Min. | Max. | | |
| CLK rising edge to C-PORT | 5 | 40 | 5 | 25 | ns | 2 x LSTTL + 20pF 1 x LSTTL + 5pF 1 x LSTTL + 5pF |
| CLK rising edge to CO | 5 | 100 | 5 | 50 | ns | |
| CLK rising edge to BFP | 5 | 100 | 5 | 50 | ns | |
| Setup CEA or CEB to CLK rising edge | 30 | | 15 | | ns | Input mode Input mode 20pF load, SV O P mode 2 x LSTTL + 20pF 2 x LSTTL + 20pF 2 x LSTTL + 20pF 2 x LSTTL + 20pF |
| Hold CEA or CEB after CLK rising edge | 40 | 0 | 20 | 0 | ns | |
| Setup A or B port inputs to CLK rising edge | 40 | 0 | 20 | 0 | ns | |
| Hold A or B port inputs after CLK rising edge | 40 | 0 | 20 | 0 | ns | |
| Setup MSA0-1, MSB, MSS, MSC, RA2-0, RS0-2, IA0-4, ISO-3, to CLK rising edge | | 0 | | 0 | ns | |
| Hold RS0-2, IA0-4 after CLK rising edge | | 3 | | 3 | ns | |
| Hold ISO-3 after CLK rising edge | | 0 | | 0 | ns | |
| Hold MSA0-1, MSB, MSS, MSC, RA0-2 after CLK rising edge | 40 | 3 | 20 | 3 | ns | |
| Setup SV to CLK rising edge | 5 | 100 | 5 | 50 | ns | |
| Hold SV after CLK rising edge | | 40 | | 25 | ns | |
| CLK rising edge to SV | | 40 | | 25 | ns | |
| OE | | 40 | | 25 | ns | |
| OE | | 40 | | 25 | ns | |
| OE | | 40 | | 25 | ns | |
| OE | | 40 | | 25 | ns | |
| Clock period (ALU & Barrel Shifter, serial mode) | 200 | | 100 | | ns | |
| Clock period (ALU & Barrel Shifter, parallel mode) | 100 | | 50 | | ns | |
| Clock high time | 40 | | 20 | | ns | |
| Clock low time | 40 | | 20 | | ns | |

- NOTES
1. LSTTL is equivalent to I_{OH} at 20µA I_{OL} of -0.4mA
 2. Current is defined as negative into the device.

PDSP1601/PDSP1601A

ORDERING INFORMATION

PDSP1601 AO AC 10MHz Military - PGA package
PDSP1601 MC GGCR 10MHz MIL883 Screened - QFP package

PDSP1601A BO AC 20MHz Industrial - PGA package



HEADQUARTERS OPERATIONS
MITEL SEMICONDUCTOR
Cheney Manor, Swindon,
Wiltshire SN2 2QW, United Kingdom.
Tel: (01793) 518000
Fax: (01793) 518411

MITEL SEMICONDUCTOR
1500 Green Hills Road,
Scotts Valley, California 95066-4922
United States of America.
Tel (408) 438 2900
Fax: (408) 438 5576/6231

Internet: <http://www.gpsemi.com>

CUSTOMER SERVICE CENTRES

- **FRANCE & BENELUX** Les Ulis Cedex Tel: (1) 69 18 90 00 Fax : (1) 64 46 06 07
- **GERMANY** Munich Tel: (089) 419508-20 Fax : (089) 419508-55
- **ITALY** Milan Tel: (02) 6607151 Fax: (02) 66040993
- **JAPAN** Tokyo Tel: (03) 5276-5501 Fax: (03) 5276-5510
- **KOREA** Seoul Tel: (2) 5668141 Fax: (2) 5697933
- **NORTH AMERICA** Scotts Valley, USA Tel: (408) 438 2900 Fax: (408) 438 5576/6231
- **SOUTH EAST ASIA** Singapore Tel:(65) 3827708 Fax: (65) 3828872
- **SWEDEN** Stockholm Tel: 46 8 702 97 70 Fax: 46 8 640 47 36
- **TAIWAN, ROC** Taipei Tel: 886 2 25461260 Fax: 886 2 27190260
- **UK, EIRE, DENMARK, FINLAND & NORWAY**
Swindon Tel: (01793) 726666 Fax : (01793) 518582

These are supported by Agents and Distributors in major countries world-wide.

© Mitel Corporation 1998 Publication No. DS3705 Issue No. 2.3 September 1996

TECHNICAL DOCUMENTATION – NOT FOR RESALE. PRINTED IN UNITED KINGDOM

This publication is issued to provide information only which (unless agreed by the Company in writing) may not be used, applied or reproduced for any purpose nor form part of any order or contract nor to be regarded as a representation relating to the products or services concerned. No warranty or guarantee express or implied is made regarding the capability, performance or suitability of any product or service. The Company reserves the right to alter without prior notice the specification, design or price of any product or service. Information concerning possible methods of use is provided as a guide only and does not constitute any guarantee that such methods of use will be satisfactory in a specific piece of equipment. It is the user's responsibility to fully determine the performance and suitability of any equipment using such information and to ensure that any publication or data used is up to date and has not been superseded. These products are not suitable for use in any medical products whose failure to perform may result in significant injury or death to the user. All products and materials are sold and services provided subject to the Company's conditions of sale, which are available on request.

All brand names and product names used in this publication are trademarks, registered trademarks or trade names of their respective owners.