silicon systems®
A TDK Group Company

SSI 32H6830
Servo DSP

Advance Information

December 1993

DESCRIPTION

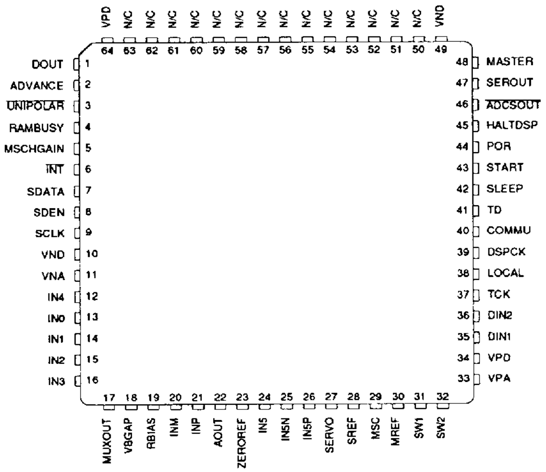
The SSI 32H6830, designated as the SEEKER™, contains a DSP, a 10-bit ADC, two 10-bit DACs, and sufficient I/O pins to perform the servo and MSC functions of a hard disk drive with no overhead to the master microprocessor. When fully programmed, the SEEKER™ performs self-contained seek, track, and spindle control functions. When the device is given a track destination, it will seek to the desired track, generate an interrupt when it arrives on track, and then servo on the track. As a spindle controller, the SEEKER™ will start the motor, spin it up to a speed indicated by the microprocessor, and generate an interrupt when proper speed has been achieved. The spindle controller is also capable of phase locking to a master spindle index and maintaining a specified phase with respect to the master index. The DSP will allow more sophisticated algorithms with better phase margin than those implemented with a microprocessor. The program and constants for the DSP are stored in the microprocessor ROM and are uploaded to the DSP. The part is optimized for use with the SSI 32H6810A motor speed commutator and servo amplifier device.

The SEEKER™ offloads all sector rate processing from the microprocessor and allows it to spend nearly 100% of its time dealing with the controller chip; an essential feature as data rates approach 48 Mbit/s. It also allows better algorithms to be implemented with less phase delay. This results in faster, quieter seek times and higher track bandwidths.

FEATURES

- **Self contained seek, track, spindle start, spindle run, and spindle sync capability**
- **Can operate at a multiple of the sector rate to reduce latency time between seek command and start of seek**
- **DSP with 16 x 16 multiply in 200 ns**
- **10-bit 2 µs ADC with 6 input MUX**
- **Two 10 bit DACs**
- **Serial µP port for uploading program memory from system µP**
- **Interfaces with pulse detectors containing on board peak detectors such as SSI 32P548 and SSI 32P4730**

PIN DIAGRAM



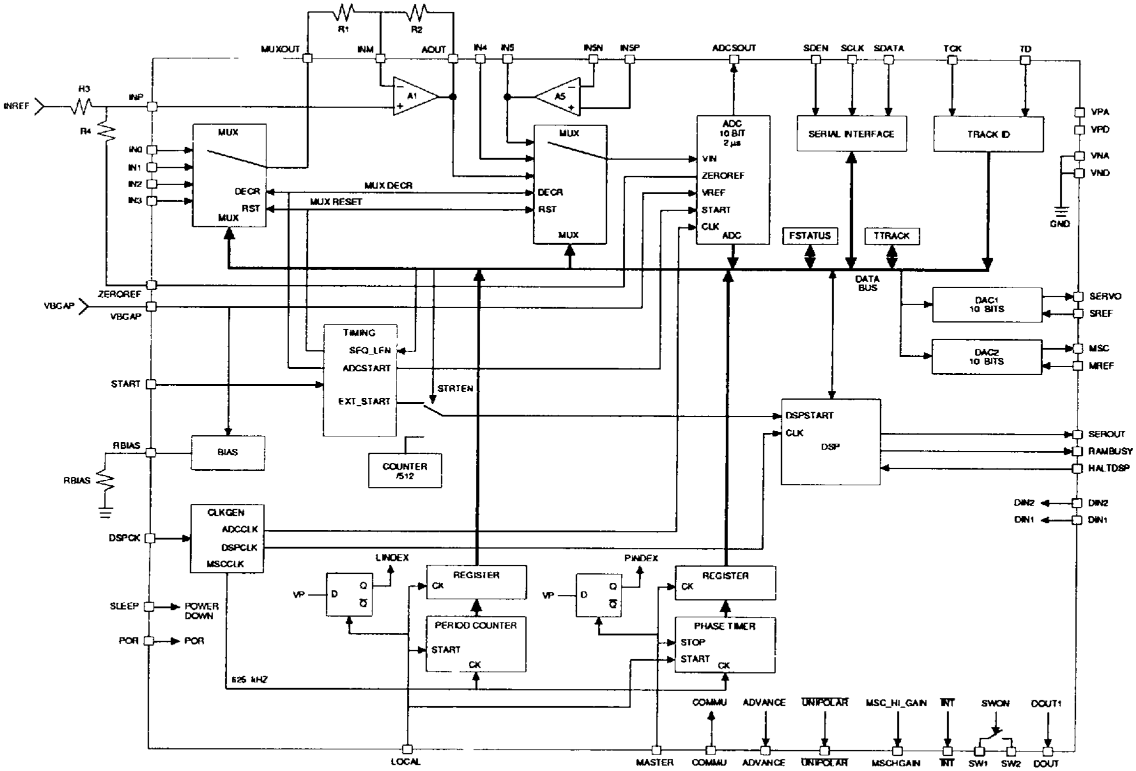
64-Lead TQFP

CAUTION Use handling procedures necessary for a static sensitive component

1293 - rev

8253965 0010063 798 SIL

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

HARDWARE FUNCTIONAL BLOCKS

Front End

The SEEKER™ front end consists of a 10-bit two's complement ADC, a 6-input MUX and some amplifiers whose gain can be set with external components. The front end is intended to be driven by pulse detectors like the SSI 32P548 that have internal peak detectors or integrators. External resistors perform gain and offset correction.

Front End Timing

A start pulse sets the MUX address to five, raises RAMBUSY, and initiates a sequence of consecutive ADC conversions. The result of each conversion is stored in the data RAM (DRAM) address 0 through 5, respectively. The MUX inputs are converted in reverse numerical order for IN0 through IN3. When the conversions are complete, other status words such as the state of the period and phase timers, the target track, and the current track are inserted in the DSP DRAM and a DSP code pass is started.

The start pulse can be generated by hardware (the START pin or an internal counter) or software (STARTBIT in FSTATUS.) If STRTEN (in FSTATUS) is high, the start signal is taken from the START pin. If it is low, the internal counter generates a start signal every 512 DSP clocks. STARTBIT generates a start regardless of any other pin or bit.

DSP Timing

When the DSP is started, the ADC values, DSPIN, TTRACK, PHTIME, PTIME, and TRACK have already been loaded in DRAM. These values will not be updated again until the next start pulse occurs. When the DSP reaches a STOP instruction, it halts, lowers RAMBUSY and waits for another start signal.

Track ID

The current track ID is demodulated and converted to binary by external circuitry. It is then supplied to the SEEKER™ through a dedicated serial port. The track ID is transferred to DRAM address 10 when a DSP start pulse is received. Depending on the TRKMSB bit in the FSTATUS word, track ID is received either LSB or MSB first. Since the track ID register can handle up to 16 bits, the unused bits can be utilized as flag bits from the demodulator. Some possible uses for the flag bits are: warning the DSP that certain sector data as bad, and in oversampling applications, indicating to the DSP which start pulses are at the beginning of a sector.

Spindle Control Hardware

The MSC portion of the seeker will support start, run, and synchronized spindle modes. The MSC hardware consists of a period timer and a phase timer. The period counter starts on the rising edge of LOCAL and transfers its count to a latch when the next rising edge occurs. A status bit, LINDEX, is set whenever new data is available at the latch. The status bit will be high for one DSP code pass. The phase timer is also started on every LOCAL rising edge. It is stopped when MASTER rises. A status bit, MINDEX, indicates when new data from the phase timer is available. By comparing the value of the period timer and the phase timer, the actual phase error between local and master can be determined. The DSP controls spindle start by monitoring COMMU and asserting ADVANCE. See the SSI 32H6810 or equivalent data sheet for more details.

µP Serial Port

Through the serial port, the µP can read FSTATUS and any DRAM or IRAM memory word. It can write the FSTATUS and TTRACK registers as well as any DRAM or IRAM word. The registers are internally double buffered and can be accessed by the µP at any time. Access to RAM must be limited to when the DSP is idle. The format of the serial port data string is consistent with other Silicon Systems serial interfaces.

DAC1 and DAC2

These DACs are 10 bits wide. They are memory mapped to DRAM address 2 and 3 for DAC 1 and 2, respectively. Each DAC has its own zero-reference input. The full scale swing of each DAC is ±VBGAP/2. These DACs are intended to drive the servo and spindle buffers.

TTRACK Memory

This 16 bit word is accessed by the µP serial port. It is double buffered so the µP does not need to synchronize to RAMBUSY. When TTRACK is programmed to a new target track, the next DSPSTART pulse will transfer it to DRAM and cause a seek to begin.

FUNCTIONAL DESCRIPTION (continued)

FSTATUS Memory

This 16-bit word can be read and written by the µP. It is double buffered so the µP does not need to synchronize to RAMBUSY.

Interrupt

The $\overline{\text{INT}}$ output is controlled by the DSP. It is typically used to indicate an event of interest to the µP such as spindle control achieving lock, or the head arriving at its target, or at other times, spindle control losing lock or the head falling off track. Upon receiving the interrupt, the µP should read FSTATUS to determine the interrupt type. Reading FSTATUS clears the $\overline{\text{INT}}$ output.

SEROUT

This is an output serial port used for diagnostic purposes. Whenever the DSP writes to this port, the bits are shifted out at the DSP clock rate. SEROUT normally sits high. A leading zero prefixes all outputs as a marker. This pin is useful for monitoring internal data words while the DSP is operating in real time.

Software Interface to µP

The µP is able to read or write any word in the IRAM or DRAM. In addition, it is able to read and write the FSTATUS register, and to write the TTRACK register.

FSTATUS Register—Read

The first 4 bits are interrupt status bits. They are written by the DSP and cause an interrupt to be initiated whenever they change state. The second four bits are information bits and do not affect $\overline{\text{INT}}$. The third group of 4 bits are interrupt flags, indicating which of the interrupt status bits caused an interrupt. The last 4 bits are for version control and future reserved functions. Note that whenever FSTATUS is read, $\overline{\text{INT}}$ is deasserted and the interrupt flags are cleared.

Bit	Name	Description
0	Spare_INT	When this bit changes state, $\overline{\text{INT}}$ is asserted.
1	Ontrack	Indicates that the head is on track. When this bit changes state, $\overline{\text{INT}}$ is asserted.
2	At_speed	Indicates that the spindle is at speed. When this bit changes state, $\overline{\text{INT}}$ is asserted.
3	Phase_lock	Indicates that the local index is phase locked to the master index. When this bit changes state, $\overline{\text{INT}}$ is asserted.
4	Track/Seek	Indicates if the DSP is in track mode.
5	RAMBUSY	Indicates the DSP RAM is servicing the DSP and is "locked out" of the µP serial interface.
6	DSTAT11	Bit 11 of DSPSTATUS
7	DSTAT12	Bit 12 of DSPSTATUS
8	INTF0	Is set when bit 0 changes state. Is cleared when FSTATUS is read.
9	INTF1	Is set when bit 1 changes state. Is cleared when FSTATUS is read.
10	INTF2	Is set when bit 2 changes state. Is cleared when FSTATUS is read.
11	INTF3	Is set when bit 3 changes state. Is cleared when FSTATUS is read.

FSTATUS Register—Read (continued)

Bit	Name	Description
12	REV0	LSB of revision number.
13	REV1	MSB of revision number.
14	RESERVED	
15	RESERVED	

FSTATUS Register—Write

0	STRTEN	Selects DSPSTART from the timing block instead of the divide-by-512 counter.
1	TRKMSB	Sets the TRACK ID serial to parallel converter to "MSB first" mode.
2	HALTBIT	Serves the same purpose as the HALTDSP pin. When asserted, the DSP will continue executing its current code pass and then will remain idle, ignoring both external and internal hardware start pulses. This bit should be set by the µP during initialization of the SEEKER™.
3	STARTBIT	Is a third way of creating a "start." When HALTBIT is asserted, this is the only way to create a "start." If this bit is programmed to 1, the usual sequence of ADC conversions and data transfers to DRAM is initiated. The DSP code pass will not be initiated, however. Instead the DSP will wait for SS pulses. STARTBIT is automatically cleared after it is written.
4	SS	This bit is ignored except if HALTBIT is asserted. In that case, every "one" written to SS causes the DSP to advance one clock cycle. SS is automatically cleared after it is written.
5	RESETBIT	Resets the SEEKER™ to the state where the µP serial interface has complete access. This bit should be set by the µP before initializing the SEEKER™.
6	RESUMEBIT	Terminates the single step mode by resuming DSP execution at full speed.
7	FS7	A spare bit for communication from µP to DSP.

FUNCTIONAL DESCRIPTION (continued)

Serial Port

The serial port is designed to be shared by other devices. For this reason, a device ID is included in the preamble. The device ID conforms to the Silicon Systems standard: 1-R/W, 2-Pulse Det, 3-Filter, 4-Data Sep, 5-ENDEC, 6-Time Base, 7-Servo/MSC. Bit 0 of the serial port is received first. Each bit string received by the serial port can be unlimited in length but consists of the following initial fields:

Bit #	Field	Description
0	R/W	Indicates whether data is to be read or written.
1..3	Device ID	Identifies the device being programmed (LSB is bit 1). Must be '7' to communicate with this part.
4..5	Type	Indicates which memory is addressed (LSB is bit 4): 0 FSTATUS 1 TTRACK (write only) 2 DRAM or ACCUMULATOR 3 IRAM or PROGRAM COUNTER
6..7	Address bank	The bank address for memories with more than 256 words. These bits are ignored if FSTATUS or TTRACK is being accessed. Note that if bank is 11, type 2 and 3 become ACCUMULATOR and PROGRAM COUNTER respectively.
8.115	Address	The RAM address (LSB is bit 8). This field must always be sent except when accessing FSTATUS or TTRACK.
16.. or 8..	Data	The data (LSB first). If RAM data is being read or written, consecutive data words can be concatenated. The address will automatically increment after each 16 (DRAM) or 20 (IRAM) bits. The address bank is automatically incremented when necessary.

DSP Memory Map

The DSP communicates with the SEEKER^{TM} chip through memory mapped regions of DRAM. The first 10 words are mapped to various hardware resources as defined below. Note that the first four words are "write protected." The remaining words are initialized at the beginning of each DSP code pass and may then be modified or overwritten by the DSP.

DRAM Address	Read by DSP	Written by DSP
0	ADC5 (write protected)	DSPSTATUS
1	ADC4 (write protected)	SEROUT
2	ADC3 (write protected)	DAC1
3	ADC2 (write protected)	DAC2
4	ADC1	
5	ADC0	
6	DSPIN	
7	TTRACK	
8	PTIME (period time)	
9	PHTIME (phase time)	
10	TRACK	

DSPSTATUS and DSPIN

DSPIN is the word the DSP uses to communicate with bits set by the µP or by SEEKER^{TM} input pins. DSPSTATUS is the word the DSP uses to control bits read by the µP and external pins controlled by the DSP. Whenever the first 4 bits of DSPSTATUS are changed by the DSP, $\overline{\text{INT}}$ is asserted. The bits in DSPIN and DSPSTATUS are defined below.

BIT	DSPIN (Read by DSP)	DSPSTATUS (Written by DSP)
0	COMMU (from COMMU pin)	SPARE_INT (to FSTATUS)
1	LINDEX (from period timer)	ONTRACK (to FSTATUS)
2	MINDEX (from phase timer)	ATSPEED (to FSTATUS)
3	DIN1 (from DIN1 pin)	PHLOCK (to FSTATUS)
4	DIN2 (from DIN2 pin)	TRACK/SEEK (to FSTATUS)
5	LOCAL (from LOCAL pin)	$\overline{\text{UNIPOLAR}}$ (to pin)
6	MASTER (from MASTER pin)	MSCHGAIN (to pin)
7	FS7 (from FSTATUS)	ADVANCE (to pin)
8		DOUT (to pin)

FUNCTIONAL DESCRIPTION (continued)

DSPSTATUS and DSPIN (continued)

BIT	DSPIN (Read by DSP)	DSPSTATUS (Written by DSP)
9		SWON (to SW1, SW2 switch)
10		DSTAT11 (to FSTATUS)
11		DSTAT12 (to FSTATUS)
12		not used
13		not used
14		not used
15		not used

PIN DESCRIPTION

The following description lists each pin, associates a pin type to it, and provides a brief description of the pin's function.

NAME	TYPE	DESCRIPTION
VPA, VPD	VCC	Analog and digital power supplies.
VNA, VND	GND	Analog and digital grounds.
IN0, IN1, IN2, IN3	Ana In	The four primary inputs automatically converted by the A/D. These inputs drive a low resistance analog switch.
MUXOUT	Ana Out	The output of the IN0..IN3 mux.
INM	Ana In	The inverting input to amplifier A1.
AOUT	Ana Out	The output of amplifier A1.
INP	Ana In	The non-inverting input to A1.
VBGAP	Ana In	The voltage reference input. This will determine the full scale swing of the ADC and the two DACs.
ZEROREF	Ana Out	The ADC zero reference output. Can be used by the A1 resistor network to level shift IN0 through IN3.
IN4	Ana In	A direct input to the ADC mux.
IN5	Ana Out	The output of amplifier A5.
IN5N, IN5P	Ana In	The inputs to amplifier A5.
RBIAS	Ana Out	A resistor to ground from this pin sets the bias current for the analog circuitry.
SERVO	Ana Out	The output of the servo DAC.
SREF	Ana In	The servo DAC reference. The DAC output swing will be SREF-0.5•VBGAP to SREF+0.5•VBGAP.
MSC	Ana Out	The output of the MSC DAC.
MREF	Ana In	The MSC DAC reference. The DAC output swing will be MREF-0.5•VBGAP to MREF+0.5•VBGAP.

NAME	TYPE	DESCRIPTION
SW1, SW2	Ana In	The two terminals of an uncommitted analog switch. The switch is controlled by the SWON bit in the DSP STATUS word.
ADCSOUT	Dig Out	A test point from the ADC. This test point is connected to the output of the ADC comparator.
START	Dig In	A rising edge on START initiates consecutive ADC conversions and causes a pass through the DSP code to begin.
DSPCK	Dig In	The master clock (20 MHz) for the chip.
SLEEP	Dig In	Reduces the supply current of the chip. All analog circuitry is deactivated, with outputs becoming high impedence. The chip clock is deactivated. No data will be lost in RAM due to the use of static RAM.
POR	Dig In	Chip reset. This pin is for diagnostic purposes and should be grounded in normal operation.
SDEN	Dig In	The µP serial interface enable. SCLK and SDATA are ignored and the interface reset when SDEN is low.
SCLK	Dig In	The µP serial interface clock.
SDATA	Dig I/O	The µP serial interface data. This pin is an input except for the data cycles of a serial read request.
TCK	Dig In	The clock for the current track serial input.
TD	Dig In	The data for the current track serial input.
SEROUT	Dig Out	The serial output from the SEROUT word memory mapped into DRAM. This output can be used to monitor the DSP during real time applications.
RAMBUSY	Dig Out	When high, the DSP is executing code. When low, it is waiting for a START pulse.
HALTDSP	Dig In	When high, halts the DSP. Similar in function to HALTBIT in FSTATUS. This pin or HALTBIT should be asserted during µP access of DRAM and IRAM.
LOCAL	Dig In	The local index pulse. The period counter measures the time between LINDEX pulses.
MASTER	Dig In	The master index pulse. The phase timer measures the time from LINDEX to MINDEX.
COMMU	Dig In	The input from the BEMF comparator on the MSC chip.
ADVANCE	Dig Out	The output to the MSC chip.
$\overline{\text{UNIPOLAR}}$	Dig Out	An uncommitted DSP output bit that can be used to command the external MSC commutator to switch to unipolar mode.
MSCHGAIN	Dig Out	An uncommitted DSP output bit that can be used to command the external MSC commutator to switch gains.
$\overline{\text{INT}}$	Dig O/D	The interrupt output for the DSP.
DOUT	Dig Out	An uncommitted output bit that can be programmed by the DSP.
DIN1, DIN2	Dig In	Uncommitted input bit that can be read by the DSP.

ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Exposure to absolute maximum rating conditions for extended periods may cause permanent damage to the device or affect device reliability.

PARAMETER			RATING
Supply voltage VPA, VPD		Vdd	-0.3 to 7.0V
Pin voltage	Ana In	Vinai	-0.3 to VDD+0.3V
	Ana Out	Vinao	-0.3 to VDD+0.3V
	Dig In	Vindi	-0.3 to VDD+0.3V
	Dig Out	Vindo	-0.3 to VDD+0.3V
Storage temperature		Tstg	-65 to 150°C
Lead temperature (10 sec duration)		Tlead	0 to 300°C

RECOMMENDED OPERATING CONDITIONS

The recommended operating conditions for the device are indicated in the table below. Performance specifications do not apply when the device is operated outside the recommended conditions.

PARAMETER			RATING
Supply voltage		Vdd	4.5 to 5.5V
Ambient temperature		Ta	0 to 70°C
Capacitive load on digital outputs		Cl	50 pF
Analog output load			
	Cl		50 pF
	Rl		20 kΩ
System clock *f*c = 20 MHz			
	Freq. tolerance	*f*c	-0.01 to 0.01%
	Pulse width	Twh,Twl	20 to 30 ns
Biasing resistor, Rbias = 56.3 kΩ		Rbias	-5 to 5%
VBGAP tolerance, VBGAP = 2.25V		VBGAP	-5 to 5%

PERFORMANCE SPECIFICATIONS

SUPPLY CURRENT (FSTART = 5 kHz, DSP ACTIVE TIME = 25 µs)

PARAMETER	CONDITIONS	MIN	NOM	MAX	UNIT
VPA Idda				24	mA
VPD Idd				13	mA
VPA, Sleep mode Iddas				TBD	mA
VPD, Sleep mode Idds				TBD	mA

DIGITAL I/O

Digital input					
Vil		0.8			V
Vih				2.0	V
Iil,Iih				1	µA
Digital Output (except $\overline{INT}$)					
Vol	Iol = 2.0 mA			0.4	V
Voh	Ioh = -100 µA	Vdd-.4			V
Digital Output ($\overline{INT}$)					
Vol	Iol = 4.0 mA			0.4	V

SERVO D/A CONVERTER

Positive Full-scale voltage Digital=0x1FF			SREF+ VBGAP/2		V
Negative Full-scale voltage Digital=0x200			SREF- VBGAP/2		V
Resolution			10		bits
Digital Delay				4	µs
LSB voltage			VBGAP/ 1024		V
Differential nonlinearity				1	LSB
Offset				TBD	mV

ELECTRICAL SPECIFICATIONS (continued)

MSC D/A CONVERTER

PARAMETER	CONDITIONS	MIN	NOM	MAX	UNIT
Positive Full-scale voltage Digital = 0x1FF			MREF+ VBGAP/2		V
Negative Full-scale voltage Digital = 0x200			MREF- VBGAP/2		V
Resolution			10		bits
Digital Delay				4	µs
LSB voltage				VBGAP/ 1024	V
Differential nonlinearity				1	LSB
Offset				TBD	mV

ADC CONVERTER

Positive full-scale input Digital Output = 0x1 FF			VBGAP		V
Negative full scale input Digital Output = 0x200			VBGAP/9		V
Resolution			10		bits
Conversion time (includes MUX delay)				2	µs
LSB voltage			VBGAP/ 1152		V
Differential nonlinearity				1	LSB

AMPLIFIERS

Gain		50			dB
Unit gain bandwidth		1			MHz
Input offset		-20		20	mV
Output swing		0.2		3.5	V
Input common mode Range		0		Vdd	V
Settling time to 0.1%full scale step, inverting unity gain				1.8	µs

MUX

On Resistance				100	Ω

SW1, SW2 SWITCH

PARAMETER	CONDITIONS	MIN	NOM	MAX	UNIT
On Resistance				100	Ω

µP SERIAL INTERFACE PORT TIMING

PARAMETER		CONDITIONS	MIN	NOM	MAX	UNIT
SCLK						
Period	TCLK		50			ns
Low Time	TCKL		15			ns
High Time	TCKH		25			ns
SDEN						
Setup Time	TSENS		20			ns
Hold Time	TSENH		60			ns
SDATA						
Setup Time	TDS		5			ns
Hold Time	TDH		2			ns
Read Delay	TPD	CL = 0 to 50 pF	4		60	ns
Disable Delay	TSENDL				25	ns

µP TRACK ID PORT TIMING

PARAMETER		CONDITIONS	MIN	NOM	MAX	UNIT
TCK						
Period	TCLK		20			ns
Low Time	TCKL		10			ns
High Time	TCKH		10			ns
TD						
Setup Time	TDS		3			ns
Hold Time	TDH		2			ns

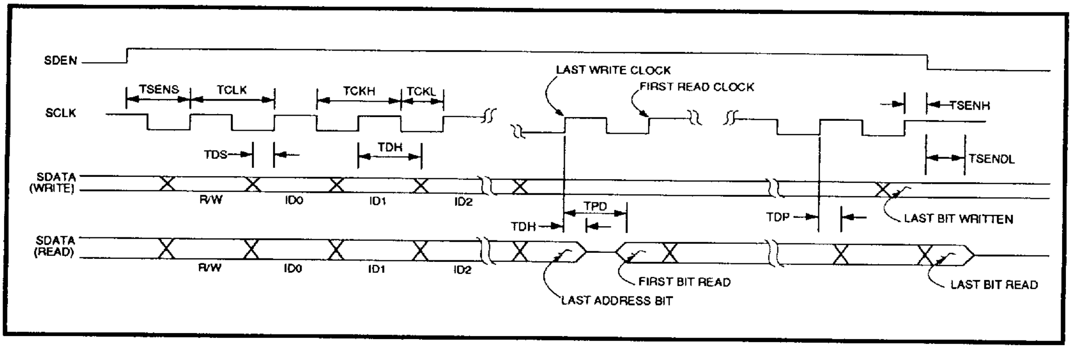


FIGURE 1: Serial Port Data Transfer Format

PROGRAMMER'S DESCRIPTION

This programmer's description of the SEEKER™ DSP contains a register level description of the DSP that is detailed enough to illustrate each opcode. It also contains a description of each opcode.

REGISTER LEVEL DESCRIPTION

The DSP consists of an arithmetic unit and an instruction unit. The arithmetic unit consists of a data RAM (DRAM), a shifter, an ALU, and an accumulator. The instruction unit consists of instruction RAM (IRAM) and a program counter. Figure 2 shows the contents of the DSP.

In a one-clock cycle, the DSP is capable of shifting a data word up to ±15 bits and adding it to the accumulator. A 4-cycle multiply is implemented in the multiply and accumulate functions.

DRAM

The DRAM stores data and coefficients for use by the DSP. The first 11 addresses of the DRAM are memory mapped to on-chip resources. In addition, a lookup command allows the DSP to use portions of DRAM as lookup tables.

DATA REG AND MULT REG

These registers store the data from the DRAM. This permits the DRAM to perform other operations during a multiply. A STORE or the setup cycle of an MLD or MADD instruction will be executed while a previous MLD or MADD instruction is completing.

SEQUENCER

This logic block controls the execution of instructions by monitoring the state of the accumulator, the DRAM, and the IRAM. When the resources required by the next instruction are available, the sequencer permits that instruction to execute.

FLAGS

Certain instructions cause one of the three flags to be updated with the sign bit of the accumulator. The flags are used by the instruction unit during conditional jumps.

SHIFTER

The shifter shifts the DATA REG word up to ±15 bits in one clock cycle. Unused left hand bits are sign extended and unused right hand bits are zero filled. The shifter has a 24-bit input and output width. The 16-bit word from DATA REG is sign extended 4 bits and padded with 4 zero bits below the LSB before entering the shifter.

ALU

The ALU is capable of performing an add, subtract, absolute value, XOR, AND, OR, and NEG function. The list of opcodes details the choices.

ACCUMULATOR

The accumulator is 16 bits with 4 extra LSB bits and 4 extra sign bits. The extra LSB bits minimize the rounding error when partial products are summed during a multiply. The top 4 sign bits are used as an aid in identifying overflow. They also are used in extended precision calculations where they are shifted to the least significant bits of the 16-bit accumulator.

MULTIPLIER DESCRIPTION

The multiplier returns the top 16 bits of a 16x16 product. An additional 4 LSBs are calculated to improve the truncation accuracy of the product. A RADIX register shifts the multiplicand (pointed to by DR) a certain number of bits to the left.

This multiplier facilitates the use of the two's complement fractional representation. For instance if RADIX is set to zero, each 16 bit number should be thought of as a signed fraction whose full scale value approaches ±1.0. The value of the fraction is calculated by the following equation:

$$\text{value} = -2^{0}b_{15} + 2^{\ 1}b_{14} + 2^{-2}b_{13} + \ldots + 2^{-15}b_{0}$$

Two such numbers multiplied together will yield an answer in the same format. For instance 0x4000 (0.5) multiplied by 0x4000 (0.5) results in 0x2000 (0.25).

Alternatively, if it is desired to represent numbers whose integer value can approach ±16, set RADIX to 4 and use the following formula:

$$\text{value} = -2^{4}b_{15} + 2^{3}b_{14} + 2^{2}b_{13} + \ldots + 2^{-11}b_{0}$$

With this format, 0x0400 (0.5) multiplied by 0x1800 (3.0) results in 0x0C00 (1.5).

OPCODE LIST

The following opcodes are implemented in the SEEKER^{TM} DSP. Parameters in [square] brackets are optional. Parameters enclosed in {curly} brackets form a list from which only one parameter may be chosen. Parameters may be placed in any order within all instructions except MLD and MADD. Opcodes, symbols and all options are case insensitive.

SHORT/LONG FORM DESCRIPTION

There is instruction memory capacity for 1K by 10-bit short form commands or 500 bytes by 20-bit long form commands. Short form is a 10-bit command which uses relative addressing, whereas long form is a 20-bit command with direct addressing.

DR, MR, and SR are three different pointers to DRAM which are updated with each reference. The assembler will insert short forms of each command if the pointer reference is close enough to its previous reference that relative addressing can be used. Every instruction has a short form except AND, OR, XOR, store commands that use /RET, and jump instructions conditional on F2 or F3. The relative address distance is listed in each opcode description.

If it is desired to force a long or short version of a particular command, an .L or .S suffix can be appended to the command. Alternatively, a short instruction can be forced by specifying a relative address such as '+1' or '+2'.

ALU INSTRUCTIONS WITH MULTIPLY

MLD *DramDR DramMR* [/F1]

Loads the data register with *DramDR* and the multiplier register with *DramMR* and initiates a multiply and load. The value in *DramDR* is shifted left by RADIX before being multiplied. The MLD instruction requires either 4 or 5 cycles to execute. The 5th cycle is not needed if the previous ALU instruction was an MLD or MADD and was followed by fewer than 4 non-ALU instructions. If F1 is specified, the F1 flag will be updated with the sign bit of the accumulator result. Relative address distance for both DR and MR addresses is +2, -1.

MADD *DramDR DramMR* [/F1]

Loads the data register with *DramDR* and the multiplier register with *DramMR* and initiates a multiply and add. The value in *DramDR* is shifted left by RADIX before being multiplied. The MADD instruction requires either 4 or 5 cycles to execute. The 5th cycle is not needed if the previous ALU instruction was an MLD or MADD and was followed by fewer than 4 non-ALU instructions. If F1 is specified, the F1 flag will be updated with the sign bit of the accumulator result. Relative address distance for both DR and MR addresses is +2, -1.

MULTIPLIER CONTROL INSTRUCTIONS

RADIX n [/RET]

Sets the amount of left shift "bias" to be performed on DR during MLD and MADD instructions. This command is effectively defining the location of the radix point in the MR word. The RET option will cause a subroutine return. The RADIX command is always short form and is the only short form instruction with a RET option.

ALU INSTRUCTIONS WITH SHIFT

LDS *DramDR* {/SHL=n, /SHR=n} [/F1] [/ABS]

Loads the contents of *DramDR* in the accumulator after being shifted as specified. Relative address distance is +2, -1. F1 indicates that the F1 flag is to be updated with the sign of the result. ABS indicates that the absolute value of *DramDR* is to be used. If F1 or ABS is specified, the long form of the instruction is used.

LDNS *DramDR* {/SHL=n, /SHR=n} [/F1]

Loads the negative contents of *DramDR* in the accumulator after being shifted as specified. Relative address distance is +2, -1. F1 indicates that the F1 flag is to be updated with the sign of the result. If F1 is specified, the long form of the instruction is used.

ADDS *DramDR* {/SHL=n, /SHR=n} [/ABS] [/F1]

Adds the contents of *DramDR* to the accumulator after being shifted as specified. F1 indicates that the F1 flag is to be updated with the sign of the result. ABS indicates that the absolute value of *DramDR* is to be used. If F1 or ABS is specified, the long form of the instruction is used. Relative address distance is +2, -1.

SUBS *DramDR* {/SHL=n, /SHR=n} [/ABS] [/F1]

Subtracts the contents of *DramDR* to the accumulator after being shifted as specified. F1 indicates that the F1 flag is to be updated with the sign of the result. ABS indicates that the absolute value of *DramDR* is to be used. If F1 or ABS is specified, the long form of the instruction is used. Relative address distance is +2, -1.

XSIGN *DramDR* {/SHL=n, /SHR=n} [/F1]

Multiplies the accumulator by the sign of *DramDR*—if *DramDR* is negative, the accumulator will be inverted. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

AND *DramDR* [/INV] {/SHL=n, /SHR=n}

ANDs the contents of *DramDR* to the accumulator after being inverted and shifted as specified. There is no short form of this instruction.

OR *DramDR* {/SHL=n, /SHR=n}

ORs the contents of *DramDR* to the accumulator after being shifted as specified. There is no short form of this instruction.

XOR *DramDR* {/SHL=n, /SHR=n}

XORs the contents of *DramDR* to the accumulator after being shifted as specified. There is no short form of this instruction.

ALU INSTRUCTIONS

LD *DramDR* [/ABS] [/F1]
Loads the contents of *DramDR* in the accumulator. If [ABS] is specified, the absolute value of *DramDR* is loaded. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

LDN *DramDR* [/F1]
Loads the negative contents of *DramDR* in the accumulator. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

ADD *DramDR* [/ABS] [/F1]
Adds the contents of *DramDR* to the accumulator. If [ABS] is specified, the absolute value of *DramDR* is added. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

SUB *DramDR* [/ABS] [/F1]
Subtracts the contents of *DramDR* from the accumulator. If [ABS] is specified, the absolute value of *DramDR* is subtracted. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

XSIGN *DramDR* [/F1]
Multiplies the accumulator by the sign of *DramDR*—if *DramDR* is negative, the accumulator will be inverted. If F1 is specified, the F1 flag is updated with the sign of the accumulator at the end of the command's execution. Relative address distance is +4, -3.

NOP
An arithmetic command that does nothing.It is sometimes useful before STO and conditional JMP commands. This command is always short.

LKUP
Loads the accumulator with the left justified DRAM value pointed to by the accumulator. The upper 8 bits of the accumulator are used as the DRAM address. The SAT module is always activated during LKUP. This command is always short.

STORE COMMANDS

STO *DramSR* [/RET] [{/F2, /F3}]
Store the accumulator in *DramSR*. If F2 or F3 is specified, the appropriate flag will be updated with the sign of the value being stored. Relative address distance is +2, -1. If RET is specified, the long form of this instruction is used.

STOSAT *DramSR* [/RET] [{/F2, /F3}]
Store the accumulator in *DramSR* with saturation logic enabled. If F2 or F3 is specified, the appropriate flag will be updated with the sign of the value being stored. Relative address distance is +2, -1. If RET is specified, the long form of this instruction is used.

STOLSW *DramSR* [/RET] [{/F2, /F3}]
Store the least significant word. This command stores the accumulator in *DramSR* and then shifts the sign bits right by 16 bits. The extra LSB bits are cleared. If F2 or F3 is specified, the appropriate flag will be updated with the sign of the value being stored. Relative address distance is +2, -1. If RET is specified, the long form of this instruction is used.

STODR *DramSR* [/RET] [{/F2, /F3}]
Stores the DATA REG contents in *DramSR*. This command permits fast data moves since the data does not have to flow through the accumulator pipeline. If F2 or F3 is specified, the appropriate flag will be updated with the sign of the value being stored. Relative address distance is +2, -1. If RET is specified, the long form of this instruction is used.

PROGRAM CONTROL

JMP *label:*

An unconditional jump. Relative address distance is +8, -7.

JSUB *label:*

An unconditional subroutine call. This is always a long instruction.

JF *label:* {/F1, /F2, /F3}

Jump if flag is one. Relative address distance is +8, -7. If F2 or F3 is specified, the long form is used.

JFB *label:* {/F1, /F2, /F3}

Jump if flag is zero. Relative address distance is +8, -7. If F2 or F3 is specified, the long form is used.

JALU

A computed jump. The jump address is taken from the bottom 9 bits of the accumulator. This is a short instruction.

STOP

Stops execution of the program. The program restarts on a rising edge of DSPSTART pulse.

COMMAND SEQUENCING

Although the SEEKER™ DSP is a pipeline architecture, instruction sequencing is unaffected except in one case.

Commands that depend on accumulator results such as STORE and conditional jump must allow one cycle to occur after the accumulator instruction completes. If the accumulator instruction requires more than one cycle to complete (MLD or MADD), the STORE or conditional jump must be placed one ALU instruction after the accumulator command. This allows the result to propagate into the accumulator before it is stored.

Example 1: Calculate C=A•B and E=A•B+D.

```
MLD   A B   ;load accumulator with A•B
ADD   D     ;add D to accumulator
STO   C     ;store A•B in C
STO   E     ;store A•B+D in E
```

Example 2: Calculate C=A•B and E=A•B+D•F.

```
MLD   A B   ;load accumulator with A•B
MADD  D F   ;add D•F to accumulator
STO   C     ;store A•B in C
NOP         ;NOP is an ALU instruction
STO   E     ;store A•B+D•F in E
```

CYCLE COUNTING

All instructions execute in one cycle except MLD and MADD instructions which require a setup cycle and 4 ALU cycles. During a multiply, the sequencer will execute up to 4 non-ALU instructions until it encounters another ALU command. Non-ALU commands include JUMP, STORE, and the setup cycle of MLD and MADD instructions. Thus it is often possible to store results and setup the next multiply without consuming clock cycles.

Example 3: Calculate a 2 pole Chebyshev low pass:

$$v_{out} = z^{-1}(v_x + K_3 v_{out})$$

$$v_x = v_x z^{-1} + v_{in}K_1 - v_{out}K_2$$

```
LD     VX        ;(1 clock) load vx
MADD   VIN K1    ;(5 clock) OLDVX + K1•VIN
MADD   VOUTNK2   ;(4 clock) the setup for this
                 instruction occurs during the
                 previous instruction.
MADD   VOUT K3   ;(4 clock) the setup for this
                 instruction occurs during the
                 previous instruction.
STO    VX        ;(0 clock) VX=OLDVX +
                 K1•VIN + K2•VOUT
LD     VX        ;(1 clock) begin new
                 calculation
STO    VOUT      ;(1 clock) VOUT=VX +
                 K3•VOUT
```

Thus this biquad requires 16 cycles to execute. Note that NK2 = -K2.

ASSEMBLER INPUT FILE

The assembler input file contains both DRAM initialization information and IRAM instructions. A typical structure for the file is:

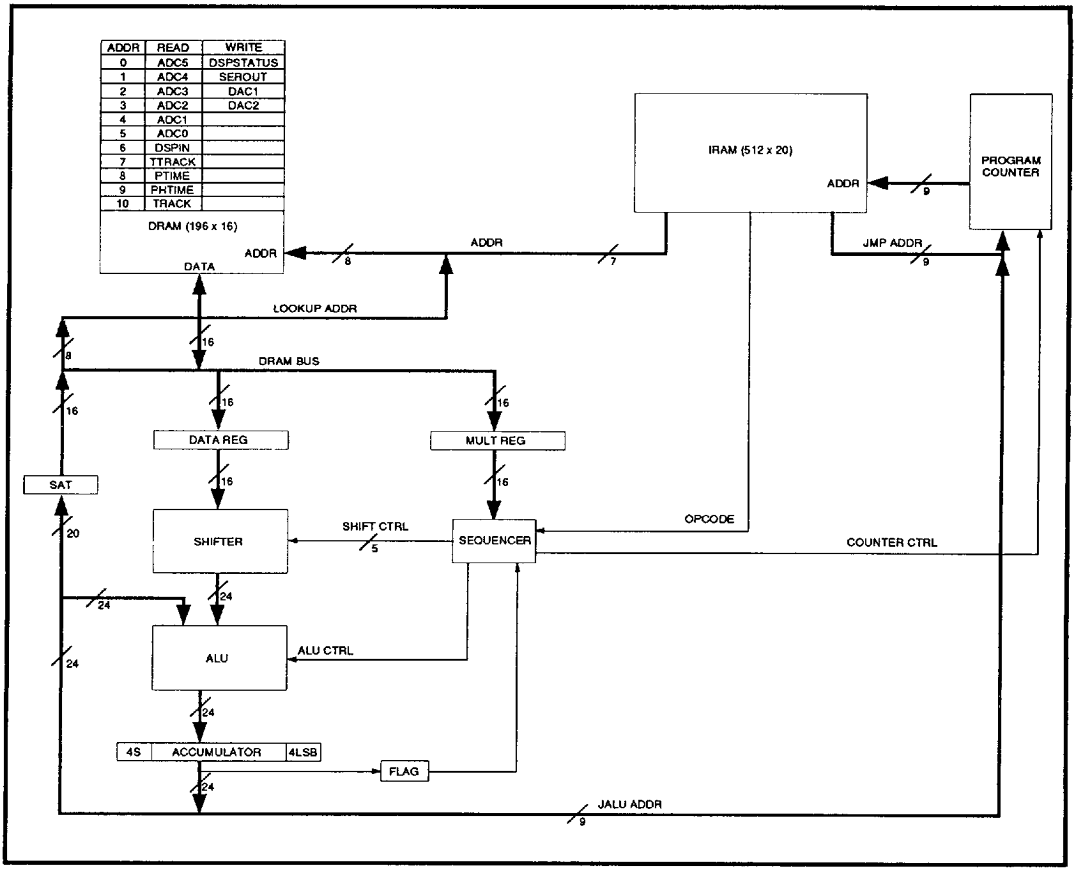
```
;*****************************sample input file********************

;sample.asm
;comments extend from ; to end of line
;
.dorg 0                 ;indicates the beginning of a DRAM section which will specify addresses
                        ;beginning with 0.  If no value is specified, the addressing will start where the last
                        ;.dorg section ended.

zero:    data 0         ;values can be expressed in decimal
qrtr:    data 0x2000    ;or hex
x2:      data 256
         data
         data           ;DRAM labels are optional
prod:    data 0
adc1:
dac1:    data 0         ;more than one label can refer to the same address
temp:    data jump1     ;a DRAM value can be another DRAM or IRAM label.

.org                ;indicates the beginning of an IRAM section.  As in the .dorg case, a value can be
                    ;specified to indicate where in the IRAM the code is to be inserted.

lds      qrtr /shl=1    ;load qrtr into accumulator after being shifted by 1 bit to the left
radix    1              ;indicate that multipliers have the radix point 1 bit from left
sto      x2             ;store 0x4000 into x2
madd     adc1 qrtr  /f1 ;multiply adc1 times qrtr times two (due to RADIX), update f1
nop                     ;wait one arith cycle before result is valid for sto or jmp
jf /f1   jump1          ;jump if f1 is one
sto      prod           ;sto prod if positive
jmp      end
jump1:                  ;labels can be on a line by themselves
sto      dac1           ;if neg, send prod to dac1
end:     stop
;******************************end of sample file******************
```



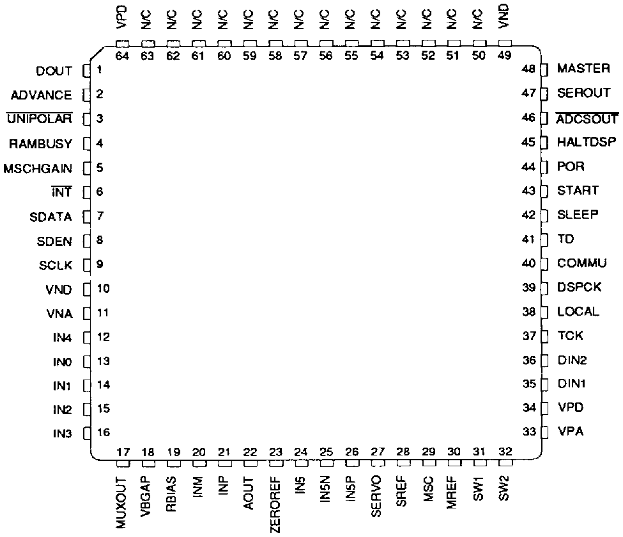
ADDR	READ	WRITE
0	ADC5	DSPSTATUS
1	ADC4	SEROUT
2	ADC3	DAC1
3	ADC2	DAC2
4	ADC1	
5	ADC0	
6	DSPIN	
7	TTRACK	
8	PTIME	
9	PHTIME	
10	TRACK	

FIGURE 2: Programmer's Model

PACKAGE PIN DESIGNATIONS

(Top View)

CAUTION: Use handling procedures necessary for a static sensitive component.



64-Lead TQFP

Advance Information: Indicates a product still in the design cycle, and any specifications are based on design goals only. Do not use for final design.

No responsibility is assumed by Silicon Systems for use of this product nor for any infringements of patents and trademarks or other rights of third parties resulting from its use No license is granted under any patents, patent rights or trademarks of Silicon Systems. Silicon Systems reserves the right to make changes in specifications at any time without notice Accordingly, the reader is cautioned to verify that the data sheet is current before placing orders

The SEEKER™ is a registered trademark of Silicon Systems, Inc

Silicon Systems, Inc. 14351 Myford Road, Tustin, CA 92680 7022 (714) 573-6000, FAX (714) 573-6914