# SSI 32C452
## Storage Controller

**Preliminary Data**

July, 1990

## DESCRIPTION

The SSI 32C452 Storage Controller is a CMOS device that provides the basis for an intelligent Winchester disk drive controller capable of non-interleaved data transfers at rates up to 20 Mbit/s. When combined with a microprocessor, memory and a buffer management device such as the SSI 32C453, the SSI 32C452 implements a powerful and cost-efficient peripheral controller solution. It also has the flexibility to be used in SCSI systems.

The SSI 32C452 includes a control sequencer with a writeable control store, and configuration/status registers which can be programmed to support standard and custom interface protocols for storage controllers. Access to the control store and registers is accomplished through the microprocessor interface which is optimized for 8 bit, multiplexed address/data bus processors such as the 8085 . It also has the flexibility to interface with most standard 8-bit microprocessors. This organization allows the controller firmware to be stored in an EPROM or the host and down-loaded to the SSI 32C452, and means wide flexibility of the control functions performed by the device.

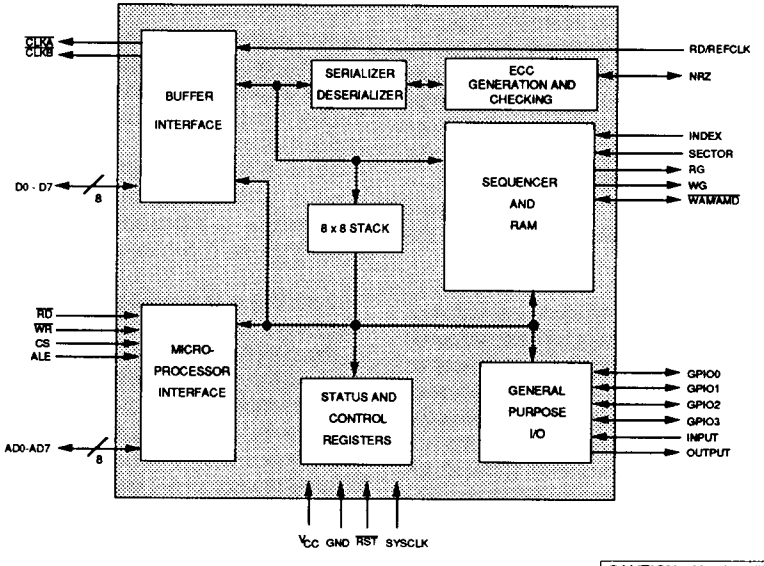
(Continued)

## FEATURES

- **Supports ST506/412, ST412HP, SA100, SMD, ESDI and custom interfaces**
- **Operates with 16 MHz microprocessors**
- **Internal RAM-based control sequencer**
- **Internal user programmable ECC to 32 bits**
- **Non-interleaved data transfer to 20 Mbit/s**
- **Hard or soft sector formats**
- **Programmable sector lengths up to a full track**
- **High performance, low power CMOS device**
- **Plug and software compatible with AIC-010F Storage Controller**
- **Single 5 volt supply**
- **Available in 44-pin PLCC or 40-pin DIP package**

7

## BLOCK DIAGRAM



## DESCRIPTION (Continued)

The SSI 32C452 performs all the controller functions for the peripheral device, such as serialization/deserialization, ECC generation and checking on the data stream. It also handles overhead information such as address marks, gaps and sector ID fields. If an ECC error is detected during a read, the syndrome is saved so that defects can be corrected. The ECC polynomial and register length can be programmed or bypassed entirely so that external ECC hardware can be used.

## FUNCTIONAL DESCRIPTION

The major functional elements and data paths of the SSI 32C452 are shown in the block diagram.

The SSI 32C452 performs the functions to interface a serial data storage device such as a Winchester disk drive, to a parallel bus interface for data processing on a byte wide basis. The functions necessary to accurately make this conversion are serialization/deserialization, error detection and correction, and data path control. The SSI 32C452 also has general purpose interface lines to further facilitate control of the data storage device or parallel interface. An eight byte stack allows data to be saved and reviewed by the microprocessor for error handling purposes. The internal sequencer performs most of the operations in conjunction with the control and status registers. The sequencer program is contained in internal sequencer RAM, which is easily (re)programmed providing almost infinite flexibility in communcations protocols and control features. A microprocessor effects both initialization and control of the SSI 32C452 by writing to and reading from the internal registers, sequencer RAM, stack and general purpose I/O circuitry. The microprocessor interface block of the SSI 32C452 provides the communication and control for the SSI 32C452 to the microprocessor.

The **buffer interface** includes a bidirectional data bus that exchanges data bytes between an external buffer memory and the serializer/deserializer. It generates two clocks, $\overline{\text{CLKA}}$ and $\overline{\text{CLKB}}$ which control all accesses to the buffer memory. All buffer memory cycles must be synchronous with $\overline{\text{CLKA}}$, which is derived from the RD/REFCLK input during data transfers and from SYSCLK otherwise. The internal register CLKCON contains control bits which define the relationship between these source clocks and $\overline{\text{CLKA}}$. The $\overline{\text{CLKB}}$ signal is asserted whenever a new data byte must be transferred (ie. when the serializer/deserializer is full during a read operation or empty during a write operation). The direction of the transfer is determined from the state of the read gate (RG) and write gate (WG) lines. A $\overline{\text{CLKB}}$ cycle is used to force the buffer control device (eg. an SSI 32C453) to reserve the next buffer memory access for the SSI 32C452, since peripheral transfers take precedence over the asynchronous host transfers. In order to allow host transfers to keep up with peripheral transfers, the $\overline{\text{CLKA}}$ rate selected should be at least twice the word transfer rate of the peripheral.

The **microprocessor interface** decodes microprocessor read and write requests and provides access to the appropriate register or internal memory location. Since both data and address information are carried on the multiplexed bus lines AD0-AD7, address information is latched from the bus on the falling edge of the microprocessor signal ALE (address latch enable). When $\overline{\text{CS}}$ is asserted along with either $\overline{\text{RD}}$ or $\overline{\text{WR}}$, the register whose address was previously latched is selected. The addresses and names of all the accessible registers are shown in the Register Address Map, Figure 1. The microprocessor should not read or write the sequencer RAM while the sequencer is running, since there is no circuitry to resolve conflicting accesses and incorrect sequencer operation will result.

The **status and control registers** make status information available to the microprocessor and allow the device to be configured for a wide variety of peripheral control applications. The microprocessor can monitor the status of transfers in progress and control the ECC register operation, the ECC polynomial, the clock generation hardware and the sequencer program execution. The microprocessor also has access to the sequencer's microprogram RAM so that it loads the microcode for all controller operations.

The **serializer/deserializer** circuit interfaces the parallel buffer memory bus to the serial NRZ data stream of the peripheral device. Byte synchronism is maintained with a bit ring, which is an 8 bit recirculating shift register clocked by the peripheral bit clock. During a sector write, the bit ring is initialized explicitly with a sequencer instruction. The bit ring continues to operate until the end of the field (ECC written or read) and causes $\overline{\text{CLKB}}$ to be asserted once for each data byte to be transferred. During write operations, the sequencer

may cause address marks and sync patterns to be loaded into the serializer instead of data bytes. These special patterns are contained in a sequencer instruction and are transferred to the serializer over an internal byte wide data path. During read operations, bytes of overhead information may be routed to the stack or sequencer for comparison against target values. This process is controlled by the control field (SEQCONF) in each sequencer instruction.

The eight byte recirculating **stack** may be used to capture read data for later examination by the microprocessor. Data is pushed onto the stack under sequencer control. The control bit STACKEN in the sequencer instruction field SEQCONF directly controls the stack. If more than 8 bytes are written to the stack, only the last 8 will be saved. When a data byte is read from the top of the stack by the microprocessor via the STACK register, the data is recirculated to the bottom of the stack, allowing the stack contents to be examined more than once without the use of temporary storage in the microprocessor or buffer.

Serial peripheral data is passed through a variable length shift register with programmable exclusive OR feedback that performs **ECC generation and checking**. The feedback taps for the desired ECC polynomial are selected in the four registers POLY0 - POLY24 and the polynomial length is determined by the LEN bits in ECCCON. In addition, the ECC register may be operated either under sequencer or microprocessor control. During read operations, the contents of the ECC register are compared to the actual ECC field read from the peripheral. If there is a mismatch, the error syndrome is available for error correction. The ECC polynomial may be reversed to allow hardware computation of the error location, relieving the microprocessor of the burden of this lengthy calculation. During writes to the peripheral, the computed ECC word can be appended to each data or address field. The sequencer data type field (SEQDATF) indicates when ECC bytes are to be written or checked during a peripheral transfer.

The **sequencer** controls the time critical operations of the SSI 32C452. It executes programs stored in the 28 word by 32 bit sequencer RAM, and can be pro-

7

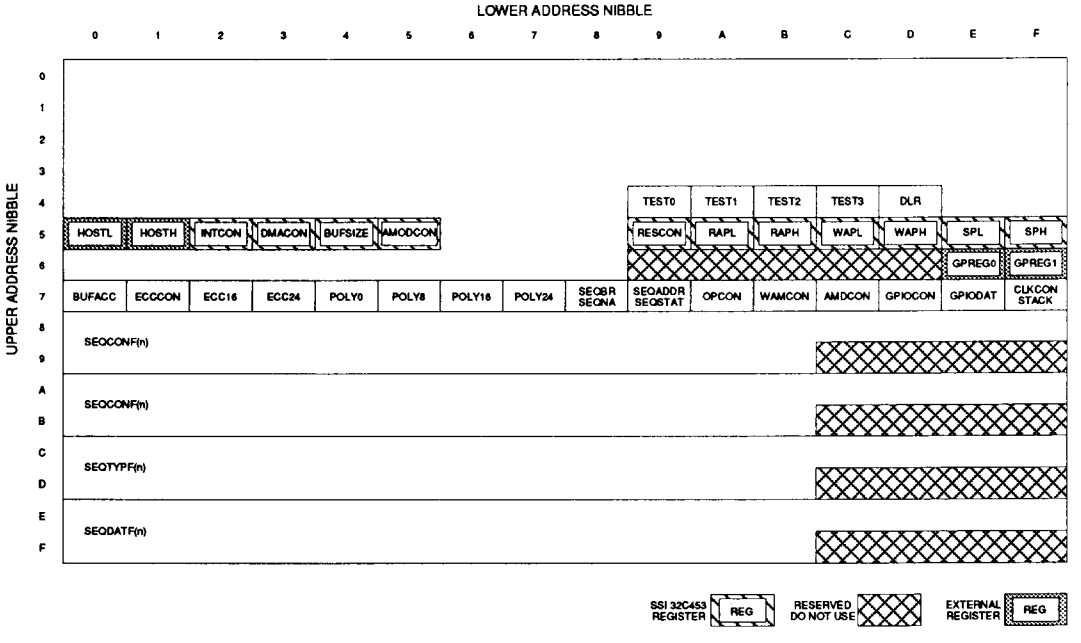



**FIGURE 1: Register Address Map**

## FUNCTIONAL DESCRIPTION (continued)

grammed to support hard and soft sectored read, write, search and verify operations for a wide variety of Winchester disk drives and other peripherals. The sequencer RAM is loaded by writing to the sequencer instruction registers as outlined in the Sequencer Instructions section of this data sheet. Each instruction is comprised of four bytes. Each of the four bytes represents a function of the sequencer operation. They are address field, control field, data type field, and data field. The organization of these fields is shown in the Register Bit Map, Figure 2. The Sequencer Registers provide control from and status to the microprocessor and sequencer. They contain branch, next and start addresses, and sequencer status information. The SEQUENCER STATUS register provides informaton on the sequencer state such as whether an ECC error occurred, a compare equal or low occurred, if the branch condition or address mark is active, or whether the sequencer is halted.

The **general purpose I/O** section has four general purpose I/O lines GPIO0 - GPIO3, and the INPUT pin which are accessible through the internal general purpose input/output registers. They are available for user defined functions such as Winchester disk or host interface control. The functionality of the GPIO0 - GPIO3 pins is programmed in the GPIOCON and GPIODAT registers. They can act as I/O's asserted or read through the GPIODAT register, or they can be programmed to decode microprocessor access to addresses 6EH and 6FH eliminating the need for external decode. The INPUT signal can be programmed in the SEQADRF RAM (registers) to affect sequencer operation and the state of the pin read from the GPIODAT register. The other general purpose line, OUTPUT is controlled directly by the sequencer to synchronize it with external circuitry. The OUT bit of the GPIODAT register reflects the state of the output pin.

## PIN DESCRIPTION

### GENERAL

| NAME | DIP | PLCC | TYPE | DESCRIPTION |
|---|---|---|---|---|
| VCC | 40 | 1 | | POWER SUPPLY +5 volts |
| GND | 20-21 | 22 | | GROUND |
| $\overline{RST}$ | 12 | 13 | I | RESET - Active low signal halts the sequencer, sets output pins RG, WG, $\overline{WAM}$ and NRZ low, forces the GPIO pins into a high impedance state and resets a number of the registers as described below. |
| SYSCLK | 13 | 14 | I | SYSTEM CLOCK - Clock input in the range of 1.5 MHz to 16 MHz |

### MICROPROCESSOR INTERFACE

| | | | | |
|---|---|---|---|---|
| ALE | 1 | 2 | I | ADDRESS LATCH ENABLE - Falling edge latches register address from AD0-7 pins. |
| CS | 29 | 33 | I | CHIP SELECT- Active high signal enables device to respond to microprocessor read or write. |

## PIN DESCRIPTION (continued)

| NAME | DIP | PLCC | TYPE | DESCRIPTION |
|---|---|---|---|---|
| $\overline{WR}$ | 30 | 34 | I | WRITE STROBE - Active low signal causes the data on the address/data bus to be written to the addressed register if CS is also active. |
| $\overline{RD}$ | 31 | 35 | I | READ STROBE - Active low signal causes the contents of the addressed register to be placed on the address/ data bus if CS is also active. |
| AD0-AD7 | 39-32 | 43-36 | I/O | ADDRESS/DATA BUS - 8 bit bus which carries register address information and bi-directional data. These pins are high impedance when not in use. |
| **GENERAL PURPOSE I/O** | | | | |
| GPIO0-3 | 4-7 | 5-8 | I/O | GENERAL PURPOSE I/O LINES - These lines can be programmed as an inputs or outputs which are accessed though the GPIODAT register. They may also be programmed to serve as active low outputs which decode microprocessor accesses to the following locations:<br><br>I/O pin — Alternate output decode<br>GPIO0 — Write to 6EH<br>GPIO1 — Read from 6EH<br>GPIO2 — Write to 6FH<br>GPIO3 — Read from 6FH |
| INPUT | 8 | 9 | I | INPUT PIN - This dedicated input line may be read through the GPIODAT register or tested directly by the control sequencer. |
| OUTPUT | 9 | 10 | O | OUTPUT PIN - Dedicated output line which is derived directly from the control sequencer instruction field. |
| **DISK DRIVE INTERFACE** | | | | |
| INDEX | 10 | 11 | I | INDEX PULSE - Active high disk drive index pulse input, must be at least one byte time long. |
| SECTOR | 11 | 12 | I | SECTOR PULSE - Active high sector pulse input from disk drives that are hard sectored, must be at least one byte time long. |
| RG | 14 | 15 | O | READ GATE - Active high output from control sequencer enables external phase-locked loop (PLL) to synchronize to read data stream from the storage device. |

## PIN DESCRIPTION (continued)

### DISK DRIVE INTERFACE (continued)

| NAME | DIP | PLCC | TYPE | DESCRIPTION |
|---|---|---|---|---|
| WG | 15 | 16 | O | WRITE GATE - Active high output from control sequencer indicates valid write data to the storage device. |
| RD/REFCLK | 26 | 30 | I | READ/REFERENCE CLOCK - This input must be externally multiplexed to provide the PLL clock when read gate is active and the write oscillator clock at all other times. This pin must always be driven with a clock signal, even when $\overline{RST}$ is active. |
| NRZ | 27 | 31 | I/O | NRZ DATA - This bi-directional pin provides write data when WG is active, and must be driven with read data when RG is active. Data must be in the NRZ format. |
| $\overline{WAM}/\overline{AMD}$ | 28 | 32 | I/O | WRITE ADDRESS MARK/ADDRESS MARK DETECT - This bi-directional pin is used to write and detect address marks. When WG is active, a low level output of one bit time on this pin indicates that an address mark must be written. When RG is active, the peripheral must provide an active low input to indicate the detection of an address mark. |

### BUFFER INTERFACE

| | | | | |
|---|---|---|---|---|
| $\overline{CLKA}$ | 2 | 3 | O | CLOCK A - Clock signal which initiates host or controller accesses to the buffer memory on its falling edge. When either RG or WG is active, this output is derived from RD/REFCLK. At all other times it is derived from SYSCLK. The clock source is divided by 2 or 4 as programmed in the CLKCON register. |
| $\overline{CLKB}$ | 3 | 4 | O | CLOCK B - This clock is used to reserve $\overline{CLKA}$ cycles for SSI 32C452 data transfers. An active low pulse spanning a falling edge of $\overline{CLKA}$ indicates that the next falling edge on $\overline{CLKA}$ will be used by the SSI 32C452 to access the buffer memory. |
| D0-D7 | 16-19<br>22-25 | 18-21<br>25-28 | I/O | BUFFER DATA BUS - Bi-directional data bus that carries data to and from the buffer memory. Bus cycles are controlled by $\overline{CLKA}$ and $\overline{CLKB}$. Direction of the transfer is determined by RG and WG. Note: refer to pin diagram for exact ordering of the pins. |
| No connects on PLCC package: 17, 23, 24, 29, 44 | | | | |

| REGISTER | ADDRESS | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | READ/ WRITE |
|---|---|---|---|---|---|---|---|---|---|---|
| TEST0 | 49H | SEQUENCER NEXT ADDRESS FIELD | | | | | | | | R |
| TEST1 | 4AH | SEQUENCER CONTROL FIELD | | | | | | | | R |
| TEST2 | 4BH | SEQUENCER COUNT/DATA TYPE FIELD | | | | | | | | R |
| TEST3 | 4CH | SEQUENCER DATA FIELD | | | | | | | | R |
| DLR | 4DH | DATA LATCH REGISTER | | | | | | | | R |
| BUFACC | 70H | BUFFER MEMORY BYTE | | | | | | | | R/W |
| ECCCON | 71H | LEN1 | LEN0 | RESET | SECTBR | CLRECC | FEEDINH | ECCSHIFT | ECCIN | R/W |
| ECC16 | 72H | ECC23 | ECC22 | ECC21 | ECC20 | ECC19 | ECC18 | ECC17 | ECC0/16 | R |
| ECC24 | 73H | ECC31 | ECC30 | ECC29 | ECC28 | ECC27 | ECC26 | ECC25 | ECC24 | R |
| POLY0 | 74H | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 | R/W |
| POLY8 | 75H | F15 | F14 | F13 | F12 | F11 | F10 | F9 | F8 | R/W |
| POLY16 | 76H | F23 | F22 | F21 | F20 | F19 | F18 | F17 | F16 | R/W |
| POLY24 | 77H | UNUSED | F30 | F29 | F28 | F27 | F26 | F25 | F24 | R/W |
| SEQBR | 78H | UNUSED | | | BRADR4 | BRADR3 | BRADR2 | BRADR1 | BRADR0 | W |
| SEQNA | 78H | TEST POINTS | | | NADR4 | NADR3 | NADR2 | NADR1 | NADR0 | R |
| SEQADDR | 79H | UNUSED | | | STADR4 | STADR3 | STADR2 | STADR1 | STADR0 | W |
| SEQSTAT | 79H | AMACTIVE | DATATRANS | BRACTIVE | STOPPED | UNUSED | ECCERR | COMPLO | COMPEQ | R |
| OPCON | 7AH | CARRYINH | UNUSED | TRANSINH | SEARCHOP | SYNDET | NRZDAT | SECTORP | INDEXP | R/W |
| WAMCON | 7BH | AM7 - AM0 | | | | | | | | R/W |
| AMDCON | 7CH | AMD7 - AMD0 | | | | | | | | R/W |
| GPIOCON | 7DH | RGFSEL | WGFSEL | RGESEL | WGESEL | GPDIR3 | GPDIR2 | GPDIR1 | GPDIR0 | R/W |
| GPIODAT | 7EH | UNUSED | | OUT | INP | GP3 | GP2 | GP1 | GP0 | R/W |
| CLKCON | 7FH | CLKF2 | CLKF1 | UNUSED | CLKFO | CLKINH | SYN2 | SYN1 | SYN0 | W |
| STACK | 7FH | TOP OF STACK | | | | | | | | R |
| SEQADDRF | 80H<br>9BH | BRCON2 | BRCON1 | BRCON0 | NEXT4 | NEXT3 | NEXT2 | NEXT1 | NEXT0 | R/W |
| SEQCONF | A0H<br>BBH | SETWG | SETRG | RESWG | STACKEN | NRZINH | OUTPIN | COMPEN | DATEN | R/W |
| SEQTYPF | C0H<br>DBH | CNT7/ DTYP2 | CNT6/ DTYP1 | CNT5/ DTYP0 | CNT4 | CNT3 | CNT2 | CNT1 | CNT0 | R/W |
| SEQDATF | | DATA FIELD | | | | | | | | R/W |

**FIGURE 2: Register Bit Map**

## REGISTER DESCRIPTION

The microprocessor which controls the system has access to all the SSI 32C452 registers and sequencer RAM through its external memory address space. The SSI 32C452 and its companion device, the SSI 32C453 Dual Port Buffer Controller, are designed to occupy a single 256 byte page. The 8 bit page address is latched from pins AD0-AD7 on a falling edge of ALE and remains valid until the next ALE falling edge.

The external registers described at the end of this section are not implemented in either the SSI 32C452 or SSI 32C453, and are assumed to be implemented in external hardware. These external registers are not required for use with the SSI 32C452, but are included as applications information.

### ECC REGISTERS

The core of the ECC circuit is a 32 bit shift register whose effective length may be programmed to be 16, 24 or 32 bits. This is accomplished in hardware by directing the input data to stage 16, 8 or 0 of the ECC shift registers, ECC16 and ECC24, while its output is always bit 31, which is bit ECC31 of register ECC24.

The ECC polynomial to be implemented is programmed by the user into the ECC feedback registers, POLY0, POLY8, POLY16 and POLY24. Each bit in these registers enables or disables exlusive OR feedback to the output of the corresponding shift register stage. The feedback signal is the exclusive OR of the serial data stream with the output of shift register stage 31. An override bit in ECCCON forces normal shift register operation, regardless of the settings of the feedback control bits.

When WG or RG are active, the ECC shift register input is the serial read or write data and the shift clock is RD/REFCLK. When an ECC word is being written, feedback is disabled and the shift register output is substituted for the data stream. At other times the microprocessor may set the ECCIN bit explicitly and cause a single shift register clocking to occur. For further information on implementing an ECC polynomial see the Applications Information Section at the end of this data sheet.

**ECCCON** **71H** **Read/Write**

| ECC CONTROL WORD | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0 | ECCIN | ECC SERIAL INPUT - When both RG and WG are inactive, this bit becomes the input bit for the ECC shift register. The RD/REFCLK must always be active for correct operation of the device. |
| 1 | ECCSHIFT | ECC SHIFT CONTROL - When both RG and WG are inactive, a single shift of the ECC register will occur when this bit is set. It is automatically cleared again when the shift is complete. |
| 2 | FEEDINH | ECC FEEDBACK INHIBIT - When this bit is set all feedback is inhibited and the ECC register functions as a simple shift register of the selected length. |
| 3 | CLRECC | CLEAR ECC - If this bit is set when either RG or WG are active, the ECC syndrome will be cleared at the end of the read/write operation. If both are inactive, the syndrome will be cleared immediately. |
| 4 | SECTBR | ENABLE SECTOR BRANCH - If the sequencer "branch on index or sector" instruction is executed and SECTBR is set, the sequencer will recognize the branch condition as true if either the INDEX or the SECTOR pin is active. If SECTBR is cleared, then the sequencer will only recognize the branch condition if the INDEX pin is active. |

**ECC REGISTERS** (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 5 | RESET | CHIP RESET - When this bit is set, the SSI 32C452 will be held in its reset state. This bit is set when $\overline{RST}$ is true. |
| 6-7 | LEN0-LEN1 | ECC REGISTER LENGTH - These two bits select the ECC register length as follows:<br>LEN1 LEN0<br>0 0 16 bit register<br>0 1 24 bit register<br>1 0 illegal combination<br>1 1 32 bit register |
| Reset State: ECCCON= 20H (ie. RESET=1) | | |

**ECC16** **72H** **Read only**

| ECC DATA | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0 | ECC0/16 | ECC REGISTER LEADING BITS - This bit reflects the OR of all the ECC register bits from the input stage through bit 16. For 16 bit operation, this is bit 16. For 24 bit operation this is bit 8 + bit 9 + .. + bit 16. For 32 bit operation, this is bit 0 + bit 1 + ... + bit 16. |
| 1-7 | ECC17-ECC23 | ECC REGISTER BITS - These bits reflect the output of ECC shift register stages 17 to 23. |
| Reset State: Unknown | | |

**ECC24** **73H** **Read only**

| ECC DATA | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | ECC24-ECC31 | ECC REGISTER BITS - These bits reflect the output of ECC shift register stages 24 to 31. |
| Reset State: Unknown | | |

**ECC REGISTERS** (continued)

**POLY0** **74H** **Read/Write**

| ECC POLYNOMIAL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | F0-F7 | ECC POLYNOMIAL FEEDBACK - These bits enable or disable exclusive OR feedback of both the shift register output (bit 31) and the serial input to the output of shift register stages 0 to 7. These settings may be overriden by the FEEDINH bit in ECCCON. For ECC register lengths of 16 or 24 bits, F0-F7 are irrelevant. |
| Reset State: POLY0=00H | | |

**POLY8** **75H** **Read/Write**

| ECC POLYNOMIAL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | F8-F15 | ECC POLYNOMIAL FEEDBACK - These bits enable or disable exclusive OR feedback to the output of shift register stages 8 to 15. For register lengths of 16 bits, F8-F15 are irrelevant. |
| Reset State: POLY8=00H | | |

**POLY16** **76H** **Read/Write**

| ECC POLYNOMIAL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | F16-F23 | ECC POLYNOMIAL FEEDBACK - These bits enable or disable exclusive OR feedback to the output of shift register stages 16 to 23. |
| Reset State: POLY16=00H | | |

**POLY24** **77H** **Read/Write**

| ECC POLYNOMIAL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-6 | F24-F30 | ECC POLYNOMIAL FEEDBACK - These bits enable or disable exclusive OR feedback to the output of shift register stages 24 to 30. |
| 7 | unused | |
| Reset State: POLY24=00H | | |

## SEQUENCER STATUS AND CONTROL REGISTERS

The sequencer controls all the time-critical interactions with the peripheral storage device being controlled by the SSI 32C452. The instructions directly control disk drive interface lines, provide data for writing or comparison, determine the number of bytes handled and control the sequence of instruction execution. It is programmed by the user for maximum capability and variability. There are 28 instructions which are 32 bits wide. They are divided in to 4 byte wide fields. These fields are sequencer address, control, data type and data fields. These may be further divided into subfields as described in detail below. Examples are shown in the Applications Information section at the end of this data sheet.

The next address field of the sequencer instruction contains address and branching information. Each instruction is executed for the duration of the number of byte times specified in its count field. The specified count is loaded into a down counter which clocks every 8 bit times. When the counter underflows execution of that instruction is terminated. A carry inhibit feature allows the counter to wrap around to a full count for fields which are more than 256 bytes long. Execution is passed to the instruction at the specified next address, unless a branch condition is specifed in the instruction (e.g., ECC error or successful data comparison). In that case, execution passes to the address specified in the SEQBR register. Sequencer operation may also be conditionally stopped. The sequencer will always stop if execution passes to address 1FH, which is outside of the 28 word instruction control store.

The control field of the sequencer instruction is used to specify the state of RG and WG, to move data to the stack and to select data transfer or data comparison operations. The count field sets the duration of each instruction in byte times and is also used to select the type of data written, such as address marks or ECC bytes.

**SEQBR** **78H** **Write only**

| SEQUENCER BRANCH ADDRESS | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-4 | BRADR0 - BRADR4 | BRANCH ADDRESS BITS - When a sequencer instruction with a branch condition is finished (i.e., the specified number of byte times have elapsed) and the specified condition did occur, execution will resume at this 5 bit address. |
| 5-7 | unused | |
| Reset State: Unknown | | |

**SEQNA** **78H** **Read only**

| SEQUENCER NEXT ADDRESS | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-4 | NADR0 - NADR4 | NEXT ADDRESS BITS- This reflects the 5 bit next address field of the sequencer instruction currently being executed. After the specified byte count, execution will proceed at this address provided no branch conditions occur. |
| 5-7 | | Internal test points |
| Reset State: Unknown | | |

**SEQUENCER STATUS AND CONTROL REGISTERS** (continued)

**SEQADDR 79H Write only**

| SEQUENCER START ADDRESS | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-4 | STADR0 - STADR4 | SEQUENCER START ADDRESS BITS - If the sequencer is currently halted, writing this register with an address in the range 00H to 1BH will cause sequencer execution to commence at that address. If this register is written with 1FH , the sequencer will halt. |
| 5-7 | unused | |
| Reset State: 00H | | |

**SEQSTAT 79H Read only**

| SEQUENCER STATUS | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0 | COMPEQ | COMPARE EQUAL - When a sequencer instruction enables the comparison operation, this bit reflects the result of all the byte comparisons performed (i.e., if it is set then all bytes compared so far have been equal.) If RG is enabled, the comparisons occur between the instruction's data field and the data bytes being read (or buffer memory if the SEARCHOP bit in OPCON is true as well). |
| 1 | COMPLO | COMPARE LOW - Similar to COMPEQ, except that it indicates that in all comparisons the data field was smaller than the compared byte. |
| 2 | ECCERR | ECC ERROR - This bit is set during RG active, upon reading the last ECC bit, if there was an error in the data read. The error syndrome will be stored in the ECC registers. |
| 3 | not used | |
| 4 | STOPPED | SEQUENCER STOPPED - This bit is set when the sequencer is stopped and its instruction address is 1FH. |
| 5 | BRACTIVE | BRANCH ACTIVE - This is set when the branch condition specified in the current instruction has been satisfied. This means that the next address used will be taken from the SEQBR register. This bit is reset when the microprocessor reads this register. |
| 6 | DATATRANS | DATA TRANSFER - This bit is set when the current sequencer instruction is causing data to be transferred between the buffer memory and the peripheral device. This distinguishes the activity from a search or verification operation. |
| 7 | AMACTIVE | ADDRESS MARK ACTIVE - This bit is set when the controller reads or writes an address mark or sync byte. It is reset after the ECC bytes are read or written, or when the sequencer is halted. |
| Reset State: 00H | | |

### SEQUENCER INSTRUCTION REGISTERS

The 4 fields of 8 bits comprising a single sequencer instruction are detailed below. They are presented as arrays of 28 bytes each, corresponding to the 28 instructions at sequencer addresses 0 to 1BH.

**SEQADRF(n) 80H-9BH Read/Write**

| SEQUENCER ADDRESS FIELD ARRAY | | | |
|---|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** | |
| 0-4 | NEXT0-NEXT4 | NEXT ADDRESS FIELD - This 5 bit field specifies the address of the next instruction to be executed when the current instruction has continued for the specified number of bytes. | |
| 5-7 | BRCON0 -BRCON2 | BRANCH CONTROL FIELD - This 3 bit field specifies the branch condition for the current instruction. When a branch condition is satisfied, execution of the current instruction is not curtailed. It continues to execute for the full byte count specifed, and then the sequencer proceeds with execution of the address specified in SEQBR. The branch condition used depends on the state of RG and data type field (see SEQTYPF). If RG is true and ECC bytes are being read, the following branch conditions apply: | |
| | | BRCON2/1/0= | 000 No branch |
| | | | 001 Stop on ECC error |
| | | | 010 Stop on comparison error |
| | | | 011 Stop on ECC or comparison error |
| | | | 100 Branch on good ECC and comparison |
| | | | 101 Branch on ECC error |
| | | | 110 Branch on comparison error |
| | | | 111 Branch on ECC or comparison error |
| | | Otherwise, the branch conditions are: | |
| | | BRCON2/1/0= | 000 No branch |
| | | | 001 Stop if INPUT pin active |
| | | | 010 Stop if INDEX or SECTOR pin active (see SECTBR bit of register ECCCON). |
| | | | 011 Stop if comparison error |
| | | | 100 Branch on carry (from byte counter). |
| | | | 101 Branch on ECC error |
| | | | 110 Branch if INDEX or SECTOR pin active (see SECTBR bit of register ECCCON). |
| | | | 111 Branch on comparison error |
| Reset State: The contents of the sequencer RAM are unchanged. | | | |

7

**SEQUENCER INSTRUCTION REGISTERS** (continued)

**SEQCONF(n) A0H-BBH Read/Write**

| SEQUENCER CONTROL FIELD ARRAY | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0 | DATEN | DATA TRANSFER ENABLE - When this bit is set, the SSI 32C452 will generate $\overline{CLKB}$ requests to transfer data bytes to or from buffer memory, depending on whether WG or RG is active. |
| 1 | COMPEN | COMPARE ENABLE - When this bit is set and RG is active, read data bytes from the peripheral will be compared with the instruction data field (SEARCHOP reset in the OPCON register) or the buffer memory data (SEARCHOP set). The results of the comparisons are OR'ed together for the duration of the instruction and can be used for a branch condition or tested by the microprocessor. |
| 2 | OUTPIN | OUPUT PIN CONTROL - This bit appears on the OUTPUT pin and may be used to synchronize external circuitry to the sequencer. |
| 3 | NRZINH | NRZ DATA INHIBIT - When RG is active and this bit is set, the NRZ data input will be ignored. This is useful while external data recovery circuits start up. |
| 4 | STACKEN | STACK WRITE ENABLE - While this bit is set, bytes of NRZ data are pushed onto the recirculating stack. |
| 5 | RESWG | RESET WRITE GATE - This bit causes the WG line to go inactive 4 bit times after the current instruction is finished (byte counter reaches 0). |
| 6 | SETRG | SET READ GATE - Provided WG is inactive, this bit sets RG, which will remain active until the ECC information is read or the sequencer is halted. |
| 7 | SETWG | SET WRITE GATE - When this bit is set and an instruction executed, the WG line will be activated after a delay of 4 bit times. WG will remain active until cleared by the RESWG bit or the sequencer is halted. WG will not be activated if RG is already active. |
| Reset State: The contents of the sequencer RAM are unchanged. | | |

**SEQTYPF(n) C0H-DBH Read/Write**

| SEQUENCER DATA TYPE FIELD ARRAY | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-4 | CNT0-CNT4 | COUNT FIELD - The current sequencer instruction is executed for the number of byte times specified by the count field. If the DATEN bit is set, the count is specified as an 8 bit quantity (CNT0-CNT7). If DATEN is reset, the count is specified as a 5 bit quantity (CNT0-CNT4), and the upper three bits of this instruction field are interpreted as data type bits, described below. |

**SEQUENCER INSTRUCTION REGISTERS** (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 5 | CNT5/DTYP0 | COUNT BIT 5 OR DATA TYPE 0 - When this bit is interpreted as a data type bit, it is used to initialize the bit ring with a single 1. This will occur at the next $\overline{CLKA}$ cycle. This starts $\overline{CLKB}$ so that write data bytes will be fetched from buffer memory. The bit ring will be cleared after the ECC is written. |
| 6 | CNT6/DTYP1 | COUNT BIT 6 OR DATA TYPE BIT 1 - When this bit is interpreted as a data type bit, it indicates that ECC information is being read or written. |
| 7 | CNT7/DTYP2 | COUNT BIT 7 OR DATA TYPE BIT 2 - When this bit is being interpreted as a data type bit it indicates that an address mark is being written. |
| Note: When DATEN is reset, and CNT5/DTYP0, CNT6/DTYP1 and CNT7/DTYP2 are being interpreted as data type select bits, the upper 3 bits of the byte counter are forced to 0 regardless of the settings of the data type bits. When all 3 data type bits are 0, the data field is interpreted as normal binary data. | | |
| Reset State: The contents of the sequencer RAM are unchanged | | |

**SEQDATF E0H-FBH Read/Write**

| SEQUENCER DATA FIELD ARRAY | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | DAT0-DAT7 | DATA FIELD - When RG is active, the byte in this field is used for comparison operations. If WG is active, DATATRANS is set and TRANSINH (Transfer Inhibit bit in OPCON register) is set, the write data will come from this field. This allows the sequencer to generate the necessary overhead bytes while writing a sector. |
| Reset State: The contents of the sequencer RAM are unchanged. | | |

**DISK DRIVE INTERFACE REGISTERS**

The disk drive interface registers provide control and status for the interface of the SSI 32C452 to the disk drive (peripheral device), and for data transfer to the buffer or host.

**OPCON 7AH Read/Write**

| OPERATION CONTROL WORD | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0 | INDEXP | INDEX PULSE DETECTED - This bit is set when an index pulse is encountered and reset each time the register is read. The bit will be reset even if the INDEX pin is true during the access. |
| 1 | SECTORP | SECTOR PULSE DETECTED - This bit is set when a sector pulse is encountered and cleared each time the register is read. The bit will be cleared even if the SECTOR pin is true during the read access. This bit is only used with hard-sectored disk drives. |

**DISK DRIVE INTERFACE REGISTERS** (continued)

| BIT | NAME | DESCRIPTION |
|---|---|---|
| 2 | NRZDAT | NRZ DATA IN - This bit is set when a rising edge is detected on the NRZ pin and RG is active. It is reset when the register is read. |
| 3 | SYNDET | SERIAL DATA SYNCHRONIZATION DETECT - Indicates that the bit ring is synchronized on byte boundaries, following detection of an address mark. |
| 4 | SEARCHOP | SEARCH OPERATION - Setting this bit will cause comparisons to occur between the contents of the buffer memory and the read data bytes from the peripheral. If SEARCHOP is reset, then read data bytes will be compared to the sequencer instruction data field. |
| 5 | TRANSINH | DATA TRANSFER INHIBIT - If WG is active and this bit is set, then the write data will come from the sequencer instruction data field instead of the buffer memory. If RG is active and this bit is set, then the read data bytes are used for comparisons only and are not written to buffer memory. Setting this bit will suppress $\overline{CLKB}$ so that no buffer memory transfers occur. |
| 6 | Unused | |
| 7 | CARRYINH | SEQUENCER COUNTER CARRY INHIBIT - When this bit is set, the sequencer will not detect a carry (underflow) in its byte counter. This bit is reset when a carry occurs. |
| Reset State: Unknown | | |

**WAMCON** **7BH** **Read/Write**

| WRITE ADDRESS MARK CONTROL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | AM0-AM7 | ADDRESS MARK BITS - When WG is active and the sequencer instruction specifies that an address mark is to be written (DATATRANS is reset, DTYP2 is set) the bits AM0-AM7 will be shifted out on the $\overline{WAM}/\overline{AMD}$ pin. The pattern is delayed by two bit times to compensate for the encoder delay. |
| Reset State: Unknown | | |

**AMDCON** **7CH** **Read/Write**

| ADDRESS MARK DETECT CONTROL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-7 | AMD0-AMD7 | ADDRESS MARK DETECT CONTROL - When RG and the $\overline{WAM}/\overline{AMD}$ input are active, the NRZ data stream is compared to the contents of this register. Byte synchronization is established when a match occurs. The number of bits used in the comparison is determined in the CLKCON register. |
| Reset State: Unknown | | |

**DISK DRIVE INTERFACE REGISTERS** (continued)

**CLKCON 7FH Write only**

| CLOCK CONTROL | | | |
|---|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** | |
| 0-2 | SYN0-SYN2 | SYNC COMPARE CONTROL - These 3 bits determine which bits in register AMDCON are used when looking for the sync byte, as follows: | |
| | | SYN2/1/0 = | 000 Bit 7 used |
| | | | 001 Bits 7,6 used |
| | | | 010 Bits 7,6,5 used |
| | | | 011 Bits 7,6,5,4 used |
| | | | 100 Bits 7,6,5,4,3 used |
| | | | 101 Bits 7,6,5,4,3,2 used |
| | | | 110 Bits 7,6,5,4,3,2,1 used |
| | | | 111 All bits used |
| 3 | CLKINH | CLOCK INHIBIT - When this bit is set, $\overline{CLKA}$ and $\overline{CLKB}$ are forced to a high impedance state. | |
| 4 | CLKF0 | CLOCK FREQUENCY SELECT - This bit sets the relationship between $\overline{CLKA}$ and RD/REFCLK when data transfers are in progress. When it is set, $\overline{CLKA}$ will be 1/4 the RD/REFCLK frequency and when it is reset, $\overline{CLKA}$ will be 1/2 the RD/REFCLK frequency. | |
| 5 | Unused | | |
| 6-7 | CLKF1-CLKF2 | CLOCK FREQUENCY SELECT - These bits determine the relationship between the frequency of $\overline{CLKA}$ and SYSCLK when no data transfers are in progress, as follows: | |
| | | CLKF2/CLKF1= | 00 1/4 frequency |
| | | | 01 1/2 frequency |
| | | | 10 same frequency |
| | | | 11 illegal combination |
| Reset State: Unknown | | | |

**STACK 7FH Read only**

| TOP OF STACK |
|---|
| This register provides the microprocessor read access to the top of the 8 byte stack. Each read operation causes the stack data to recirculate, with the top of the stack moving to the bottom. When the sequencer writes data to the stack, the byte on the bottom of the stack is lost. |

7

### GENERAL PURPOSE INPUT/OUTPUT REGISTERS

**GPIOCON 7DH Read/Write**

| GENERAL PURPOSE I/O CONTROL | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-3 | GPDIR0 -GPDIR3 | GENERAL PURPOSE I/O LINE DIRECTION- These bits program the direction of lines GPIO0 to GPIO3. The direction bits are set for outputs and reset for inputs. |
| 4 | W6ESEL | $\overline{W6E}$ SELECT - If this bit is set along with GPDIR0, the GPIO0 pin becomes an active low output signal decoding a microprocessor write to location 6EH. |
| 5 | R6ESEL | $\overline{R6E}$ SELECT - If this bit is set along with GPDIR1, the GPIO1 pin becomes an active low output signal decoding a microprocessor read from location 6EH. |
| 6 | W6FSEL | $\overline{W6F}$ SELECT - If this bit is set along with GPDIR2, the GPIO2 pin becomes an active low output signal decoding a microprocessor write to location 6FH. |
| 7 | R6FSEL | $\overline{R6F}$ SELECT - If this bit is set along with GPDIR3, the GPIO3 pin becomes an active low output signal decoding a microprocessor read from location 6FH. |
| Reset State: Unknown | | |

**GPIODAT 7EH Read/Write**

| GENERAL PURPOSE I/O DATA | | |
|---|---|---|
| **BIT** | **NAME** | **DESCRIPTION** |
| 0-3 | GP0-GP3 | GENERAL PURPOSE I/O PIN STATUS - These bits represent the state or output data for the GPIO0 to GPIO3 pins, depending on the direction programmed in the GPIOCON register. |
| 4 | INPUT | INPUT PIN STATUS - This bit reflects the data on the INPUT pin. |
| 5 | OUT | OUTPUT PIN STATUS - This bit reflects the data on the OUTPUT pin. The OUTPUT pin is actually written to by the sequencer. |
| 6-7 | Unused | |
| Note: The GPIOCON register must be initialized before GPIODAT is accessed. | | |
| Reset State: Unknown | | |

**MICROPROCESSOR INTERFACE REGISTERS**

**DLR** **4DH** **Read only**

| DATA LATCH REGISTER |
|---|
| When a microprocessor read from location 70H is detected, the data on the buffer memory bus (D0-D7) is latched by the SSI 32C452 into the DATA LATCH REGISTER. When the microprocessor accesses DLR this data is placed on the address/data bus (AD0-AD7). |

**SPECIAL ADDRESS DECODES 50H-51H** **Read/Write**

| Special decodes |
|---|
| Microprocessor accesses to these locations will cause the address/data bus (AD0-AD7) and the buffer data bus (D0-D7) to be bridged together internally (see external register description). |

**BUFACC** **70H** **Read/Write**

| BUFFER ACCESS |
|---|
| Microprocessor accesses to this location cause the address/data bus (AD0-AD7) and the buffer data bus (D0-D7) to be bridged together internally. If a read cycle is performed, the data present will be latched into register DLR as well. |

**TEST REGISTERS**
These registers may not be accessed while the sequencer is running.

**TEST0** **49H** **Read only**

| TEST REGISTER 0 |
|---|
| Access to the Next Address field of the current sequencer instruction. |

**TEST1** **4AH** **Read only**

| TEST REGISTER 1 |
|---|
| Access to the Control field of the current sequencer instruction. |

**TEST2** **4BH** **Read only**

| TEST REGISTER 2 |
|---|
| Access to the Count/Data Type field of the current sequencer instruction. |

**TEST3** **4CH** **Read only**

| TEST REGISTER 3 |
|---|
| Access to the Data field of the current sequencer instruction. |

**EXTERNAL REGISTERS** (for reference only)

**HOSTL** **50H** **Read/Write**

| HOST BUS (LOWER BYTE) |
|---|
| External hardware may be used to connect the lower byte of the host bus to the buffer memory when this address is accessed. |

**HOSTH** **51H** **Read/Write**

| HOST BUS (UPPER BYTE) |
|---|
| External hardware may be used to connect the upper byte of the host bus to the buffer memory when this address is accessed. |

**GPREG0** **6EH** **Read/Write**

| GENERAL PURPOSE REGISTER 0 |
|---|
| Systems which need extra I/O on the microprocessor data bus may take advantage of the strobes available on pins GPIO0 (write) and GPIO1 (read) to add an expansion port at this address. |

**GPREG1** **6FH** **Read/Write**

| GENERAL PURPOSE REGISTER 1 |
|---|
| Systems which need extra I/O on the microprocessor data bus may take advantage of the strobes available on pins GPIO2 (write) and GPIO3 (read) to add an expansion port at this address. |

## ELECTRICAL SPECIFICATIONS

### ABSOLUTE MAXIMUM RATINGS

| PARAMETER | RATING | UNIT |
|---|---|---|
| Ambient Temperature Under Bias | 0 to 70 | °C |
| Storage Temperature | -65 to 150 | °C |
| Voltage On Any Pin With Respect To Ground | GND -0.5 or VCC + 0.5 | V |
| Power Supply Voltage | 7.0 | V |
| Max Current Injection | 25 | mA |

NOTE: Stress above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### RECOMMENDED OPERATING CONDITIONS

| PARAMETER | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| VCC Supply Voltage | | 4.75 | | 5.25 | V |
| TA Operating Free Air Temp. | | 0 | | 70 | °C |
| Input Low Voltage | | 0 | | 0.4 | V |
| Input High Voltage | | 2.4 | | VCC | V |

7

### D. C. CHARACTERISTICS

TA = 0°C to 70°C, VCC = 5V ± 5%, unless otherwise specified.

| PARAMETER | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| VIL Input Low Voltage | | -0.5 | | 0.8 | V |
| VIH Input High Voltage | | 2.0 | | VCC + .5 | V |
| VOL Output Low Voltage | IOL = 4 mA for RG and WG | | | 0.45 | V |
| | IOL = 2 mA all others | | | | |
| VOH Output High Voltage | IOH = 400 mA | | | 2.4 | V |
| ICCS Supply Current Standby | Inputs at GND or VCC | | | 25 | mA |
| ICC Supply Current | | | | 85 | mA |
| Power Dissipation | | | | 500 | mW |

**D. C. CHARACTERISTICS** (continued)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| IL | Input Leakage | 0V < Vin < VCC | -10 | | 10 | µA |
| IOL | Output Leakage | 0.45V < Vout < VCC | -10 | | 10 | µA |
| Cin | Input Capacitance | | | | 10 | pF |
| Cout | Output Capacitance | | | | 10 | pF |

## A. C. TIMING CHARACTERISTICS

TA = 0°C to 70°C, VCC = 5v ± 5%, unless otherwise specified.
Load conditions for all pins - 30pF. Timing measurements are made at 50% of rising or falling edge.
Note: ↓ indicates falling edge; ↑ indicates rising edge.

**MICROPROCESSOR INTERFACE TIMING** (See Figure 3)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| S | SYSCLK Period | | 50 | | | ns |
| S/2 | SYSCLK Assert to De-assert | | 18 | | | ns |
| S/2 | SYSCLK Rise and fall | Sr = Sf, S = 60 ns | | | 5 | ns |
| Ta | ALE Width | | 45 | | | ns |
| Taw | ALE ↓ to $\overline{WR}$ ↓ | | 25 | | | ns |
| Tar | ALE ↓ to $\overline{RD}$ ↓ | | 25 | | | ns |
| Tw | $\overline{WR}$ Width | | 200 | | | ns |
| Tr | $\overline{RD}$ Width | | 200 | | | ns |
| As | AD0 - AD7 in Valid to ALE ↓ | | 7.5 | | | ns |
| Ah | ALE ↓ to AD0 - AD7 in Invalid | | 20 | | | ns |
| Cs | CS ↑ to ALE ↓ | | 7.5 | | | ns |
| Ch | $\overline{RD}$ ↑ or $\overline{WR}$ ↑ to CS ↓ | | 0 | | | ns |
| Wds | AD0 - AD7 in Valid to $\overline{WR}$ ↑ | | 70 | | | ns |
| Wdh | $\overline{WR}$ ↑ to AD0 - AD7 in Invalid | | 10 | | | ns |
| Tda | $\overline{RD}$ ↓ to AD0 - AD7 out Valid | | | | 145 | ns |
| Tdh | $\overline{RD}$ ↑ to AD0 - AD7 out Invalid | | | | 50 | ns |

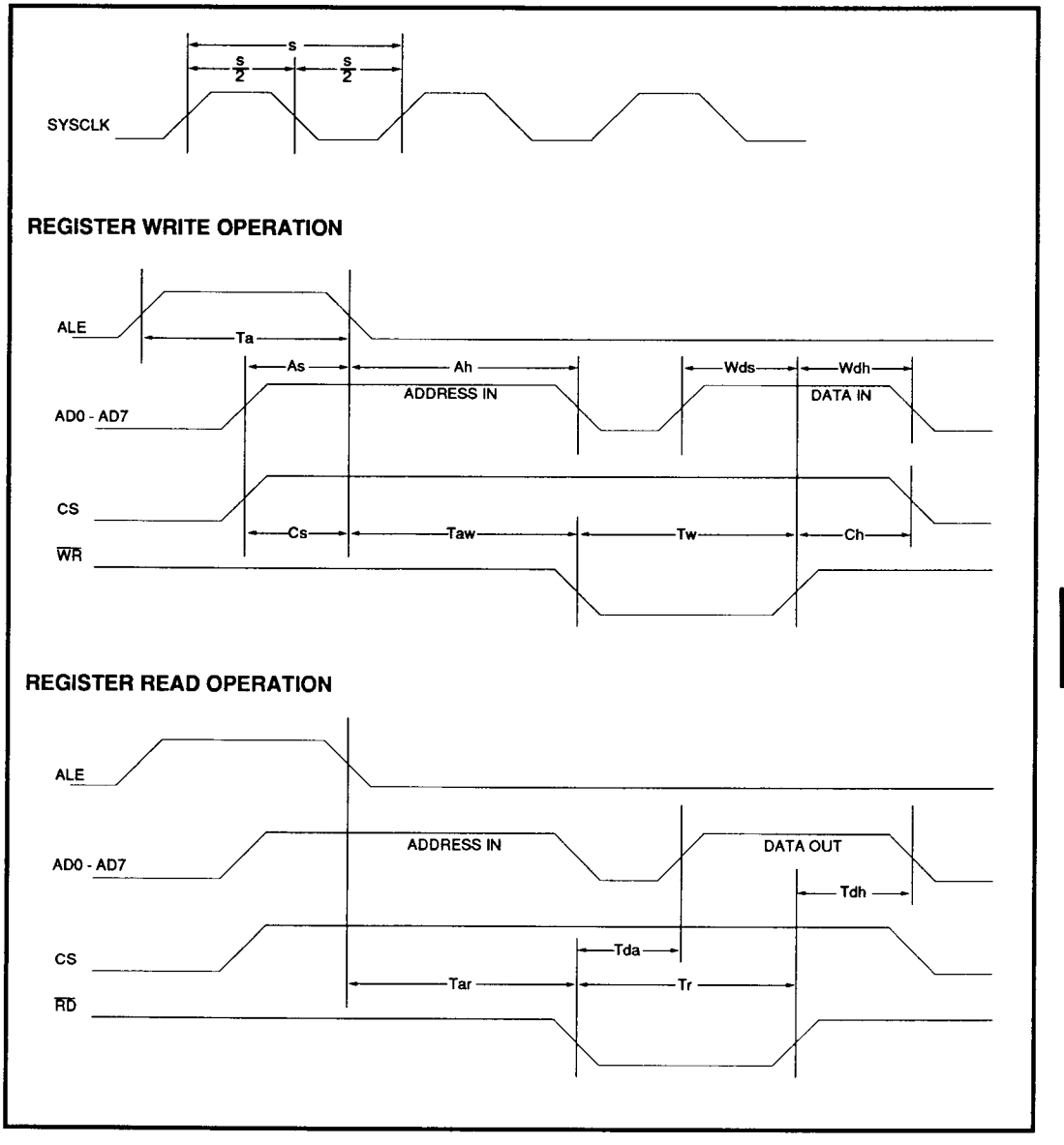



**FIGURE 3: Microprocessor Interface Timing**

## A. C. TIMING CHARACTERISTICS (continued)

**PERIPHERAL DEVICE INTERFACE TIMING** (See Figure 4)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| T | RD/REFCLK Period | | 50 | | | ns |
| T/2 | RD/REFCLK Assert to De-assert | | 18 | | | ns |
| Tr | RD/REFCLK Rise Time | T = 62.5 ns | | | 5 | ns |
| Tf | RD/REFCLK Fall Time | T = 62.5 ns | | | 5 | ns |
| Ds | NRZ in Valid to RD/REFCLK ↑ | Set-up time | 10 | | | ns |
| Dh | RD/REFCLK ↑ to NRZ in Invalid | Hold time | 7 | | | ns |
| As | $\overline{AMD}$ ↓ to RD/REFCLK ↑ | Set-up time | 10 | | | ns |
| Dv | RD/REFCLK ↑ to NRZ out | | 10 | | 40 | ns |
| Wv | RD/REFCLK ↑ to $\overline{WAM}$ ↓ | | 10 | | 40 | ns |
| Wvr | RD/REFCLK ↑ to $\overline{WAM}$ ↑ | | 10 | | 40 | ns |

**BUFFER INTERFACE TIMING** (See Figure 5)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| T | $\overline{CLKA}$ Period | | 100 | | | ns |
| T/2 | $\overline{CLKA}$ Assert to De-assert | | 40 | | | ns |
| Tba | $\overline{CLKB}$ ↓ to $\overline{CLKA}$ ↓ | | 40 | | | ns |
| Tab | $\overline{CLKA}$ ↓ to $\overline{CLKB}$ ↑ | | 40 | | | ns |
| Dov | $\overline{CLKA}$ ↑ to D0 - D7 out Valid | | 0 | | 50 | ns |
| Doh | $\overline{CLKA}$ ↑ to D0 - D7 out Invalid | | 0 | | 50 | ns |
| Dis | D0 - D7 in Valid to $\overline{CLKA}$ ↓ | | 25 | | | ns |
| Dih | $\overline{CLKA}$ ↓ to D0 - D7 in Invalid | | 10 | | | ns |

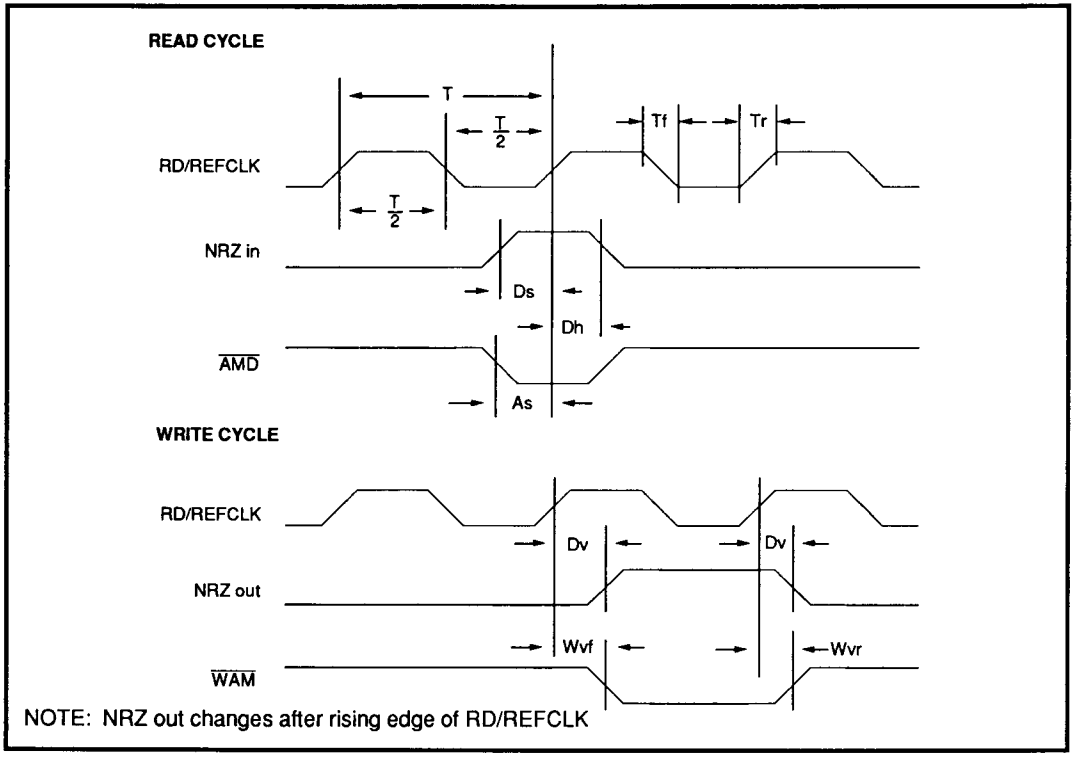



**FIGURE 4: Peripheral Device Interface Timing**

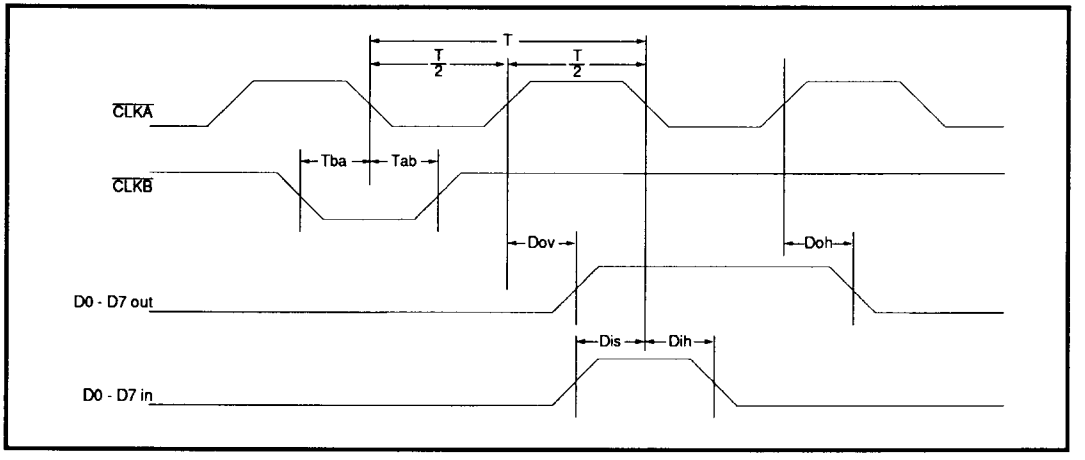



**FIGURE 5: Buffer Interface Timing**

## A. C. TIMING CHARACTERISTICS (continued)

**EXTERNAL REGISTER TIMING** (See Figure 6)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| Tda | D0 - D7 in Valid to AD0 - AD7 out Valid | | | | 55 | ns |
| Tra | $\overline{RD}$ ↓ to AD0 -AD7 out Valid | D0-D7 setup before $\overline{RD}$ ↓ | | | 85 | ns |
| Trh | $\overline{RD}$ ↑ to AD0 - AD7 out Invalid | | | | 50 | ns |
| Tad | AD0 - AD7 in Valid to D0 - D7 out Valid | | | | 59 | ns |
| Twd | $\overline{WR}$ ↓ to D0 - D7 out Valid | AD0-AD7 setup before $\overline{WR}$ ↓ | | | 60 | ns |
| Twh | $\overline{WR}$ ↑ to D0 - D7 out Invalid | | 45 | | | ns |

**ADDRESS DECODE 6E AND 6F TIMING** (See Figure 7)

| PARAMETER | | CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| Tdf | $\overline{RD}$ or $\overline{WR}$ ↓ to Strobe ↓ | | | | 40 | ns |
| Tdr | $\overline{RD}$ or $\overline{WR}$ ↑ to Strobe ↑ | | | | 40 | ns |

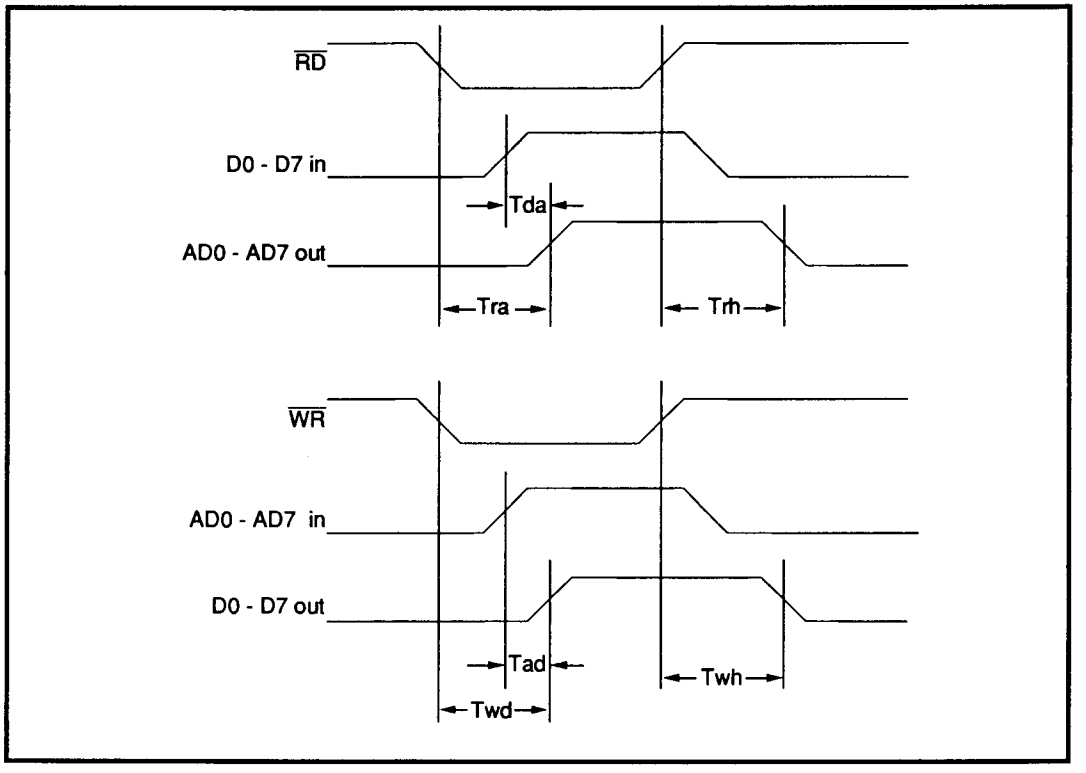



**FIGURE 6: External Register Timing**

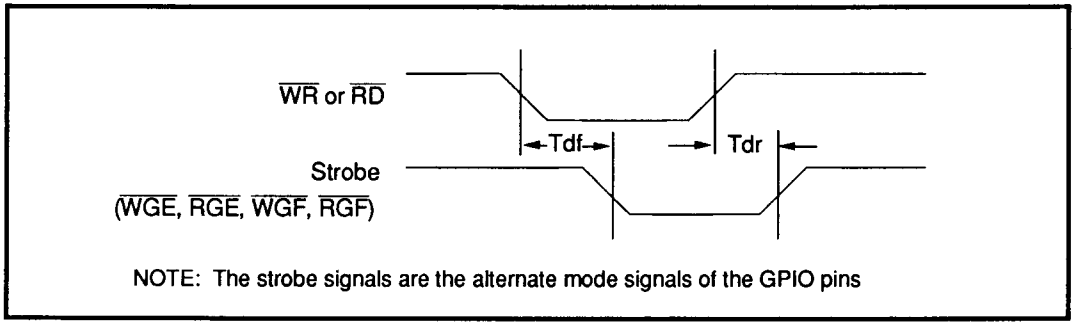



**FIGURE 7: Address Decode 6E and 6F Timing**

## APPLICATIONS INFORMATION

### SEQUENCER PROGRAMMING EXAMPLES

This section describes how specific controller functions are implemented with the SSI 32C452. Sequencer programming examples for the specific case of an ST-506 Winchester disk drive are given. For convenience, all the code samples start at sequencer address 00H. In an actual implementation, the sequencer intructions would be distributed throughout the sequencer RAM, with common portions reused, so that the code for all operations would be resident simultaneously. All example values are hex quantities.

### SECTOR ID

There are two types of Sector ID operation. In the first, the Sector ID field is read and saved by the controller for examination by the microprocessor. The 8 byte internal stack is used for this type of operation and read data is pushed to the stack under the control of the sequencer. In the second, the sector ID field is compared to a desired value in preparation for some other operation, such as sector read or sector write. In this case, the ID field parameters are compared to the data field of the controller instructions. A sequencer branch instruction is used to test for a positive field ID comparison and no ECC error before the rest of the operation proceeds. The microprocessor must program the SEQBR register with the address of the code for the following operation.

The controller establishes byte synchronism by searching for an address mark after RG is asserted. The data pattern of the address mark is specified in the AMDCON register and the number of bits actually used in the pattern is selected by the bits SYN2/1/0.

**ADDR INSTRUCTION (SEQADRF SEQCONF SEQTYPF SEQDATF)**

```
                                ; ST-506 Sector Identification example. Assumes AMDCON=A1H, SYN2/1/
                                  0=7
00    60 00 00 00               ; Loop here until Index Pulse (SEQBR=01H)
01    02 40 00 00               ; Turn on RG
02    03 02 80 A1               ; Look for address mark ( A1H in ST-506 format)COMPEN=1DTYP2=1
                                  (Address Mark - Data Separator will detect deliberate coding violations and
                                  assert WAM/AMD pin).
03    04 02 00 FE               ; Look for 2nd byte of address mark (FEH - written as normal data - no coding
                                  violations) COMPEN=1
05    06 12 00 NCYL             ; Compare cylinder number (NCYL) and save too.COMPEN=1, STACKEN=1
06    07 12 00 NHEAD            ; Compare head number (NHEAD) and save too.COMPEN=1, STACKEN=1
07    08 12 00 NSECT            ; Compare sector number (NSECT) and save too.COMPEN=1, STACKEN=1
08    89 10 41 00               ; Check ID field ECC and save ECC bytes.Branch to read or write operation
                                  if positive comparison on field ID and if ECC was good (SECTBR indicates
                                  condition for desired sector operation).DTYP1=1 (ECC byte),
                                  STACKEN=1COUNT=1
0A                              ; Here if sector ID did not match target. Actual ID field and ECC bytes are
                                  available on the stack for microprocessor check.
```

### SECTOR READ

Once the sector ID field has been verified, the data field may be read. Detection of the address mark for the data portion of the sector proceeds as for ID field address mark, and causes the serializer/deserializer to be correctly synchronized with the incoming data bytes. At the end of a sector read, the microprocessor may check the ECC result to determine if a reread or error correction computation is required.

**ADDR INSTRUCTION (SEQADRF SEQCONF SEQTYPF SEQDATF)**

```
                        ; ST-506 Sector Read example assumes that sector ID field verification has
                          been performed.
00    01 40 00 00       ; Turn on RG
01    02 02 A0 A1       ; Look for data field Address Mark (A1H)COMPEN=1DTYP2=1 (AM byte),
                          DTYP0=1, enable CLKB when synchronization occurs.
02    E3 02 00 F8       ; Check second byte of AM. Must be F8H for ST-506 data field. Branch if AM
                          bytes bad.COMPEN=1
03    04 01 FF 00       ; Transfer 256 data bytes DATEN=1COUNT=FFH
04    A5 00 41 00       ; Read ECC bytes, branch on error DTYP1=1 (ECC)COUNT=1
05                      ; Here if read was error free.
```

**SECTOR WRITE**

Sector writes proceed in a similar fashion to reads. Once the sector ID field has been verified, the sequencer writes a short gap (the 'write splice') and then the sector data, followed by ECC bytes and another gap.

**ADDR INSTRUCTION (SEQADRF SEQCONF SEQTYPF SEQDATF)**

```
                        ; ST-506 Sector Write exampleAssumes that sector ID field verification has
                          been performed.
00    01 00 02 00       ; Skip 3 bytes
01    02 80 0C 00       ; Turn on WG and write 13 bytes of 00HCOUNT=0CHSETWG=1
02    03 00 A0 A1       ; Write first data AM byte (A1H)DTYP2=1 (AM), DTYP0=1 (Start CLKB*)
03    04 00 00 F8       ; Write second data AM byte (F8H)DTYP2=0 since this byte is written as
                          normal data (no coding violations).
04    05 01 FF 00       ; Write 256 data bytes DATEN=1 (transfer enabled, data comes from buffer
                          memory)COUNT=FFH
05    06 00 41 00       ; Write 2 ECC bytes COUNT=1DTYP1=1 (ECC)
06    07 00 02 00       ; Write three bytes of 00HCOUNT=2
07    08 20 00 00       ; Turn WG off RESWG=1
08                      ; Here when sector write is finished
```

**OPERATIONAL INFORMATION**

**Sector formatting** is similar to sector writing, except that the sector ID field is written in addition to the data field. The data field is also written with a fixed value instead of data transferred from buffer memory. Examples of sequencer code to write specific data are given under sector write. When an entire track is to be written, the microprocessor may update ID field information in the sequencer RAM to reflect the next sector while the sequencer is writing the current data field. This allows an entire track to be formatted in one continuous write operation. Formatting begins after the sequencer detects an index pulse.

A **data search** operation can be implemented by a simple modification to the sequencer programming for sector read operations. When the COMPEN bit of the sequencer control field is enabled, incoming data will be compared to buffer data instead of being stored. This allows the sector to be searched for specific data. (The SEARCHOP bit in the OPCON register must also be set for searches).

**Data verification** can be performed during a sector read if the TRANSINH bit (data transfer inhibit) of OPCON is enabled, because no data will be written to the buffer. However, ECC checking will continue so that at the end of the sector, the ECC result can be verified.

The controller can support **extended sector sizes** of greater than 256 bytes. One simple way to achieve larger sector sizes is to use several sequencer data transfer instructions in a row. The size of the data block that results will be the sum of the counts for each transfer instruction. Large sectors may also be implemented with a single sequencer instruction by using

the CARRYINH bit in OPCON. Sequencer instructions terminate when the carry caused by an underflow of the byte counter is detected. When CARRYINH is set, this carry will not be recognized, so the counter (which is initially loaded with the value specified in each instruction's count field) will wrap around to a full count (FFH). The CARRYINH bit is cleared by an underflow, so that if it is not set again by the microprocessor, the sequencer instruction will terminate after an additional 256 bytes. This permits the sector length to be extended in multiples of 256 bytes.

**Multi-sector reads and writes** are accomplished in a similar manner to full track formatting. The sequencer is programmed as for a single sector operation. However, when the microprocessor detects that the DATA-TRANS bit in the SEQSTAT register is set (implying that a data transfer is in progress), it alters the ID field information in the sequencer's instruction RAM. When the data transfer for a particular sector is completed, the sequencer is looped back to the same sector ID routine. It will then start a new sector operation using the ID information just loaded by the microprocessor. This type of operation may proceed for an entire track.

## ECC IMPLEMENTATION

The ECC hardware may be used for error correction as well as checksum generation. An algorithm for locating and correcting read errors is described below. The algorithm assumes the use of a 32 bit ECC polynomial capable of correcting a single burst of up to 8 bit errors. Longer bursts or multiple bursts may be incorrectable.

1. If an ECC error is detected (ECCERR is set in SEQSTAT) and error correction is needed (ie. multiple reads from the same sector have failed) the error syndrome must be read from the ECC shift register and reloaded in bit-reversed order, as follows:
   - 1.1 Set FEEDINH in ECCCON
   - 1.2 Read and save top 8 bits of shift register from ECC24
   - 1.3 Set ECCSHIFT in ECCCON 8 times
   - 1.4 Repeat 1.2 and 1.3 until all 4 bytes of the syndrome are RAM
   - 1.5 Copy each syndrome bit, starting with the least significant, to ECCIN and set ECCSHIFT after each copy. After 32 such operations the ECC shift register will contain the bit reversed polynomial.
2. The reverse ECC generator polynomial must be written to the ECC generator.
   - 2.1 Configure the bit-reversed polynomial in the 4 feedback registers, POLY0, POLY8, POLY16 and POLY24. This step is not equivalent to bit reversing the feedback register contents, since the coefficients for $x^0$ and $x^{32}$ are fixed in hardware. The reverse polynomial is generated by subtracting the exponents from 32. The following is a numerical example to illustrate the programming of forward and reverse polynomials for the 32 bit computer-generated code:
     forward:
     $x^0+x^4+x^6+x^{13}+x^{15}+x^{22}+x^{26}+x^{30}+x^{32}$;
     reverse:
     $x^{32}+x^{28}+x^{26}+x^{19}+x^{17}+x^{10}+x^6+x^2+x^0$;

     | | Forward | Reverse |
     |---|---|---|
     | POLY0 | 28H | 22H |
     | POLY8 | 50H | 02H |
     | POLY16 | 20H | 05H |
     | POLY24 | 22H | 0AH |
   - 2.2 Reset FEEDINH and ECCIN in the ECCCON register.
3. The ECC shift register is operated until either the number of shifts exceeds the number of bits in the read block or the 24 least significant bits of the ECC register are zero.
   - 3.1 Compute block length in bits, including ECC and overhead bits.
   - 3.2 Initialize a shift counter to zero.
   - 3.3 Set ECCSHIFT to shift the ECC registers by one, and increment the shift counter.
   - 3.4 If the shift counter exceeds the block length, stop the computation as this means the errors are uncorrectable. Otherwise, if register ECC16 is non-zero, repeat step 3.3.
4. At this point, ECC24 contains the bit-reversed error pattern and the shift counter indicates its displacement from the end of the block. The pattern must be mirrored and aligned to byte boundaries so that the errors in the buffer storage may be corrected.
   - 4.1 Subtract 7 from the shift counter, to compensate for a hardware offset internal to the SSI 32C452.
   - 4.2 Subtract 32 from the shift counter. (This is the number of the ECC bits). If the result is less than zero then no further action is required, since the errors occurred in the ECC portion of the block.
   - 4.3 Read the contents of ECC24 into RAM and bit-reverse this 8 bit quantity.

4.4 Form a 16 bit word with the reversed error pattern as its lower byte and zero as its upper byte.
4.5 If the lowest three bits of the shift counter are non-zero, left shift the 16 bit word and decrement the shift counter.
4.6 Repeat 4.5 until the shift counter's three least significant bits are zero.
4.7 Divide the shift counter by 8, to convert bits into bytes.

5. The position and nature of the errors are now known, so they may be corrected as follows:
5.1 Exclusive OR the lower byte of the error word with the data byte whose offset from the end of the data block is given by the value of the shift counter.
5.2 Exclusive OR the upper byte of the error word with the data byte whose offset from the end of the data block is one more than the value of the shift counter.

The above procedure will correct a single burst of errors, provided that the degree of the error is within the capability of the chosen code. The code whose polynomial is illustrated above is capable of correcting a single burst of up to 8 error bits.
Since the error correction process is time consuming and ties up the ECC hardware, blocks with errors should be re-read to ensure that the errors observed are in fact hard errors.

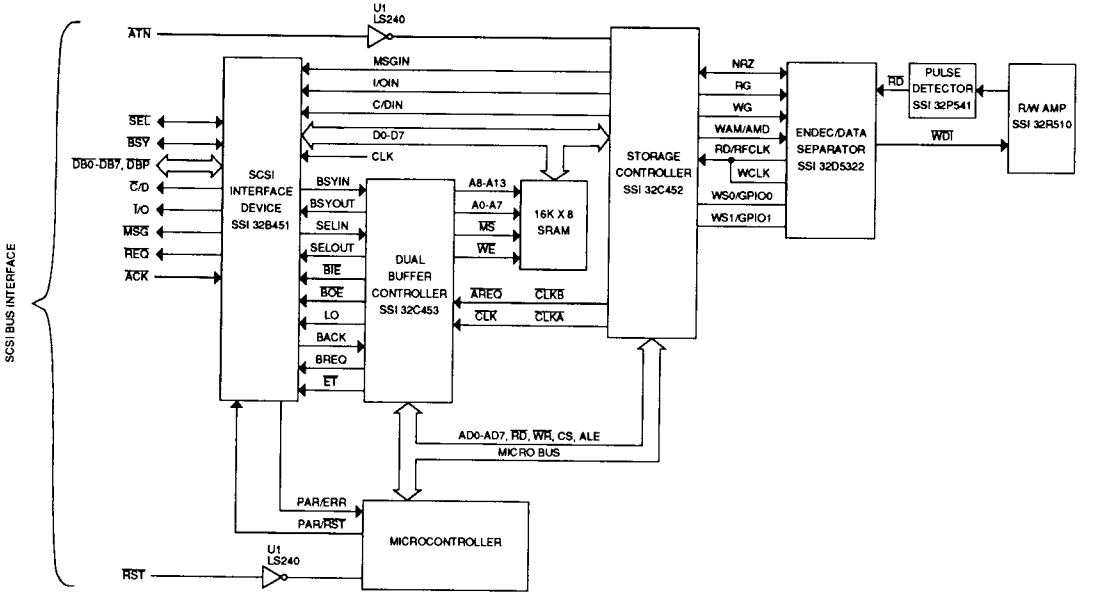
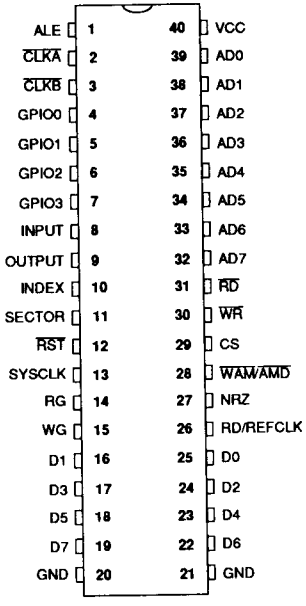



**FIGURE 8: Partial Schematic for SCSI Implementation with Arbitration Support using Silicon Systems Microperipheral Devices**

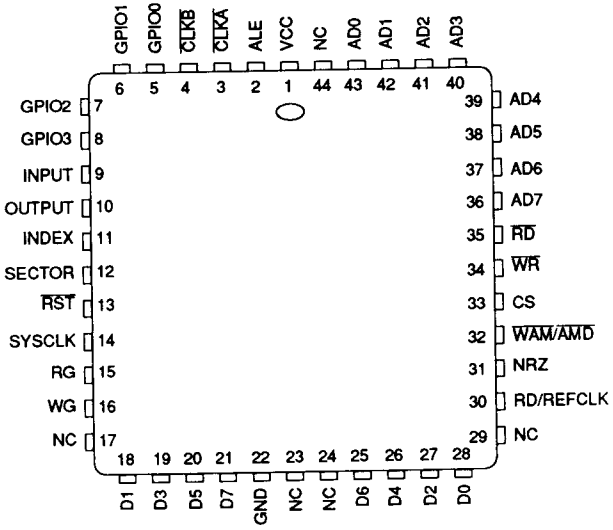
7

# SSI 32C452
# Storage Controller

## PACKAGE PIN DESIGNATIONS
(TOP VIEW)




40-pin DIP




44-pin PLCC

## ORDERING INFORMATION

| PART DESCRIPTION | | ORDER NO. | PKG. MARK |
|---|---|---|---|
| SSI 32C452 Storage Controller | 40 Pin DIP | SSI 32C452-CP | 32C452-CP |
| | 44 Pin PLCC | SSI 32C452-CH | 32C452-CH |

**Preliminary Data:** Indicates a product not completely released to production. The specifications are based on preliminary evaluations and are not guaranteed. Small quantities are available, and Silicon Systems should be consulted for current information.



Silicon Systems, Inc. 14351 Myford Road, Tustin, CA 92680, (714) 731-7110, FAX (714) 731-5457