1. INTRODUCTION

The LC8950 and LC8951 Real-Time Error Correction and Host Interface Processors (RCHIP) are dedicated single-chip LSIs for use in CD-ROM (Mode 1) and CD-I (Mode 2, Forms 1 and 2) disc formats. The LC8951 is hardware and software compatible with the LC8950. The differences in functionality between the two chips is described in appendix E.

The LC8950 RCHIP performs real-time data decoding (error detection and correction) and implements the host interface, which incorporates a first-in first-out (FIFO) command buffer suitable for Small Computer System Interface (SCSI) applications.

The following figure shows a typical system implementation for a CD-ROM disc drive using the LC8950.

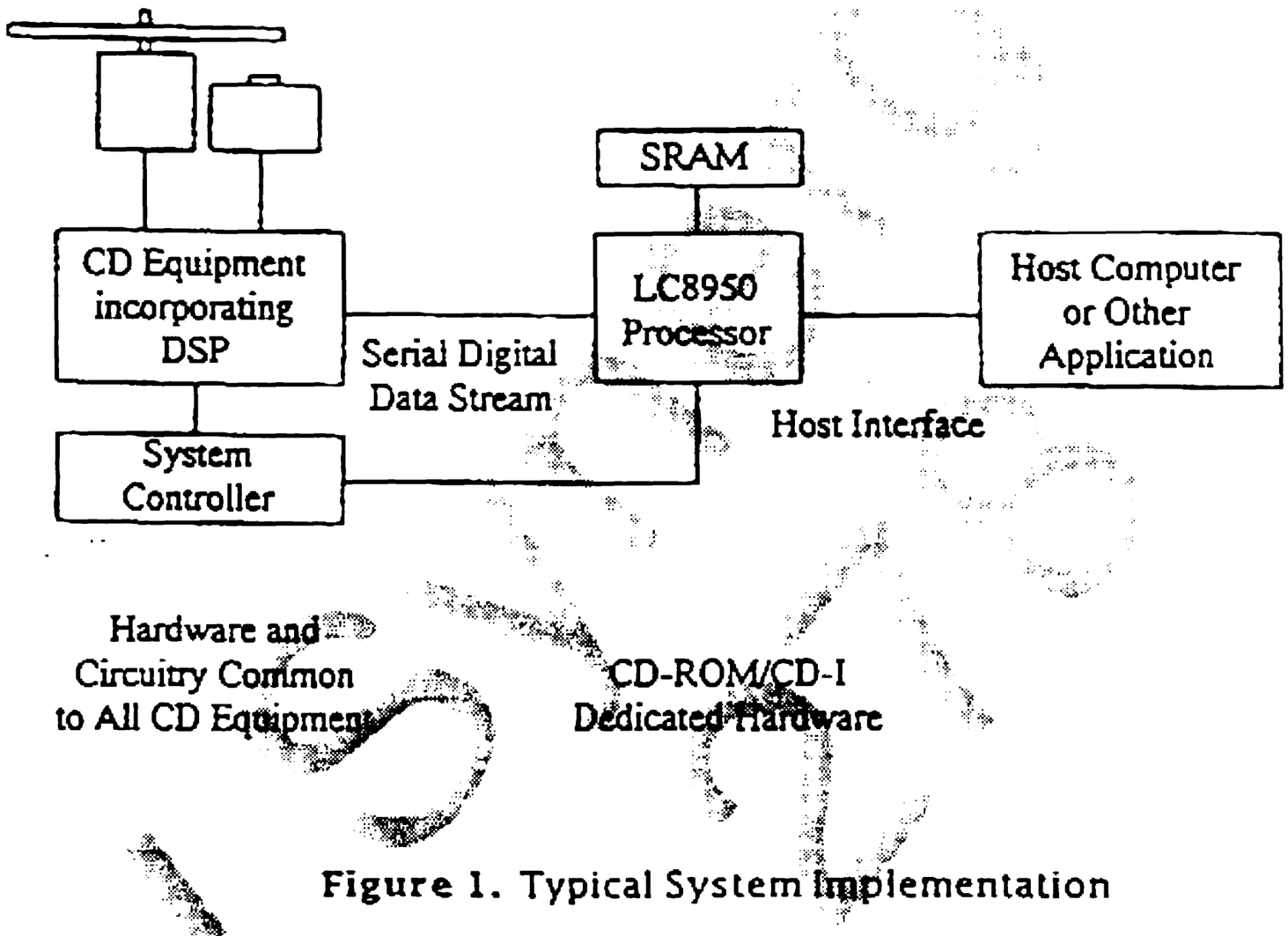


Figure 1. Typical System Implementation

As the LC8950 RCHIP eliminates many of the discrete ICs or gate arrays and additional software control provisions previously required by implementing them in hardware. The net result is faster product design and development cycles, better cost-efficiency and smaller equipment size, as well as improved functionality and reliability.

The LC8950 RCHIP processor is designed to be used with an external controller and external data buffer. This distributed architecture allows both configuration and control parameters to be defined during product development, ensuring that the device can be easily incorporated into existing and future applications.

The LC8950 RCHIP is fabricated using a low-power CMOS process, operates on a standard single +5 V power supply, and is available from stock in 80-pin plastic flatpacks.

1.1 Features

- Suitable for both CD-ROM (Mode 1) and CD-I (Mode 2, Forms 1 and 2) disc formats
- All dedicated CD-ROM/CD-I functions provided on-chip
- Real-time error detection and correction in hardware without controller intervention using pipeline processing architecture
- Transparent design separates command functions from hardware control
- Host system sees decoded data only in final error-free form
- Burst data transfers of up to 2.3 MB/s to host computer
- Multi-block catching (transfer data buffering) accommodates low read-speed hosts.
- Support for real-time erasure correction, detect-and-correct algorithms and CRC error detection, without interruption of host data transfers.
- Host interface incorporates FIFO command buffer suitable for SCSI applications.
- Single-chip LSI
- Low-power CMOS process
- Single +5 V power supply
- Available in 80-pin plastic flatpacks

2. BLOCK DIAGRAM

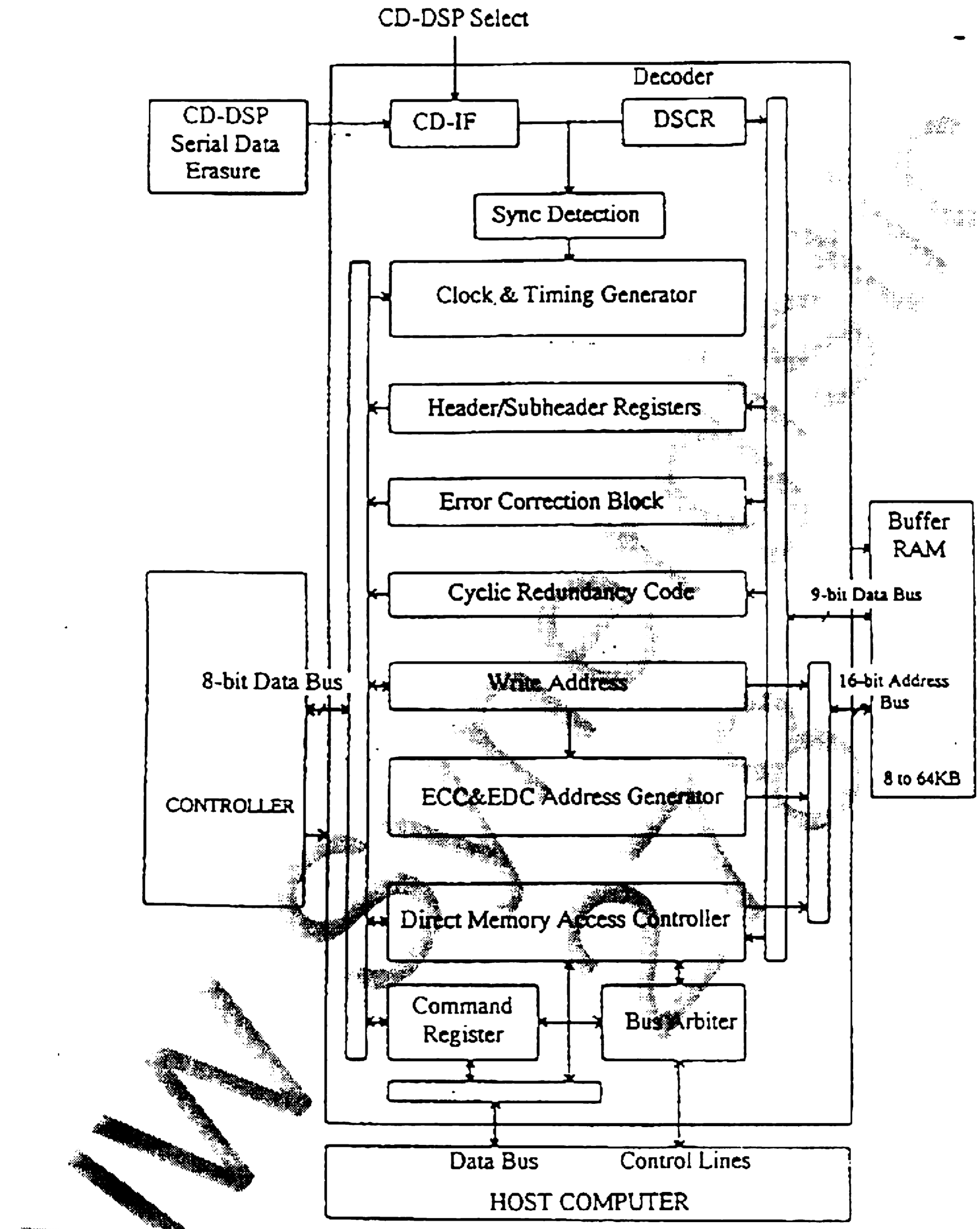
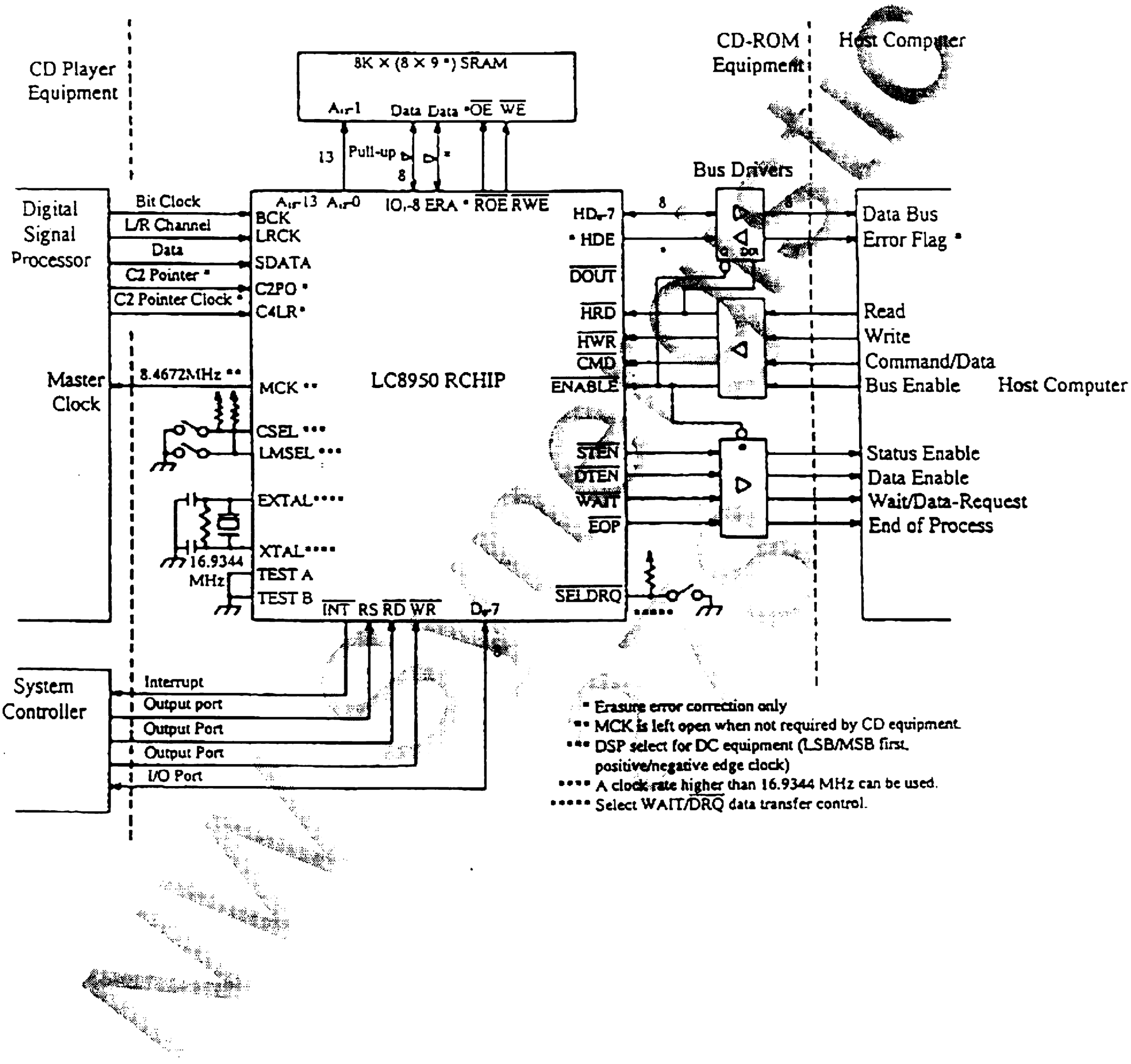


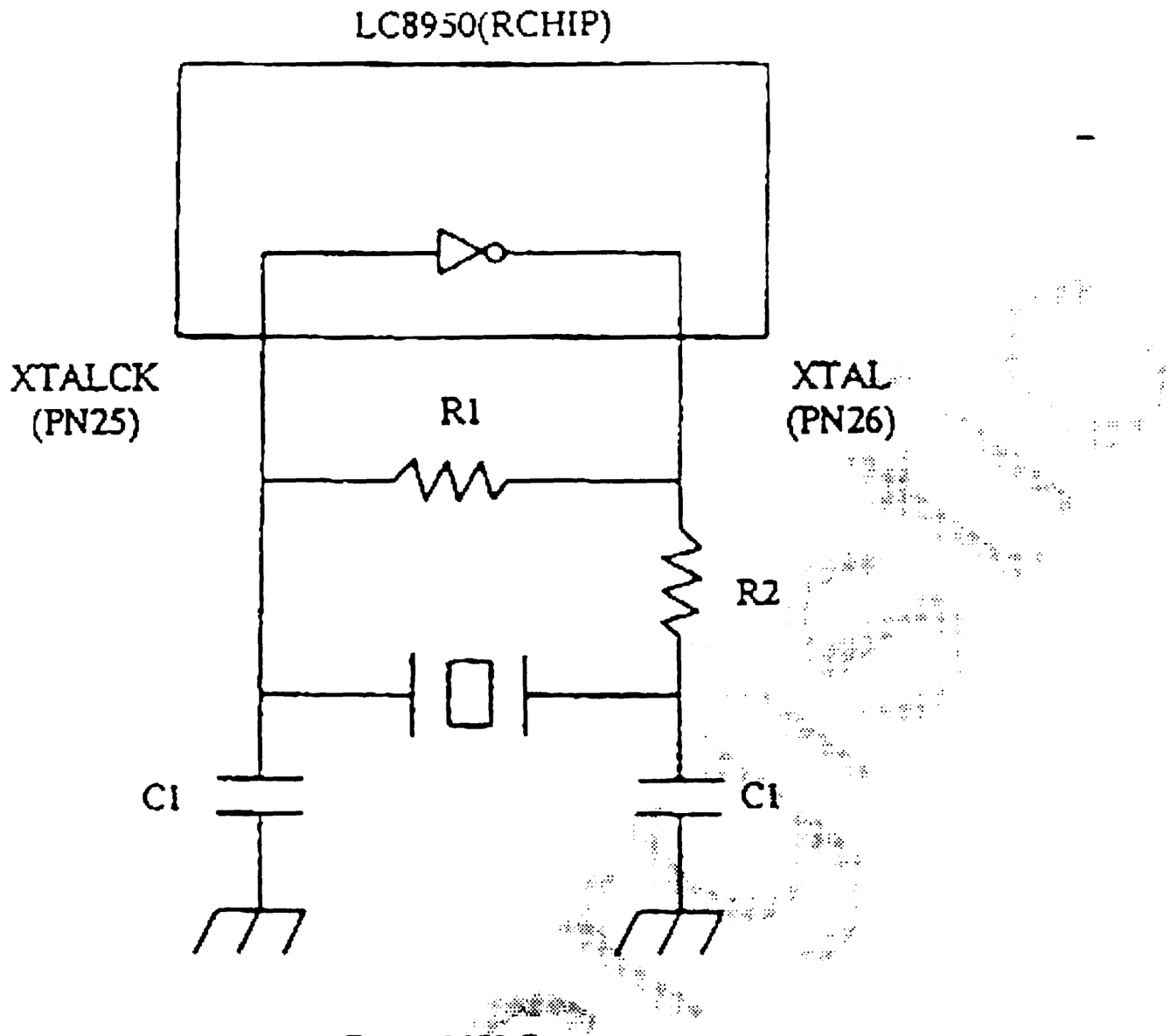
Figure 2. LC8950 Real-Time Error Correction and Host Interface Processor

3. SYSTEM CONFIGURATION



* Erasure error correction only
** MCK is left open when not required by CD equipment.
*** DSP select for DC equipment (LSB/MSB first, positive/negative edge clock)
**** A clock rate higher than 16.9344 MHz can be used.
***** Select WAIT/DRQ data transfer control.

Figure 3. LC8950 System Configuration



R1 = 120K Ω
R2 = 47 Ω
C1 = 30pF
Crystal frequency = 16.93444MHz

Figure 4. Typical Clock Circuit

4. FUNCTIONAL OVERVIEW

Data input, decoding and data transfer in the LC8950 RCHIP take place simultaneously and in parallel using a pipeline architecture. These operations proceed in real time under synchronous clocking, without the intervention of the external controller.

The LC8950 is divided into three functional blocks.

- Data-input block
- Error-correction block, and
- Host-interface block

The operation of each of these blocks is described in detail below.

4.1 Data-Input Block

This section of the LC8950 interfaces with the serial data output of the digital signal processing (DSP) unit in the CD equipment.

The LC8950 data input interface can be easily adapted to a variety of CD equipment. Simply by setting the CSEL and LMSEL pins, the equipment developer can select one of three serial data input formats.

The LC8950 synchronizes on-chip operations with the incoming data stream by detecting the sync patterns at the head of every data block. Programmable functions for pattern recognition and sync protection can be turned on and off as required.

Once detected, input data pass through a decoder and are written to a 64 Kbytes (max.) data buffer, together with the C2 pointer (error flag) received from the CD equipment. The entire 2352 bytes of data in each block, including the sync pattern, header, subheader and parity data, are written continuously into the data buffer.

When the C2 pointer is enabled, the RAM data buffer can be configured for 9-bit words; when disabled, an 8-bit RAM architecture can be selected. However, inhibiting the C2 pointer disables the player's double error-correction function.

The LC8950 also has a master clock output (MCK) which can be connected to the clock input of the digital signal processor on the CD equipment. As the clock signal to MCK passes through a binary prescaler, the clock frequency for the LC8950 should normally be set at twice that required for the CD-DSP clock signal.

4.2 Error-Correction Block

The LC8950 carries out error detection and correction in real time, immediately after a 2352-byte block has been written to the data buffer. Incoming data is corrected and sent to the host computer with no drop in data transfer rate. A simple detect-and-correct algorithm, which does not use error flags, can correct one symbol error per codeword.

The LC8950 error-correction block can also examine the C2 pointer and erase error bytes flagged by the CD equipment. When erasure correction is

enabled, two symbol errors per codeword can be corrected. Erasure error correction is also performed in real time without degrading data throughput.

This correction algorithm is programmable, and can be instructed to perform repeated or QP/PQ correction strategies to achieve higher data reliability.

The LC8950 decodes the error-correcting code. Error-detection code is used with a 32-bit cyclic redundancy code (CRC) to check errors still remaining in the current block. During these operations, the header and subheader data are stored in separate registers for use by the controller.

A decoding-complete interrupt is sent to the controller when the CRC check is completed and the current data block is ready for transfer to the host. The remaining in the current block, controller can then read out the header and subheader data, the address of the block head in the data buffer, and the status information about the decoded data.

4.3 Host-Interface Block

The host interface allows data transfers in bursts at up to 2.3 MB/s. In addition, a data buffer of up to 64 Kbytes can be allocated, allowing the LC8950 to cache up to 27 decoded data blocks. This buffer can also be implemented as a disc cache memory for CD-ROM drive applications.

The host interface also has a built-in 8-byte FIFO command buffer to receive instructions from the host computer. When the host signals the LC8950 using the $\overline{\text{HWR}}$ pin, command bursts of up to eight bytes in length can be written to the buffer. When the host writes to the command buffer, the LC8950 sends a command interrupt to the controller. Note, however, that the LC8950 itself does not interpret commands written to the command buffer.

The LC8950 can send data to the host either under software with WAIT control, or with DMA transfers using data request controls. The $\overline{\text{SELDRQ}}$ pin is used to control the type of data transfer.

When $\overline{\text{SELDRQ}}$ is HIGH, the LC8950 uses software transfer mode. In this mode, the controller places the number of bytes to be transferred and the data buffer address of the block head into the appropriate control registers, and then writes to the transfer-start register. This action sets the pin $\overline{\text{DTEN}}$ to LOW, informing the host that data transfer will start. When the host sees that $\overline{\text{DTEN}}$ is LOW, it sets $\overline{\text{CMD}}$ to HIGH, instructing the LC8950 to transfer successive bytes.

If the read signals from the host exceed the LC8950's maximum data rate (about 2.3 MB/s), the LC8950 sets the $\overline{\text{WAIT}}$ pin to LOW. The host must then hold $\overline{\text{HRD}}$ LOW to delay the read until the $\overline{\text{WAIT}}$ pin goes HIGH. Note that the controller is not directly involved in transfer operations and simply waits to receive the transfer-complete interrupt.

When the $\overline{\text{SELDRQ}}$ pin is LOW, the LC8950 performs DMA transfers using [illegible] control and operates in a manner similar to a DMA controller.

When the LC8950 sends a data request signal, the host applies signals at $\overline{\text{HRD}}$ to control the data transfer. When the last byte is transferred, the number in the byte counter reaches the number specified by the controller. This condition sets the $\overline{\text{EOP}}$ pin to LOW during the request signal output. At the same time, $\overline{\text{DTEN}}$ goes HIGH and a transfer-complete interrupt is sent to the controller.

The controller can also use the LC8950 as an intermediary for communicating the CD-ROM drive status or the results of a block decoding operation to the host. The controller writes data to a one-byte status register, and the LC8950 then transmits these data to the host, without modifying their content. The controller and host perform handshaking using signals at the $\overline{\text{STEN}}$ pin.

Since the LC8950 neither interprets nor acts on the command or status registers, equipment developers can freely define their own conventions. This facilitates both the development of new CD-ROM applications and the incorporation of the LC8950 into existing designs.

5. PACKAGE DIMENSIONS

Scale: 5:1

Unit: mm

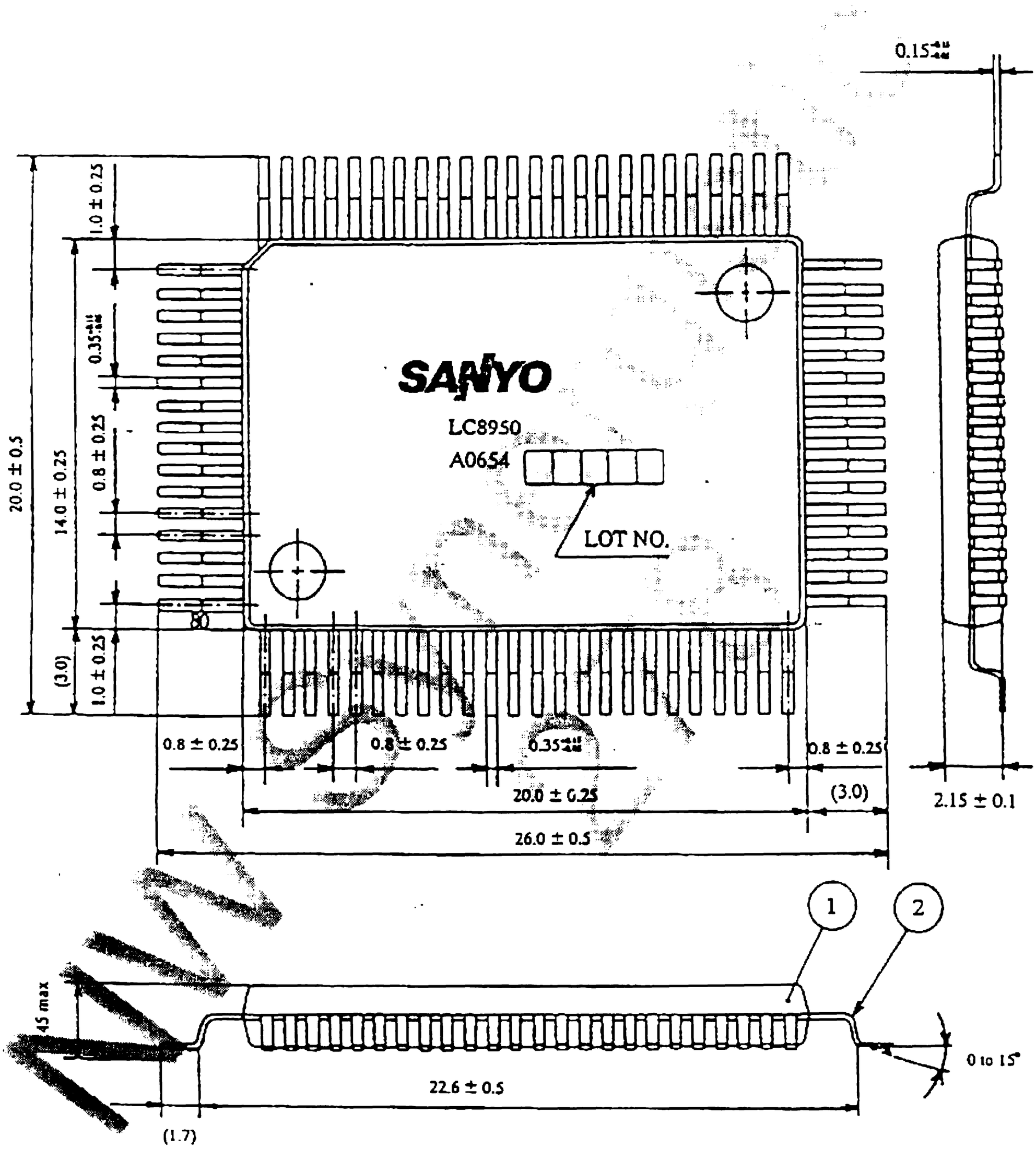


Figure 5. 80-Pin Plastic Flatpack Dimensions

6. PIN CONFIGURATION

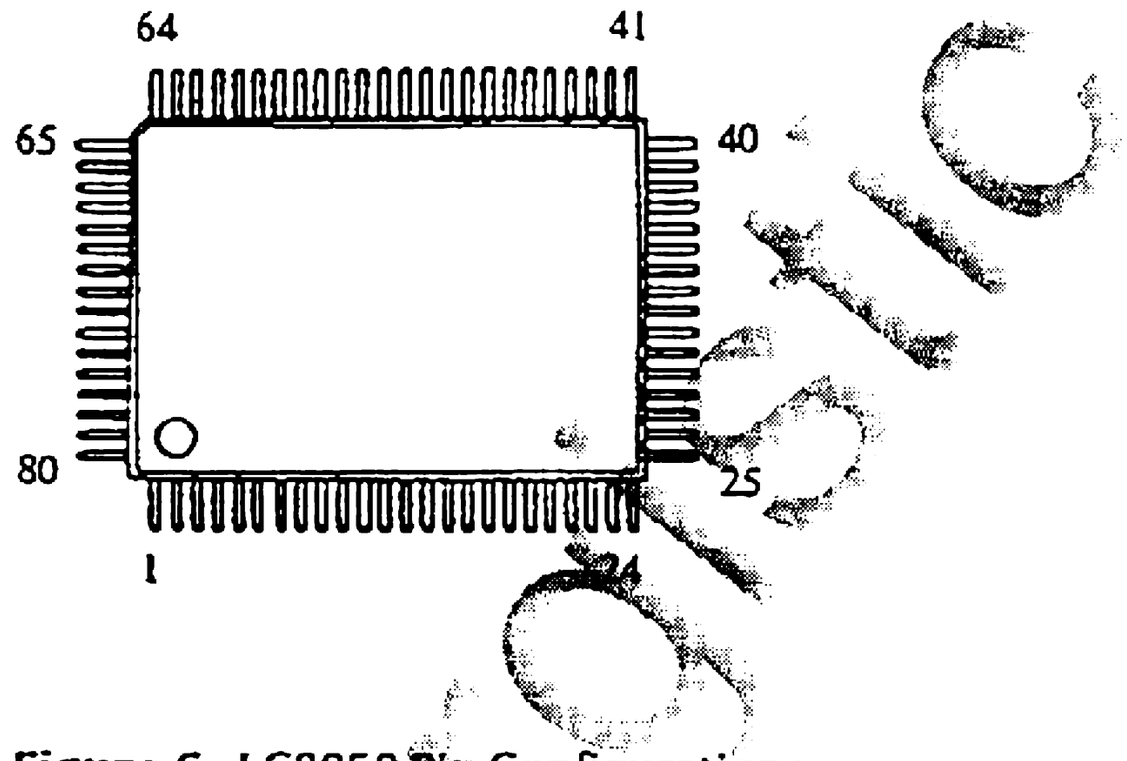


Figure 6. LC8950 Pin Configuration

The following designations are used for the pin types shown in table 2 overleaf.

TABLE 1. Pin Designators

Designator	Pin Type
I	Input
O	Output
B	Bidirectional
P	Power supply*

*V_{SS} should normally be tied to GND.

TABLE 2. Pin Numbers, Names and Designations

Pin #	Name	Designator	Pin #	Name	Designator
1	GND	P	80	RA5	O
2	RA6	O	79	RA4	O
3	RA7	O	78	RA3	O
4	RA8	O	77	RA2	O
5	RA9	O	76	RA1	O
6	RA10	O	75	RA0	O
7	RA11	O	74	$\overline{\text{SELDRQ}}$	I
8	RA12	O	73	VDD	P
9	RA13	O	72	HD0	B
10	RA14	O	71	HD1	B
11	RA15	O	70	HD2	B
12	$\overline{\text{RWE}}$	O	69	HD3	B
13	GND	P	68	HD4	B
14	$\overline{\text{ROE}}$	O	67	HD5	B
15	ERA	B	66	HD6	B
16	IO8	B	65	HD7	B
17	IO7	B	64	GND	P
18	IO6	B	63	$\overline{\text{HDE}}$	O
19	IO5	B	62	$\overline{\text{DOUT}}$	O
20	IO4	B	61	$\overline{\text{EOP}}$	O
21	IO3	B	60	$\overline{\text{STEN}}$	O
22	IO2	B	59	$\overline{\text{DTEN}}$	O
23	IO1	B	58	$\overline{\text{WAIT}}$	O
24	GND	P	57	$\overline{\text{CMD}}$	I
25	XTALCK	I	56	$\overline{\text{HRD}}$	I
26	XTAL	O	55	$\overline{\text{HWR}}$	I
27	TEST1	I	54	$\overline{\text{ENABLE}}$	I
28	TEST2	I	53	$\overline{\text{RESET}}$	I
29	CSEL	I	52	GND	P
30	EMSEL	I	51	$\overline{\text{INT}}$	O
31	VDD	P	50	$\overline{\text{CS}}$	I
32	LRCK	I	49	$\overline{\text{WR}}$	I
33	SDATA	I	48	$\overline{\text{RD}}$	I
34	BCK	I	47	RS	I
35	C4LR	I	46	D7	B
36	C2PO	I	45	D6	B
37	MCK	O	44	D5	B
38	D0	B	43	D4	B
39	D1	B	42	D3	B
40	D2	B	41	GND	P

7. PINOUTS

TABLE 3. Pin Numbers, Names and Functions

Pin #	Pin Name	I/O	Function
1	GND	—	Ground
2	RA6	O	Data buffer RAM and erasure flag RAM address outputs
3	RA7	O	
4	RA8	O	
5	RA9	O	
6	RA10	O	
7	RA11	O	
8	RA12	O	
9	RA13	O	
10	RA14	O	
11	RA15	O	
12	$\overline{RWE}$	O	RAM write enable
13	GND	—	Ground
14	$\overline{ROE}$	O	RAM output enable
15	ERA	I/O	Erasure flag RAM data input/output
16	IO8	I/O	Data buffer RAM input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
17	IO7	I/O	
18	IO6	I/O	
19	IO5	I/O	
20	IO4	I/O	
21	IO3	I/O	
22	IO2	I/O	
23	IO1	I/O	
24	GND	—	Ground
25	XTALCK	I	Clock input
26	XTAL	O	Clock output
27	TEST1	I	Test inputs. Normally held LOW.
28	TEST2	I	
29	CSEL	I	Serial data clock phase select
30	LMSEL	I	Serial data byte order control (LSB/MSB first)

Pin #	Pin Name	I/O	Function
31	VDD	—	Positive power supply connection
32	LRCK	I	44.1 KHz left & right channel separator strobe
33	SDATA	I	Serial data input
34	BCK	I	Buffer input clock
35	C4LR	I	C2 error flag pointer strobe
36	C2PO	I	C2 flag pointer
37	MCK	O	Master clock output
38	D0	I/O	Controller data input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
39	D1	I/O	
40	D2	I/O	
41	GND	—	Ground
42	D3	I/O	Controller data input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
43	D4	I/O	
44	D5	I/O	
45	D6	I/O	
46	D7	I/O	
47	RS	I	Register select
48	$\overline{RD}$	- I	Controller data read
49	$\overline{WR}$	I	Controller data write
50	$\overline{CS}$	I	Controller chip select
51	$\overline{INT}$	O	Controller interrupt
52	GND	—	Ground
53	$\overline{RESET}$	I	LC8950 chip reset
54	$\overline{ENABLE}$	I	Host interface enable
55	$\overline{HWR}$	I	Host data write input
56	$\overline{HRD}$	I	Host data read input
57	CMD	I	Host command/data select
58	$\overline{WAIT}$	O	Data transfer WAIT signal/DRQ signal
59	$\overline{DTEN}$	O	Data enable output
[illegible]	$\overline{STEN}$	O	Status enable output

Pin #	Pin Name	I/O	Function
61	$\overline{EOP}$	O	End-of-process flag output
62	$\overline{DOUT}$ ($\overline{RCS}$)	O	Host computer data buffer control output (LC8951 RAM chip select)
63	$\overline{HOE}$	O	Host data erasure flag tristate output
64	GND	—	Ground
65	HD7	I/O	Host data input/output
66	HD6	I/O	
67	HD5	I/O	
68	HD4	I/O	
69	HD3	I/O	
70	HD2	I/O	
71	HD1	I/O	
72	HD0	I/O	
73	VDD	—	Positive power supply connection
74	$\overline{SELDRQ}$	I	Data transfer mode select (WAIT control/DRQ control)
75	RA0	O	Data buffer RAM and erasure flag RAM address outputs
76	RA1	O	
77	RA2	O	
78	RA3	O	
79	RA4	O	
80	RA5	O	

8. PIN DESCRIPTIONS

GND — Ground pins

RA0 to RA15 — Data buffer RAM and erasure flag RAM address outputs

The LC8950 uses these 16 outputs to specify addresses in the data buffer or erasure data memory, allowing the use of an 8 Kbyte to 64 Kbyte external buffer.

$\overline{RWE}$ — RAM write enable output

The LC8950 sets this output to LOW when writing data to the external data buffer or erasure flag RAM.

$\overline{ROE}$ — RAM output enable output

The LC8950 sets this output to LOW when reading data from the external data buffer or erasure flag RAM.

ERA — Erasure flag RAM data I/O pin

This pin is the data bus for the erasure flag RAM. As this pin is a tristate type, a pull-up resistor should be used.

IO1 to IO8 — Data buffer RAM I/O pins

These pins are the data bus to the external data buffer. As these pins are tri-state types, pull-up resistors should be used.

XTALCK, XTAL — External clock signals

As the LC8950's clock circuit includes an on-chip binary prescaler, a 16.9344 MHz crystal should be used to generate the required 8.4672 MHz master clock (MCK) signal.

XTALCK can be driven by a 50±10 % duty external signal. This generates a master clock signal one-half the frequency of the external signal. In this case, the XTAL pin should be left floating.

The clock frequency needs to be exactly 16.9344 MHz only when the master clock output is used. Otherwise, select a frequency that results in both the LOW- and HIGH-level signals at LRCK having widths longer than 192 clock pulses.

TEST1, TEST2 — Test inputs

Test pins. Usually tied to GND

CSEL, LMSEL — Clock phase select and serial data byte order inputs

These signals select the read timing at the SDATA and C2PO inputs as follows:

A. When both CSEL and LMSEL are LOW:

   i. SDATA is read at the falling edge of BCK.

   ii. The data at SDATA are processed beginning from the most-significant bit.

   iii. The edge of LRCK ends the reading of 2-byte words at SDATA, and starts the reading of the next word.

iv. When more than 16 clock pulses are detected at BCK within one-half cycle of LRCK, only the data read in during the 16 clock pulses immediately before the LRCK edge are significant.

v. A HIGH-level signal at LRCK identifies the data at SDATA as belonging to the left audio channel.

vi. Data at C2PO is read in at the rising edge of C4LR.

vii. Signals applied to C2PO flag the most-significant byte then the least-significant byte of the 2-byte words at SDATA.

viii. In this mode, the LC8950 can be interfaced directly with the Sony CXD1125 digital signal processor.

B. When CSEL is LOW and LMSEL is HIGH:

i. SDATA is read at the falling edge of BCK.

ii. The data at SDATA are processed beginning from the least-significant bit.

iii. The edge of LRCK marks the start of reading one 2-byte word at SDATA. When reading is completed, the read operation is terminated.

iv. When more than 16 clock pulses are detected at BCK within one-half cycle of LRCK, only the data read in during the 16 clock pulses immediately after the LRCK edge are significant.

v. A HIGH-level signal at LRCK identifies the data at SDATA as belonging to the left audio channel.

vi. Data at C2PO is read in at the fifth and thirteenth BCK pulses after the LRCK edge. C4LR is not used to supply the timing.

vii. Signals applied to C2PO flag the least-significant byte then the most-significant byte of the 2-byte words at SDATA.

viii. In this mode, the LC8950 can be interfaced directly with the Sanyo LC7860, Matsushita MN6617 or Yamaha TM3815 digital signal processors.

C. When CSEL is HIGH and LMSEL is LOW:

i. SDATA is read at the rising edge of BCK.

ii. The data at SDATA are processed beginning from the most-significant bit.

iii. The edge of LRCK marks the end of reading one 2-byte word at SDATA, and starts the reading of the next word.

iv. When more than 16 clock pulses appear at BCK within one-half cycle of LRCK, only the data read in during the 16 clock pulses immediately before the LRCK edge are significant.

v. A HIGH-level signal at LRCK identifies the data at SDATA as belonging to the left audio channel.

vi. Data at C2PO is read in at the rising edge of C4LR.

vii. Signals applied to C2PO flag the least-significant byte then the most-significant byte of the 2-byte words at SDATA.

viii. In this mode, the LC8950 can be interfaced directly with the Toshiba TC9200F digital signal processor.

D. No operational mode is defined when CSEL and LMSEL are both HIGH. As this mode is reserved for future use, it should not be used.

VDD — Positive power supply connections

VDD should be in the range of 5.0±10 % V

LRCK — Left and right channel separator clock input

Supplies the signal which separates CD audio data into those for the left and right channels.

In a half cycle of LRCK, 16 bits of data are read in at SDATA. The rising and falling edges of LRCK are used to distinguish between the first and last bits.

SDATA — Serial data input

This serial input receives the data after CIRC (Cross Interleaved Reed-Solomon Code) decoding. The BCK pin controls the flow of data to the LC8950. The received data are stored in the external data buffer.

BCK — Buffer input clock

The clock signal for reading in serial data at SDATA is applied to this input. At least 16 BCK clock pulses should be generated within one-half cycle of LRCK.

C4LR — C2 error flag pointer strobe input

Used to supply the strobe signal for reading in the C2 error flag. When unused, it should be tied to either VCC or GND.

C2PO — C2 error flag pointer input

This input is used for the C2 error flag input during CIRC decoding. The flag is non-inverting; a HIGH level input indicates an error condition. The data at C2PO are read in at the rising edge of the signal at C4LR, if present, or on an internal timing signal generated at BCK and LRCK, and are stored temporarily in a one-bit buffer.

MCK — Master clock output

This output supplies a master clock signal of exactly one-half the frequency of the crystal connected at XTALCK. This output is used to provide a common clocking signal for the clock input of digital signal processor in the CD equipment.

D0 to D7 — Controller data pins

These pins are the data bus to the external controller. As these pins are high-impedance types, pull-up resistors should be used.

RS, $\overline{RD}$ and $\overline{WR}$ — Controller interface control inputs

These are the register select, read and write inputs used to control the operation of the controller interface. They are active only when $\overline{CS}$ is LOW.

RS selects either the address register (AR) or one of the LC8950's 32 registers. AR is a special register used to select one of the 32 read or write registers on-chip.

RS = LOW selects direct addressing (AR), whereas RS = HIGH selects indirect addressing (the register pointed to by AR).

$\overline{CS}$ — Controller chip select input

The controller sets this input to LOW to enable the LC8950's controller interface.

$\overline{INT}$ — Interrupt output

This output transfers interrupt requests from the LC8950 to the controller when LOW.

$\overline{RESET}$ — Reset input

Forcing this input LOW resets the LC8950.

$\overline{ENABLE}$ — Host interface enable input

This input selects the LC8950. The host computer sets this input to LOW to enable the host interface.

$\overline{HWR}$, $\overline{HRD}$ and $\overline{CMD}$ — Host interface control inputs

These inputs control the operation of the host interface. $\overline{HWR}$ is the host data write input, $\overline{HRD}$, the host data read input, and $\overline{CMD}$, the host command/data select input. They are active only when $\overline{ENABLE}$ is LOW.

i. Setting $\overline{HWR}$ LOW, $\overline{HRD}$ HIGH and $\overline{CMD}$ LOW instructs the LC8950 to enter command write mode. The data at HD1 to HD7 are written into the LC8950's command register.

ii. Setting $\overline{HWR}$ HIGH, and $\overline{HRD}$ and $\overline{CMD}$ LOW instructs the LC8950 to enter status-read mode. The data previously written by the controller to the LC8950's status register are output at HD0 to HD7.

iii. Setting $\overline{HRW}$ HIGH, $\overline{HRD}$ LOW and $\overline{CMD}$ HIGH instructs the LC8950 to enter data transfer mode. Data in the buffer are read by the LC8950 and output to the host at HD0 to HD7. The erasure flag status is output at HDE.

All other combinations of $\overline{HWR}$ LOW and $\overline{CMD}$ HIGH, and both $\overline{HWR}$ and $\overline{HRD}$ LOW are undefined and should not be set, as the result of their use cannot be guaranteed.

$\overline{WAIT}$ — Wait output

The function of this output depends on the state of the $\overline{SELDRQ}$ input.

i. When $\overline{SELDRQ}$ is HIGH (that is, during software transfer mode), the LC8950 sets the $\overline{WAIT}$ output to LOW to signal the host to suspend the data transfer.

   $\overline{WAIT}$ is held HIGH while $\overline{DTEN}$ is HIGH, and while the LC8950 is not transferring data to the host.

ii. When $\overline{SELDRQ}$ is LOW (that is, during DMA transfer mode), $\overline{WAIT}$ functions as a DRQ (data request) output to the host computer.

$\overline{WAIT}$ remains LOW while $\overline{DTEN}$ is HIGH, and while the LC8950 is not transferring data to the host.

$\overline{DTEN}$ — Data enable output

This output is set to LOW to signal the host computer that data is ready to be transferred.

$\overline{STEN}$ — Status enable output

This output is set to LOW to signal the host computer that the status byte is ready to be read out.

$\overline{EOP}$ — End-of-process flag output

The LC8950 sets this flag to LOW on sending the last byte to the host computer using either software or DMA data transfers.

$\overline{DOUT}$ — Host computer data buffer control output*

This output is set to LOW to request data from the host computer data buffer when $\overline{ENABLE}$ and $\overline{HRD}$ are both LOW.

HDE — Host data error flag output

This output is used to output error flag signals to the host computer. The error flag is output during data transfers when the corresponding data bytes are being output at HD0 to HD7. When the LC8950 is in status-read mode, HDE is LOW.

HD0 to HD7 — Host data I/O pins

These pins are the data bus to the host computer. As these pins are tristate types, pull-up resistors should be used.

$\overline{SELDRQ}$ — Data transfer mode select input

When HIGH, this input selects software transfer mode for data transfers to the host (WAIT control); when LOW, DMA transfer mode (DRQ control) is selected.

---

* See appendix E for the LC8951.

9. ELECTRICAL SPECIFICATIONS

TABLE 4. Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Test Conditions
Supply voltage	V_{DD} max	−0.3 to 7.0	V	Ta=25 deg. C
Input/output voltages	V_I, V_O	−0.3 to V_{DD}+0.3	V	Ta=25 deg. C
Power dissipation	P_d max	350	mW	Ta≤70 deg. C
Operating temperature	T_{opg}	−30 to 70	deg. C	
Storage temperature	T_{stg}	−55 to 125	deg. C	
Soldering temperature		260	deg. C	For 10 s

TABLE 5. Recommended Operating Conditions

(Over operating free-air temperature range)

Parameter	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{DD}	4.5	5.0	5.5	V
Input voltage range	V_{IN}	0	—	V_{DD}	V

TABLE 6. Electrical Characteristics

(i) LC8950

(Input/output level V_{DD}=4.5 to 5.5V, over operating free-air temperature range)

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions	Pins
HIGH-level input voltage HIGH	V_{IH}	2.2	—	—	V		All input and bus pins, except XTALCK
LOW-level input voltage	V_{IL}	—	—	0.8	V		
HIGH-level output voltage	V_{OH}	2.4	—	—	V	I_{OH}=−0.4 mA	All output and bus pins, except XTAL
LOW-level output voltage	V_{OL}	—	—	0.4	V	I_{OL}=2.0 mA	
Input leak current	I_L	−25	—	25	μA	V_{IN}=V_{SS},V_{DD}	All inputs
Output leak current	I_{OZ}	−100	—	100	μA	At high impedance	HDE and bus pins

(ii) LC8951

(Input/output level V_{DD}=4.5 to 5.5V, over operating free-air temperature range)

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions	Pins
HIGH-level input voltage	V_{IH1}	2.2	—	—	V		All input pins except XTALCK, BUS pins and $\overline{RESET}$
LOW-level input voltage	V_{IL1}	—	—	0.8	V		
HIGH-level input voltage	V_{IH2}	2.5	—	—	V		
LOW-level input voltage	V_{IL2}	—	—	0.6	V		
HIGH-level output voltage	V_{OH1}	2.4	—	—	V	I_{OH}=−0.4 mA	All output and bus pins, except XTAL
LOW-level output voltage	V_{OL1}	—	—	0.4	V	I_{OL}=2.0 mA	
LOW-level output voltage	V_{OL}	—	—	0.4	V	I_{OL}=−2.0 mA	$\overline{INT}$ (Open drain output)
Input leak current	I_L	−25	—	25	μA	V_{IN}=V_{SS},V_{DD}	All inputs
Output leak current	I_{OZ}	−100	—	100	μA	At high impedance	HDE and bus pins

10. REGISTERS

TABLE 7. LC8950 Registers During Reads

RS	AR	Number	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	—	AR	0	0	0	0	A_3	A_2	A_1	A_0
	0000	R0	COMIN	msb	—	—	—	—	—	—	lsb
	0001	R1	IFSTAT	$\overline{CMDI}$	$\overline{DTEI}$	$\overline{DECI}$	1	$\overline{DTBSY}$	$\overline{STBSY}$	$\overline{DTEN}$	$\overline{STEN}$
	0010	R2	DBCL	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
	0011	R3	DBCH	DTEI	DTEI	DTEI	DTEI	B_{11}	B_{10}	B_9	B_8
	0100	R4	HEAD0	msb	—	—	—	—	—	—	lsb
	0101	R5	HEAD1	msb	—	—	—	—	—	—	lsb
	0110	R6	HEAD2	msb	—	—	—	—	—	—	lsb
	0111	R7	HEAD3	msb	—	—	—	—	—	—	lsb
	1000	R8	PTL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1001	R9	PTH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1010	R10	WAL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1011	R11	WAH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1100	R12	STAT0	CRCOK	ILSYNC	NOSYNC	LBLK	WSHORT	SBLK	ERABLK	UCEBLK
	1101	R13	STAT1	MINERA	SECERA	BLKERA	MODERA	SH0ERA	SH1ERA	SH2ERA	SH3ERA
	1110	R14*	STAT2					MODE	FORM		
	1111	R15*	STAT3	$\overline{VALST}$	WLONG						

Key

1	Always read as one
0	Always read as zero
—	Used
BLANK	Not used, don't care

* See appendix E for the LC8951.

TABLE 8. LC8950 Registers During Writes

RS	AR	Number	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	—	—	AR					A_3	A_2	A_1	A_0
	0000	R0	SBOUT	msb	—	—	—	—	—	—	lsb
	0001	R1	IFCTRL	CMDIEN	DTEIEN	DECIEN	$\overline{\text{CMDBK}}$	$\overline{\text{DTWAI}}$	$\overline{\text{STWAI}}$	DOUTEN	SOUTEN
	0010	R2	DBCL	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
	0011	R3	DBCH					B_{11}	B_{10}	B_9	B_8
	0100	R4	DACL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	0101	R5	DACH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	0110	R6	DTTRG	—	—	—	—	—	—	—	—
0	0111	R7	DTACK	—	—	—	—	—	—	—	—
	1000	R8	WAL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1001	R9	WAH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1010	R10*	CTRL0	DECEN	EDCRQ	E01RQ	AUTORQ	ERAMRQ	WRRQ	QRQ	PRQ
	1011	R11*	CTRL1	SYIEN	SYDEN	DSCREN	COWREN	MODRQ	FORMRQ		SHDREN
	1100	R12	PTL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1101	R13	PTH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1110	R14									
	1111	R15	RESET	—	—	—	—	—	—	—	—

Key

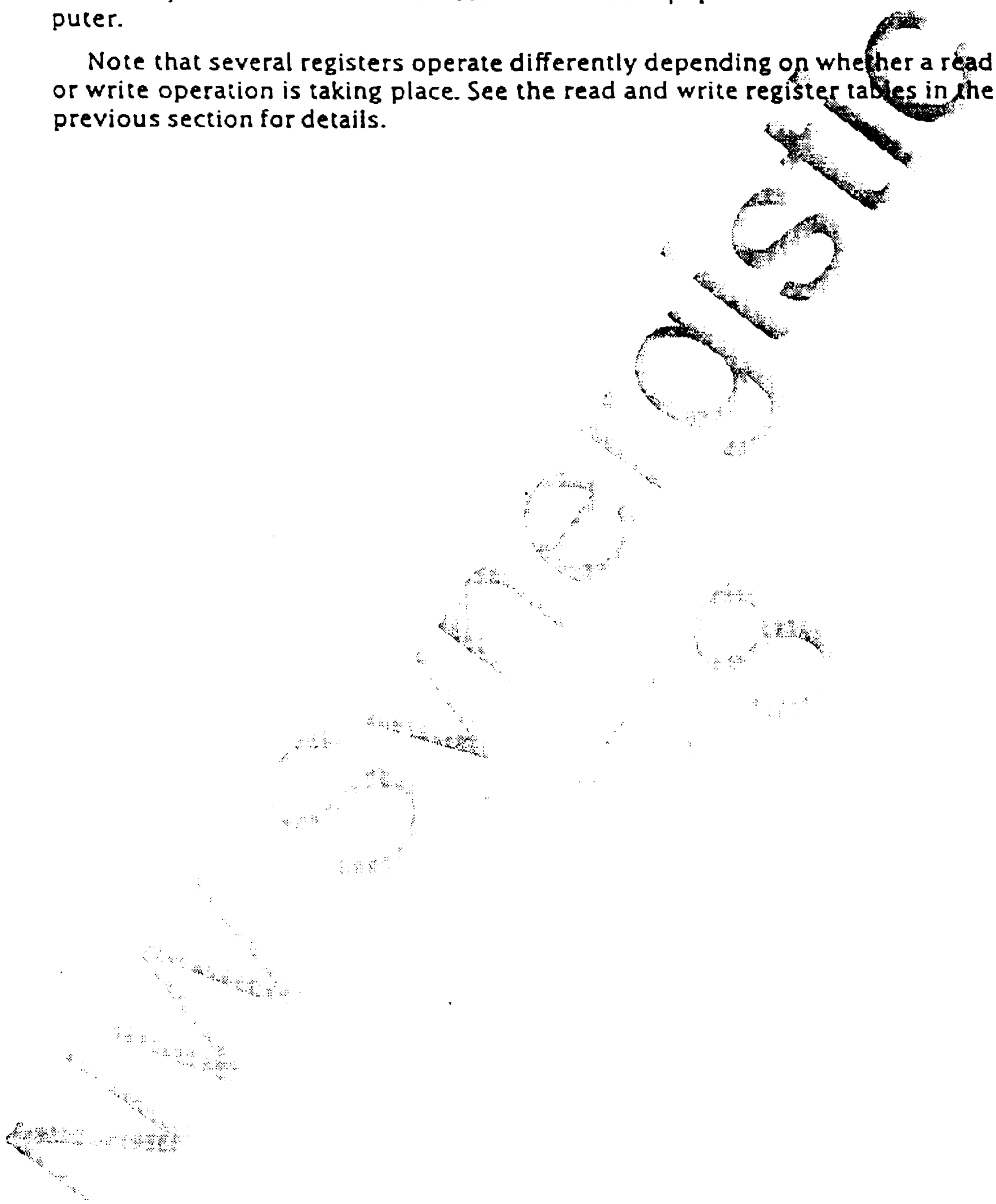
—	Used
BLANK	Not used, don't care. However, writing 0 is recommended to ensure future compatibility.

* See appendix E for the LC8951.

11. REGISTER DESCRIPTION

This section describes each of the registers on the LC8950 RCHIP. A range of register options including levels of error correction, can be selected. These allow fully customized interfaces for both the CD equipment and the host computer.

Note that several registers operate differently depending on whether a read or write operation is taking place. See the read and write register tables in the previous section for details.



11.1 AR — Address Register

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
				Register Address				R/W

This 4-bit register holds the addresses used to specify the LC8950's 16 internal data registers (R0 to R15).

A register address must be loaded into AR before reading from or writing to that register.

AR increments each time a read or write operation is performed at registers R1 to R15. When the register address reaches R15, the next I/O resets it to 0. Register I/O does not increment AR when the register address is R0.

AR is selected when RS and $\overline{CS}$ are LOW.

11.2 COMIN/SBOUT — Command Input Register/Status-Byte Output Register (Register R0)

During reads, this register is COMIN, the command input register; during writes it is SBOUT, the status-byte output register. A read/write operation to this register does not increment AR.

11.2.1 COMIN — Command Input Register

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
msb							lsb	R

This register is a 8-byte circular buffer between the host computer and the controller. It is configured as a FIFO memory.

When commands from the host computer fill one or more bytes of the FIFO buffer, the $\overline{\text{CMDI}}$ bit in the IFSTAT register is set to 0, forcing the $\overline{\text{INT}}$ pin LOW.

When the command buffer is emptied, $\overline{\text{CMDI}}$ is set to 1, and reading the COMIN register in this state returns a value of FFH.

11.2.2 SBOUT — Status-Byte Output Register

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
msb							lsb	W

This one-byte register buffers status bytes to the host computer. Multiple status bytes are passed to the host byte-by-byte.

When the SOUTEN bit in the IFCTRL register has been set to 1, writing to the SBOUT register sets the $\overline{\text{STBSY}}$ bit to 0. If the $\overline{\text{STWAI}}$ bit is 0, the $\overline{\text{STEN}}$ pin is immediately set to LOW to inform the host computer that the status byte is ready to be read out.

If the $\overline{\text{STWAI}}$ bit is set to 1 and the $\overline{\text{DTEN}}$ bit in the IFSTAT register is also 1, both the $\overline{\text{STEN}}$ pin and the $\overline{\text{STBSY}}$ go LOW when writing to SBOUT is completed. However, if the $\overline{\text{STWAI}}$ bit is set to 0 and the $\overline{\text{DTEN}}$ bit is 0, the $\overline{\text{STEN}}$ pin is held HIGH until the $\overline{\text{DTEN}}$ bit goes HIGH; thereafter it goes LOW. See figure 13 *Status-Byte Processing during Normal Operation* and figure 14 *Status-Byte Processing when WAIT is Enabled.*

The $\overline{\text{STEN}}$ pin (and the $\overline{\text{STEN}}$ bit in the IFSTAT register) goes HIGH under the following conditions.

- The host computer reads the status byte while the $\overline{\text{STEN}}$ pin is LOW.
- The SOUTEN bit in the IFCTRL register is 0.
- The host writes a command while the $\overline{\text{CMDBK}}$ bit in the IFCTRL register is 0.
- The chip is reset.

11.3 IFSTAT/IFCTRL — Interface Status Register/Interface Control Register (Register R1)

During reads this register is IFSTAT, the interface status register; during writes it is IFCTRL, the interface control register. A read/write operation to this register automatically increments AR.

11.3.1 IFSTAT — Interface Status Register

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
$\overline{\text{CMDI}}$	$\overline{\text{DTEI}}$	$\overline{\text{DECI}}$	1	$\overline{\text{DTBSY}}$	$\overline{\text{STBSY}}$	$\overline{\text{DTEN}}$	$\overline{\text{STEN}}$	R

This register indicates the status of the controller and host interfaces.

The operation of the $\overline{\text{DTEN}}$ and $\overline{\text{STEN}}$ bits resembles that of the $\overline{\text{DTEN}}$ and $\overline{\text{STEN}}$ pins. They differ in that the rising edges of signals at the $\overline{\text{DTEN}}$ and $\overline{\text{STEN}}$ pins are generated at the falling edge of the signal at pin $\overline{\text{HRD}}$, while the states of the $\overline{\text{DTEN}}$ and $\overline{\text{STEN}}$ bits change at rising edge of $\overline{\text{HRD}}$. See figure 18 *Transfer Start Delays* and figure 21 *Data-Transfer Completion.*

Bit 4 is unused and always returns a value of 1.

1. $\overline{\text{CMDI}}$ — Command Interrupt

Status	Operation
0	Interrupt set Indicates that there are command bytes present in the COMIN register.
1	Interrupt cleared Indicates that there are no command bytes in the COMIN register.

The interrupt clears when the controller empties the FIFO command buffer by reading successive command bytes. See figure 11 *Interrupt Control Flags.*

2. $\overline{\text{DTEI}}$ — Data-Transfer End Interrupt

Status	Operation
0	Set interrupt set
1	Set interrupt cleared

Writing to the DTACK register clears the interrupt. See figure 11 *Interrupt Control Flags.*

3. $\overline{\text{DECI}}$ — Decoder Interrupt

Status	Operation
0	Interrupt set
1	Interrupt cleared

Reading the STAT3 register clears the interrupt. See figure 11 *Interrupt Control Flags.*

4. $\overline{\text{DTBSY}}$ — Data Busy

Status	Operation
0	Data-transfer hardware busy
1	Data-transfer hardware idle

Sets $\overline{\text{DTBSY}}$ to 0. See figures 16 and 17 *Initiating Data Transfer (1)* and *(2).*

5. $\overline{\text{STBSY}}$ — Status Busy

Status	Operation
0	Status-byte readout hardware busy
1	Status-byte readout hardware idle

6. $\overline{\text{DTEN}}$ — Data Enable

Status	Operation
0	Data transfer in progress
1	No data transfer in progress

7. $\overline{\text{STEN}}$ — Status Enable

Status	Operation
0	Status-byte transfer in progress
1	No status-byte transfer in progress

11.3.2 IFCTRL — Interface Control Register

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
CMDIEN	DTEIEN	DECIEN	$\overline{\text{CMDBK}}$	$\overline{\text{DTWAI}}$	$\overline{\text{STWAI}}$	DOUTEN	SOUTEN	W

This register provides control functions for the host interface.

Resetting the chip clears all bits in this register.

1. CMDIEN, DTEIEN, DECIEN — Enable bits for CMDI, DTEI and DECI

   These are all interrupt mask bits. Note that even when a mask bit is set to 1, the corresponding interrupt bit in the IFSTAT register is unmasked. Setting an interrupt forces the $\overline{\text{INT}}$ pin to LOW. See figure 11 *Interrupt Control Flags.*

   CMDIEN, DTIEN and DECIEN do not clear interrupt requests. The $\overline{\text{INT}}$ pin will immediately go LOW if an interrupt is ended (by setting a bit to 1) during an interrupt request.

2. $\overline{\text{CMDBK}}$ — Command Break

   This bit enables/disables the command break function.

Status	Operation
0	Command break enabled
1	Command break disabled

   A command break is the suspension of data or status-byte transfer to the host computer caused when the host computer writes a command byte to the command buffer. The data-transfer end interrupt, $\overline{\text{DTEI}}$ is not generated by a command break.

3. $\overline{\text{DTWAI}}$ — Data-Transfer Wait

   This bit enables/disables the data-transfer wait function.

Status	Operation
0	Data-wait function enabled
1	Data-wait function disabled

   The data wait function allows the LC8950 to delay the hardware execution of a data transfer until a status-byte transfer (a $\overline{\text{STEN}}$ pin LOW condition) is cleared. In other words, if the DTTRG register is written to while the $\overline{\text{STEN}}$ bit in the IFSTAT register is 0, the $\overline{\text{DTEN}}$ bit is not cleared until the $\overline{\text{STEN}}$ bit is set to 1.

   Disabling the data-transfer wait function allows data transfers to take place independently of status-byte transfers. Also note that these enable bits do not clear interrupt requests.

   The $\overline{\text{INT}}$ pin will immediately go LOW if an interrupt is enabled (by setting a bit to 1) during an interrupt request.

4. $\overline{\text{STWAI}}$ — Status-Transfer Wait

This bit enables/disables the status-transfer wait function.

Status	Operation
0	Status-wait function enabled
1	Status-wait function disabled

The status-transfer wait function allows the LC8950 to delay the hardware execution of a status transfer until a data transfer (a $\overline{\text{DTEN}}$ pin LOW condition) is cleared. In other words, if the $\overline{\text{SBOUT}}$ register is written to while $\overline{\text{DTEN}}$ bit in the IFSTAT register is 0, the $\overline{\text{STEN}}$ bit is not cleared until the $\overline{\text{DTEN}}$ bit is set to 1.

Disabling the data wait function allows status-byte transfers to take place independently of data transfers.

5. DOUTEN, SOUTEN — Data Output Enable and Status Output Enable

DOUTEN enables/disables data transfers; SOUTEN enables/disables status-byte transfers.

Status	Operation
0	Output disabled
1	Output enabled

When DOUTEN is set to 0, any data transfers in progress are aborted. Similarly, when SOUTEN is set to 0, any status-byte transfers in progress are aborted.

11.4 DBCL, DBCH — Data Byte Counters (Registers R2 & R3)

DBCL	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	lsb								R/W

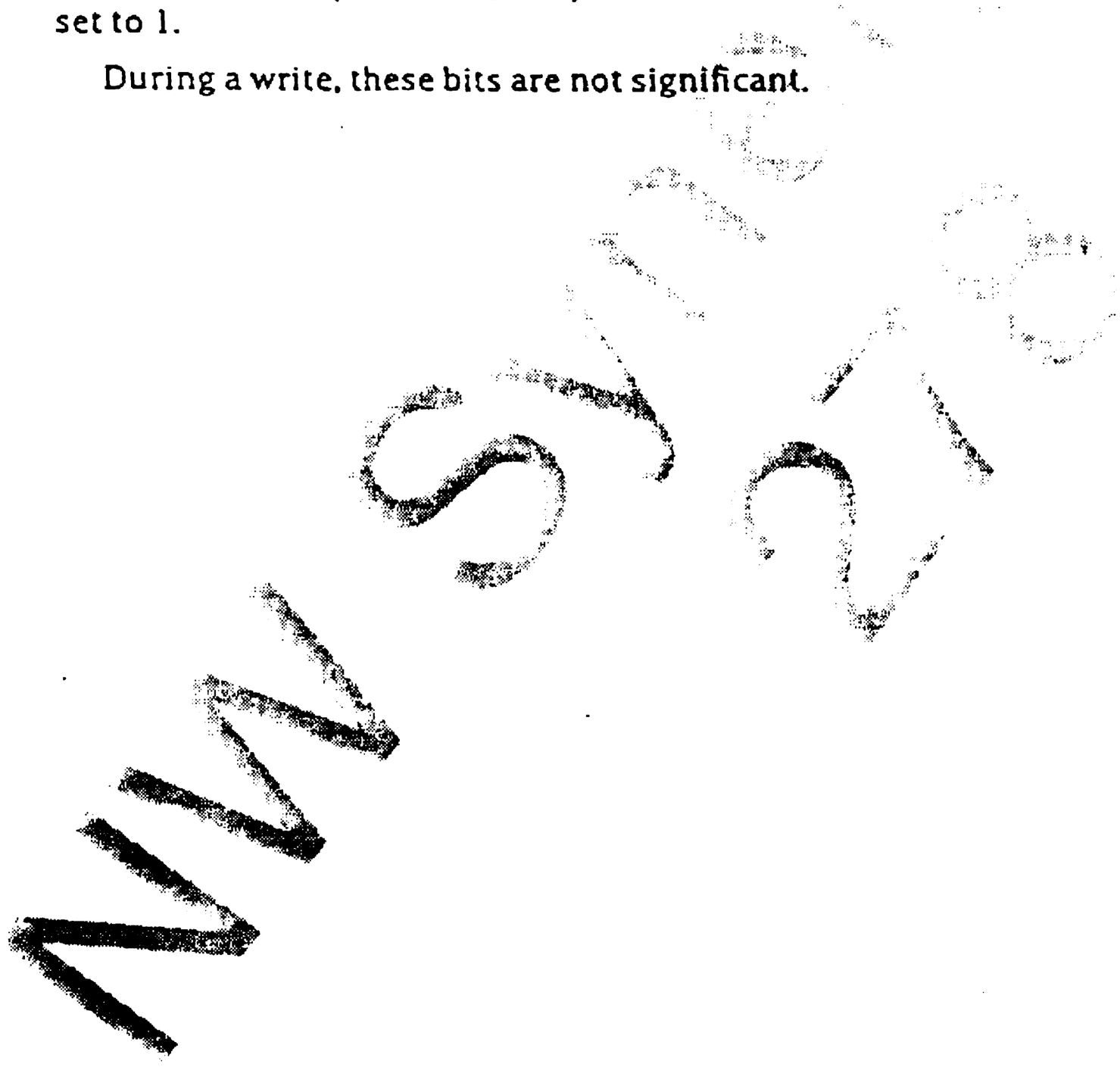
DBCH	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	DTEI during reads				msb				R/W

DBCL and DBCH constitute a 12-bit counter used to set or monitor the number of bytes sent to the host computer during data transfers. DBCL (R2) and DBCH (R3) are the lower- and upper-byte counters, respectively.

To initiate a data transfer to the host computer, the controller writes the number of bytes to be transferred minus one to this counter. Then, while the $\overline{\text{DTEN}}$ pin remains LOW, the counter transfers bytes to the host, decrementing the counter by one each time a byte is passed. When the count reaches -1, that is, when a borrow occurs, the transfer halts and a data-transfer end interrupt is generated.

During a read, Bits 4 to 7 of the DBCH register have the same value as the four bits in DTEI (Bit 6 in IFSTAT) and indicate the end of a transfer when all are set to 1.

During a write, these bits are not significant.



11.5 DACL, DACH — Data Address Counter (Registers R4 & R5 during writes)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
DACL								lsb	W

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
DACH	msb								W

This 16-bit write-only register is used to specify the RAM buffer address of the head byte of the data blocks to be transferred to the host.

Once the address has been set, it is incremented automatically as successive bytes are transferred into the 2-byte FIFO transfer buffer. Note, however, that the number of times the counter is incremented may not match the number of read operations performed by the host. As loading and transfer are performed asynchronously, byte counts may not match.

11.6 DTTRG — Data Transfer Trigger (Register R6 during writes)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
								W

The controller writes to this register to activate the hardware for data transfer to the host. The content of the data written is not significant.

For the write to take place, the DOUTEN bit in the IFCTRL (R1–WR) register must be set to 1. When the data transfer hardware is operating, the $\overline{\text{DTBSY}}$ bit of the IFSTAT (R1–RD) register is set to 0, and the DTTRG register must not be written to.

11.7 DTACK — Data Transfer End Acknowledge (Register R7 during writes)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
								W –

The controller writes to this register to clear the data-transfer end interrupt condition at the $\overline{\text{DTEI}}$ bit in the IFSTAT register and at the $\overline{\text{INT}}$ pin.

The content of the data written is not significant.

11.8 HEAD0 to HEAD4 — Header/Subheader Data (Registers R4 to R7 during reads)

HEAD0	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	msb	MINUTES/FILE NUMBER						lsb	R

HEAD1	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	msb	SECONDS/CHANNEL NUMBER						lsb	R

HEAD2	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	msb	BLOCKS/SUBMODE						lsb	R

HEAD3	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	msb	MODE/CODING INFORMATION						lsb	R

These registers are used to hold the header and subheader data of the current block.

To read the header data set, the SHDREN bit in the CTRL1 register is set to 0; to read the subheader data, set SHDREN to 1. The following byte numbers correspond to the block (sector) byte numbers in Modes 1 and 2 in which the head of sync block is Byte 0.

- When SHDREN is 0:

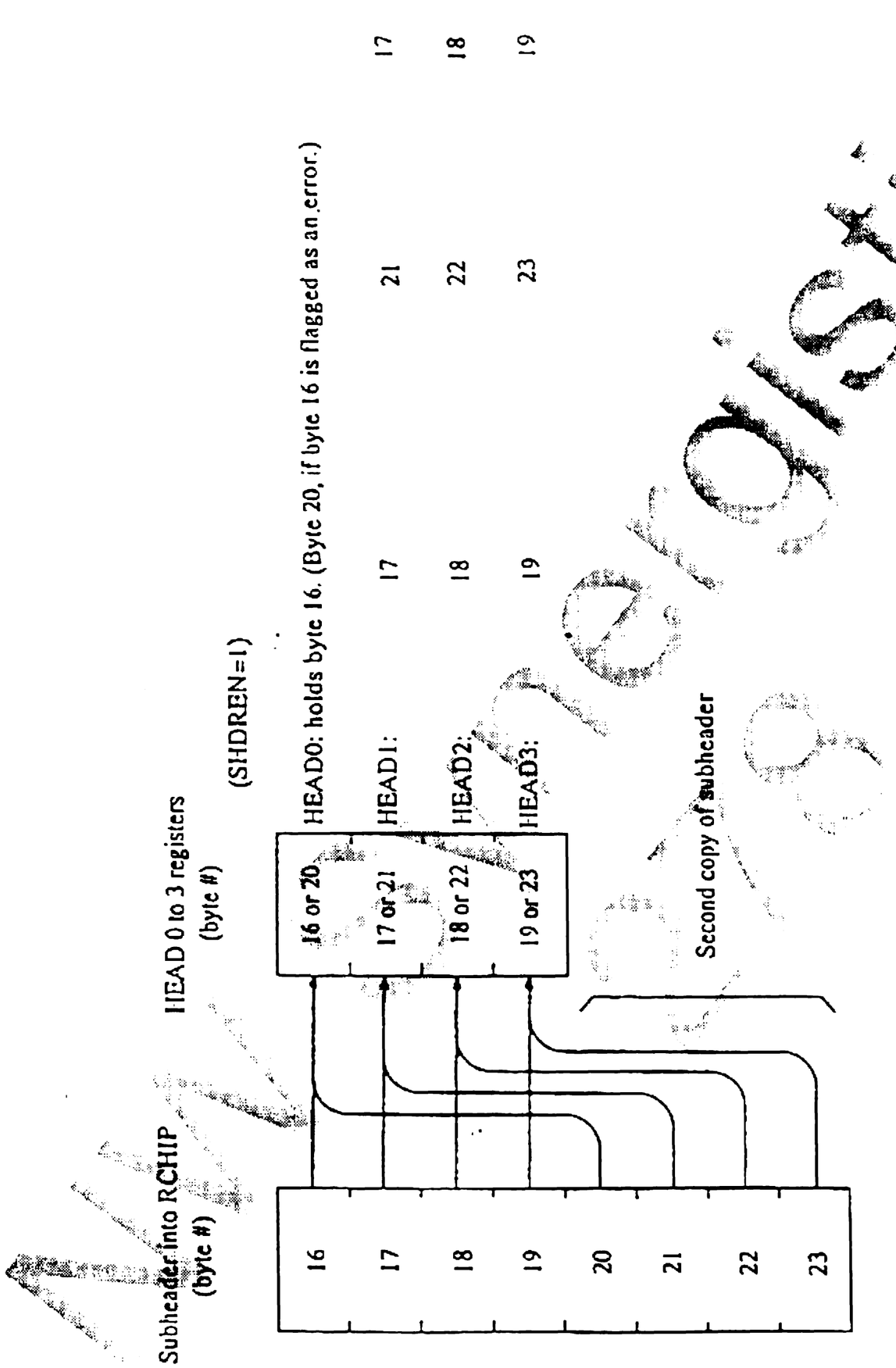
  HEAD0: Header MINUTES (byte 12)
  HEAD1: Header SECONDS (byte 13)
  HEAD2: Header BLOCKS (byte 14)
  HEAD3: Header MODE (byte 15)

- When SHDREN is 1:

  HEAD0: Subheader FILE NUMBER (byte 16 or 20)
  HEAD1: Subheader CHANNEL NUMBER (byte 17 or 21)
  HEAD2: Subheader SUBMODE NUMBER (byte 18 or 22)
  HEAD3: Subheader CODING INFORMATION (byte 19 or 23)

These registers normally hold subheader data from Bytes 16 to 19. If the error flags for any of these bytes are set, the data for that byte are read from the corresponding location in Bytes 20 to 23.

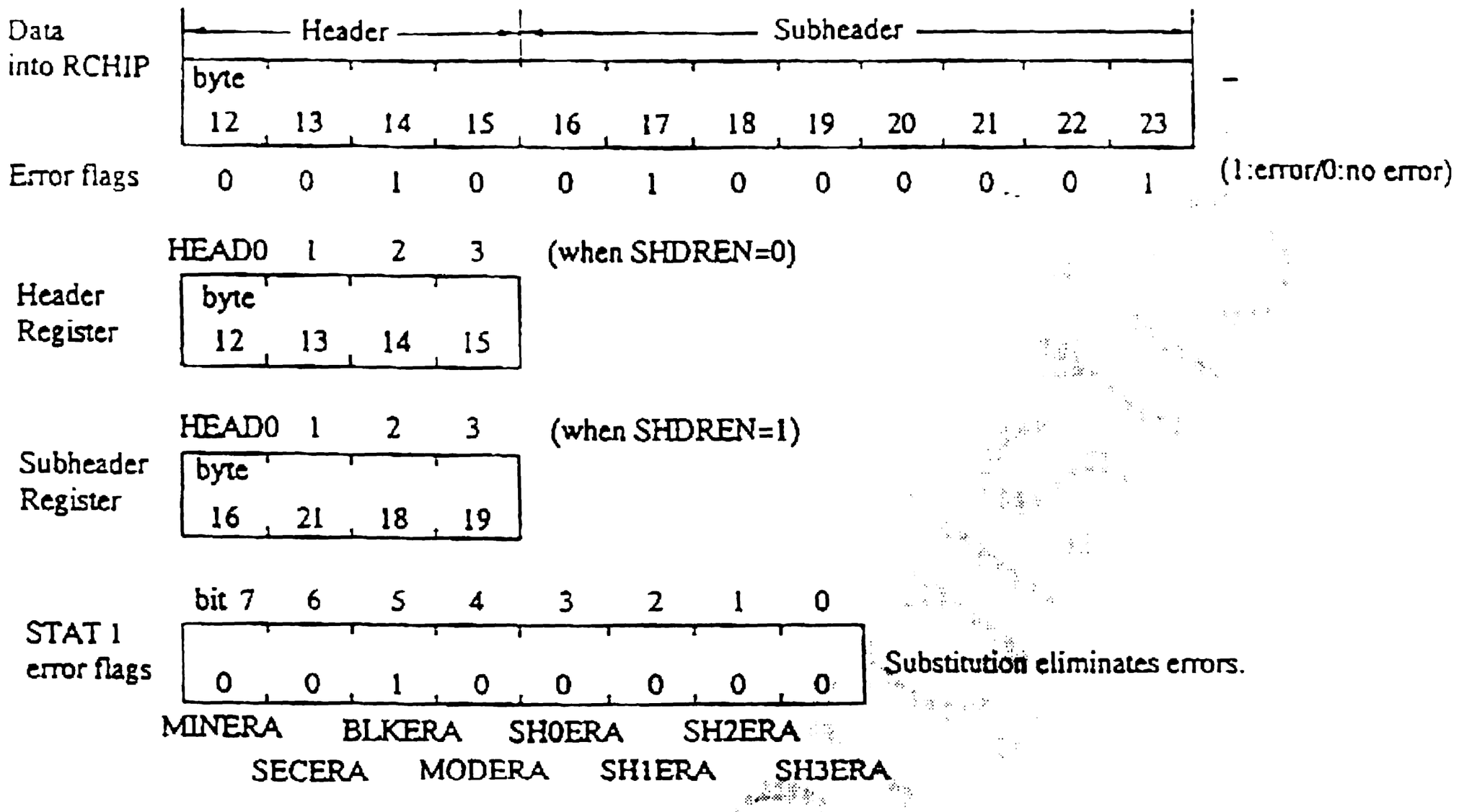
If errors occur both in the original data and in the designated corresponding bytes, Bytes 20 to 23 are read for the subheader data. See figure 7 *Subheader Data*. The error flags for the header and subheader bytes can be read from the STAT1 register. After error correction, these registers display the corrected header and subheader data of the current block.



Note:
The CD-I disc specification provides duplicate copies of the subheader on disc.

Figure 7. Subheader Data

Example 1



Example 2

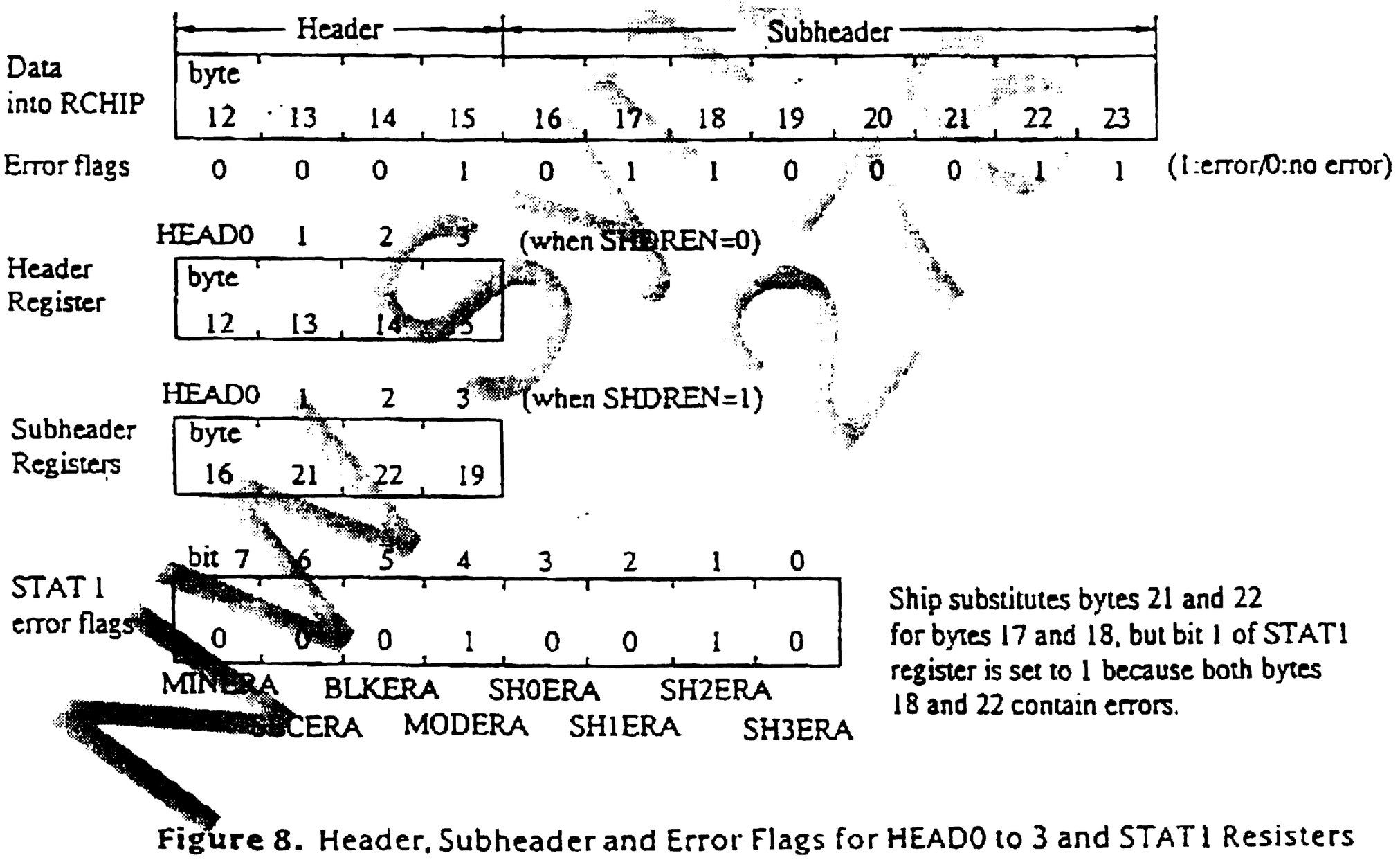


Figure 8. Header, Subheader and Error Flags for HEAD0 to 3 and STAT1 Resisters

When the LC8950 is set to monitor-only mode, these registers display the header and subheader of the block currently being written to the buffer RAM. In other modes, these registers display the block currently being decoded.

After a reset, the register state is undefined.

11.9 PTL, PTH — Block Pointers (Register R8 & R9 during reads; Registers R12 & R13 during writes)

PTL	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
								lsb	R/W

PTH	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
	msb								R/W

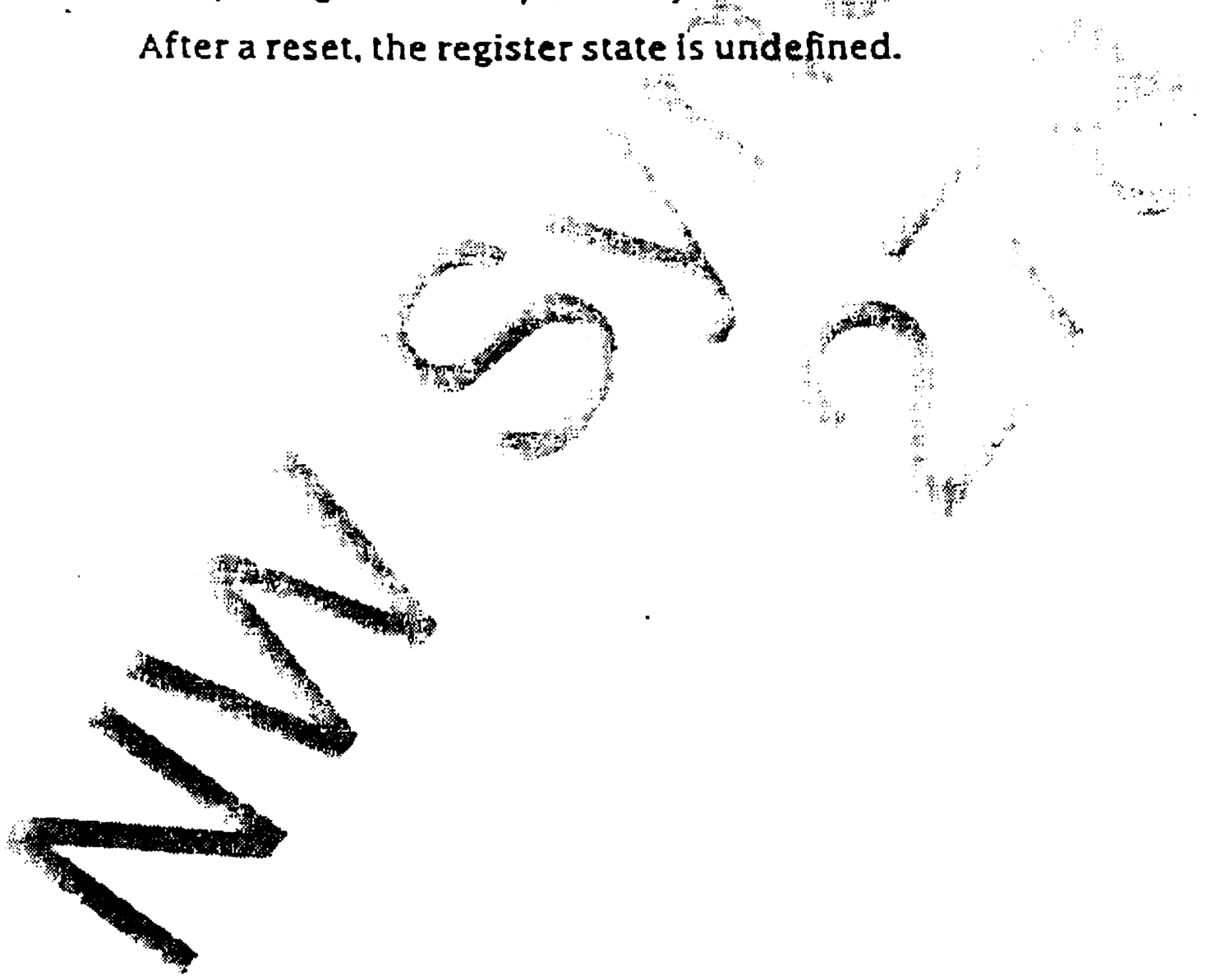
PTL and PTH are the lower and upper bytes, respectively, of this pair of registers. They hold a 16-bit pointer to the RAM address of the head of the current data block after correction. Writing to this register directs the LC8950 to perform error correction on any block in the buffer RAM.

The LC8950 defines the MINUTES byte in the header (Byte 12) at the head of the block and the 12 sync bytes (Bytes 0 to 11) are at the tail of the block. Each block contained in the buffer is taken to be the 2352 bytes from the MINUTES byte of the header through the 12 sync bytes of the next block.

The controller can transfer the decoding block back to the host by copying the address in this register and writing it to the DACL/DACH register immediately after decoder interrupts.

When the WRRQ bit in the CTRL0 register is set to 1, this pointer is updated at the sync signal of every 2352-byte block.

After a reset, the register state is undefined.



11.10 WAL, WAH — Write Address Register (Registers R10 & R11 during reads; Registers R8 & R9 during writes)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
WAL								lsb	R/W

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
WAH	msb								R/W

These registers contain a pointer into write the address to the RAM buffer for the raw data input from the CD device. WAL and WAH are the lower and upper bytes, respectively.

This pointer is automatically incremented during data transfer—that is, while the DECEN and WRRQ bits in the CTRL0 register are set to 1, a read operation on the write address register will not return a meaningful value. To ensure that the returned value is accurate, the register contents should be read only after DECEN or WRRQ has been set to 0. For the same reason, either DECEN or WRRQ should be set to 0 before writing an address to this register.

After a reset, the register contents are undefined.

11.11 CTRL0 — Control 0 (Register R10 during writes)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
DECEN	EDCRQ	E01RQ	AUTORQ	ERAMRQ	WRRQ	QRQ	PRQ	W

Resetting the chip sets all the bits in this register to 0.

1. DECEN — Decoder Enable

   This bit enables/disables decoding functions.

Status	Operation
0	Disable decode.
1	Enable decode.

   Setting this bit to 1 enables the decoding functions, including write to buffer, decoding EDC (error detection code), and decoding ECC (error correction codes such P-codes and Q-codes). This also enables the WRRQ, EDCRQ, QRQ and PRQ bits in the CTRL0 register.

2. EDCRQ — EDC Decoding Request

   This bit is set to 1 while DECEN is 1 to enable the error detection (decoding EDC) after and decoding ECC.

   The EDC decoding, which is timed using the LC8950's internal block-sync signal, is performed after all other error correction processes have been completed.

   The completion of the EDC decoding generates a decoder interrupt.

3. E01RQ — Error Detect-and-Correct Request

   This bit allows system designers to select the most appropriate error handling.

Status	Operation
0	Disable error correction of bytes for which an error has been detected, but not yet corrected.
1	Enable error correction of bytes for which an error has been detected, but not yet corrected.

   Setting bit to 0 can suppress miscorrections resulting from too many errors when an efficient error detection algorithm is implemented in the CIRC (Cross-Interleaved Reed-Solomon Code) decoder, and when errors are localized. Error correction improves significantly at higher error rates when an efficient CIRC decoding scheme is implemented.

   Erasure is disabled when the ERAMRQ bit in the CTRL0 register is 0. Setting E01RQ to 0 suppresses error correction.

4. AUTORQ — Automatic Correction Mode Selection Request

Setting this bit to 1/0 during MODE 2 enables/disables automatic error correction mode when the FORM bit in the subheader is read.

This bit is significant only during MODE 2 error correction—that is, when the MODRQ bit in the CTRL1 register is set to 1.

Status	Operation
0	Disable automatic error correction.
1	Enable automatic error correction.

The CRC check performed by the EDC also differs in accordance with the FORM flag setting.

5. ERAMRQ — Erasure RAM Request

This bit enables/disables reading of the error flag data in the erasure RAM, that is the one-bit buffer RAM for C2PO.

Status	Operation
0	Disable error flag reads.
1	Enable error flag reads.

Setting ERAMQ to 0 disables the erasure flag, causing read operations performed on the erasure RAM to return a 0 (indicating no errors), so that only error detection and correction are allowed. The 0 setting is used when error, interpolation, or previous-value-hold flags in the CIRC C2 decoding phase are not outputted by the CIRC decoder synchronized with the output data blocks from the CD equipment.

6. WRRQ — Write Request

This bit enables/disables writes from the CD equipment to the buffer. It also enables/disables updating of the block pointer (PTL/PTH) each time a data block is received.

Status	Operation
0	Disable data writes to the buffer and PTL/PTH updates.
1	Enable data writes to the buffer and PTL/PTH updates.

PTL/PTH updates are timed using the LC8950's internal block-sync signal.

If error correction is enabled and WRRQ is 0, the same data block will be processed repeatedly. This allows correction of errors that could not be handled in the single pass performed during real-time processing. See section 12.5.3 *Repeated-correction mode.*

7. QRQ, PRQ — Q-code/P-code Correction Requests

These bits enable/disable error correction using the Q- and P-codes, respectively.

Status	Operation
0	Disable Q-code (or P-code) error correction
1	Enable Q-code (or P-code) error correction

The QRQ and PRQ setting operations are timed to the internal block-sync signal.

The block being corrected can be determined by reading the header registers while the decoder interrupt is set. The block's location in the buffer can be determined by checking the read pointer (PTL/PTH) contents.

The specified error correction process is performed in the order QD, PD, QE and PE within each block, where Q and P indicate the Q- and P-codes and D and E, error correction and erasure. D for single-error correction with detect-and-correct algorithm and E for single- or double-error correction, also with erasure flags.

The detect-and-correct algorithm corrects one symbol error per code word; erasure correction corrects up to two symbol errors per code word.

Note that the detect-and-correct algorithm can also detect and correct CIRC decoder miscorrection errors which are unflagged by C2PO. Erasure correction, on the other hand, does not perform any error detection.

When the detect-and-correct algorithm detects a single error in a code word, it also examines the erasure flag to ascertain if the location for the corrected error is correct. The error byte (symbol) is taken to be correct and the erasure flag cleared if the erasure flag and its location are identical to that of the corrected error. When they are not, this indicates that further errors are present at other locations, which cannot be determined. Moreover, the LC8950 leaves the errors and the erasure flag, as these errors will be corrected during the next error correction process.

11.12 CTRL1 — Control 1 (Register R11 during writes)*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
SYIEN	SYDEN	OSCREN	COWREN	MODRQ	FORMRQ	MBCKRQ	SHDREN	W

The reset function clears all the flags in this register.

1. SYIEN — Internal Block-Sync Insertion Enable

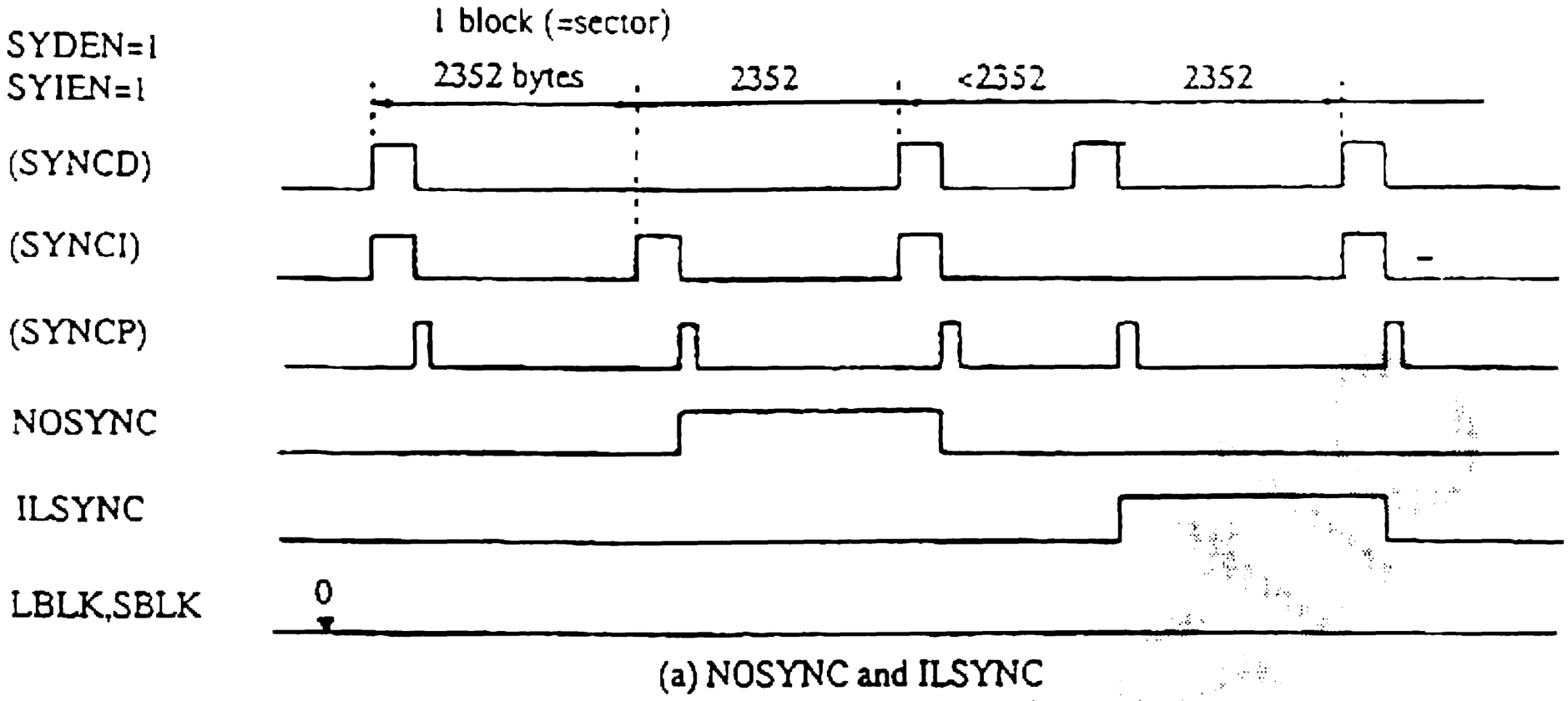
   This bit enables/disables the internal block-sync insertion.

Status	Operation
0	Disable internal block-sync insertion.
1	Enable internal block-sync insertion.

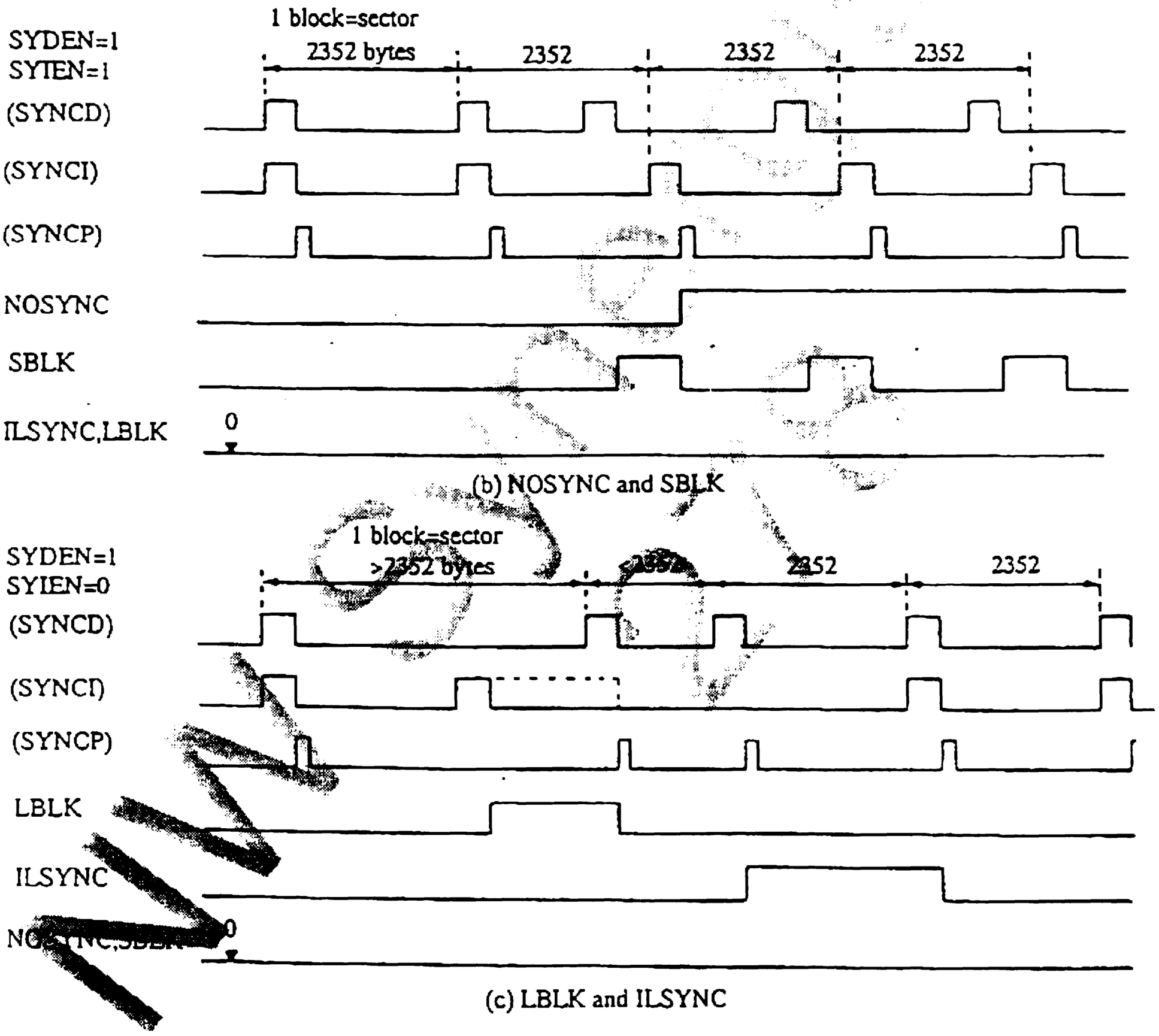
As each word consists of two bytes, every [illegible] words counted define a block boundary. Enabling SYIEN prevents loss of synchronization when an error occurs in a sync pattern during data read.

Note that the LC8950 is fully synchronized with both the inserted sync and the detected sync when both are enabled. See figure 9 *Inserted and Detected Sync Synchronization* below.

* See appendix E for the LC8951.



(a) NOSYNC and ILSYNC



(b) NOSYNC and SBLK

(c) LBLK and ILSYNC

Figure 9. Inserted and Detected Sync Synchronization

2. SYDEN — Sync-Detected Enable

This bit enables/disables sync detection from the input data.

Status	Operation
0	Disable sync detection.
1	Enable sync detection.

Enabling this bit synchronizes decoder operations with the sync pattern detected in the input data.

Note that the LC8950 is fully synchronized with both the inserted sync and the detected sync when both are enabled. See figure 9 *Inserted and Detected Sync Synchronization.*

3. DSCREN — Descrambling Enable

This bit enables/disables the descrambling function.

Status	Operation
0	Disable descrambling.
1	Enable descrambling.

Setting DSCREN to 1 allows the LC8950 to read the raw data on disc, even if they are CD digital audio signals.

4. COWREN — Correction Write Enable

This bit enables/disables rewriting of error bytes in the buffer during error correction.

Status	Operation
0	Disable error-byte rewriting.
1	Enable error-byte rewriting.

Setting COWREN to 0 and EDCRQ to 1, allows CRC checks without error correction. Disc block error rates can be checked using raw disc data and by monitoring the CRCOK flag in the STAT0 register.

5. MODRQ — Mode Request

This bit sets the decoding mode.

Status	Operation
0	Mode 1
1	Mode 2

This bit should be set manually to match the CD equipment source mode. Note that the correct decoder mode setting is *not* determined automatically.

The Mode byte in the header does not change as pre- and post-gap sectors and track number identifiers separate the sectors of different modes. Also, errors resulting from mode detection before correction must be

avoided. In the case of Mode 2, in particular, the Mode byte cannot be corrected because of its error correcting code definition.

6. FORMRQ — Form Request

This flag sets the decoding form. It is significant only when the AUTORQ bit in the CTRL0 register is 0 and the MODRQ in the CTRL1 register is 1.

Status	Operation
0	Mode 2, Form 1
1	Mode 2, Form 2

Decoding form settings are used in MODE 2 only and should be set manually.

7. SHDREN — Subheader Read Enable

This bit toggles header and subheader data between registers HEAD0 to HEAD3.

Status	Operation
0	Header data
1	Subheader data

Note that the LC8950 has separate 4-byte registers for the header and subheader data. Both the header and subheader of the current block can be read by toggling the SHDREN bit from 0 to 1.

11.13 STAT0 — Status 0 (Register R12 during reads)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
CRCOK	ILSYNC	NOSYNC	LBLK	WSHORT	SBLK	ERABLK	UCEBLK	R

Resetting the chip clears all bits in this register.

1. CRCOK — Cyclic Redundancy Check OK

   Set by EDC in accordance with the results of the CRC check.

Status	Operation
0	Incorrect CRC result
1	Correct CRC result

2. ILSYNC — Illegal Sync

   This bit is set to 1 when the sync pattern is detected at an incorrect location (that is, at word counts between 0 and 1174), and the decoder has been retimed.

3. NOSYNC — No Sync

   This bit is set to 1 if the word counter reaches 1175 and no sync pattern has been detected in the input data. It indicates that the decoder timing is being controlled by the LC8950's internal block-sync signal when SYIEN in the CTRL1 register is set to 1.

4. LBLK — Long Block

   This bit is set to 1 if no sync signal has been detected while the SYIEN bit of the CTRL1 register is 0. This condition causes the block length to be extended.

5. WSHORT — Word Short

   This bit is set to 1 if word lengths shorter than 192 pulses of the XTALCK external clock signal (that is a half cycle at pin LRCK) are detected during decoding of the current block.

   Setting this bit suspends decoding of the current block, and is a fatal system error.

6. SBLK — Short Block

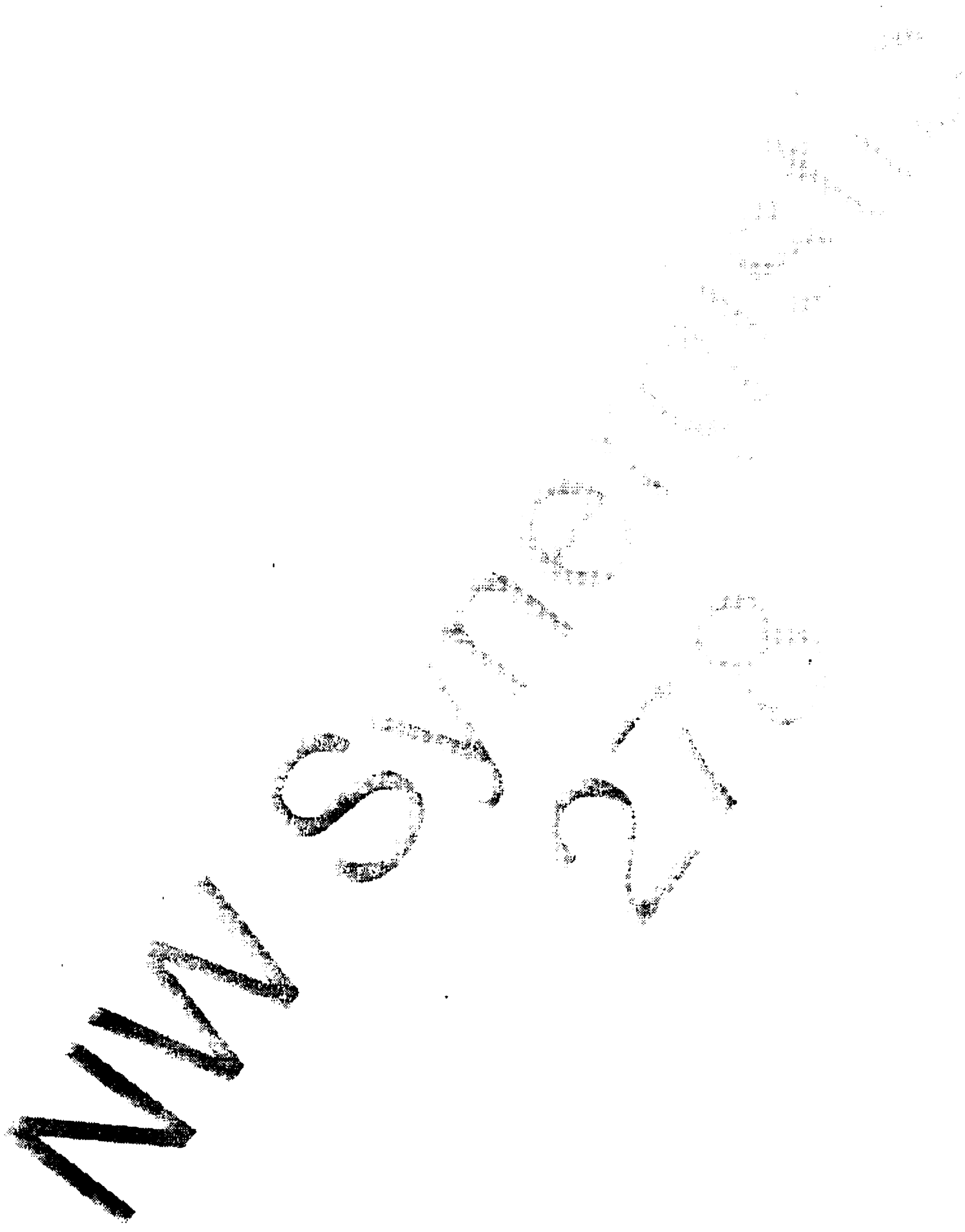
   This bit is set to 1 if the decoder is not retimed when a sync pattern is detected in an incorrect word counter location and is ignored while the SYDEN bit of the CTRL1 register is 0.

7. ERABLK — Erasures in Block

   This bit is set to 1 when one or more error flags remain after the error-correction process and are present between the first header byte and last EDC parity byte in the current block.

8. UCEBLK — Uncorrectable Errors in Block

   This bit is set to 1 when errors remain in one or more code words within the current block, which includes the ECC parity area, after the final ECC decoding process has been completed.



11.14 STAT1 — Status 1 (Register R13 during reads)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
MINERA	SECERA	BLKERA	MODERA	SH0ERA	SH1ERA	SH2ERA	SH3ERA	R

The bits in this register indicate the reliability of the data in the HEAD0 to HEAD3 registers.

Resetting the chip clears all bits in this register.

MINERA, SECERA, BLKERA and MODERA indicate errors in the MINUTES, SECONDS, BLOCKS and MODE bytes in the header of the current block.

SH0ERA, SH1ERA, SH2ERA and SH3ERA indicate errors in the FILE NUMBER, CHANNEL NUMBER, SUBMODE and CODING INFORMATION bytes of the subheader. These flags are set only if errors appear in both bytes of the doubly written subheader data; they remain 0 if none or only a single-byte error occurs.

11.15 STAT2 — Status 2 (Register R14 during reads)*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
RMOD3	RMOD2	RMOD1	RMOD0	MODE	NOCOR	RFORM1	RFORM0	R

The bits in this register indicate the mode and form settings of the current block.

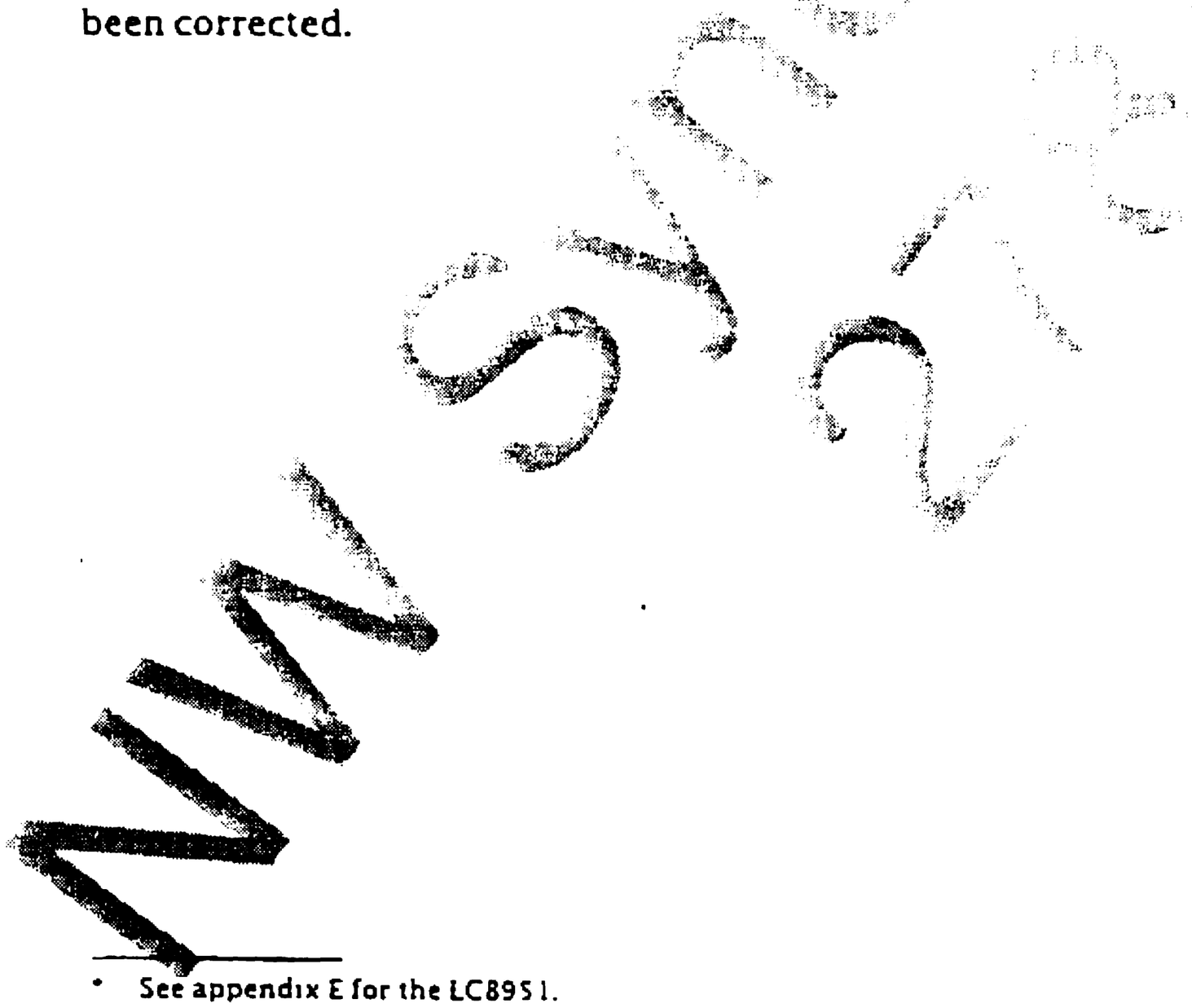
Bits	Setting
0X	Mode 1
10	Mode 2, Form 1
11	Mode 2, Form 2

X Don't care

Settings are determined by the status of the AUTORQ bit in the CTRL0 register, the MODRQ and FORMRQ bits in the CTRL1 register and the Form bit readout from the CD equipment.

Resetting the chip clears all the bits in this register.

When AUTORQ is set to 1, the mode and form data actually decoded from the current block can be checked. If the Form bit in this register differs from the Form bit in the subheader, it indicates that the subheader Form bit has been corrected.



* See appendix E for the LC8951.

11.16 STAT3 — Status 3 (Register R15 during reads)*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	R/W
$\overline{\text{VALST}}$	WLONG	CBLK						R

Reading this register sets the $\overline{\text{INT}}$ pin to HIGH and the $\overline{\text{DECI}}$ flag to 1 in the IFSTAT register, clearing a decoder interrupt.

The unused bits in this register are not significant.

1. $\overline{\text{VALST}}$ — Valid Status

   This bit is an valid/invalid flag for registers related to the decoder interrupt.

Status	Operation
0	Valid registers
1	Invalid registers

   The $\overline{\text{VALST}}$ flag goes to 0 at the same time as the decoder interrupt is generated, and doesn't go to 1 when the STAT3 register is read.

   The controller should read all decoder interrupt-related registers until $\overline{\text{VALST}}$ sets to 0. The $\overline{\text{VALST}}$ flag goes to 1 shortly before the internal sync is generated. See section 12.6 *Decoder Control*.

2. WLONG — Word Long

   This flag is set to 1 if word lengths longer than 192 pulses of the master clock are detected when LRCK is either HIGH or LOW.

   This flag has no effect on decoding operations.

3. CBLK — Corrected Block Flag

---

* See appendix E for the LC8951.

12. FUNCTIONAL DESCRIPTION AND OPERATION

12.1 Command Processing

To send a command, the host computer sets the $\overline{\text{ENABLE}}$ and $\overline{\text{CMD}}$ pins to LOW, and loads one or more command bytes into the COMIN register.

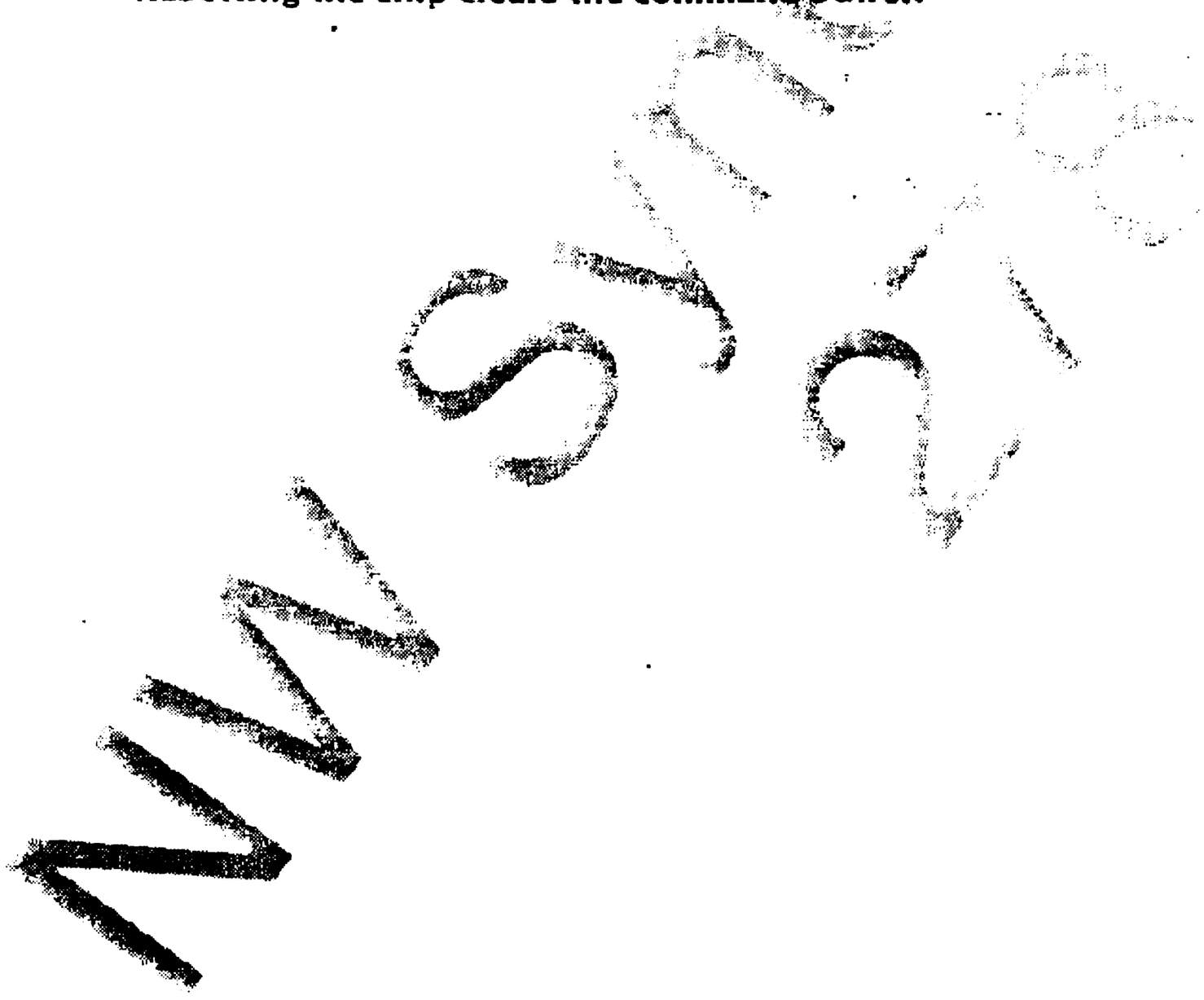
The COMIN register is an 8-byte FIFO buffer used by command write commands of the host and the read commands of the controller. This register allows these operations to be performed using completely independent timing. See figure 12 *Command Processing (1)*.

If the host attempts to send a command byte when the command buffer is full (that is, the buffer is full at the falling edge of the $\overline{\text{HWR}}$ signal), that command is ignored. See figure 12 *Command Processing (2)*. As the LC8950 does not support an external buffer-full signal, the controller should be designed to process commands promptly to prevent buffer overflow.

If the controller tries to read the command buffer when there are no commands (that is, the buffer is empty at the falling edge of the $\overline{\text{RD}}$ signal), the buffer returns a value of FFH. See figure 12 *Command Processing (2)*.

Whenever commands are present in the buffer, the LC8950 sets the command interrupt bit ($\overline{\text{CMDI}}$) to 0. If the command interrupt mask bit (CMDIEN) is set to 1, the $\overline{\text{INT}}$ pin will go LOW, signaling the interrupt to the controller.

Resetting the chip clears the command buffer.



12.2 Status Byte Processing

If the status-byte output enable bit (SOUTEN) is set to 1, the controller can transfer status bytes to the host. When the controller has written a status byte to the SBOUT register, the LC8950 completes the transfer to the host. The host reads this byte using the $\overline{HRD}$ pin when the $\overline{ENABLE}$ and $\overline{CMD}$ pins are LOW.

The controller should write to SBOUT only while the $\overline{STBSY}$ bit of the IFSTAT register is 1. Writing a byte to SBOUT while $\overline{STBSY}$ is set to 0 will disrupt the transfer in progress and the new status byte will be lost.

12.2.1 Normal operation

If the status-wait bit ($\overline{STWAI}$) is 1, thereby inhibiting the status-wait function, a write to SBOUT sets the $\overline{STBSY}$ and $\overline{STEN}$ bits in the IFSTAT register to 0. The $\overline{STEN}$ pin goes LOW, it signals the host that a status bit is ready for transfer. While $\overline{STEN}$ is LOW, the host can read the contents of the SBOUT register using the $\overline{HRD}$ pin when the $\overline{ENABLE}$ and $\overline{CMD}$ pins are LOW.

When the host starts reading the status byte, the LC8950 sets the $\overline{STBSY}$ bit to 1 to signal the controller that the SBOUT register is being read. The $\overline{STEN}$ pin goes HIGH. The $\overline{STEN}$ bit resets to 1 when the status-byte transfer has been completed, signaling the controller that the SBOUT register has been read.

The host should read the status byte only when the $\overline{STEN}$ pin is LOW. If the host tries to read SBOUT while $\overline{STEN}$ is HIGH and if the controller completes a write to SBOUT while the read operation is in progress, the $\overline{STBSY}$ bit will set to 0, but the $\overline{STEN}$ pin and $\overline{STEN}$ flag bit will go LOW just after reading is completed — indicating that a status byte is ready to send. Note, however, that the integrity of this status byte cannot be guaranteed. See figure 13 *Status Byte Processing during Normal Operation.*

12.2.2 When status byte wait is enabled

When the $\overline{STWAI}$ bit in the IFSTAT register is set to 0, the LC8950 checks the $\overline{DTEN}$ bit before sending the status byte. If the controller has written a status byte to SBOUT while $\overline{DTEN}$ is 1 (that is, there are no data ready for transfer), the status byte is transferred as described in section 12.2.1 above.

When SBOUT is written while $\overline{DTEN}$ is 0, the $\overline{STBSY}$ bit is set to 0. However the $\overline{STEN}$ bit remains 1 and the $\overline{STEN}$ pin is held HIGH until the data read operation by the host is complete. When $\overline{DTEN}$ resets to 1, the $\overline{STEN}$ bit sets to 0 and the $\overline{STEN}$ pin goes LOW. The transfer described in section 12.2.1 can now take place. See figure 14 *Status Byte Processing when WAIT is Enabled.*

12.2.3 Aborting status byte transfer

Three methods are available to block transfer of the status byte:

1. Reset.

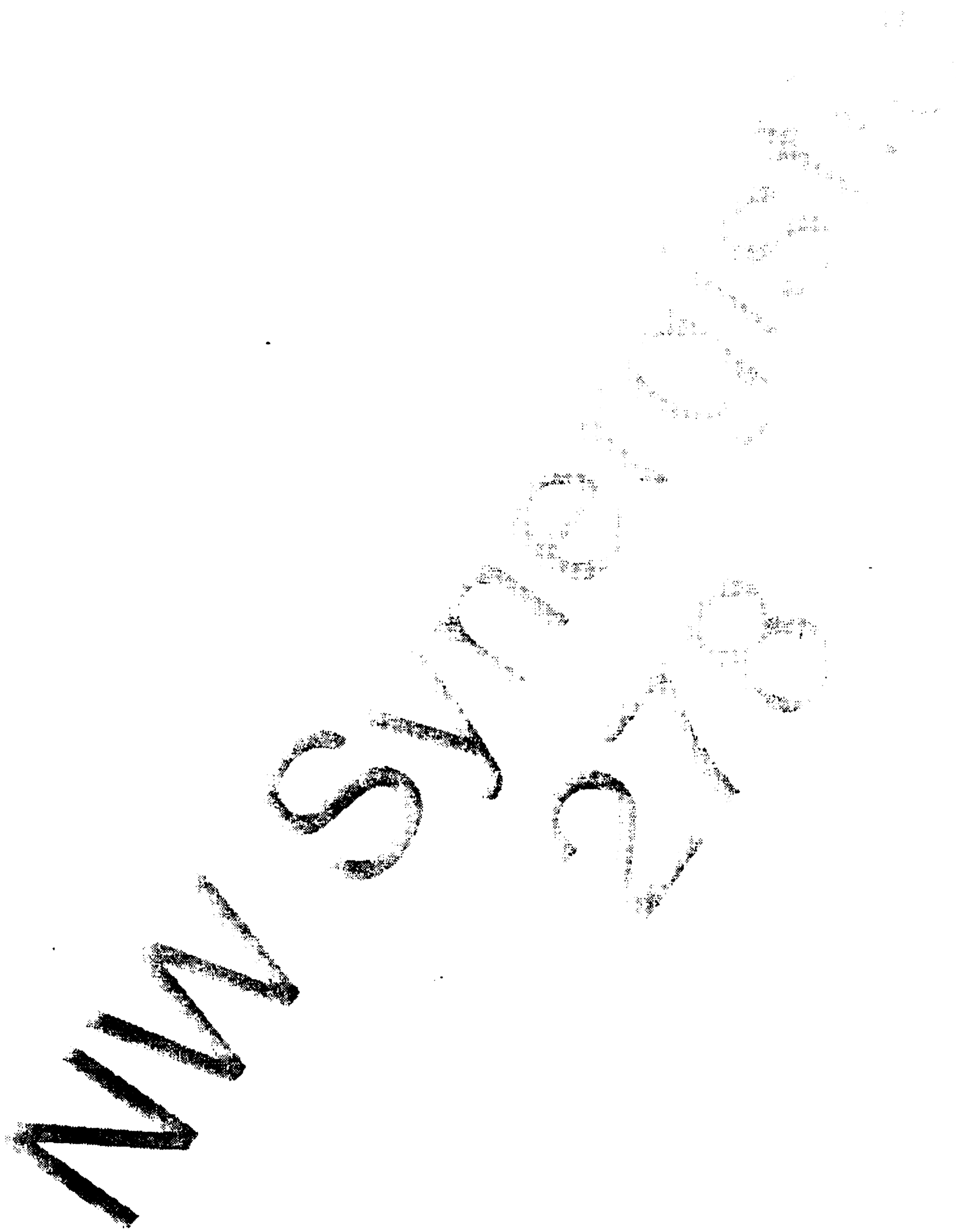
   Forcing the $\overline{RESET}$ pin to LOW or writing to the RESET register resets the LC8950, thus terminating status-byte transfers. See figure 15 *Status Byte Transfer Abort (1 & 2).*

2. Setting the SOUTEN bit to 0.

   Setting the SOUTEN bit in the IFCTRL register to 0 will abort status-byte transfers. See figure 15 *Status Byte Transfer Abort (3).*

3. Setting a command break.

   Setting the $\overline{\text{CMDBK}}$ bit of the IFCTRL register to 0 enables the host to send commands, inhibiting status byte transfers. See figure 15 *Status Byte Transfer Abort (4).*



12.3 Data Transfer

12.3.1 Summary

The LC8950 uses a special built-in 9-bit × 2-word alternating FIFO buffer register for data transfers.

When the controller instructs the LC8950 to carry out a data transfer, the transfer hardware writes the data and erasure flag at the specified address in the external data buffer RAM to the FIFO transfer buffer, then sets the $\overline{\text{DTEN}}$ pin to LOW, signaling the host that the LC8950 has data ready to be read. These data are output from the transfer buffer to the data bus using the $\overline{\text{HRD}}$ pin when the host sets the $\overline{\text{ENABLE}}$ pin to LOW and the $\overline{\text{CMD}}$ pin to HIGH.

The $\overline{\text{SELDRQ}}$ pin controls the data transfer mode. When $\overline{\text{SELDRQ}}$ is HIGH, software transfer with WAIT control is used. When $\overline{\text{SELDRQ}}$ is LOW, direct memory access (DMA) transfer with data request ($\overline{\text{DRQ}}$) control is used.

The data-transfer hardware reads data from the data buffer RAM and writes them to the FIFO buffer until the transfer buffer fills. When the LC8950 has transferred the number of bytes specified by the controller, the $\overline{\text{DTEN}}$ pin goes HIGH, and a data-transfer end interrupt is generated (the $\overline{\text{INT}}$ pin goes LOW and the $\overline{\text{DTEI}}$ flag in the IFSTAT register goes to 0).

12.3.2 Initiating data transfer

Data transfer is initiated using the following procedure:

1. The data-transfer hardware is enabled when a 1 is written to the DOUTEN bit in the IFCTRL register. When a reset or other operation sets DOUTEN to 0, it must be set to 1 before a transfer begins.
2. The number of bytes to be transferred minus one is written to the transfer byte counter. The lower byte is written to DBCL, the upper byte, to DBCH. Only the four lower bits of DBCH are significant—they limit a single transfer to 4096 bytes.
3. The memory address of the data to be transferred is set in the data address register. The lower byte is written to DACL, the upper byte to DACH.
4. A write is performed on the DTTRG register to initialize the FIFO transfer buffer and start the data-transfer operation, and the DTBSY bit in the IFSTAT register goes LOW. The content of the data written to DTTRG is not significant.

Caution

Under no circumstances should the controller be allowed to write to DBCL, DBCH, DACL, DACH or DTTRG while a transfer is in progress.

When the first byte has been written to the FIFO transfer buffer, the $\overline{\text{DTEN}}$ bit in the IFSTAT register sets to 0 and the $\overline{\text{DTEN}}$ pin goes LOW. The host can read data from the transfer buffer while $\overline{\text{DTEN}}$ is low. See figure 16 *Initiating Data Transfer (1)*.

While the host is reading data from the transfer buffer, the $\overline{\text{ENABLE}}$ and $\overline{\text{HRD}}$ pins are held LOW, and the $\overline{\text{CMD}}$ and $\overline{\text{HWR}}$ pins, HIGH. When the host reads the last byte in the transfer buffer, the $\overline{\text{DTEN}}$ pin goes HIGH and the $\overline{\text{DTEN}}$ bit in the IFSTAT register resets to 1. See figure 17 *Initiating Data Transfer (2).*

12.3.3 Transfer start delays

If the controller instructs the LC8950 to transfer data while the $\overline{\text{DTWAI}}$ bit in the IFCTRL register is 1, the transfer takes place as described in section 12.3.2 above.

When $\overline{\text{DTWAI}}$ has been set to 0, the transfer hardware moves the first bit into the transfer buffer and checks the $\overline{\text{STEN}}$ bit in the IFSTAT register to see if a status-byte transfer operation is in progress. If $\overline{\text{STEN}}$ is 0 (that is, a status byte is still held in the SBOUT register and has not yet been transferred to the host), the $\overline{\text{DTEN}}$ pin is held HIGH and the $\overline{\text{DTEN}}$ bit at 1 until the status-byte transfer is completed. After the host has read the status-byte and $\overline{\text{STEN}}$ goes HIGH, the $\overline{\text{DTEN}}$ pin is set to LOW and the $\overline{\text{DTEN}}$ bit, to zero. See figure 18 *Transfer Start Delays.*

This mode is used to block data transfers when the $\overline{\text{DTEN}}$ and $\overline{\text{STEN}}$ pins cannot both be LOW at the same time, for example, when the host cannot handle data and status-byte transfers simultaneously.

12.3.4 Software transfer using WAIT control

The LC8950 enters software transfer mode when $\overline{\text{SELDRQ}}$ is HIGH.

In this mode, the LC8950 sends a WAIT signal by setting the $\overline{\text{WAIT}}$ pin to LOW during read operations by the host when the transfer buffer is empty, thereby controlling data transfers. The status of the $\overline{\text{WAIT}}$ pin is significant only while the $\overline{\text{DTEN}}$ pin is LOW and a host read is in progress; at all other times it is inactive (HIGH).

If the host reads data relatively slowly (that is, the interval between host read operations is more than seven XTALCK clock cycles), $\overline{\text{WAIT}}$ signals may not be generated. When the host read interval exceeds seven XTALCK clock cycles, the transfer buffer remains filled, masking the hardware operation that would set the $\overline{\text{WAIT}}$ pin to LOW. Note that this $\overline{\text{WAIT}}$ pin activation is a glitch, because both host and FIFO transfer buffer reads occur asynchronously. See figure 19 *Software Transfer using WAIT Control.*

12.3.5 DMA transfer using DRQ control

The LC8950 enters DMA transfer mode when $\overline{\text{SELDRQ}}$ is LOW.

In this mode, the LC8950 generates a HIGH-level DRQ signal to the $\overline{\text{WAIT}}$ pin. The DRQ signal is significant only while $\overline{\text{DTEN}}$ is LOW and the host has performed a data read. At other times $\overline{\text{WAIT}}$ is held LOW. A DMA controller in the host is assumed to output the $\overline{\text{HRD}}$ read signals when the DRQ signals are generated for each byte. Note that DRQ is activated after a delay time of two to four XTALCK periods after $\overline{\text{HRD}}$ rises. Figure 20 *DMA Transfer using DRQ Control* shows DMA transfer timing.

12.3.6 Data-transfer completion

The data transfer operation halts when the number of bytes specified by the controller has been transferred to the host.

When the host begins to read the final byte of a data transfer, the $\overline{\text{DTEN}}$ pin goes HIGH. During this final read operation, the $\overline{\text{EOP}}$ (End-of-Process) pin remains LOW while the $\overline{\text{HRD}}$ pin is LOW. $\overline{\text{EOP}}$ can be used to generate the interrupt to the host CPU or DMA controller signaling the completion of the data transfer. The rising edge of the signal at $\overline{\text{HRD}}$ sets the $\overline{\text{DTEN}}$ bit to 1, and a data-transfer end interrupt is generated. If the mask bit DTEIEN is 1 (that is, masking is not set), the $\overline{\text{INT}}$ pin goes LOW. $\overline{\text{DTEI}}$ in the IFSTAT register is cleared to 1 and the $\overline{\text{INT}}$ pin goes HIGH by writing to the DTACK register. See figure 21 *Data Transfer Completion.*

12.3.7 Data-transfer Abort

Three methods are available to abort data transfers:

1. Reset.

   Forcing the $\overline{\text{RESET}}$ pin to LOW or writing to the RESET register will reset the LC8950, terminating data transfers. See figure 22 *Data Transfer Abort (1 & 2).*

2. Setting the DOUTEN bit to 0.

   Setting the DOUTEN bit in the IFCTRL register to 0 aborts data transfers. See figure 22 *Data Transfer Abort (3).*

3. Setting the command break.

   Setting the $\overline{\text{CMDBK}}$ bit in the IFCTRL register to 0 allows data transfers to be aborted by sending a command from the host. See figure 22 *Data Transfer Abort (4).*

The data transfer end interrupt is not generated when any of these methods has been used to abort a data transfer.

12.4 Reset

The LC8950 can be reset by setting the $\overline{\text{RESET}}$ pin to LOW, or by writing to the RESET register.

During a reset, the following operations take place.

- All tristate outputs enter high-impedance state.
- The COMIN register is cleared.
- All bits in the IFCTRL, CTRL0 and CTRL1 registers are set to 0.
- The MCK clock output is suppressed.

12.5 Decoder Operation Modes

The decoder block of the LC8950 writes the input data stream from the CD equipment's digital signal processor to the data buffer RAM and decodes data blocks. In this context, decoding refers to error detection and/or correction.

Setting the decoder enable bit DECEN to 1 enables the decoder interrupt and decoder operation. The decoder functional block generates the decoder interrupt for the controller each time it completes error detection and correction of a data block. When DECEN and DECIEN are both 1, decoder interrupts are generated even during correction inhibit (that is, when QRQ, PRQ and EDCRQ set to 1), or during input inhibit (when WRRQ set to 0).

The transparent pipeline processing architecture of the LC8950 allows input buffering, decoder operation and data transfers to the host to take place simultaneously. The following table shows the basic decoder modes that can be set by the DECEN, QRQ, PRQ, EDCRQ and WRRQ control bits. Other modes can be set by combining these bits.

TABLE 9. Decoder Modes

Control Bits					Mode Selected
DECEN	WRRQ	PRQ	QRQ	EDCRQ*	
0	X	X	X	X	Decoder-disable
1	0	1	1	1	Repeated-correction
1	0	0	0	0	Monitor-only
1	1	0	0	0	Write-only
1	1	1	1	1	Real-time correction
1	0/1	1	0	1	P-only correction
1	0/1	0	1	1	Q-only correction

Key

0/1: 1 = real-time correction
0 = non-real time correction

X: Not significant

12.5.1 Decoder-disable mode

In this mode, the decoder is disabled and does not generate decoder interrupts. Decoder-disable mode is invoked by setting DECEN to 0.

12.5.2 Repeated-correction mode

CD-ROMs use doubly encoded Reed-Solomon code. The two corrections (P and Q) can be performed alternately to raise the level of data integrity.

While the WRRQ flag is set to 0, the PTL and PTH block pointers pointing to the first address of the current block are not updated, even although the correction pass has been completed. This allows successive corrections to be

---

* See appendix E for the LC8951.

applied to the current block only in order to obtain highly reliable data. However, in the case of very high error rates, correction of every error cannot be guaranteed by this process on every occasion. The number of retries allowed by the controller and should be monitored by counting decoder interrupts and should be limited.

Note that entering repeated-correction mode in order to set WRRQ to 0 should be carried out during the decoder-interrupt interval, that is, when $\overline{\text{VALST}}$ is at 0. When $\overline{\text{VALST}}$ goes to 1 while WRRQ is 1, the block pointer is updated, and decoding of the next block begins.

12.5.3 Monitor-only mode

In this mode, the block header and subheader data from the input data stream are loaded into the HEAD0 to HEAD3 registers, and the corresponding error flags (read at pin C2PO) are latched in the STAT1 register. The data blocks themselves, however, are not stored in the buffer RAM and are not subjected to error detection or correction processing.

Since error correction is not performed, the header registers hold the unaltered and uncorrected data from the CD digital signal processor.

Another significant difference between monitor-only mode and the modes available when WRRQ is 1 (excluding decoder-disable mode) is the time delay in updating the header and subheader register data.

Under normal buffered operation, the LC8950's pipeline architecture writes the header and subheader to the appropriate registers (HEAD0 to HEAD3 and STAT1) after the entire block has been written into the buffer RAM and has undergone error detection and correction processing. The register data for a given block is available only after the input time for one block has passed.

In monitor-only mode, however, register data are written directly from the input data stream. Register data are therefore available one block earlier than in the other operating modes.

12.5.4 Write-only mode

In this mode, errors in the input data are neither detected nor corrected, but simply written to the external buffer RAM.

Errors can not be detected and corrected unless there is a full block of data in the buffer RAM. Thus before starting decoding, the decoder should enter write-only mode for at least a one-block period until a decoder interrupt is generated.

Otherwise, when using correction mode only, the first two decoded blocks should be ignored as the first of these is not the input block and the second may be unreliable as result of correction errors in decoding the first block.

Write-only mode can also be used when error detection and correction is to be performed by the host computer.

12.5.5 Real-time correction mode

This is most commonly used decoder mode. Errors in the CD data written to the LC8950 are detected and corrected in real-time. Operations are performed on each block in the order QDL, QDH, PDL, PDH, QEL, QEH, PEL, PEH and EDC, where these codes represent combinations of the operations in table 10.

TABLE 10. Real-Time Correction Mode Codes

Code	Operation
Q	Q-code
P	P-code
D	Detection correction (one symbol correction per word)
E	Erasure correction (two symbol corrections per word)
L	LSB-byte plane
H	MSB-byte plane
EDC	CRC check

12.6 Decoder Control

The decoder searches for a sync pattern in each block of data sent from the CD equipment and uses a sync signal generated from this pattern to time its operation.

Decoder interrupts are generated for the controller after the decoding of each block. Interrupts are enabled by writing a 1 to the DECIEN bit in the IFCTRL register. This sets the $\overline{INT}$ pin to LOW and the $\overline{DECI}$ flag in the IFSTAT register to 0 when requesting interrupt processing to the controller after decoding as long as DECEN (Bit 7 in CTRL 0) is set to 1. See figure 11 *Interrupt Control Flags*.

The duration of the active LOW at the $\overline{INT}$ pin depends on the operation mode in effect. See table 11.

TABLE 11. Period of Active LOW at $\overline{INT}$

DECEN	EDCRQ	MODE	FORM	Active LOW (≃μs)	Interrupt Cycle (blocks)
0	X	X	X	0	∞
1	0	X	X	2750	1
1	1	0	X	437*	1
1	1	1	0	428*	1
1	1	1	1	114*	
	1				

Decoder interrupts become inactive immediately after the STAT3 register is read, or when DECEN is set to 0.

The following read registers and flags related to the decoder should be read out whenever a decoder interrupt is generated.

HEAD0, HEAD1, HEAD2, HEAD3
PTL, PTH
STAT0, STAT1, STAT2, STAT3

Note that when WRRQ in the CTRL0 register is 1, the contents of the WAL and WAH write address registers constantly changes. Accordingly, WRRQ should be set to 0 before reading is attempted.

The controller should read all decoder-related registers while $\overline{VALST}$ in the STAT3 register is 0. Data reads, while $\overline{VALST}$ is 1, are not guaranteed to be accurate.

$\overline{VALST}$ goes to 0 immediately after a decoder interrupt is generated, but differs from the interrupt ($\overline{INT}$ pin) in that reading STAT3 does not cause it to reset to 1.

* See appendix E for the LC8951.

12.7 Block Synchronization

The LC8950 synchronizes its operations to the blocks in the data stream received from the digital signal processor of the CD equipment by generating a block-sync signal whenever it detects the sync pattern at the head of each data block.

Since synchronized decoding is vital for the correct recovery of CD-ROM and CD-I data, the LC8950 has on-chip block-sync signal circuitry to handle sync-signal pattern errors caused by disc flaws, scratches, or other media defects.

It generates an internal block-sync signal each time that a 2352-byte block of data is received. Thus, if the sync signal derived from the incoming data is lost, the internal block-sync signal can be used to maintain correct data synchronization.

Enable/disable flags — SYDEN and SYIEN — can be set for both the internal- and external-sync signals respectively. In combination these flags allow selection of three synchronization modes. SYDEN and SYIEN should not both be set to 0 as this combination will inhibit both sync signals, and risk complete loss of synchronization.

The LC8950 is generally operated in real-time decoding mode with both SYDEN and SYIEN set to 1. This provides the most reliable synchronization scheme and allows rapid recovery of synchronization, even when detracking or sync pattern errors occur.

The following four synchronization modes are supported.

- No sync mode (SYDEN=SYIEN=0)
  - — No decoding
  - — Rough seeks in subcode Q state
  - — In-track jumps for seek adjustment in order to pick up the destination sector
- Sync hunt mode (SYDEN=1, SYIEN=0)
  - — Enters on-track play state and waits for real synchronization after seeks or track jumps
- Protected sync mode (SYDEN=SYIEN=1)
  - — All decoder modes except repeated-correction mode
- Inner sync mode (SYDEN=0, SYIEN=1)
  - — Repeated-correction mode
  - — PQ error-correction strategy (Q-only correction precedes P-only correction)

In repeated-correction mode only, the internal block-sync signal should be used; that is, SYIEN should be set to 1 and SYDEN to 0. This setting completely ignores the external-sync signal, since any loss of the external sync would interfere with error correction.

To assist in handling sync errors, the LC8950 maintains NOSYNC, ILSYNC, LBLK and SBLK flags in the STAT0 register. The meaning of each flag is described below. See also figure 23 *Block Synchronization (1)* to *(3)*.

12.7.1 NOSYNC — No sync

This flag indicates that an external-sync signal was not detected and that the internal block-sync signal is being used. When this flag is set, it generally means that noise or media flaws have caused errors in the sync pattern that prevent correct recognition.

If the NOSYNC flag is set each time a decoder interrupt occurs, it signifies a malfunction related to a sync-detected SYDEN flag. A SYDEN flag setting of 0 means synchronization between the internal sync signal and incoming data has been lost. A setting of 1 means that no CD-ROM or CD-I data are present.

12.7.2 ILSYNC — Illegal sync

This flag indicates that a sync pattern which was out of phase with the internal block-sync signal has been detected. This flag is significant only when SYDEN is set to 1. In other sync modes, ILSYNC is 0.

When ILSYNC is set to 1, the LC8950 resets its internal block-sync signal to the new timing, truncates the current data block and suppresses error detection and correction for the current block because there are insufficient data to decode the block completely. See figure 23 *Block Synchronization (1)*.

12.7.3 LBLK — Long block

This flag is set while SYDEN is 1 and SYIEN is 0 when a sync signal has not been detected after 2352 bytes of the current block have been received.

When LBLK is set, only the 2325 bytes of data from the block head (the first byte of the header) are used by the LC8950. Remaining data are discarded, but excess data are written to the buffer.

The extra data could be transferred to the host computer, but this transfer is rarely implemented. Also, the excess data in the buffer are kept separate from, and has no effect on the decoding of the next block.

A long block can be caused either by an error in the sync pattern which prevents sync detection, or by detracking during a block. In the former case, the block data will be accurate; in the latter, erroneous.

Setting SYIEN to 1 (that is, enabling the internal block-sync signal) will prevent LBLK from being set after the next sync pattern is detected. See figure 23 *Block Synchronization (2)*.

12.7.4 SBLK — Short block

This flag indicates that a sync pattern detected in the middle of a block was ignored. It is enabled while SYDEN is 0 and SYIEN is 1.

In other sync modes, it remains 0.

If this flag appears at every block, it indicates a complete loss of synchronization. See figure 23 *Block Synchronization (3)*.

12.8 Buffer Operation

The LC8950 RCHIP uses an external buffer RAM to temporarily hold the incoming data received from the digital signal processor of the CD equipment. Writes to this buffer are always performed in bytes.

As the LC8950 has a transparent bus architecture, data input, the read and write operations involved in data decoding (error detection and correction) and data transfers to the host can take place simultaneously without wait states.

Together the buffering and decoding operations create a time lag of 1/75 seconds (corresponding to the timing for a one block transfer from the point when a data block from the CD equipment enters the LC8950's processing pipeline to the point when it exits the pipeline and is transferred to the host).

This real-time decoding delay is small enough to meet the CD-I specifications in the *Green Book*.

12.9 Caching

The external RAM buffer has a minimum size of 8 Kbytes (64 Kbytes), which accommodates three 2352-byte blocks with a 1136-byte margin.

When a single block caching (8 Kbyte) buffer is used, one block is allocated to hold the incoming data stream, one block is used to hold the sector being decoded (error detection and correction), and one block is allocated to hold the sector being transferred to the host. See figure 24 *Single Block Caching (Decoder-Interrupt Control)*. The processing of these three blocks takes place simultaneously and transparently. Note, however, that the host must read out one block of data during each one-block processing period in order to prevent buffer overflow.

A larger buffer can hold additional blocks and data transfers to the host can be delayed in order to service interrupts.

Figure 25 *An Example of Block Caching in Real-Time Correction Mode 1* shows the flowchart of the error correction process. In real-time correction mode, one block of storage is allocated to the incoming data stream, one block for decoding, and four blocks to data transfer. The four-block data transfer area is occupied by untransmitted blocks, the block currently being transferred, and already transferred blocks. This configuration, known as four-block caching, allows the LC8950 to transfer four blocks of data to the host at any time during a four-block period, reducing overhead.

The buffer should be checked for an overflow condition, either when each block is decoded (decoder interrupt-controlled caching), or when the transfer of each block to the host is completed (data transfer complete-controlled caching).

When an 8 Kbyte buffer is used, one block of data must be transferred each time a decoder interrupt is received. If a larger buffer is employed with multi-block caching, checks should be performed to ensure that the total number of blocks being transmitted and awaiting transmission does not exceed the specified limit.

Even when a one-block overflow condition is detected at the time of a decoder interrupt, the block being transferred may not yet have been overwritten at the time the decoder interrupt was generated, and the block of data already transmitted may be correct. Figure 24 *Single Block Caching* and figure 25 *An Example of Block Caching in Real-Time Correction Mode 1* show at which point during an overflow condition the block currently being transferred is actually overwritten.

Note that the controller is unable to read the write address registers (WAH and WAL) correctly when buffering data while WRRQ is 1. This because the data in WAH and WAL may be in transition (for example, being updated) when the read is attempted.

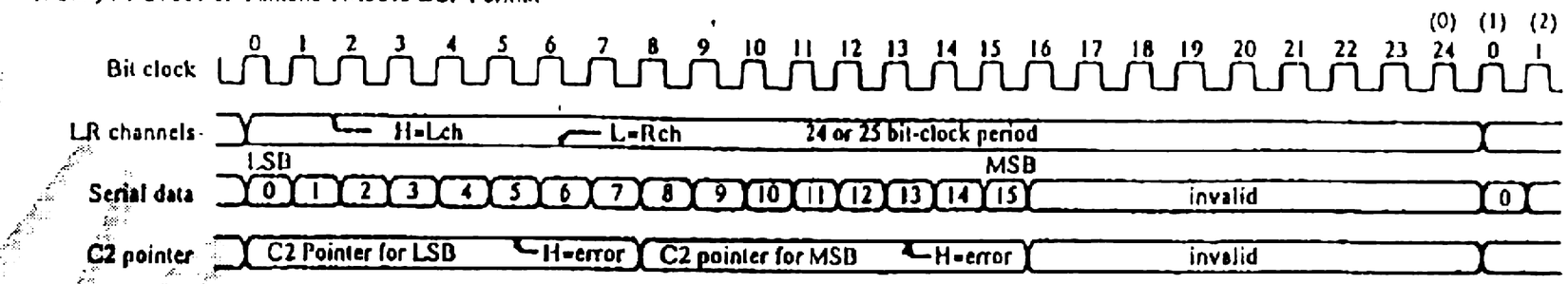
When overflow checking is performed after the transfer-complete interrupt is received, the controller in a system configured with a one-block caching buffer should check after each interrupt to confirm that the decoder has not yet completed its error detection and correction of the next block. This limit is managed by software that counts the number of blocks to be held in the buffer RAM. See figure 26 *Error Checking in Full Sync Mode* and figure 27 *Repeated Error Correction in Mode 1*.

When a multi-block caching buffer is used, the controller should check after each block transferred that the number of error-corrected blocks awaiting transmission does not exceed the specified limit.

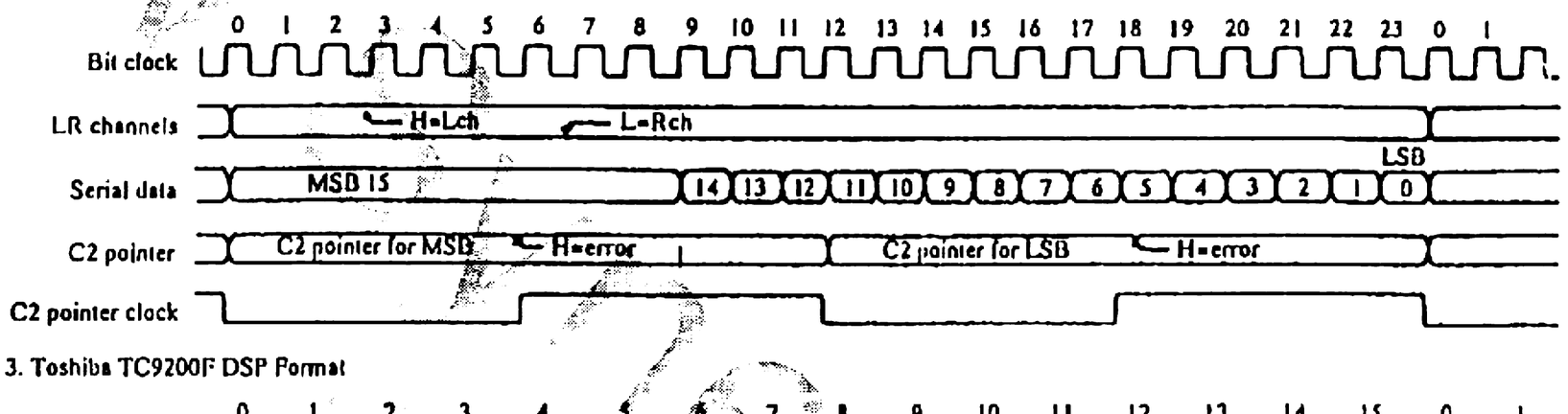
When the latter method—data transfer complete-controlled caching—is used, one block of data is sent whether an overflow condition occurs or not. The status signaling the overflow occurs and is sent to the host using status-byte processing. The latter part of the block previously transferred to the host is overwritten. The host should discard the entire block already received and, using the LC8950 command registers, send a repeat request to the controller by COMIN, the chip's command register.

APPENDIX A: TIMING DIAGRAMS

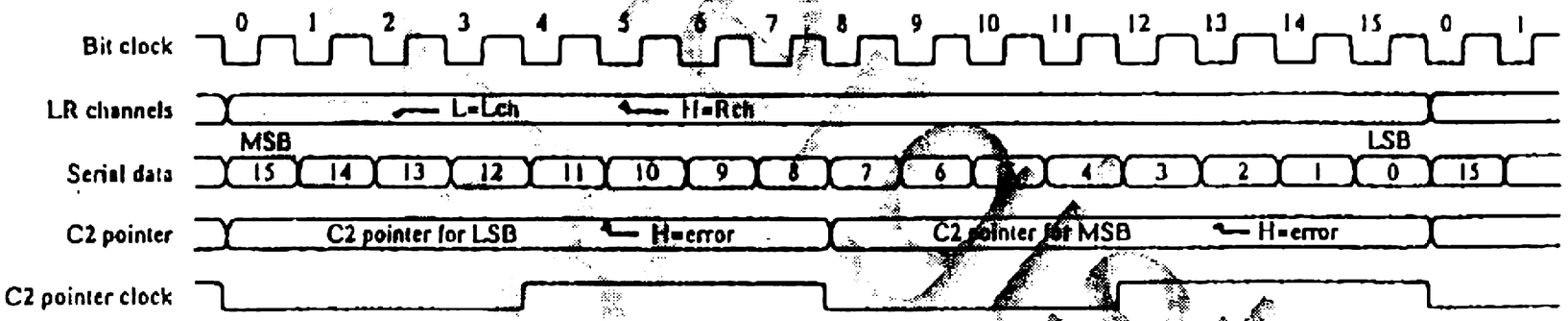
1. Sanyo LC7860 & Yamaha YM3815 DSP Format



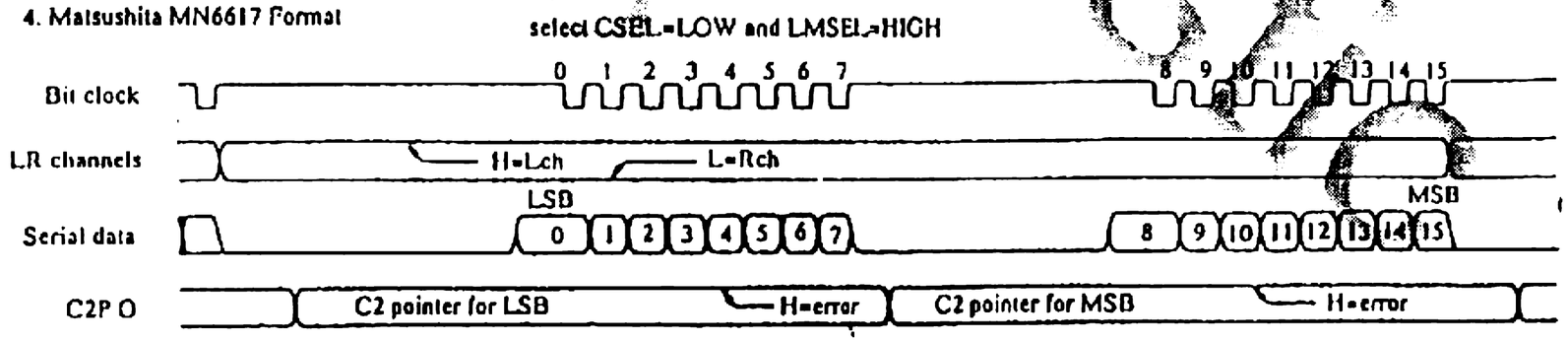
2. Sony CXD1125 DSP Format



3. Toshiba TC9200F DSP Format



4. Matsushita MN6617 Format



Note Connection pins between MN6617 and RCHIP are as follows.
BCK: inverted SRCK, LRCK: inverted L/R, SDATA: SRDATX and C2P O: IPBYTE.
The internal clock replaces C4LR (C2 pointer clock).

Figure 10. Four Data Input Modes

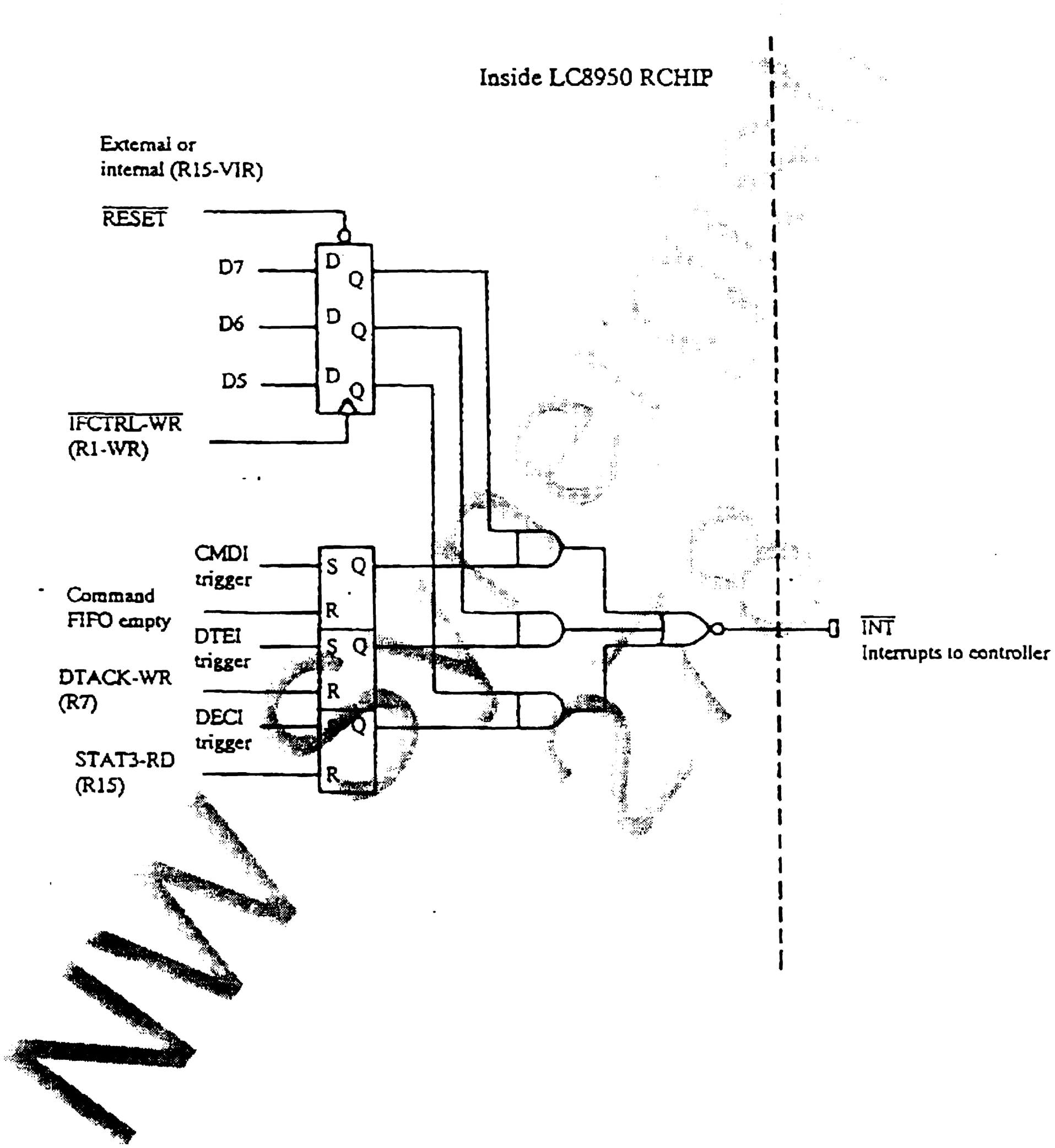


Figure 11. Interrupt Control Flags

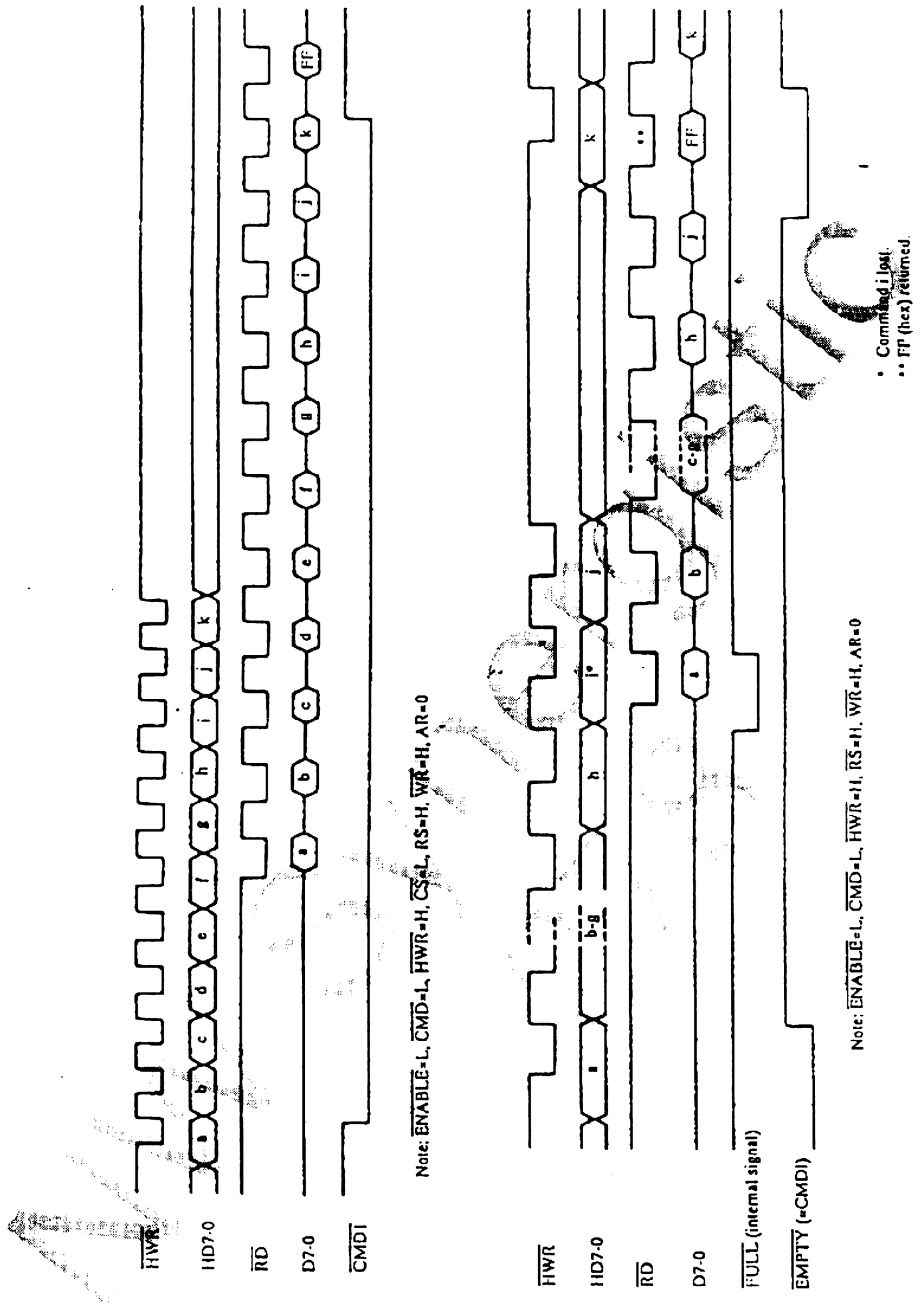


Figure 12. Command Processing (1 & 2)

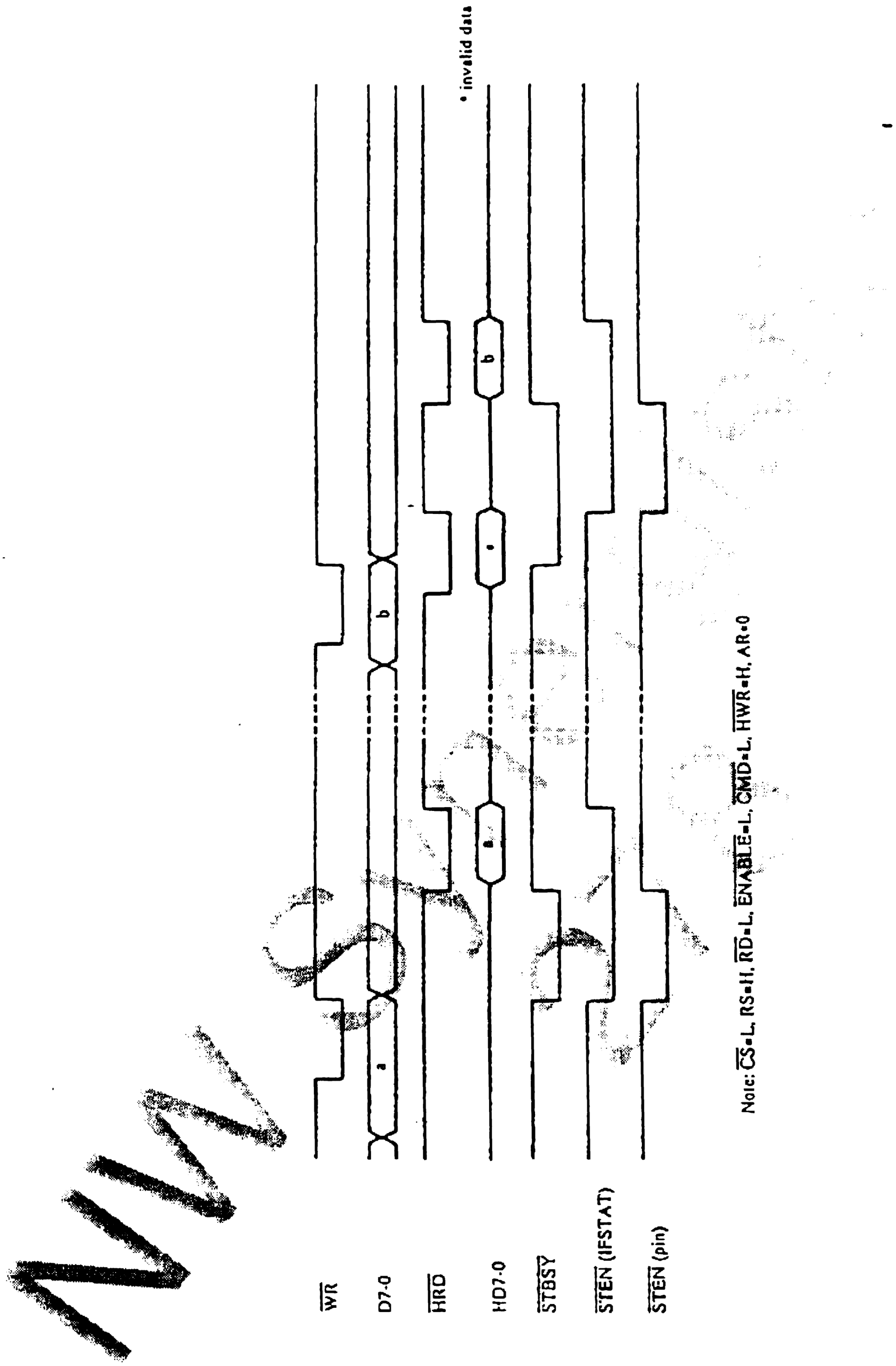


Figure 13. Status-Byte Processing during Normal Operation

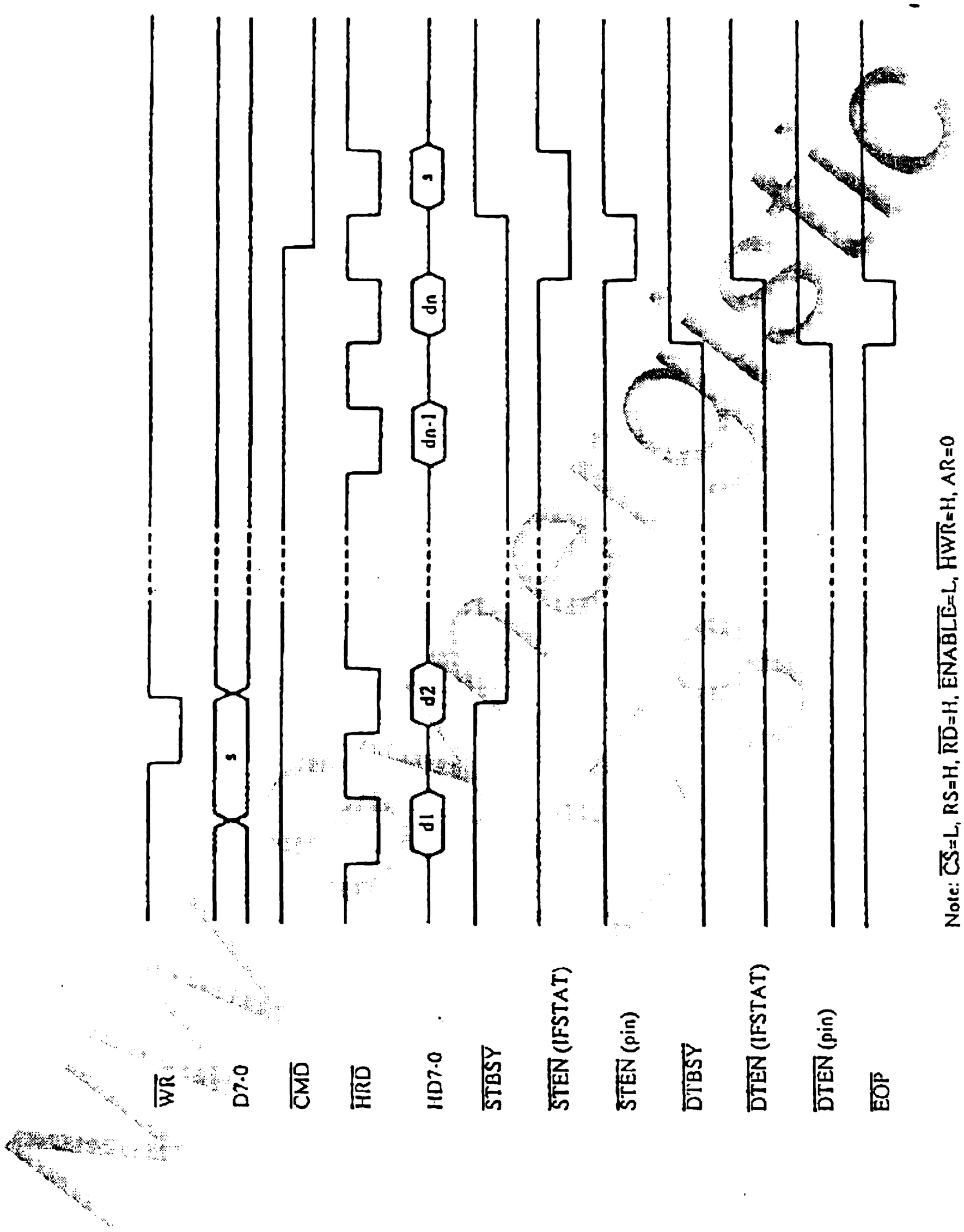


Figure 14. Status-Byte Processing when WAIT is Enabled

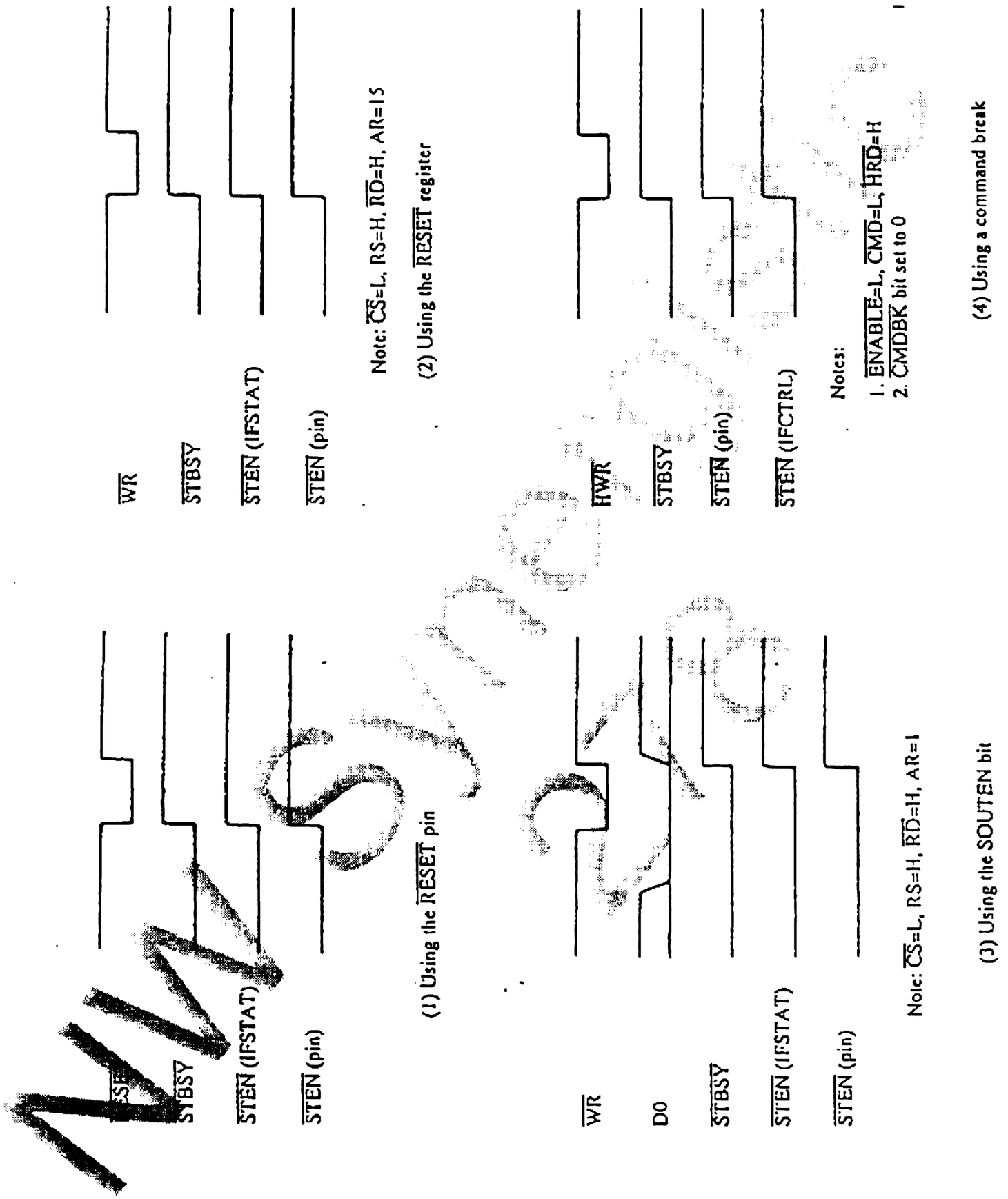
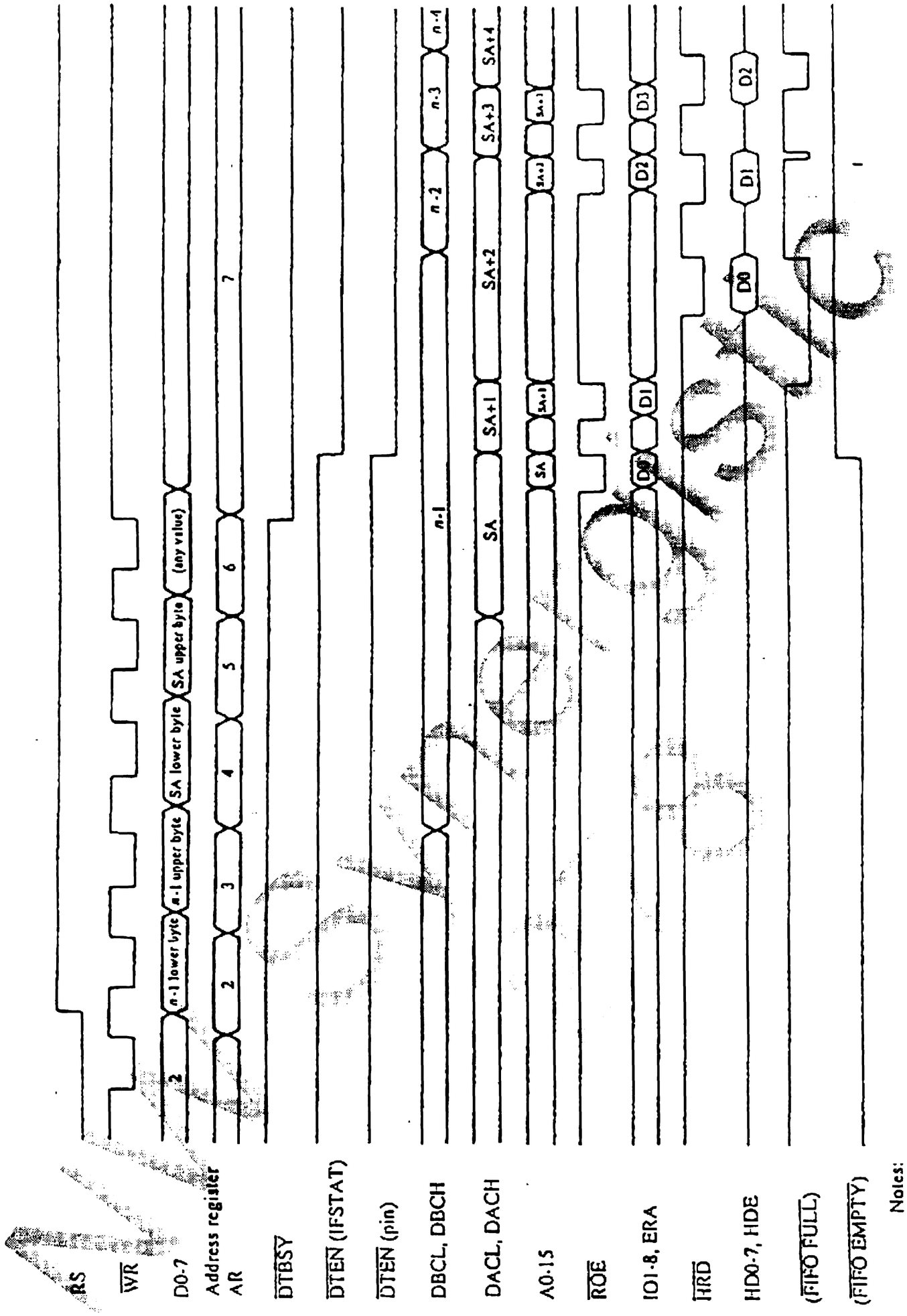


Figure 15. Status Byte Transfer Abort (1 to 4)



Notes:

1. $\overline{CS}$=L, $\overline{RD}$=H, $\overline{ENABLE}$=L, $\overline{CMD}$=H, $\overline{HWR}$=H
2. *n* bytes transferred from SA

Figure 16. Initiating Data Transfer (1)

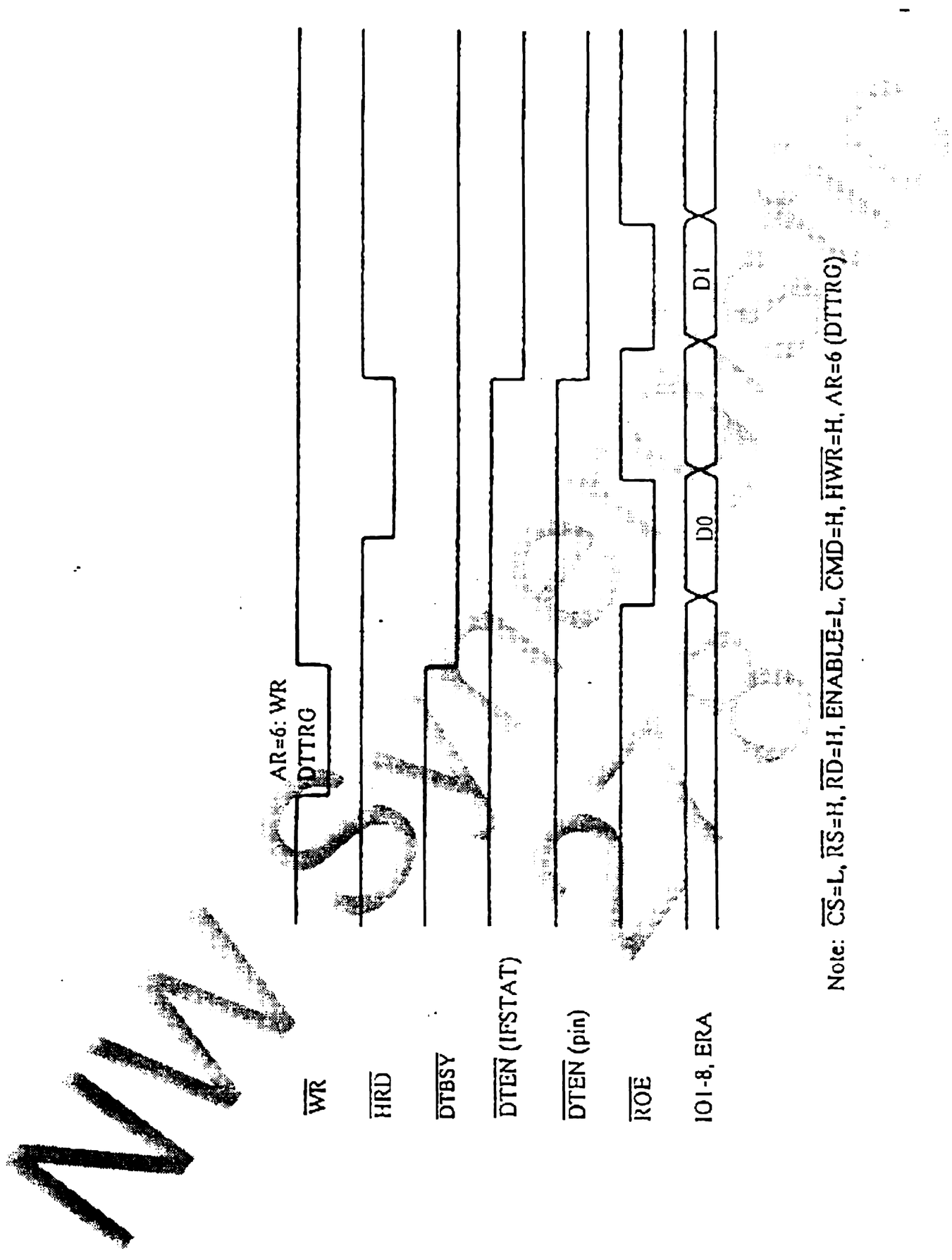


Figure 17. Initiating Data Transfer (2)

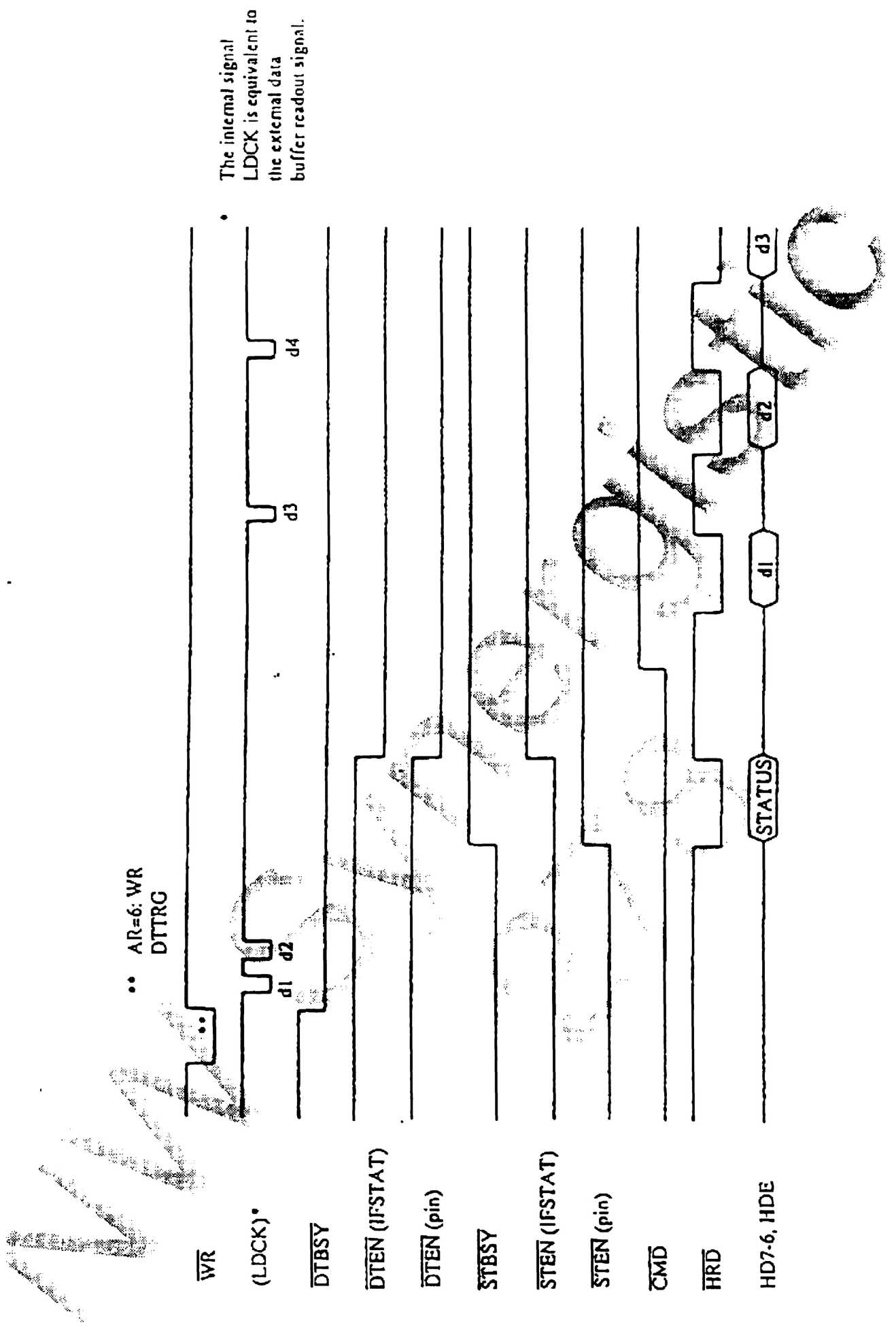
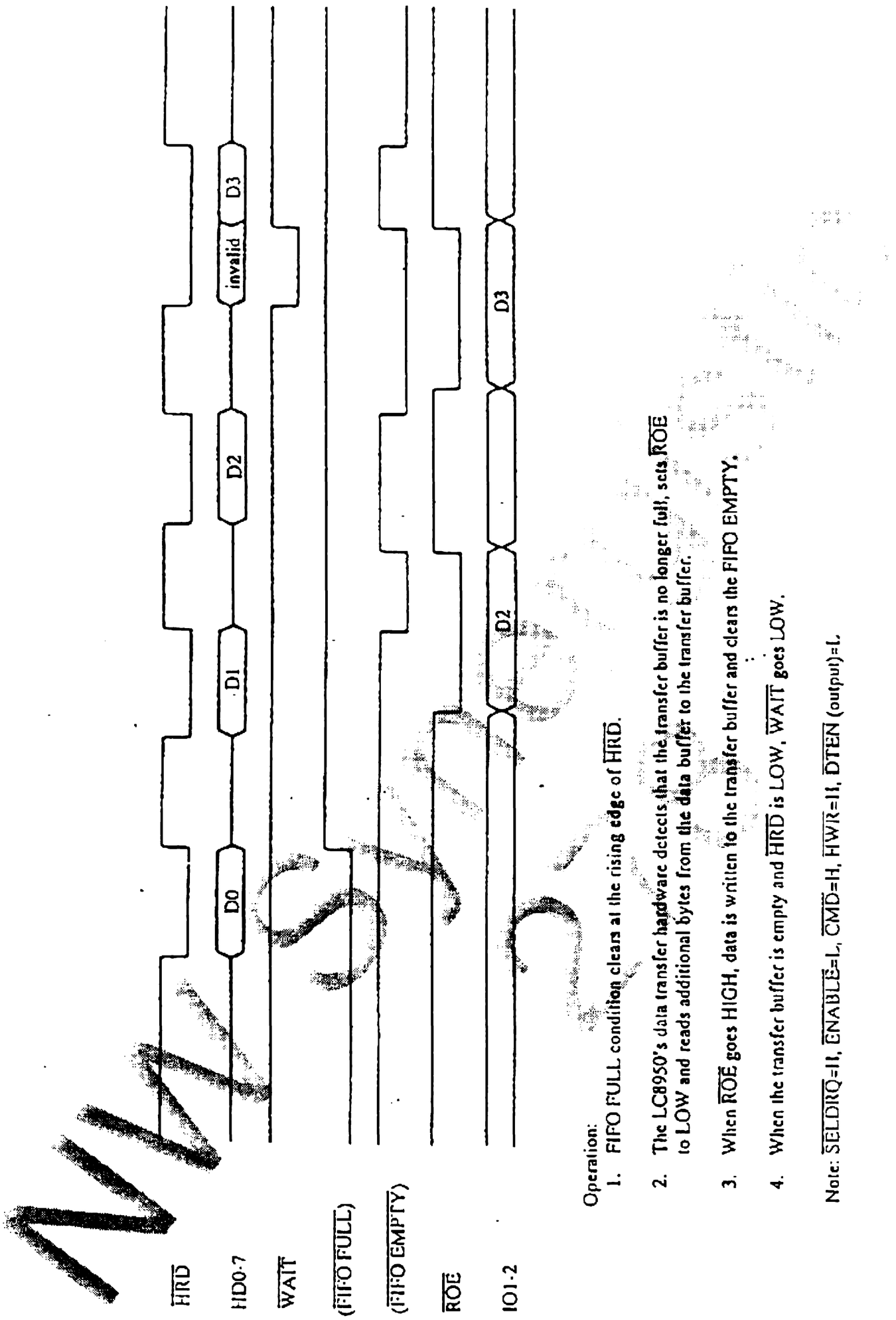


Figure 18. Transfer Start Delays

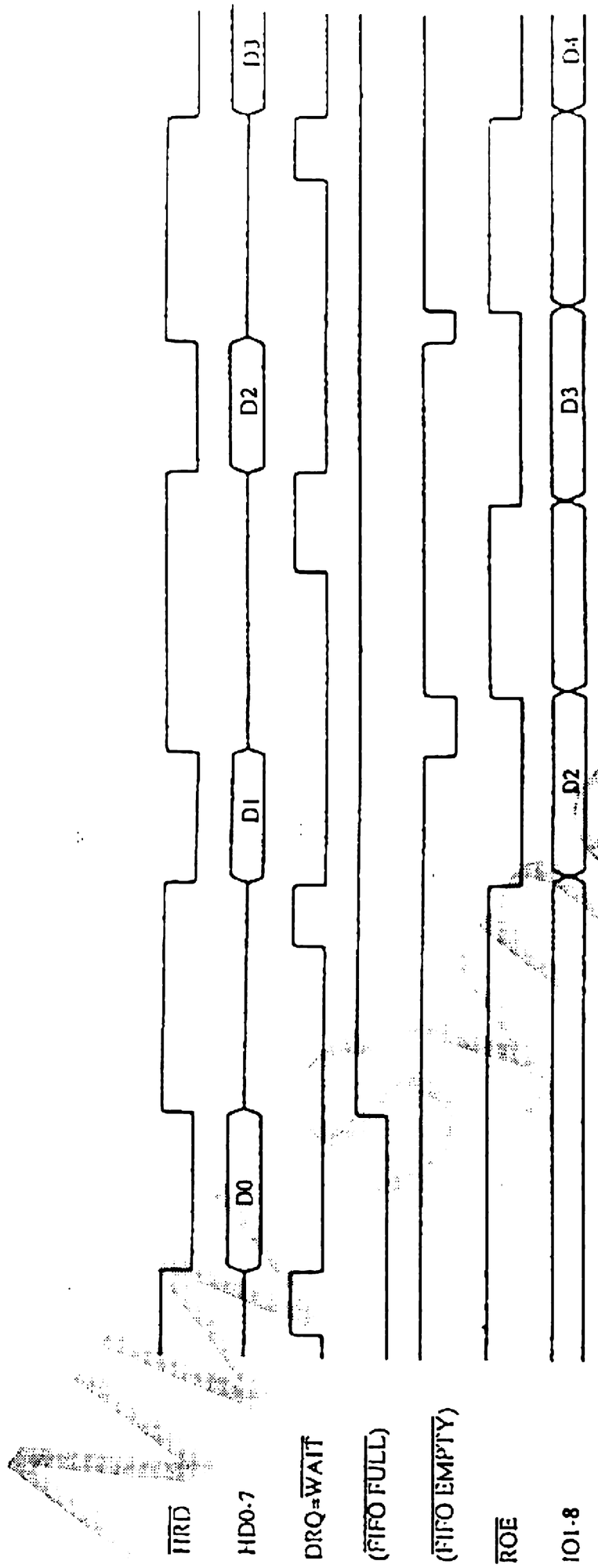


Operation:

1. FIFO FULL condition clears at the rising edge of $\overline{HRD}$.
2. The LC8950's data transfer hardware detects that the transfer buffer is no longer full, sets $\overline{ROE}$ to LOW and reads additional bytes from the data buffer to the transfer buffer.
3. When $\overline{ROE}$ goes HIGH, data is written to the transfer buffer and clears the FIFO EMPTY.
4. When the transfer buffer is empty and $\overline{HRD}$ is LOW, $\overline{WAIT}$ goes LOW.

Note: $\overline{SELDRQ}$=H, $\overline{ENABLE}$=L, $\overline{CMD}$=H, $\overline{HWR}$=H, $\overline{DTEN}$ (output)=L.

Figure 19. Software Transfer using WAIT Control



Operation:

1. DRQ goes HIGH at the falling egde of DTEN.
2. The host sets HRD to LOW.
3. DRQ goes LOW.
4. The host sets HRD to HIGH.
5. When the LC8950 detects that HRD is HIGH, DRQ also goes HIGH.

   However, if the transfer buffer is EMPTY, DRQ does not go HIGH. Instead the LC8950 data transfer hardware reads additional bytes into the transfer buffer. When ROE goes HIGH, DRQ also goes HIGH.

Notes:

1. SELDRQ=L, ENABLE=L, CMD=L, HWR=H, DTEN (output)=L
2. In this mode, the WAIT pin functions as DRQ.

Figure 20. DMA Transfer using DRQ Control

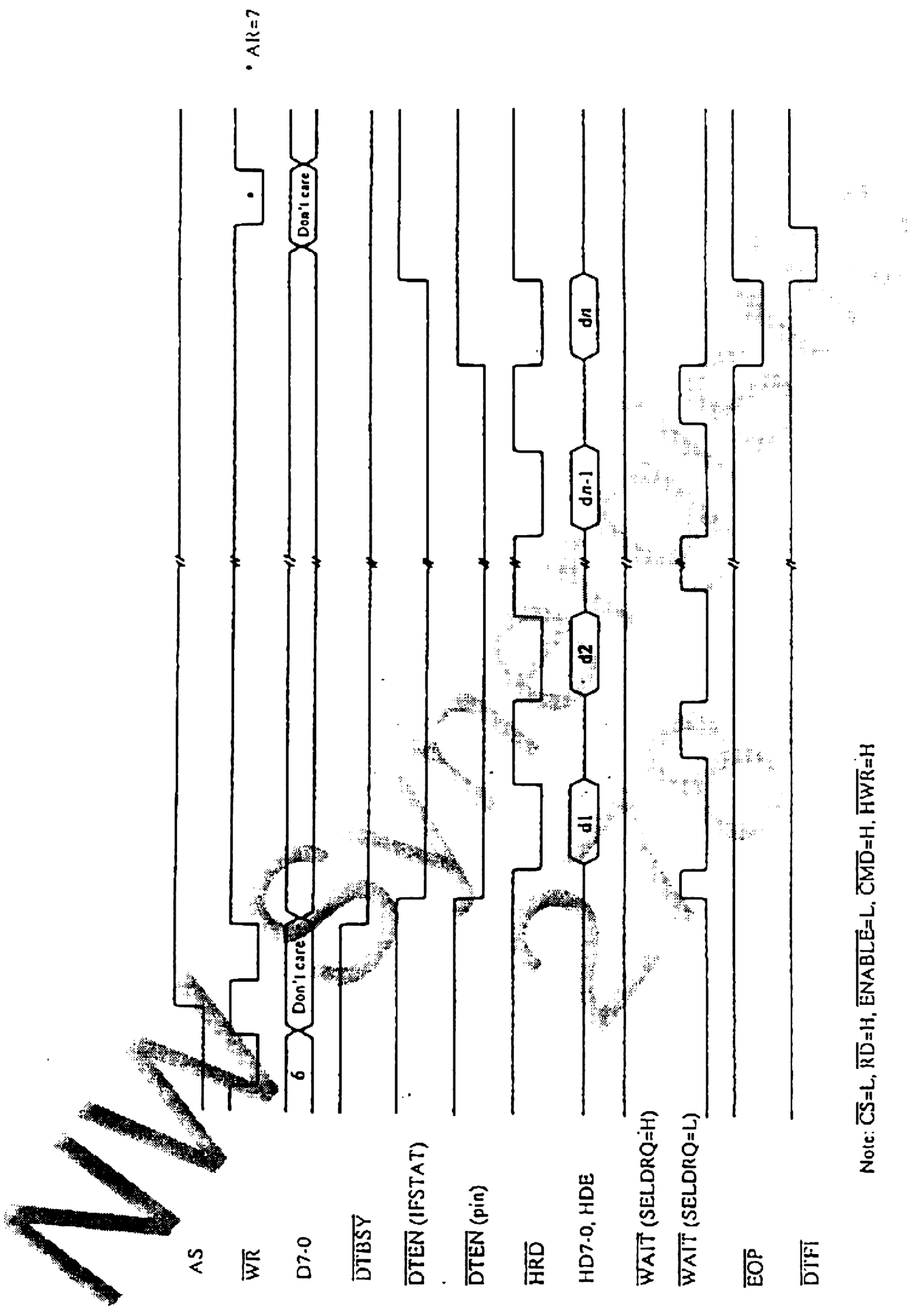


Figure 21. Data Transfer Completion

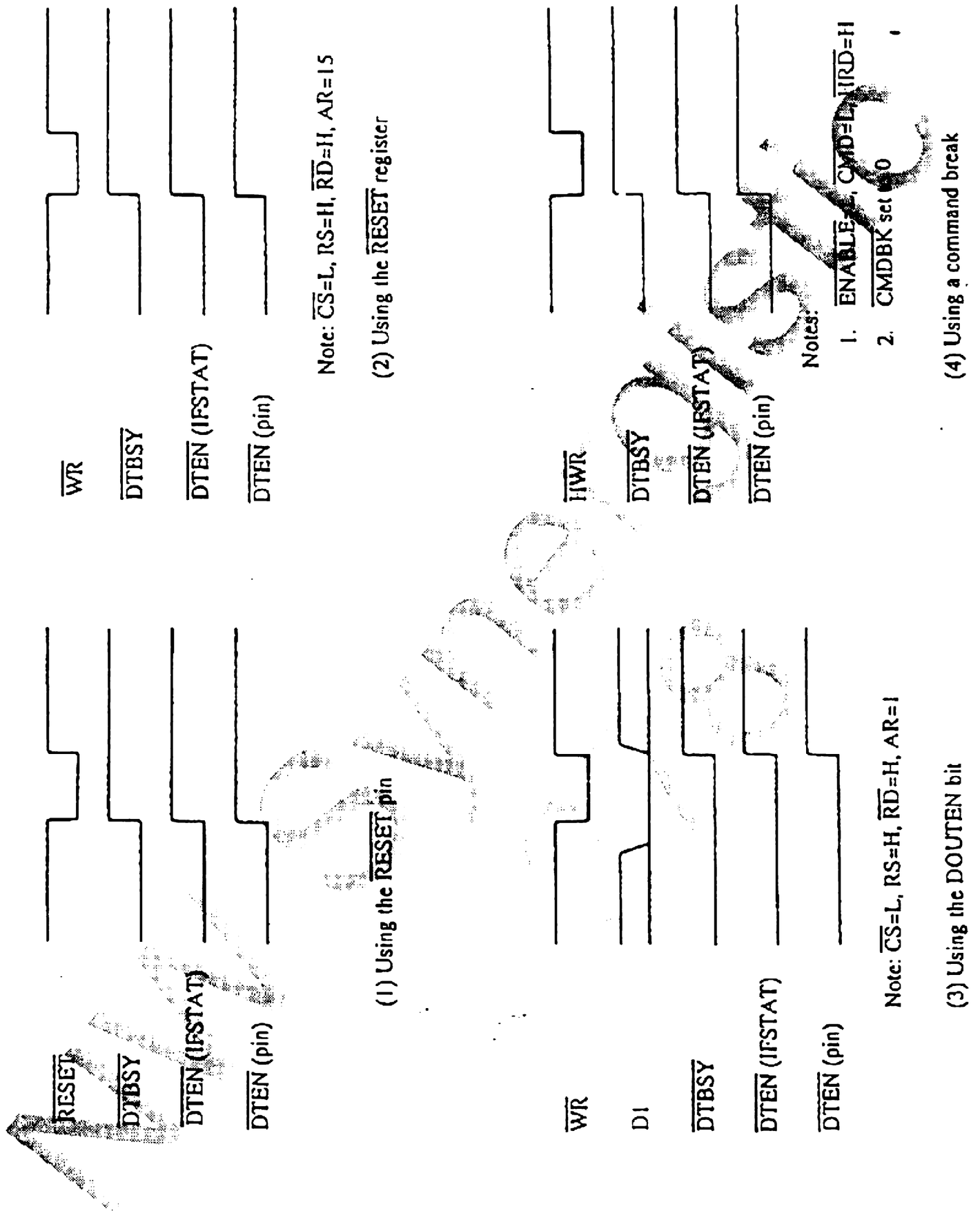


Figure 22. Data Transfer Abort (1 to 4)

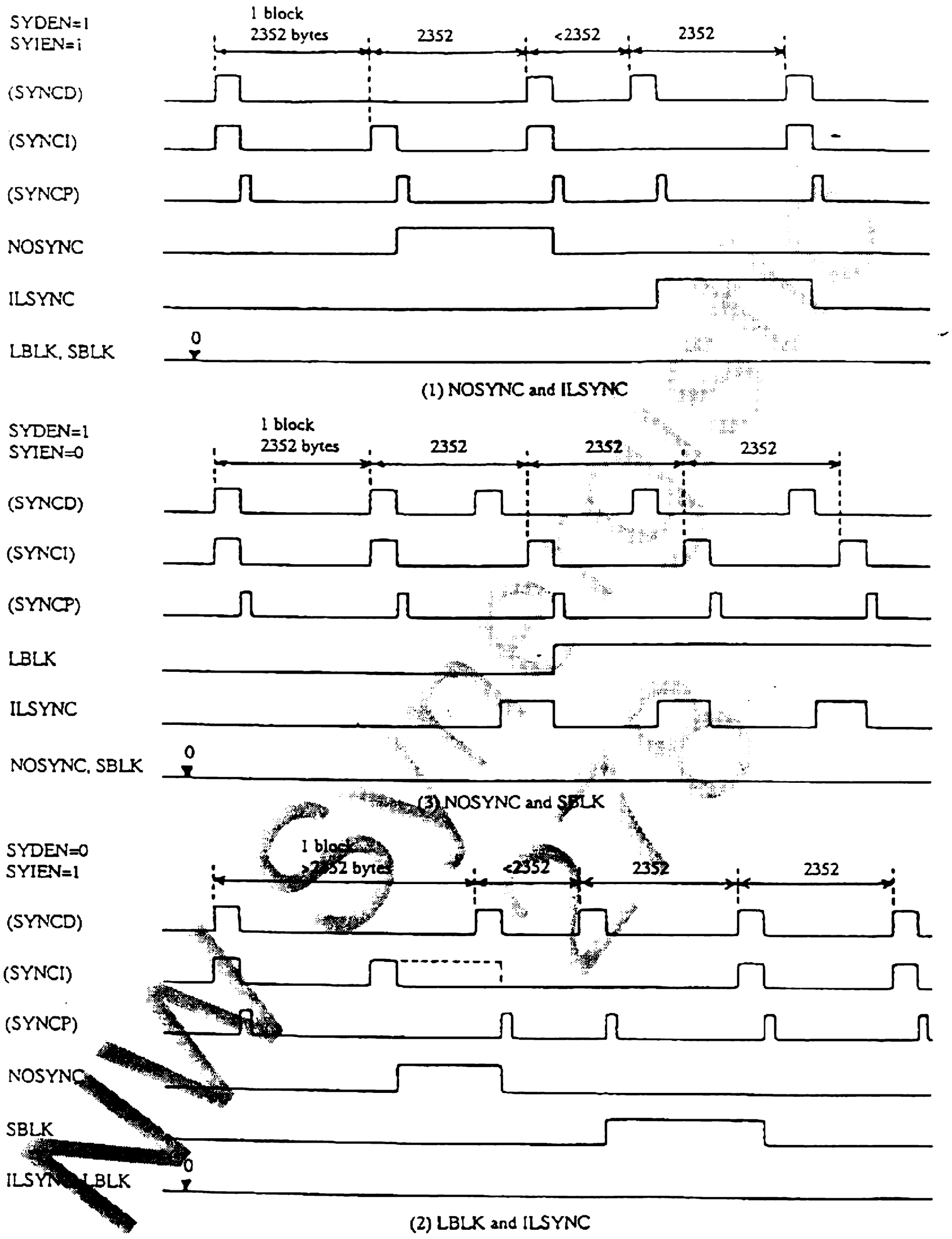


Figure 23. Block Synchronization (1 to 3)

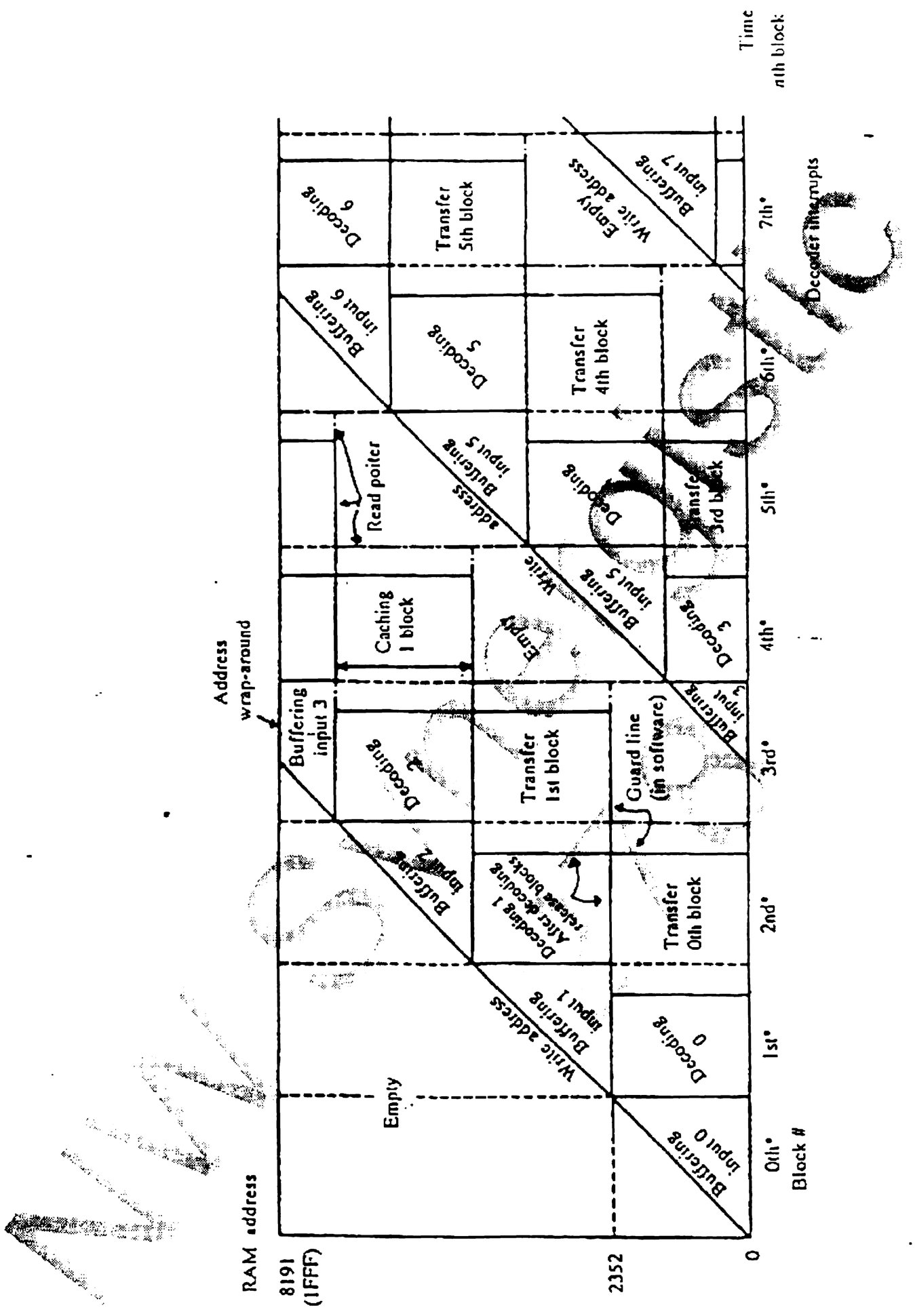


Figure 24. Single-Block Caching (Decoder Interrupt Control)

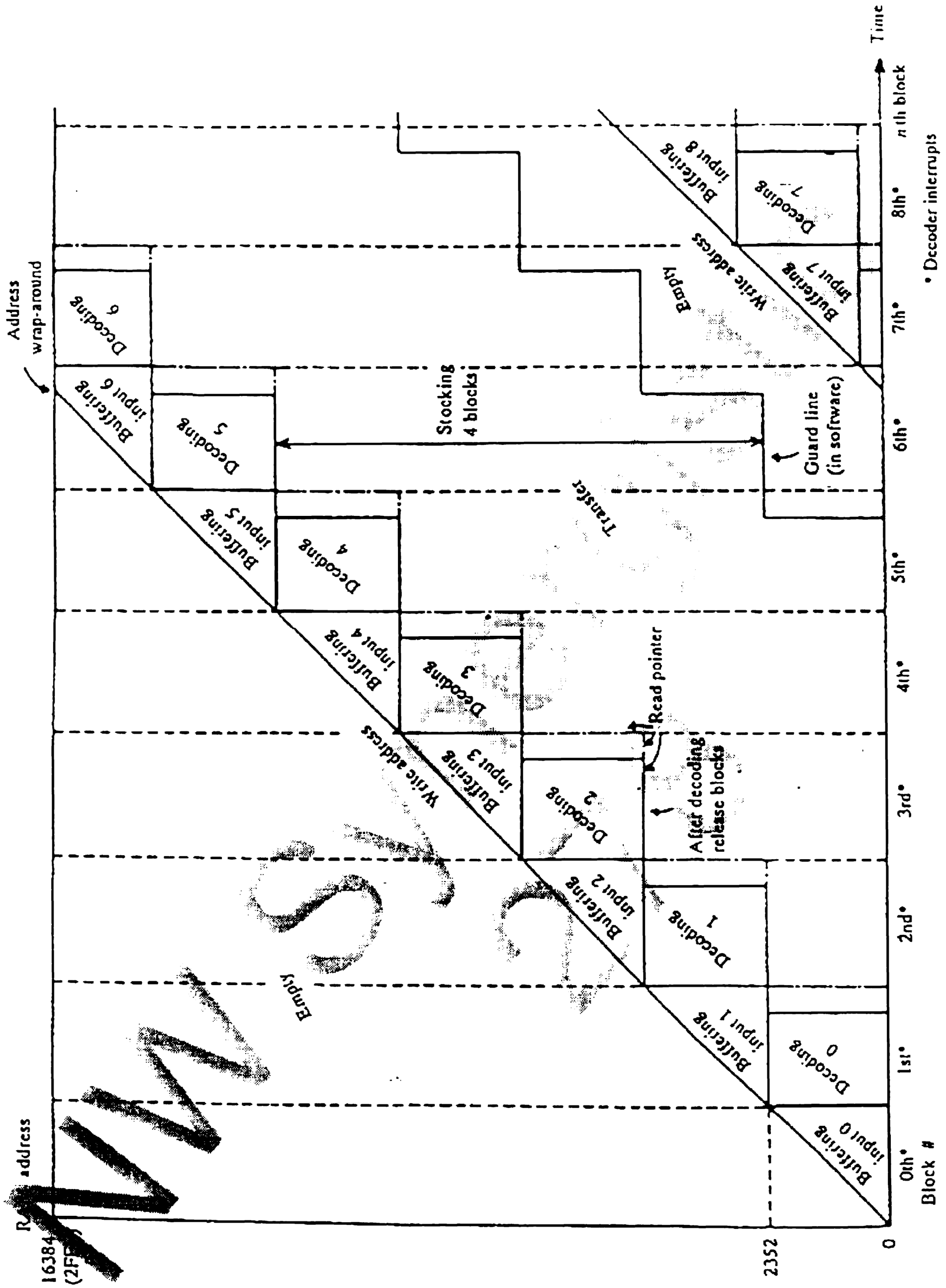


Figure 25. An Example of Block Caching in Real-Time Correction Mode 1 (Block-by-Block Transfer)

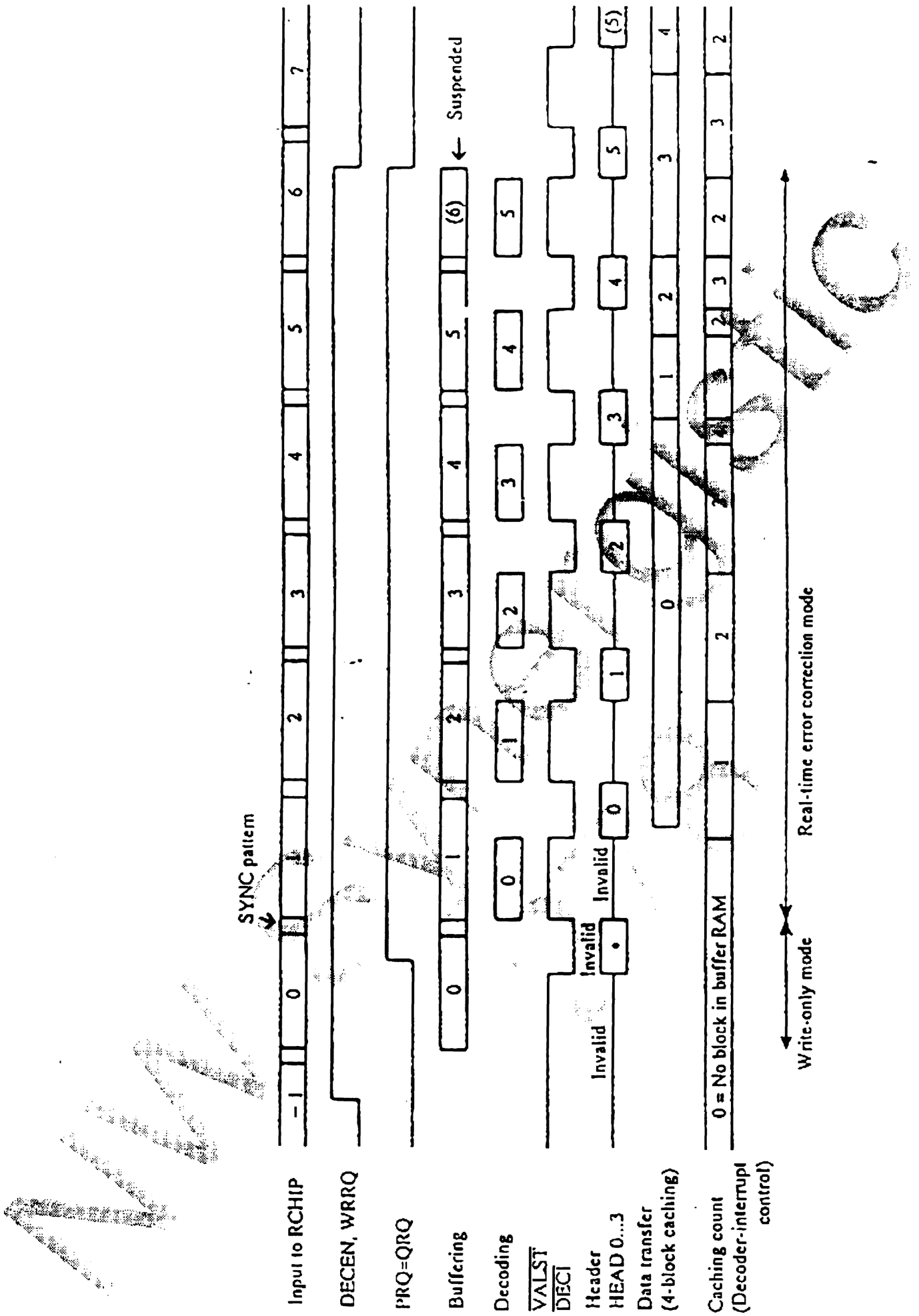


Figure 26. Error Checking in Protected Sync Mode (SYIEN=SYDEN=1)

Repeated correction can be implemented by modifying the Error Checking shown in figure 25.

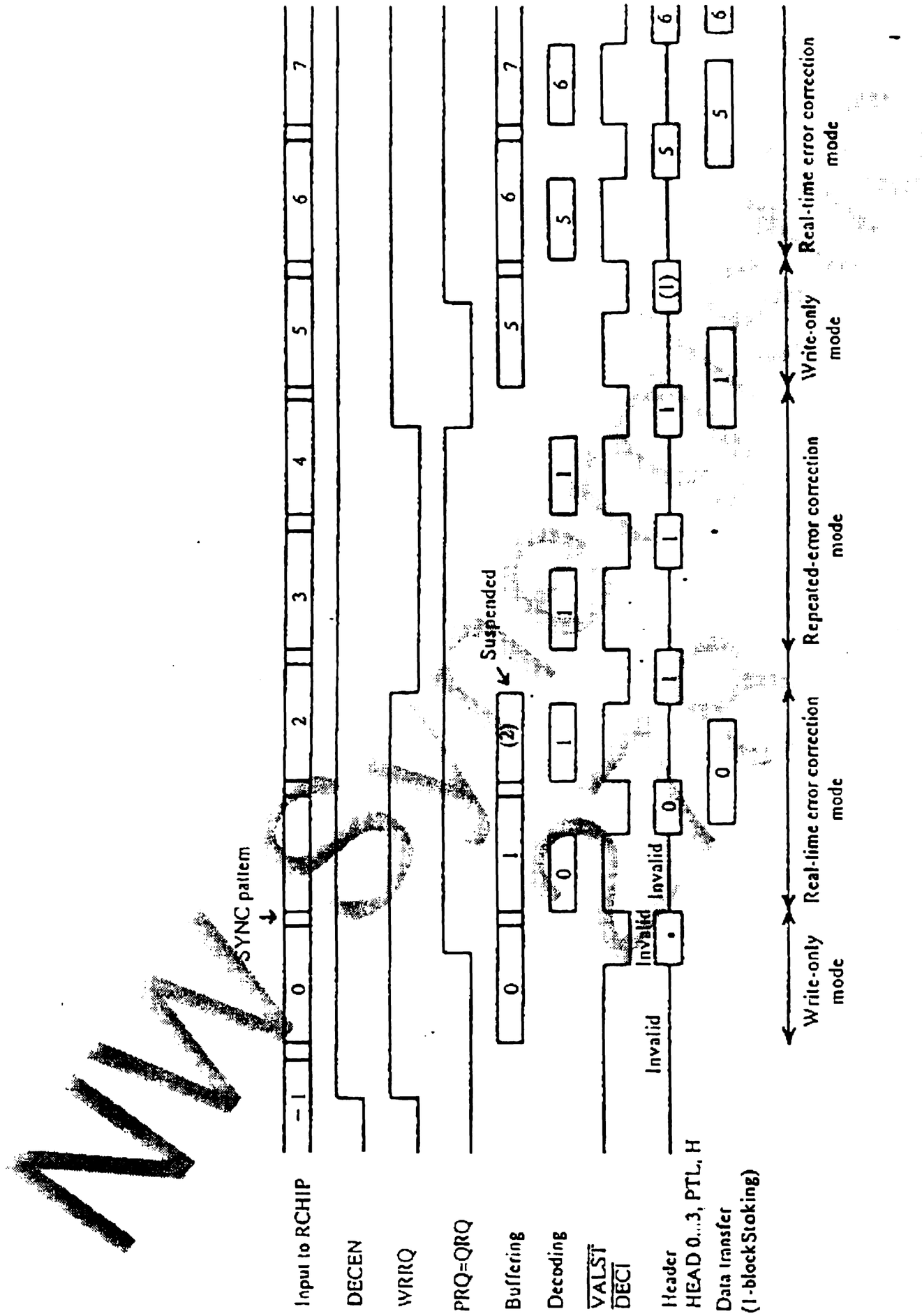
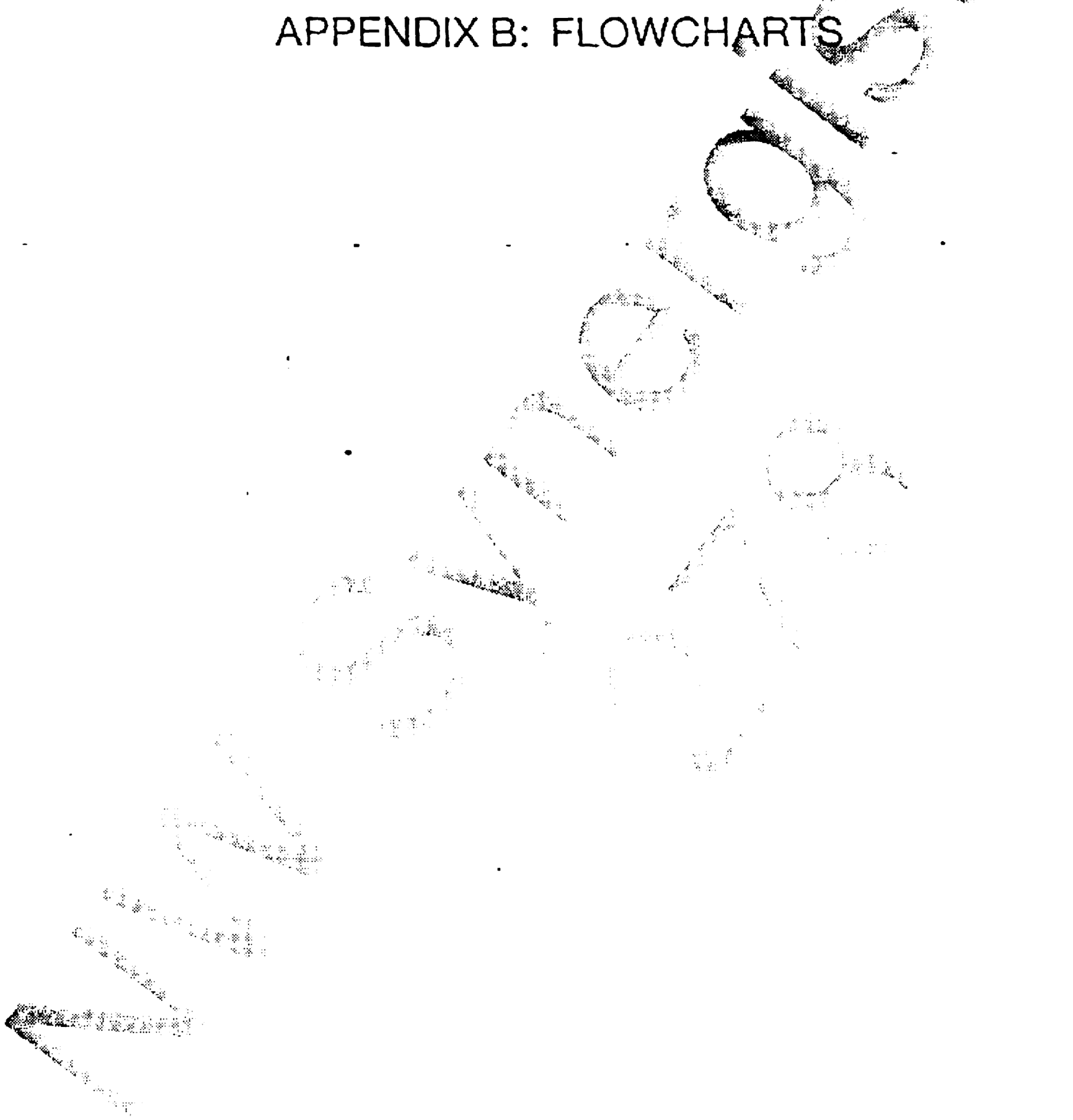


Figure 27. Repeated Error Correction in CD-ROM

APPENDIX B: FLOWCHARTS



TYPICAL CONTROLLER PROGRAMS

- Real-Time Error Detection and Correction in Mode 1

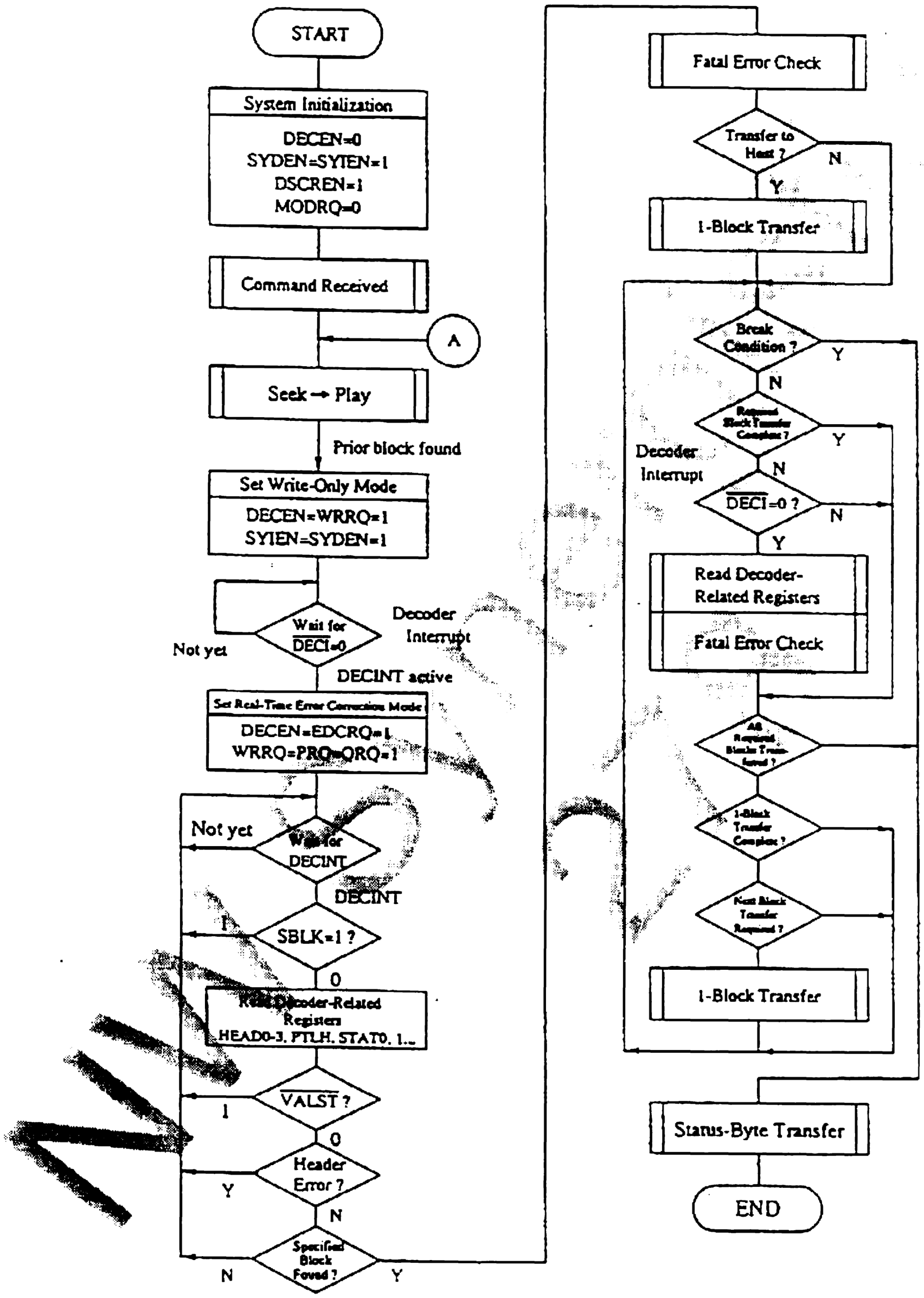


Figure 28. Non-Cached Block-by-Block Transfer

FLOWCHART DESCRIPTION

The steps shown in the preceding flowchart are as follows;

i. Initialize system settings

Disable the decoder by setting DECEN to 0, set SYIEN and SYDEN to 1 to set full sync mode, and enable the descramble function by setting DSCREN to 1.

ii. Data request received from the host computer.

iii. Seek the specified block using subcode.

In actual operation, the controller seeks a block slightly ahead of the specified block.

iv. When a block slightly ahead of the desired block is located, enable the PLAY function and read out a continuous data stream.

v. Set DECEN to 1 to enable the decoders write-only mode. Error correction cannot be performed until at least one block of data has been written to the buffer. Write-only mode is maintained until this one-block write is complete.

Use of sync hunt mode (SYDEN=1, SYIEN=0) is recommended. When the LC8950 hunts a real sync from the data stream, this is indicated as a NOSYNC=0, ensuring that the device is really synchronized. Protected sync mode (SYDEN=SYIEN=1) should then be set.

vi. After the write-only mode has continued for at least one block, enable real-time error correction mode and set EDCRQ to 1 immediately after the decoder interrupt.

vii. Continuing PLAY mode, check the header registers (HEAD0 to HEAD3) immediately after each decoder interrupt to see if the desired block has been decoded. Also check the header error flags in STAT1.

viii. If the desired block has not been located, wait for the decoder interrupt at the next block and check again.

ix. When the desired block is located, read STAT0 to check the block for fatal errors. If CRCOK is 1 and WSHORT, ILSYNC, UCEBLK, and ERABLK are all 0, the current block is valid data.

In the decoder's protected sync mode (SYDEN=SYIEN=1), the LBLK and SBLK flags do not set to 1 on errors, so they do not require checking. If the NOSYNC flag is set, the host does not need to be informed of blocking errors immediately as long as the header is checked successively.

If WSHORT is 1, decoding of the current block is aborted as it is not possible to perform error detection and correction and CRC checking on the block. All bytes received, however, are retained in the buffer. Check the relationship between XTALCK and LRCK.

If ILSYNC is 1, an insufficient number of bytes have been read in to complete the current block, making error detection and correction and CRC checking impossible. If this incomplete block were to be transferred to the host computer, the transferred block would include the head of the next block.

x. If there are no errors in the current block, compare the block's head address in the data buffer—that is, the pointer in PTL and PTH—with the address of the blocks in the transfer queue to check for a buffer overflow.

xi. Transfer the current block to the host computer. Copy the contents of PTL and PTH into DACL and DACH and the number of bytes into DBCL and DBCH. Write to DTTRG to enable data transfer. The host-interface control signals are used to send the data as described in section 12.3 *Data Transfer.*

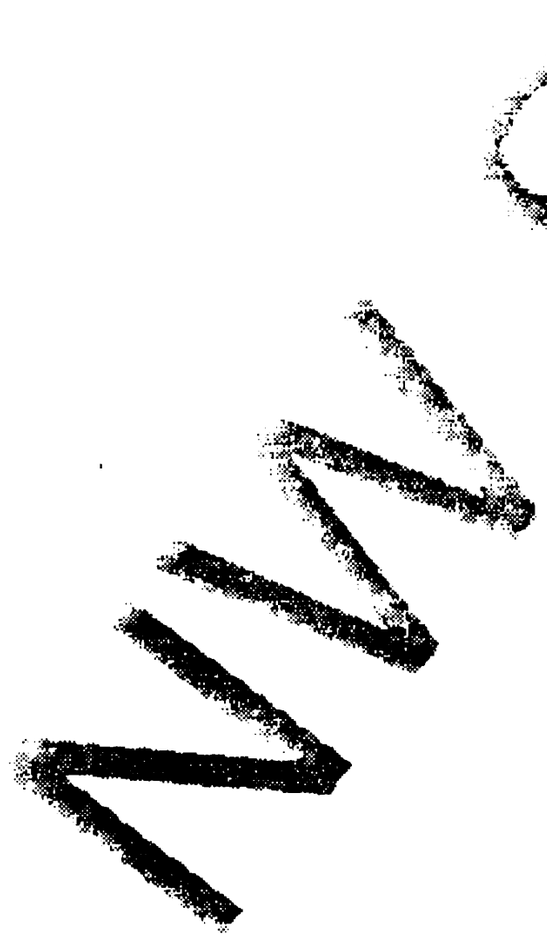
xii. Process the decoder interrupt, the transfer-complete interrupt and the command-received interrupt. These interrupts may be processed asynchronously and in order. Note that the $\overline{INT}$ pin generates only a single interrupt signal, even when two interrupts occur at the same time. The controller should read IFCTRL to ascertain whether another interrupt is pending after the processing of the current interrupt is completed.

xiii. When the desired number of blocks have been read out, disable decoder operations and decoder interrupts, and set the CD equipment to PAUSE.

xiv. If the host generates a status request command, suspend decoding, and write the status byte to the SBOUT register, and transfer the required status information.

xv. The error handling procedures can be designed to meet the needs of the application. Typical responses to block errors include:

- Perform repeated-error correction for as long as specified and then seek the next block.
- Give priority to real-time processing. Blocks containing errors are passed to the host together with their error status.
- Seek and re-read the block containing the error.



- Real-Time Error Detection and Correction in Mode 1

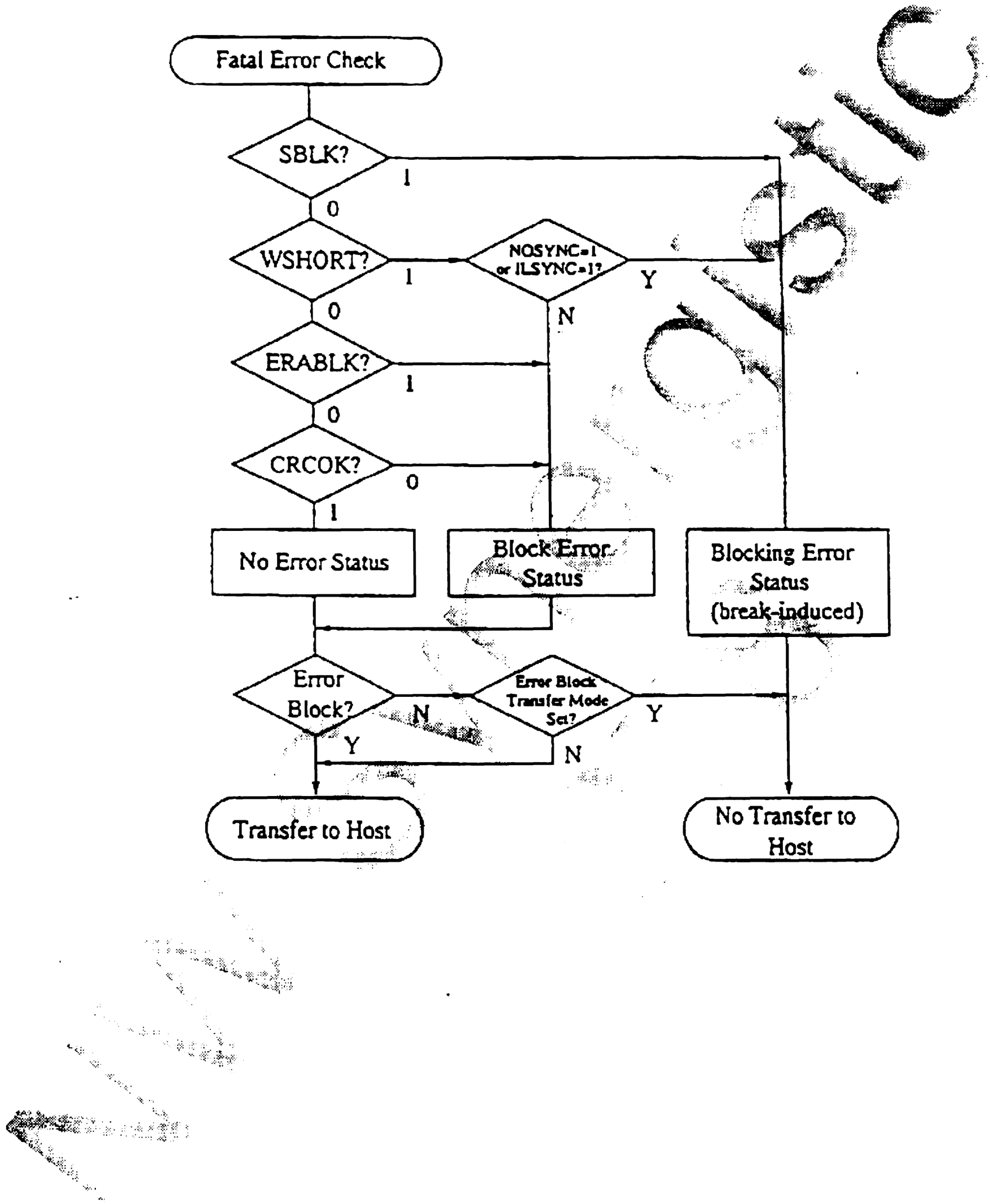


Figure 29. Fatal Error Checking

- Repeated Error Correction in Mode 1

  Repeated error correction can be implemented by modifying the routine for fatal-error checking in the previous flowchart as shown below.

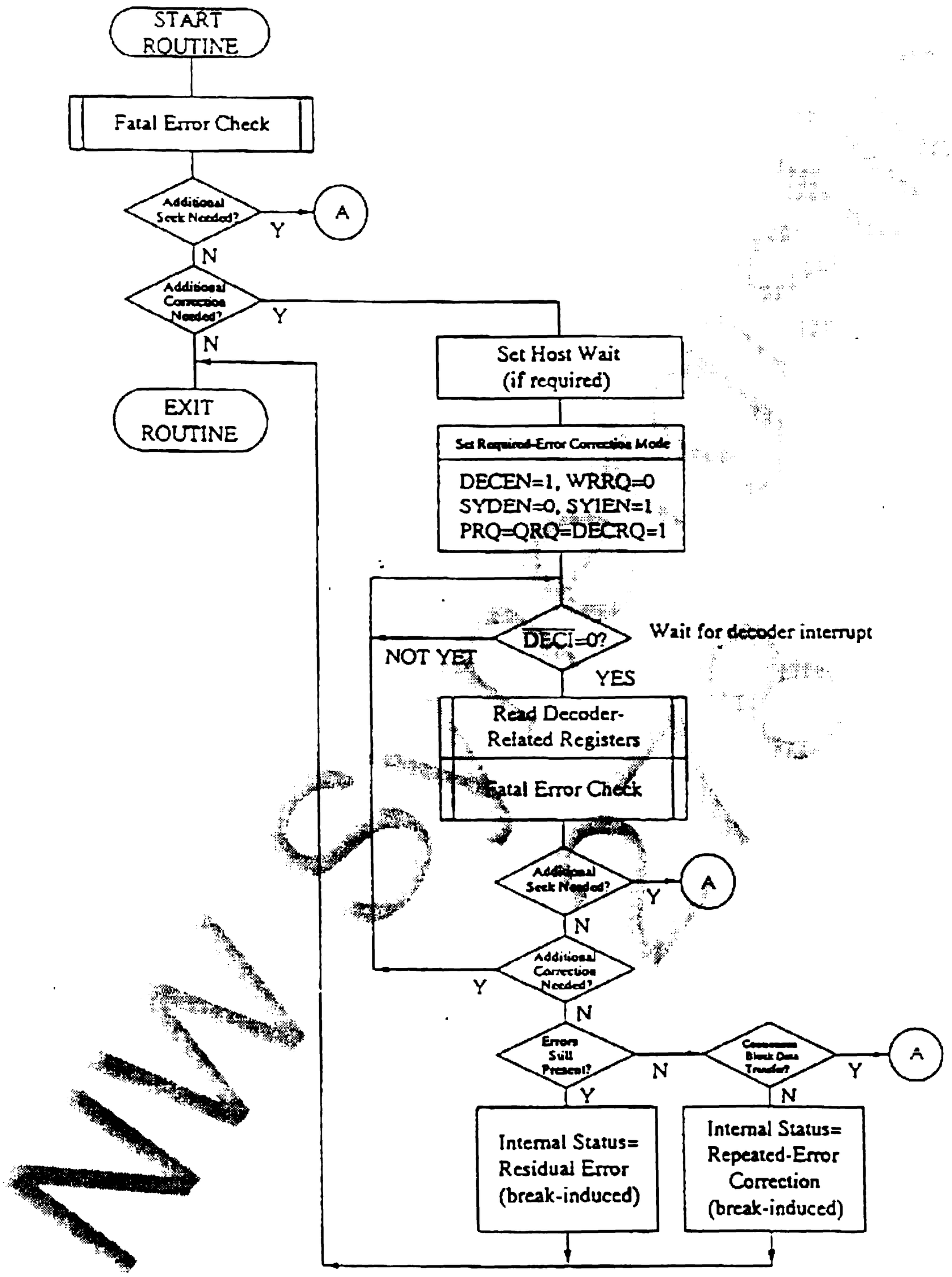


Figure 30. Repeated Error Correction

- Real-Time Error Detection and Correction in Mode 2

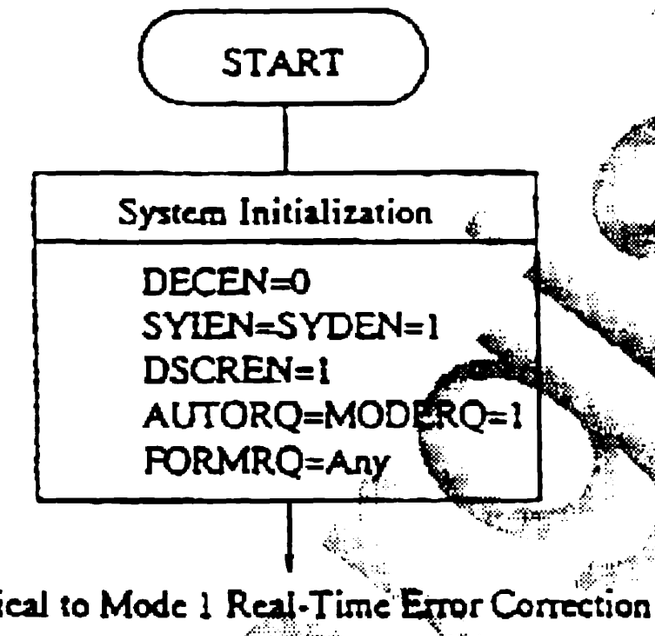


Figure 31. Automatic Form Checking

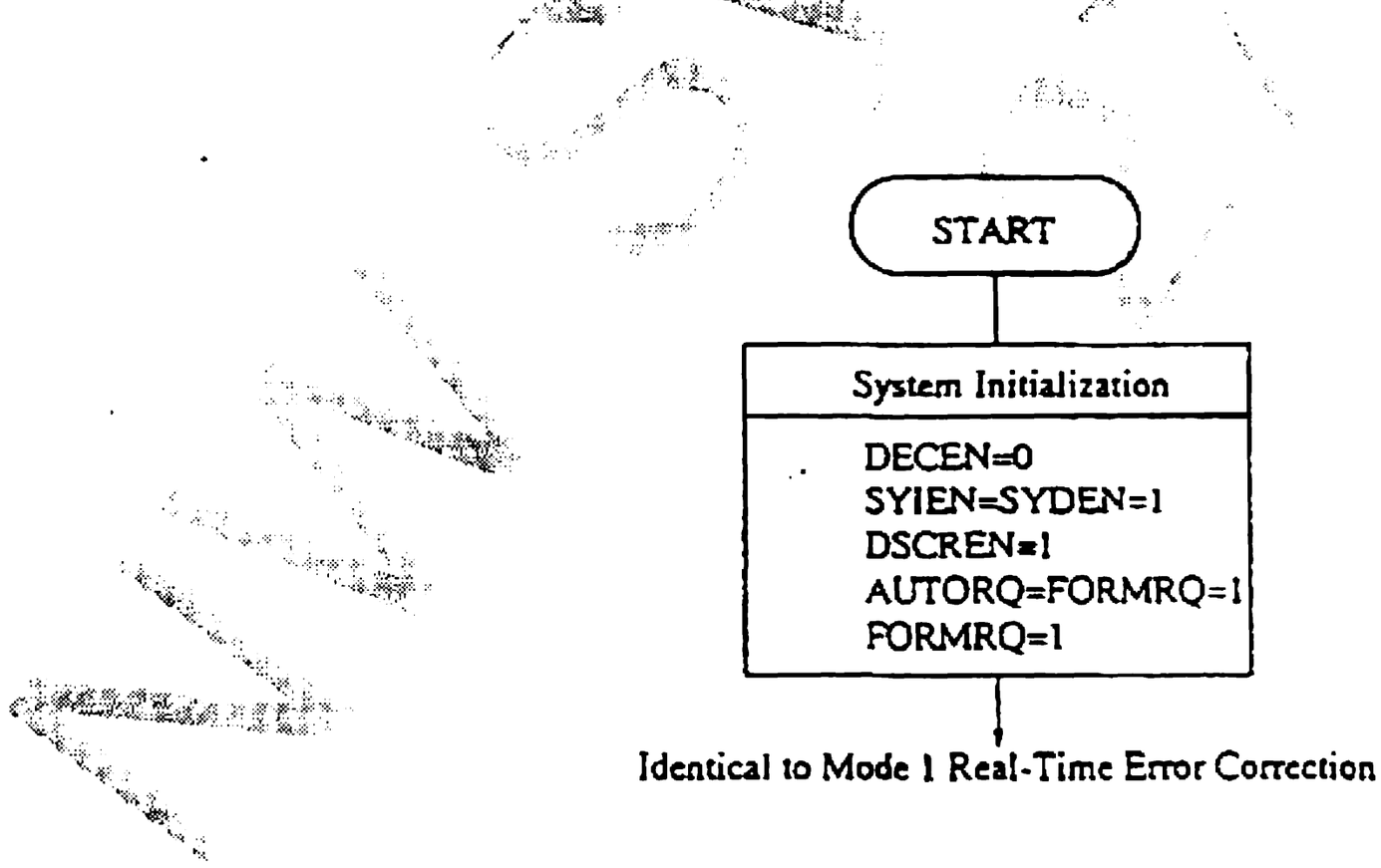


Figure 32. Real Time Error Correction in Form 1

- Data Transfers in Modes 1 and 2

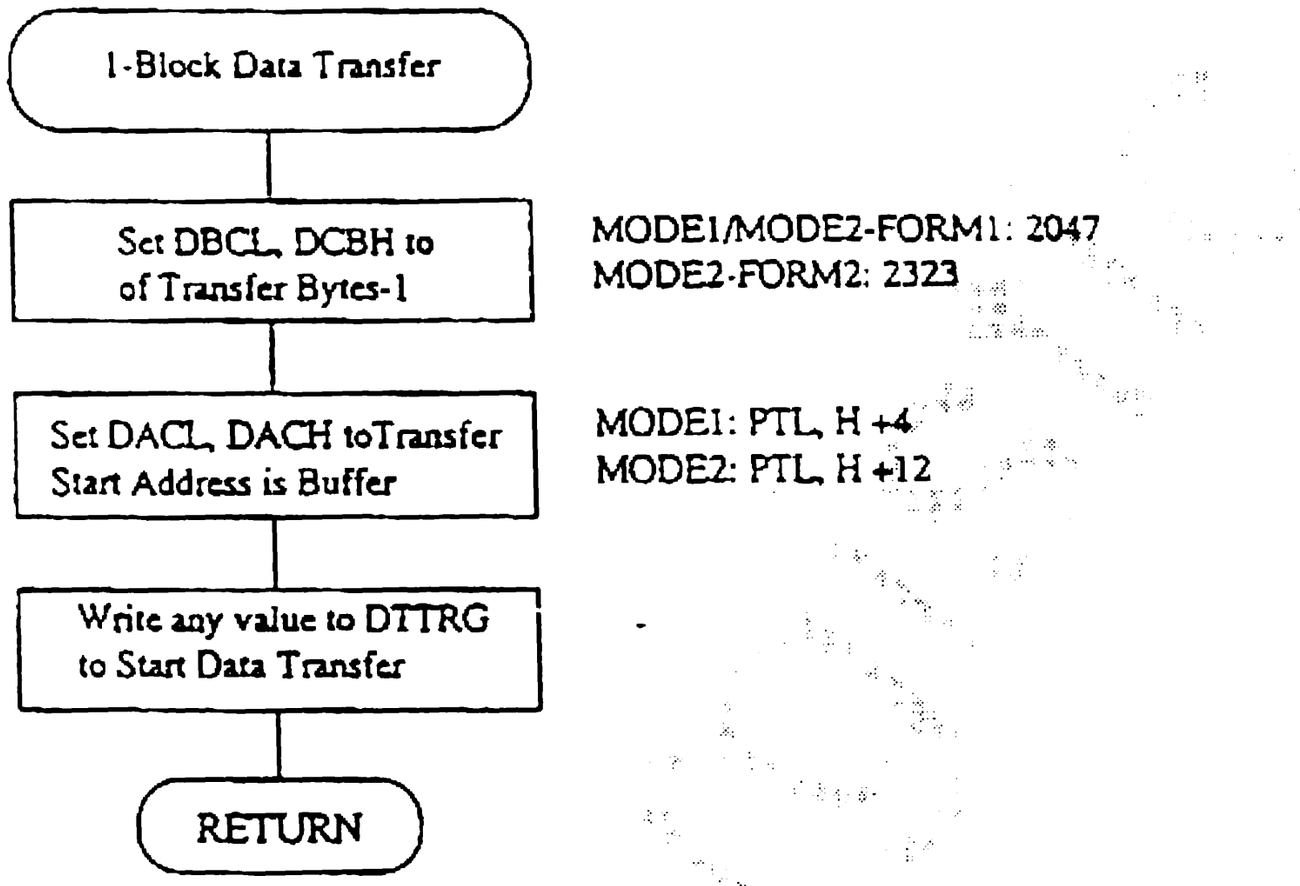


Figure 33. Single Block Data Transfers

APPENDIX C: CONFIGURATION EXAMPLES

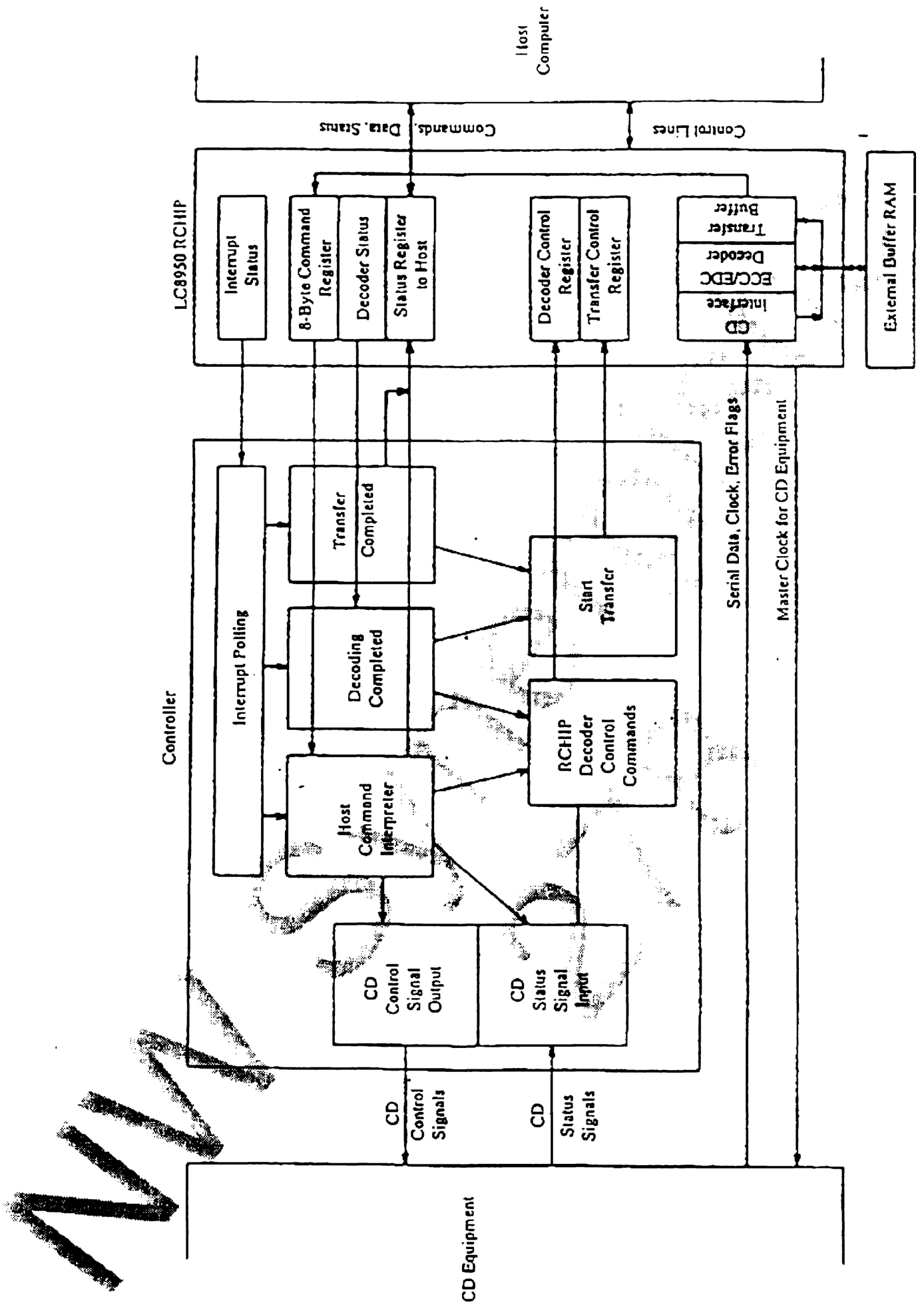


Figure 34. Typical LC8950 Controller

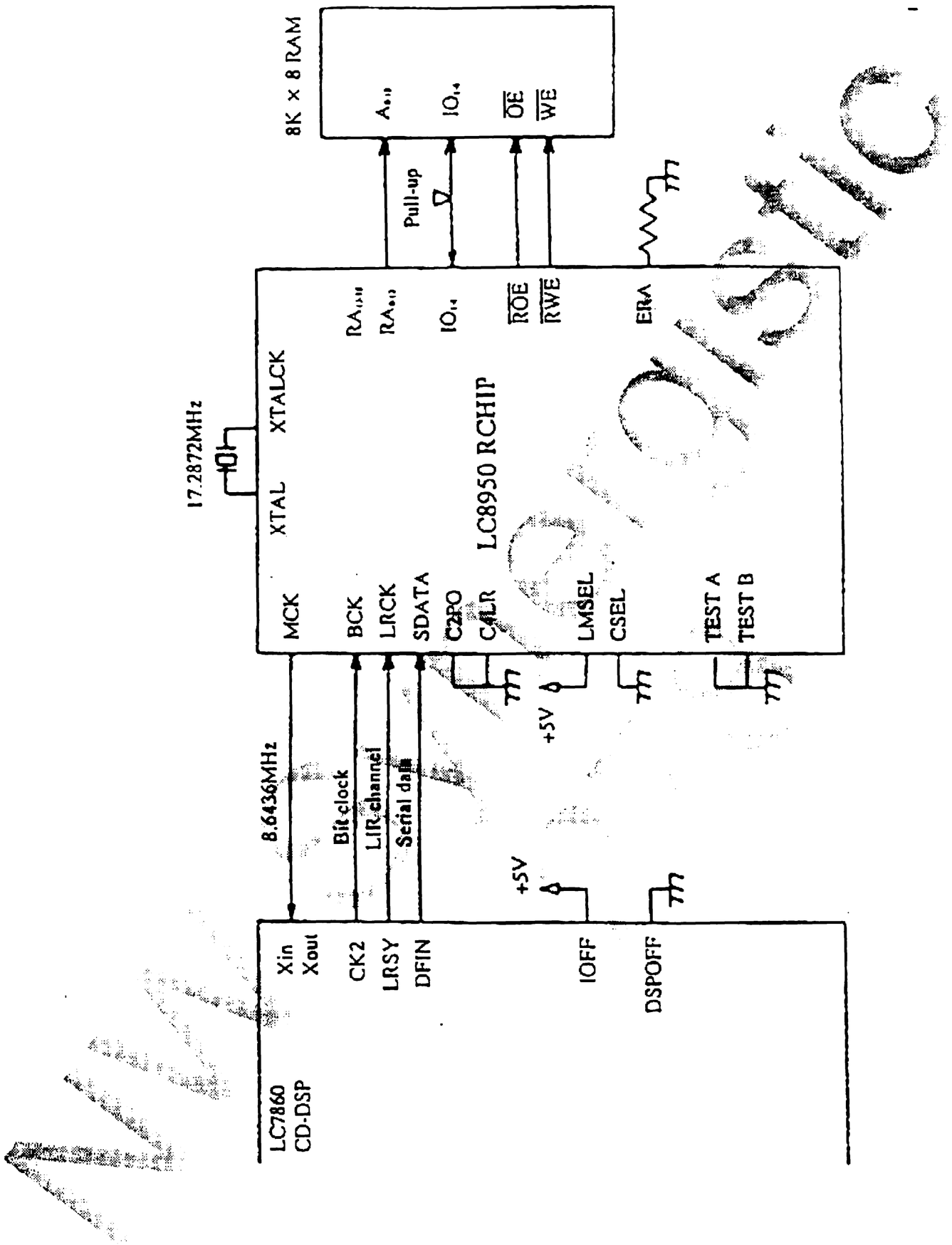


Figure 35. CD-DSP (Sanyo LC7860) Interface

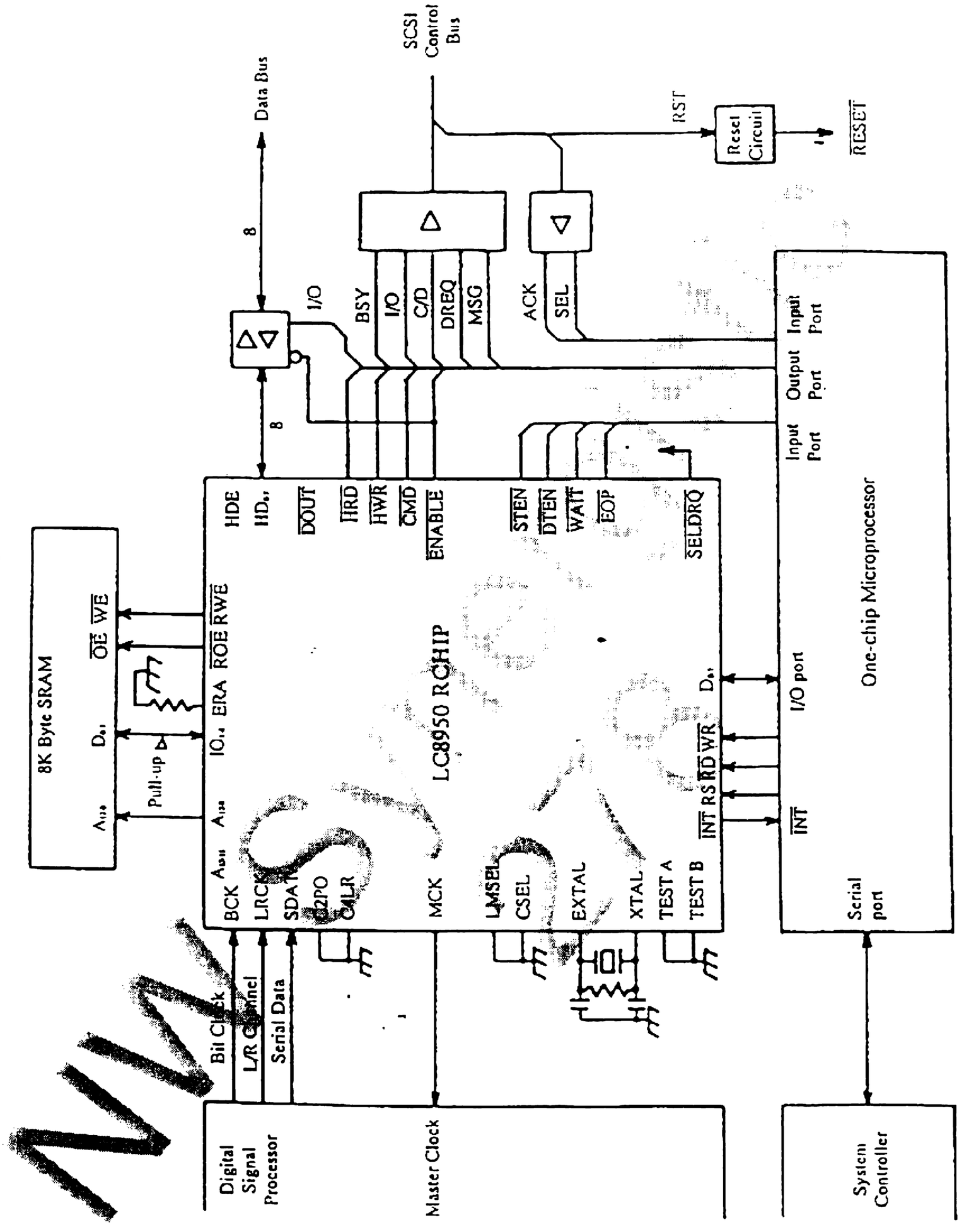


Figure 36. SCSI System Configuration

APPENDIX D: APPLICATION NOTES

The information in this section does not apply to the LC8951. See appendix E.

1. Buffer Access Time

   The access time of the SRAM devices interfaced to I/O Pins IO1 to IO8 must be 120 ns or less. The access time of the SRAM devices interfaced to the ERA pin must be 100 ns or less.

2. Decoder Mode Operation

   When setting the decoder enable bit CTRL0-DECEN (Register 10 during writes) to 1, either or both of the buffered write enable bit (CTRL0-WRRQ) or EDC decoder enable bit (CTRL0-EDCRQ) must be held at 1.

   Note that in the operation mode defined by DECEN=1, WRRQ=0, and EDCRQ=0, the header and subheader data cannot be read correctly. Monitor-only mode should not be used.

3. MCK Output during RESET

   The clock signal output at MCK stays LOW while the $\overline{\text{RESET}}$ input is held LOW. If one or more of the peripheral devices requires clock pulses while the LC8950's $\overline{\text{RESET}}$ pin is LOW, the LC8950's $\overline{\text{RESET}}$ pin and the RESET inputs on peripherals should be separated. In this case, the LC8950's RESET should be set to HIGH first, followed by that of the peripheral.

4. Controlling SRAM Chip-Select Inputs

   Figure 37 *RA0 to RA15, $\overline{\text{RWE}}$ and $\overline{\text{ROE}}$ Timing* shows the timing relationships between the RAM read address outputs, RA0 to RA15, and the write and output enable signals, $\overline{\text{RWE}}$ and $\overline{\text{ROE}}$.

   When the read address outputs RA0 to RA15 are being decoded and applied to the chip select inputs of two or more SRAM chips, $\overline{\text{RWE}}$ should be delayed by at least t + 40 ns, where t is the address decoder delay time.

   There may be a point when $\overline{\text{RWE}}$ and $\overline{\text{ROE}}$ are both active. If this overlap disrupts the operation of the RAM buffer, $\overline{\text{RWE}}$ should be delayed by 40 to 60 ns.

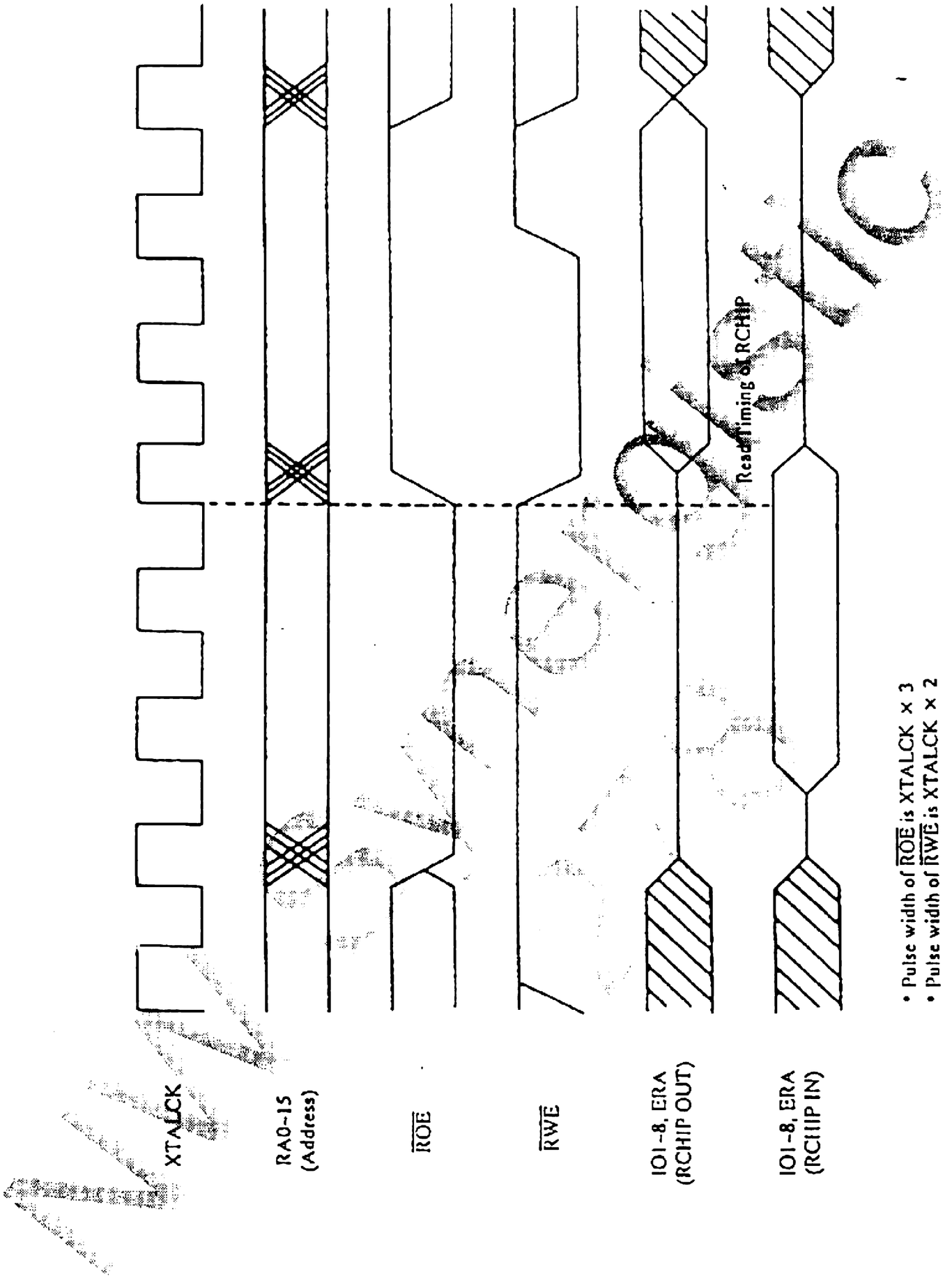


Figure 37. LC8950 RA0 to RA15, $\overline{RWE}$ and $\overline{ROE}$ Timing

APPENDIX E: LC8950 AND LC8951 FUNCTIONALITY

Section 8 — Pin Descriptions

C4LR — C2 error flag pointer strobe input

It is no longer necessary to provide this input as the LC8951 reads in C2PO data using internal timing based on BCK. As a result, the minimal signal connections to the CD equipment are as follows.

LRCK, BCK, SDATA, C2PO	Erasure connection algorithm only
LRCK, BCK, SDATA	Detect-and-correct algorithm only

Pin 35 should be tied to V_{DD} or V_{SS}.

C2PO — C2 error flag pointer input

[...] The data at C2PO are read in at the rising edge of the signal at C2LR (if present) or [...]

$\overline{RCS}$ — RAM chip select

The LC8951 uses this pin differently from the LC8950's $\overline{DOUT}$ pin. Tying it to the $\overline{CS}$ pin decreases the power consumption of the external RAM buffer. If not used for this purpose, the pin should be left open.

Section 11.11 — CTRL0 (R10 during writes)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
DECEN	*	E01RQ	AUTORQ	ERAMRQ	WRRQ	QRQ	PRQ

The LC851 does not use the EDCRQ bit. Instead, it automatically includes EDC decoding is when any other type of decoding is requested—that is, when at least one of the lowest three bits (WRRQ, QRQ and PRQ) is non-zero and decoding is enabled (DECEN=1).

Section 11.12 — CTRL1 (R11 during writes)

7. MBCKRQ — Mode Byte Check Request

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SYIEN	SYDEN	DSCREN	COWREN	MODRQ	FORMRQ	MBCKRQ	SHDREN

This bit, defined for the LC8951 only, enables/disables the mode check function. If the mode byte in the raw data is not compatible with the mode specified by the MODRQ bit (0 for Mode 1, 1 for Mode 2), this function disables error detection and correction (EDAC) processing for the block.

Status	Operation
0	Disable function.
1	Enable function.

The LC8950 does not use this bit or have this function.

Section 11.13 — STAT3 (R15 during reads)

CBLK — Corrected Block Flag

The LC8951 sets this bit to 1 if it has corrected the current block.

Section 11.15 — STAT2 (R14 during reads)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
RMOD3	RMOD2	RMOD1	RMOD0	MODE	NOCOR	RFORM1	RFORM0
$\overline{\text{VALST}}$	WLONG	CBLK	•	•	•	•	•

RMOD 3, 2, 1 and 0 — Raw Mode Byte

These four bits, defined for the LC8951 only, contain the results of a preliminary check on the raw mode byte in the next block to be decoded.

RMOD3 = BIT7 + BIT6 + BIT5 + BIT4 + BIT3 + EF
RMOD2 = BIT2 + EF
RMOD1 = BIT1 + EF
RMOD0 = BIT0 + EF

where + denotes a logical-OR, and EF is the byte's error flag (from C2P0).

If RMOD3=0, the other three bits uniquely determine the mode (0 to 7); otherwise, they just give the lower three bits of a mode in the range 08H to 0FFH. If all four bits are 1, however, there is an additional interpretation: the mode byte contains an error.

RMOD3	RMOD2	RMOD1	RMOD0	
0	0	0	0	Mode 0
0	0	0	1	Mode1
0	0	1	0	Mode 2
0	0	1	1	Mode 3
0	1	0	0	Mode 4
0	1	0	1	Mode 5
0	1	1	0	Mode 6
0	1	1	1	Mode 7
1	0	0	0	Mode $ x 0 or $ x 8
1	0	0	1	Mode $ x 1 or $ x 9
1	0	1	0	Mode $ x 2 or $ x A
		:		
1	1	1	0	Mode $ x 6 or $ x E
1	1	1	1	Mode $ x 7 or $ x F of mode byte error

These bits do not reflect the status of the current block, but that of the *next* block, the one to be decoded after the next decoder interrupt.

NOCOR — No Correction

The LC8951 sets this bit to 1 if ECC processing is not possible for the current block. Such processing is blocked for:

- an illegally synchronized block (ILSYNC=1)
- a data block specified in software as Mode 2, form 2
- a Mode 2, form 2 data block automatically detected during processing
- a Mode 2 submode byte error automatically detected during processing
- a mode mismatch detected by the mode check function (MCHQRQ=1)
- a mode byte error detected by the mode check function (MCHQRQ=1)

NOCOR is invalid if COWREN=0 or PRQ=QRQ=0.

This usage represents an extension of the LC8950's FORM bit, which was formerly only applicable to Mode 2 (CD-I) blocks, to Mode 1 (CD-ROM) data.

RFORM — Raw Form Bit

These two bits, defined for the LC8951 only, contain a preview of the FORM bit in the submode byte for the next block to be decoded.

RFORM1: Error flag (from C2PO) for the byte
RFORM2: Form bit from byte

RFORM1	RFORM0	
0	0	Form 1
0	1	Form 2
1	—	Error

These bits do not reflect the status of the current block, but that of the *next* block, the one to be decoded after the next decoder interrupt.

These bits are only meaningful for Mode 2 (CD-I) blocks.

Section 12.3.3 — Transfer start delays

Revision of the output timing for the IO1 to IO8, $\overline{RDE}$, $\overline{RWE}$, and ERA pins has eliminated both the need for a delay circuit on the $\overline{RWE}$ pin and the risk of collisions on the 9-line (IO1 to IO8 plus ERA) data bus to the external RAM buffer. See appendix F. *AC Characteristics.*

As a result, SRAMs with access times of 120 ns or less may be directly connected to the LC8951.

Pull-up resistors

All data bus pins joining the LC8951 to the external SRAM (IO1 to IO8 plus ERA) and the host (HD0 to HD7) have 20 kΩ pull-up resistors. The $\overline{INT}$ pin has a similar 20 kΩ pull-up resistor and provides open-drain output.

12.4 — Reset

A reset triggered by pulling the $\overline{RESET}$ pin LOW no longer affects the MCK clock output.

Section 12.5 — Decoder Modes

DECN	WRRQ	PRQ+ORQ	CRCK, ERABLK	HEAD0 to 3, STAT1	Decoder Mode
0	—	—	Invalid	Invalid	No decoding
1	0	0	Invalid	Raw	Monitor only mode
1	1	0	Valid	Decoded	Write only mode
1	1	1	Valid*	Decoded	Real-time correction mode
1	0	1	Valid**	Decoded	Repeated correction mode

* After WRRQ=0 or DECEN=0, the first one block should be ignored.

** After WRRQ=0 or DECEN=0, the first two blocks should be ignored.

Section 12.5.4 — Monitor only mode

In this mode, the LC8951 loads the raw block header and subheader from the input data stream into the registers HEAD0 to HEAD3 and latches the corresponding error flags (from the C2PO pin) into the RMOD and RFORM bits of the STAT1 register, but does not store the data block in the RAM buffer or subject it

to any EDAC processing.

These bits do not reflect the status of the current block, but that of the *next* block, the one to be decoded after the after decoder interrupt.

Section 12.6 — Decoder Control

The duration of the decoder interrupt (active low at the $\overline{INT}$ pin) depends on the decoder mode.

WRRQ+PRQ+QRQ	MODE	FORM	ACTIVE (µs)
0	X	X	About 2750 (Same as LC8950)
1	0	X	About 1800
1	1	0	About 1800
1	1	1	About 1500

The last three values are considerably longer than those for the LC8950.

APPENDIX F: LC8950 AND LC8951 AC CHARACTERISTICS

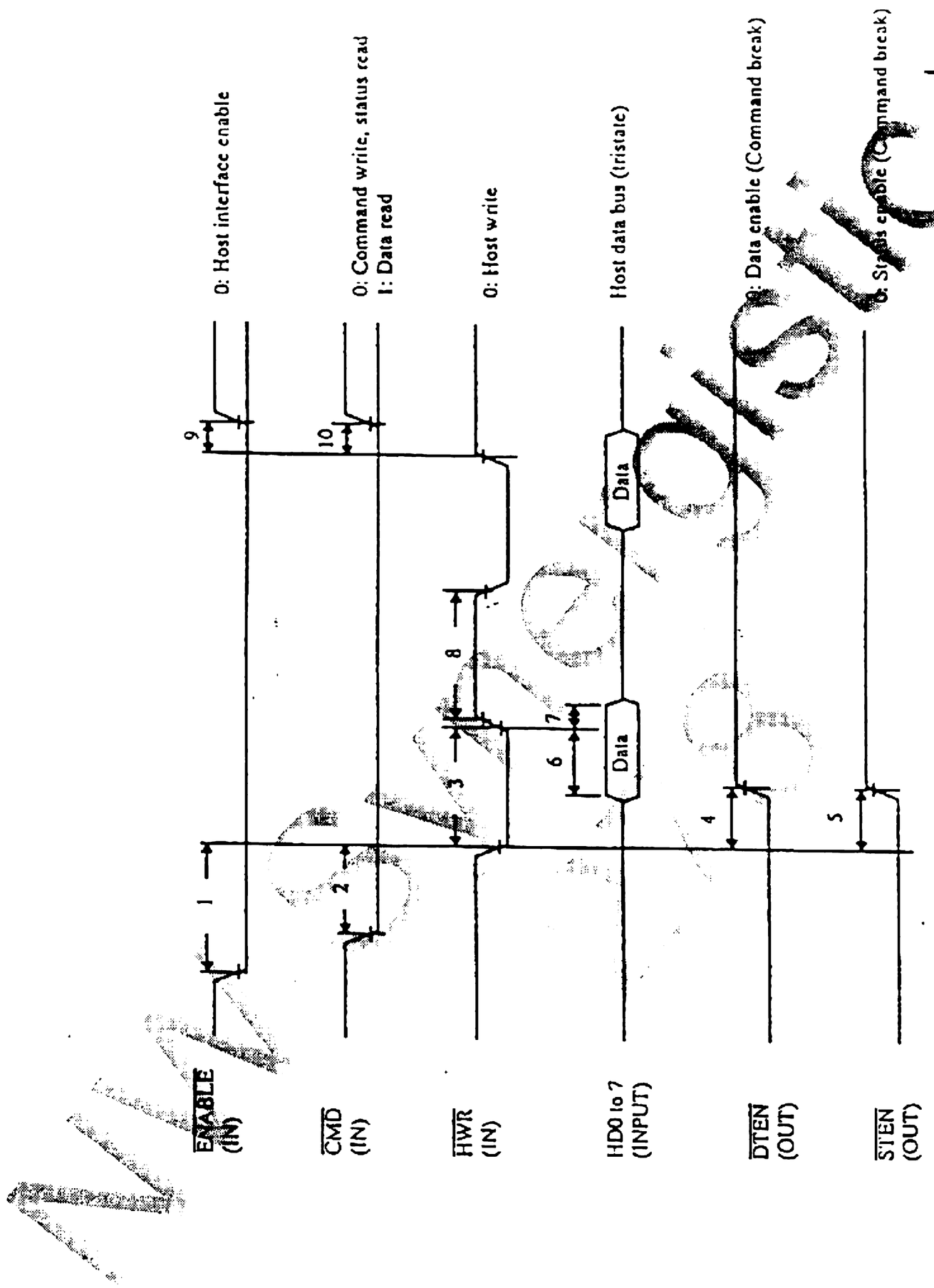
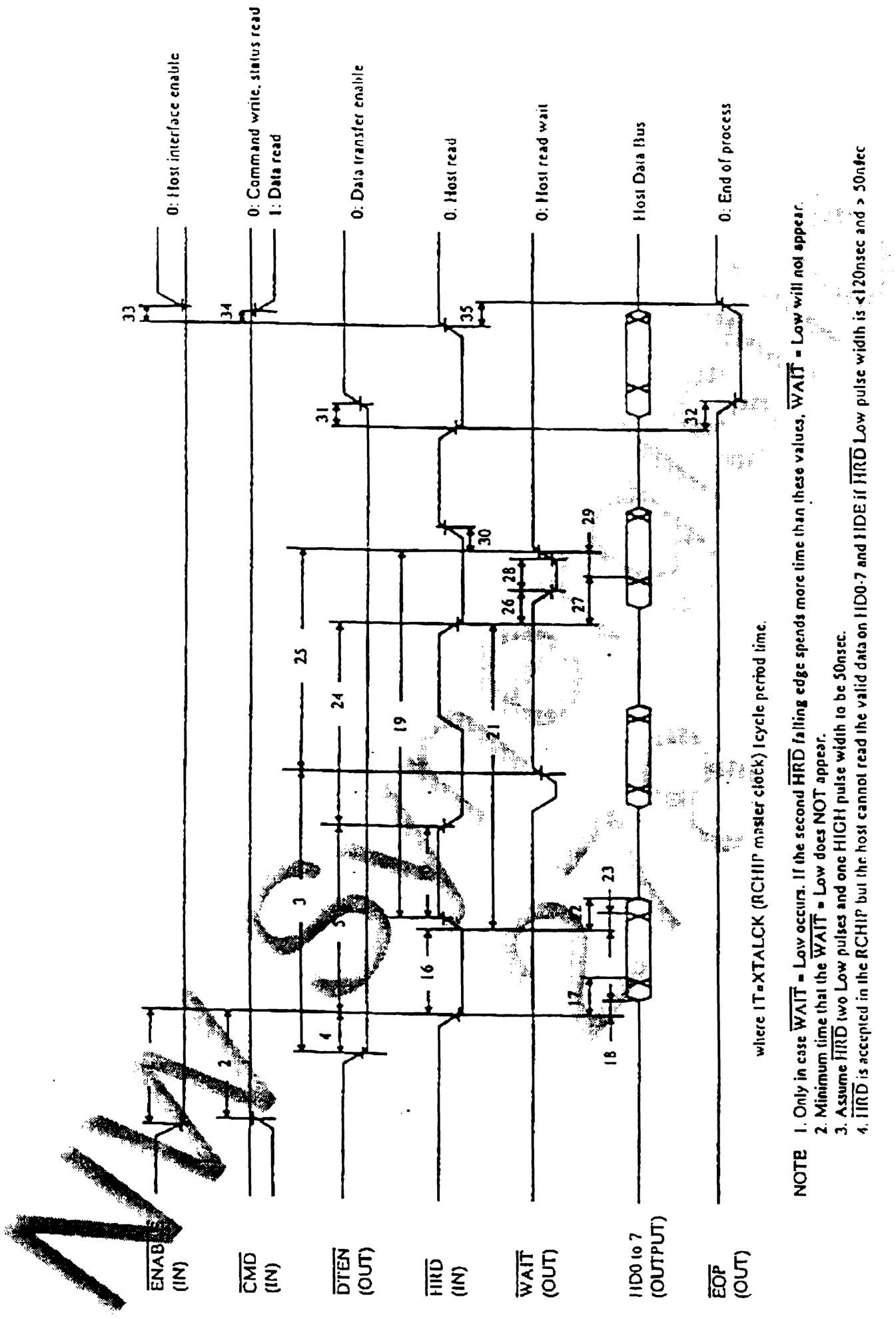


Figure 38. LC8950/LC8951 Command Write



where 1T=XTALCK (RCHIP master clock) 1cycle period time.

NOTE 1. Only in case $\overline{WAIT}$ = Low occurs. If the second $\overline{HRD}$ falling edge spends more time than these values, $\overline{WAIT}$ = Low will not appear.
2. Minimum time that the $\overline{WAIT}$ = Low does NOT appear.
3. Assume $\overline{HRD}$ two Low pulses and one HIGH pulse width to be 50nsec.
4. $\overline{HRD}$ is accepted in the RCHIP but the host cannot read the valid data on HD0-7 and HDE if $\overline{HRD}$ Low pulse width is <120nsec and > 50nsec

Figure 39. LC8950/LC8951 Data Read with WAIT Control

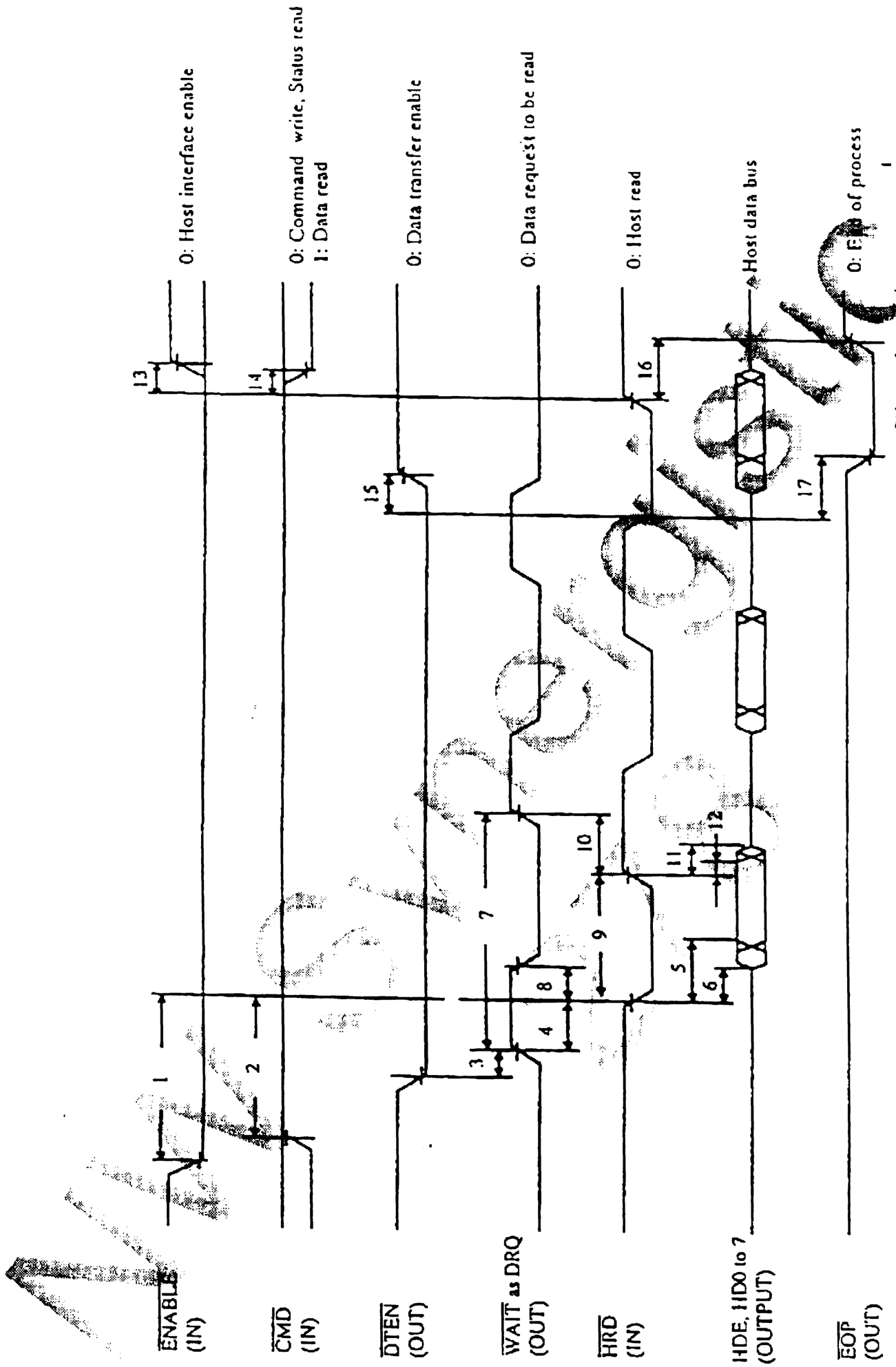


Figure 40. LC8950/LC8951 Data Read with DRQ Control

TABLE 12. TABLE 12. AC Electrical Characteristics

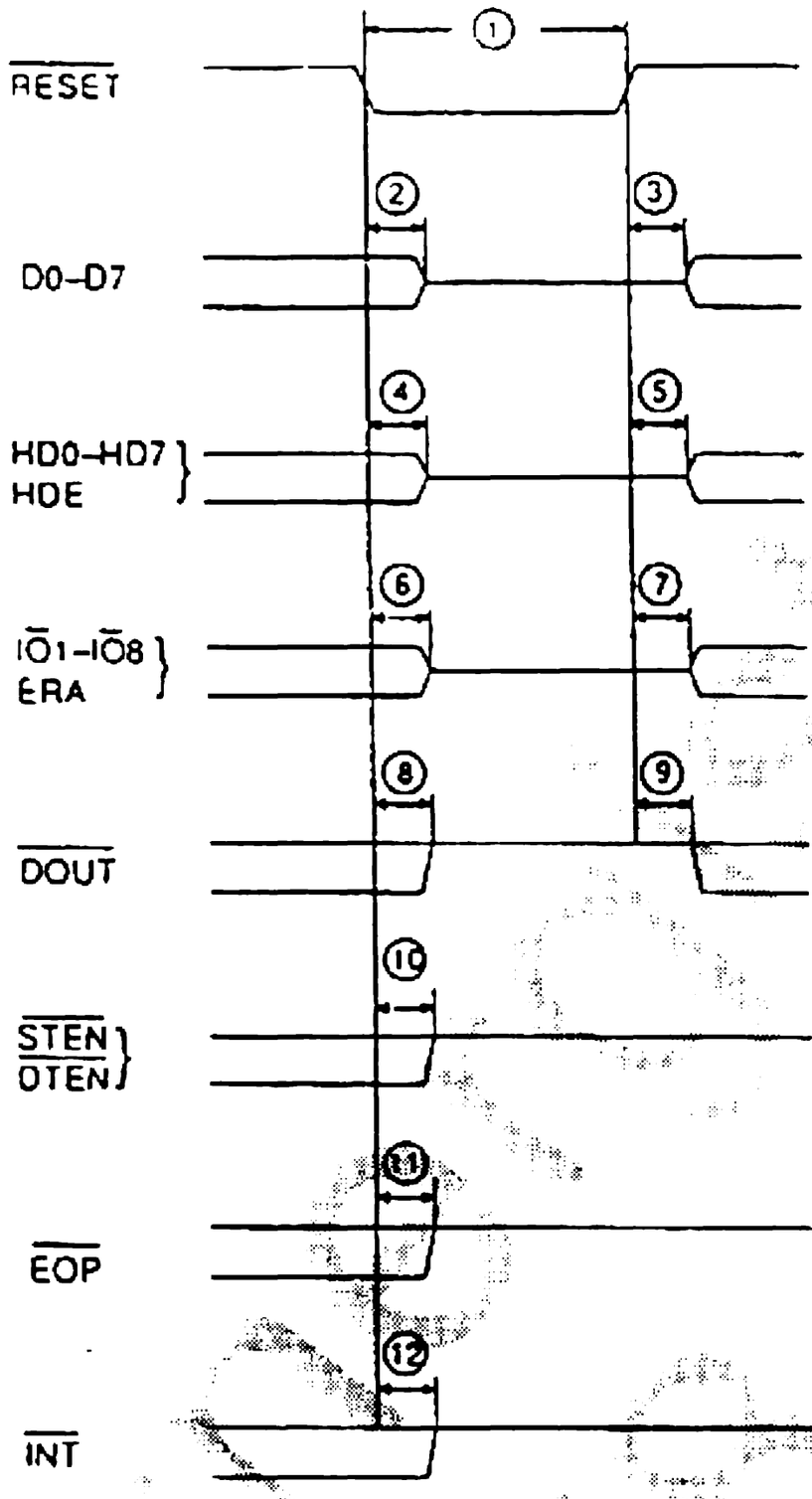
Number	Figure	Characteristic	Range				Unit
			LC8950		LC8951		
			min	max	min	max	
1	38	$\overline{\text{ENABLE}}$ to $\overline{\text{HWR}}$ (Host Write) setup	30	—	30	—	ns
2	38	$\overline{\text{CMD}}$ to $\overline{\text{HWR}}$ setup	15	—	15	—	ns
3	38	$\overline{\text{HWR}}$ low period	50	—	50	—	ns
4	38	$\overline{\text{HWR}}$ to $\overline{\text{DTEN}}$ change	—	105	—	105	ns
5	38	$\overline{\text{HRW}}$ to $\overline{\text{STEN}}$ change	—	105	—	105	ns
6	38	HD0–7 to $\overline{\text{HWR}}$ inactive	50	—	50	—	ns
7	38	$\overline{\text{HWR}}$ inactive to End of Data	20	—	20	—	ns
8	38	$\overline{\text{HWR}}$ high period	50	—	50	—	ns
9	38	$\overline{\text{HWR}}$ inactive to $\overline{\text{ENABLE}}$ inactive	0	—	0	—	ns
10	38	$\overline{\text{HWR}}$ inactive to $\overline{\text{CMD}}$ inactive	5	—	5	—	ns
11	39, 40	$\overline{\text{ENABLE}}$ to $\overline{\text{HRD}}$ setup	30	—	30	—	ns
12	39	$\overline{\text{CMD}}$ to $\overline{\text{HRD}}$ setup	15	—	15	—	ns
13	39	$\overline{\text{DTEN}}$ to $\overline{\text{WAIT}}$ inactive	6T–20 Note 1	6T+50	6T–20 Note 2	6T+50	ns
14	39	$\overline{\text{DTEN}}$ to $\overline{\text{HRD}}$ setup	0	—	0	—	ns
15	39	$\overline{\text{HRD}}$ period	6T–50 Note 2	—	6T–50 Note 2	—	ns
16	39	$\overline{\text{HRD}}$ low period	6T–50 Note 4	—	6T–50 Note 4	—	ns
17	39	$\overline{\text{HRD}}$ to valid data setup with $\overline{\text{WAIT}}$ inactive	—	120	—	No max	ns
18	39, 40	$\overline{\text{HRD}}$ to invalid data setup	5	100	5	100	ns
19	39	$\overline{\text{HRD}}$ inactive to the second $\overline{\text{WAIT}}$ inactive	10T+180 Note 1	—	10T+180 Note 1	—	ns
20	39	$\overline{\text{HRD}}$ high period	50	—	50	—	ns
21	39	$\overline{\text{HRD}}$ inactive to the second $\overline{\text{HRD}}$ active transition	6T+180 Note 1	—	6T+180 Note 2	—	ns
22	39, 40	$\overline{\text{HRD}}$ inactive to end of data	5	60	5	60	ns
23	39	$\overline{\text{HRD}}$ inactive to end of valid data	0	—	0	—	ns
24	39	$\overline{\text{HRD}}$ active to the next $\overline{\text{HRD}}$ active	6T+50 Note 2	—	6T+50 Note 2	—	ns
25	39	$\overline{\text{WAIT}}$ inactive to the next $\overline{\text{WAIT}}$ inactive	6T–20 Note 2	6T+50	6T–20 Note 2	6T+50—	ns
26	39	$\overline{\text{HRD}}$ active to $\overline{\text{WAIT}}$ active setup	80	—	80	—	ns
27	39	$\overline{\text{HRD}}$ active to valid data setup with $\overline{\text{WAIT}}$ active	— Note 3	10T+160	— Note 3	10T+160	ns
28	39	$\overline{\text{WAIT}}$ active period	0 Note 3	10T+30	0 Note 3	10T+30	ns
29	39	Valid data to $\overline{\text{WAIT}}$ inactive	-50	—	-50	—	ns
30	39	$\overline{\text{WAIT}}$ inactive to $\overline{\text{HRD}}$ inactive	50	—	50	—	ns

Number	Figure	Characteristic	Range				Unit
			LC8950		LC8951		
			min	max	min	max	
31	39	$\overline{HRD}$ active to $\overline{DTEN}$ inactive	—	120	—	120	ns
32	39, 40	$\overline{HRD}$ to $\overline{EOP}$ setup	—	120	—	120	ns
33	39, 40	$\overline{HRD}$ inactive to $\overline{ENABLE}$ inactive	0	—	0	—	ns
34	39, 40	$\overline{HRD}$ inactive to $\overline{CMP}$ active	5	—	5	—	ns
35	39, 40	$\overline{HRD}$ inactive to $\overline{EOP}$ inactive	—	120	—	120	ns
36	40	$\overline{CMD}$ inactive to $\overline{HRD}$ active	15	—	15	—	ns
37	40	$\overline{DTEN}$ active to $\overline{WAIT}$ inactive	5	25	5	25	ns
38	40	$\overline{WAIT}$ period	6T−20	—	6T−20	—	ns
39	40	$\overline{WAIT}$ inactve to $\overline{HRD}$ active	0	—	0	—	ns
40	40	$\overline{HRD}$ to $\overline{WAIT}$ setup	—	80	—	80	ns
41	40	$\overline{HRD}$ active period	2T	—	2T	—	ns
42	40	$\overline{HRD}$ to valid data setup	—	120	—	120	ns
43	40	$\overline{HRD}$ inactive to $\overline{WAIT}$ inactive	2T+10	4T−75	2T+10	4T−75	ns
44	40	Valid data to end of data	0	—	0	—	ns
44	40	$\overline{HRD}$ active to $\overline{DTEN}$ inactive	6T−20	—	6T−20	—	ns

Notes:

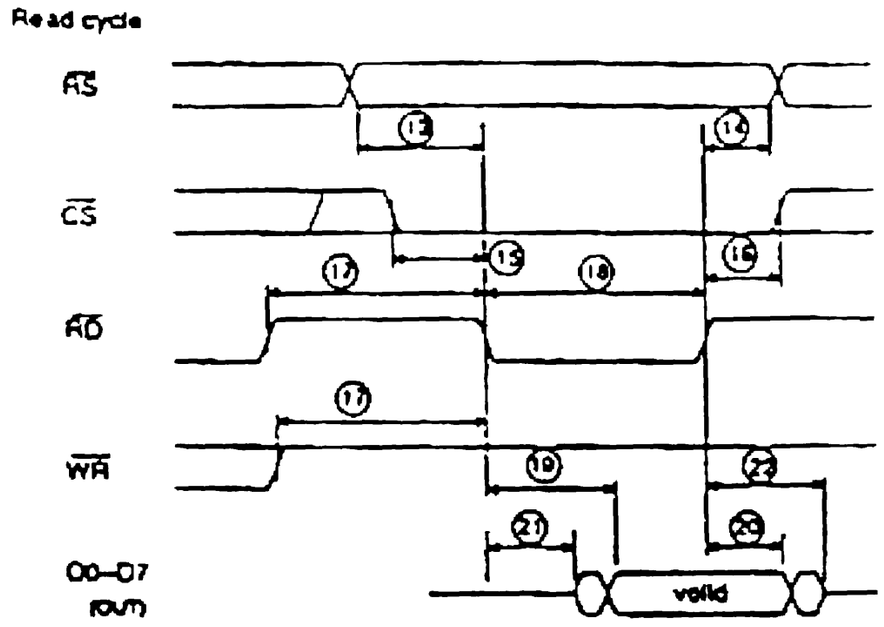
1. Only if $\overline{WAIT}$=LOW occurs. $\overline{WAIT}$=LOW does not appear if the second $\overline{HRD}$ falling edge exceeds these times.
2. Minimum time for which $\overline{WAIT}$=LOW does not occur.
3. Assume both $\overline{HRD}$ LOW and HIGH pulses to be 50 ns.
4. If the $\overline{HRD}$ LOW pulse width, $50 < t_{\overline{HRD}} < 120$ ns, the host cannot read the valid data on HD0 to 7 and HDE.

- Reset Timing



	Description	Rating min.	Rating max.	Unit
1	Reset pulse width	100	—	ns
2	Data hold	—	50	
3	Data delay	—	50	
4	Host data hold	—	50	
5	Host data delay	—	50	
6	RAM data hold	—	50	
7	RAM data delay	—	50	
8	$\overline{DOUT}$ hold	—	100	
9	$\overline{DOUT}$ hold delay	—	100	
10	$\overline{STEN}$, $\overline{DTEN}$ hold	—	100	
11	$\overline{EOP}$ hold	—	100	
12	$\overline{INT}$ hold	—	100	

Figure 41. LC8950/LC8951 Reset Timing



See table 12 for AC characteristics.

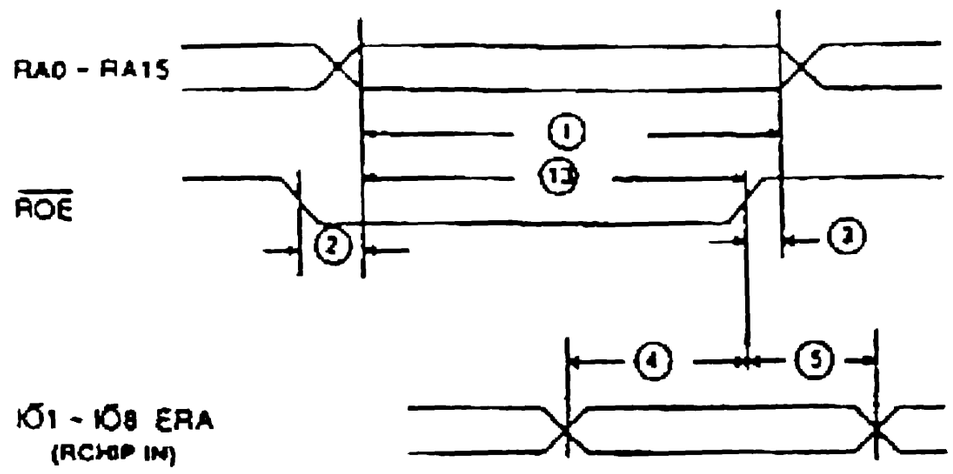
	Description	Rating min.	Rating max.	Unit
13	RS setup	40	—	ns
14	RS hold	10	—	
15	$\overline{CS}$ setup	30	—	
16	$\overline{CS}$ hold	5	—	
17	$\overline{RD}$, $\overline{WR}$ pulse	50	—	
18	$\overline{RD}$, $\overline{WR}$ pulse width	50	—	
19	Read data access	—	120	
20	Read data hold	10	—	
21	Data out delay	5	—	
22	Data out HIGH Z	10	80	
23	Write data setup	30	—	
24	Write data hold	20	—	

Notes

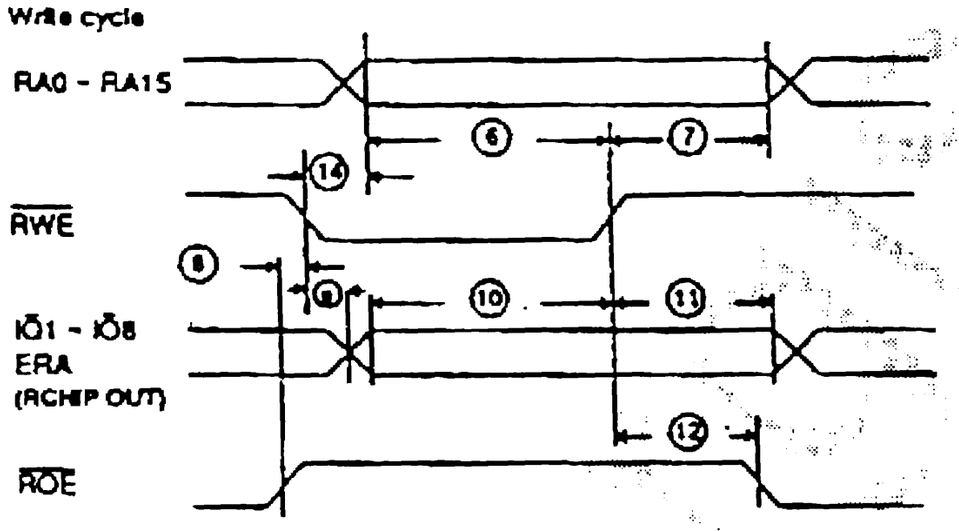
1. Timing intervals #13 and #14 are not absolute values, but intervals inserted to protect #19 and #20. In itself, #18 does not ensure validity of the values pins D0 to D7.
2. The maximum is 80 ns for the LC8951.

Figure 42. LC8950/LC8951 Host Interface

Read cycle



Write cycle



See table 12 for AC characteristics.

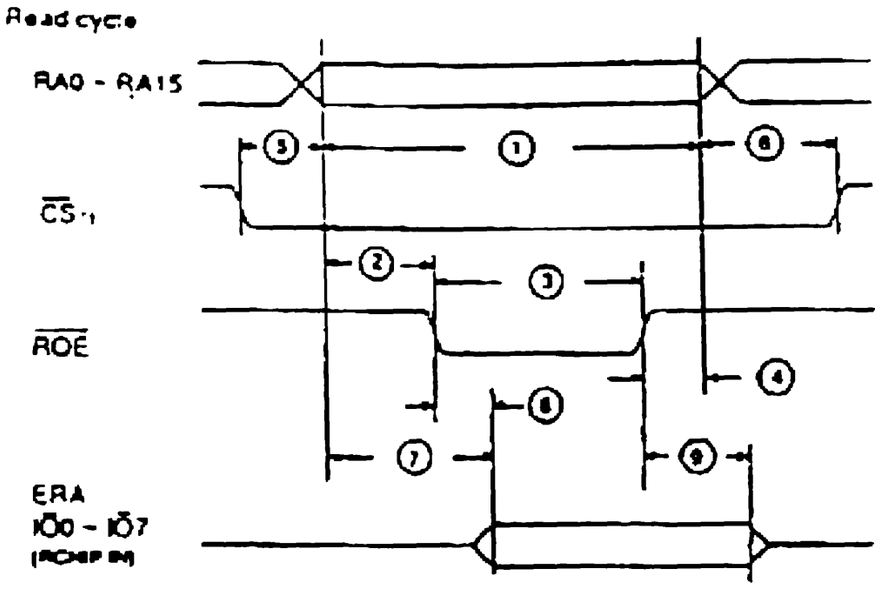
	Description	Rating min.	Rating max.	Unit
1	Read cycle	3T–30	—	
2		—	45	
3		—	55	
4	Data setup for $\overline{ROE}$	25	—	
5	Data hold for $\overline{ROE}$	0	—	
6		2T–50	—	
7		1T	—	ns
8		–10	—	
9		5	30	
10		2T–30	—	
11		1T	1T+30	
12		—	1T+35	
13		3T–30	—	
14		—	35	

Notes:

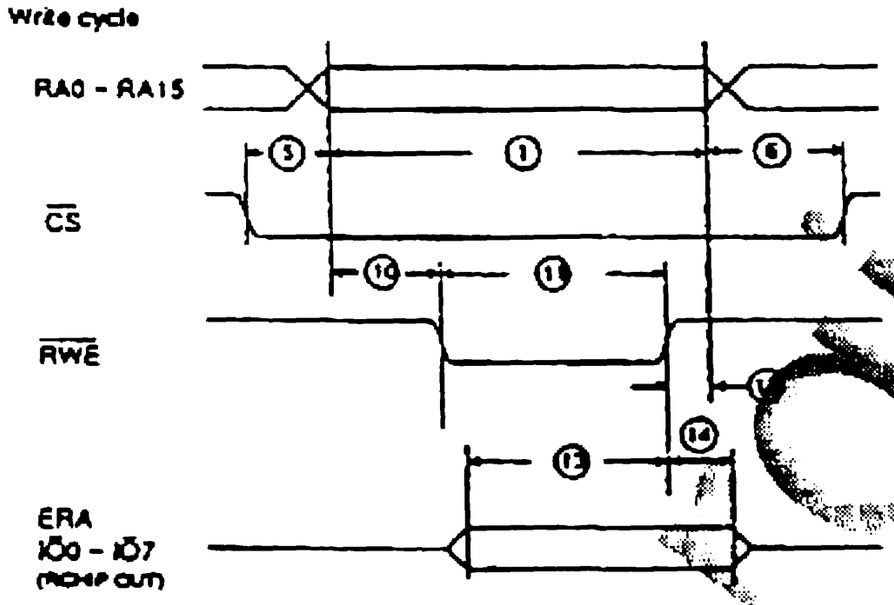
1. Pins IO1 - IO8 may be directly connected to SRAMs with access times of 100 ns or less; ERA, to SRAMs with access times of 100 ns or less.
2. $\overline{RWE}$ and $\overline{ROE}$ should not be LOW at the same time. $\overline{RWE}$ should go LOW 40 to 60 ns after $\overline{ROE}$.
3. T stands for one execution cycle—that is, the inverse of the crystal frequency. For most implementations it will be 60 ns.

Figure 43. LC8950 SRAM Interface (Preliminary)

Read cycle



Write cycle



See table 12 for AC characteristics.

	Description	Rating min.	Rating max.	Unit
1	Read/write cycle	3T-30	—	
2		35	—	
3		2T	—	
4		5	—	
5		1T	—	
6		2T-25	—	
7		—	130	ns
8		—	95	
9		0	—	
10		30	—	
11		2T	—	
12		10	—	
13		80	—	
14		5	—	

Notes

1. The LC8951 $\overline{CS}$ pin is PIN62, the LC8950 $\overline{DOUT}$ pin.
2. *t* stands for one execution cycle—that is, the inverse of the crystal frequency. For most implementations it will be 60 ns.
3. The numbers on these timing charts are totally independent of those on the preceding page.

Figure 44. LC8951 SRAM Interface

SANYO SEMICONDUCTORS

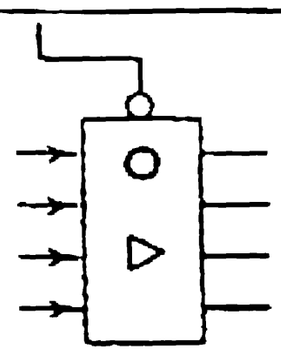
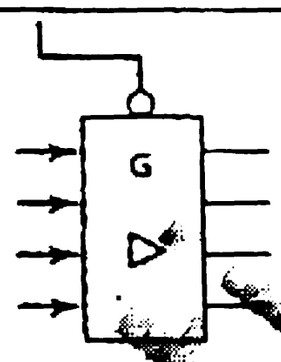
For technical information, please contact:

- JAPAN: SANYO ELECTRIC CO., LTD.
  Natsume Bldg., 18-6, 2-chome, Yushima, Bunkyo-ku, Tokyo 113 Japan
  Tel: 03(818)1144, Tlx: J26463 TKYSANYO, Cable Address: SEMICONSANYO, Fax: 03(818)1274
- HONG KONG: SANYO SEMICONDUCTOR (H.K.) CO., LTD.
  Room 612, Harbour Crystal Centre, 100 Granville Road, Tsimshatsui East, Kowloon, Hong Kong
  Tel: 3-3111198, Tlx: 56370 SSHK HX, Fax: 3-3110900
- HONG KONG: SHIN-NICHI ELECTRONICS DEVICE (H.K.) LTD.
  Room 1912-13 Park-in Commercial Centre, 56 Dundas Street, Kowloon, Hong Kong
  Tel: 3-7800359/0, 3-7701367/8, Tlx: 49652 SNEDVHX, Fax: 3-7802401
- SINGAPORE: SANYO SEMICONDUCTOR(S) PTE, LTD.
  Singapore Office, 112 Lavender Street, #03-00, Chuan Building, Singapore 1233
  Tel: 2926501, Tlx: SANSEMI RS23576, Fax: 2919677
- SINGAPORE: SHIN-NICHI ELECTRONICS (SINGAPORE) PTD. LTD.
  4 Leng Kee Road, #04-06 Thye Hong Centre, Singapore 0315
  Tel: 4722277, Tlx: RS20257 SNBSIN, Fax: 4735053
- KOREA: SANYO SEMICONDUCTOR (H.K.) CO., LTD.
  Seoul Branch, Room #1201 Samjung Bldg., 69-5, 2-KA, Taepyung-Ro, Chung-ku, Seoul, Korea
  Tel: Seoul 753-3415, 756-6337, 774-0298 ~ 0298, Tlx: SITUCON K22920, Fax: Seoul 752-9790
- TAIWAN: SANYO SEMICONDUCTOR TAIPEI CORPORATION
  Room 606, Chia Hsin Bldg., 96, Chung Shan Rd, N.SEC. 2, Taipei, Taiwan
  Tel: 02-551-5866, 02-541-6320, Fax: 02-541-7649
  Kaohsiung Office, Room 603, 61, Chung Cheng 3rd Road, Kaohsiung, Taiwan
  Tel: 07-201-5592, Fax: 07-201-5593
- TAIWAN: TONG SAN ELECTRIC CO., LTD.
  Room 406, No. 372 Lin Sen N. Rd., Taipei, Taiwan, R.O.C.
  Tel: 02-561-0381, Tlx: 23588 TONSANCO, Fax: 02-543-5431
- TAIWAN: O.S. SEMICONDUCTOR CO., LTD.
  Room No. 4, 4fl No. 200, Sung Chiang Road, Taipei, Taiwan R.O.C.
  Tel: 02-561-1247, Fax: 02-537-4084
- UNITED KINGDOM: SANYO SEMICONDUCTOR (EUROPE) GMBH
  UK Representative Office, The Marlborough Rooms, 68 High Street Weybridge, Surrey, KT13 8BL, U.K.
  Tel: 0932-855445, 855585, Tlx: 267470 SANYO G, Fax: 0932-855487
- DENMARK: LARS VALLENTIN ELECTRONICS APS
  Stenlese Center 18, DK-3660 Stenlese, Denmark
  Tel: 02-17 24 20, Tlx: 42547 VALLEK DK, Fax: 02-17 15 18
- F.R. GERMANY: SANYO SEMICONDUCTOR (EUROPE) GMBH
  Frankfurter STR. 7-9, 6236 Eschborn/TS, F.R. Germany
  Tel: (06196) 4-88-26, Tlx: 4072605 SSE D, Fax: (06196) 4-36-21
- F.R. GERMANY: HERIBERT LENNER GMBH
  Dorfstrasse 22, D-2351 Wiemersdorf, F.R. Germany
  Tel: 04192-4071, Tlx: 2180148 LEHN D, Fax: 04192-4031
- FRANCE: ERN
  237, Rue Fourny-Z A de BUC, 78530 BUC, France
  Tel: 39.56.00.11, Tlx: 698627 F
- PORTUGAL: NIPOSOM-J, NABAIS, LTDA.
  Rua Czsimiro Freire, 9-A 1900 Lisboa, Portugal
  Tel: 803738-896810, 887677, Tlx: 14028 NIP P, Cable Address: NIPOSOM
- U.S.A.: SANYO SEMICONDUCTOR CORPORATION
  New Jersey Office, 7 Pearl Court, Allendale NJ 07401, U.S.A.
  Tel: (201)825-8080, Tlx: 135138 SANYOSEMI ALND, Fax: (201)825-0193
  California office, 453 Ravendale Drive, Suite G, Mountain View, CA 94043, U.S.A.
  Tel: (415)960-8582, Fax: (415)960-8591
  Illinois Office, 415 W. Golf Road, Suite 60, Arlington Heights, IL60005, U.S.A.
  Tel: (708)364-7766, Fax: (708)364-7837
- BRAZIL: INDUSTRIA ELECTRONICA SANYO DO BRASIL LTDA.
  Rua Galvão Bueno, 430, Bairro Liberdade, São Paulo, S.P., CEP: 01506, Brazil
  Tel: 278-7111, 279-5511, Tlx: 1123255 ILSB BR, Fax: 1270-3938
- MEXICO: M.H. INTERNACIONAL
  Rio Guadalquivir 28-702, 06500 Mexico D.F., Mexico
  Tel: 514-4792, Tlx: 170991 INTIME

SANYO Electric Co., Ltd. Semiconductor Division

Specifications and information herein are subject to change without notice.

List of Errata

Contents		False	Truth	Page
Figure 3	SRAM	8K x (8 x 9*) SRAM	8K x 8 (x 9*) SRAM	4
	Bus Drivers (The 3rd from the top)			
Figure 4		Crystal frequency = 16.93444MHz	Crystal frequency = 16.9344MHz	5
Table 4	Symbol	V_I, V_O	V_I, V_O	20
	Test Conditions	Ta ≤ 70 deg. C	Ta ≤ 70 deg. C	
Figure 25		Stocking 4 Blocks	Caching 4 Blocks	85
Figure 27		Data Transfer (1 Block Stocking)	Data Transfer (1 Block Caching)	87
Figure 35		LIR channel →	L/R channel →	98

Sanyo LC8950 Real-Time Error Correction & Host Interface Processor

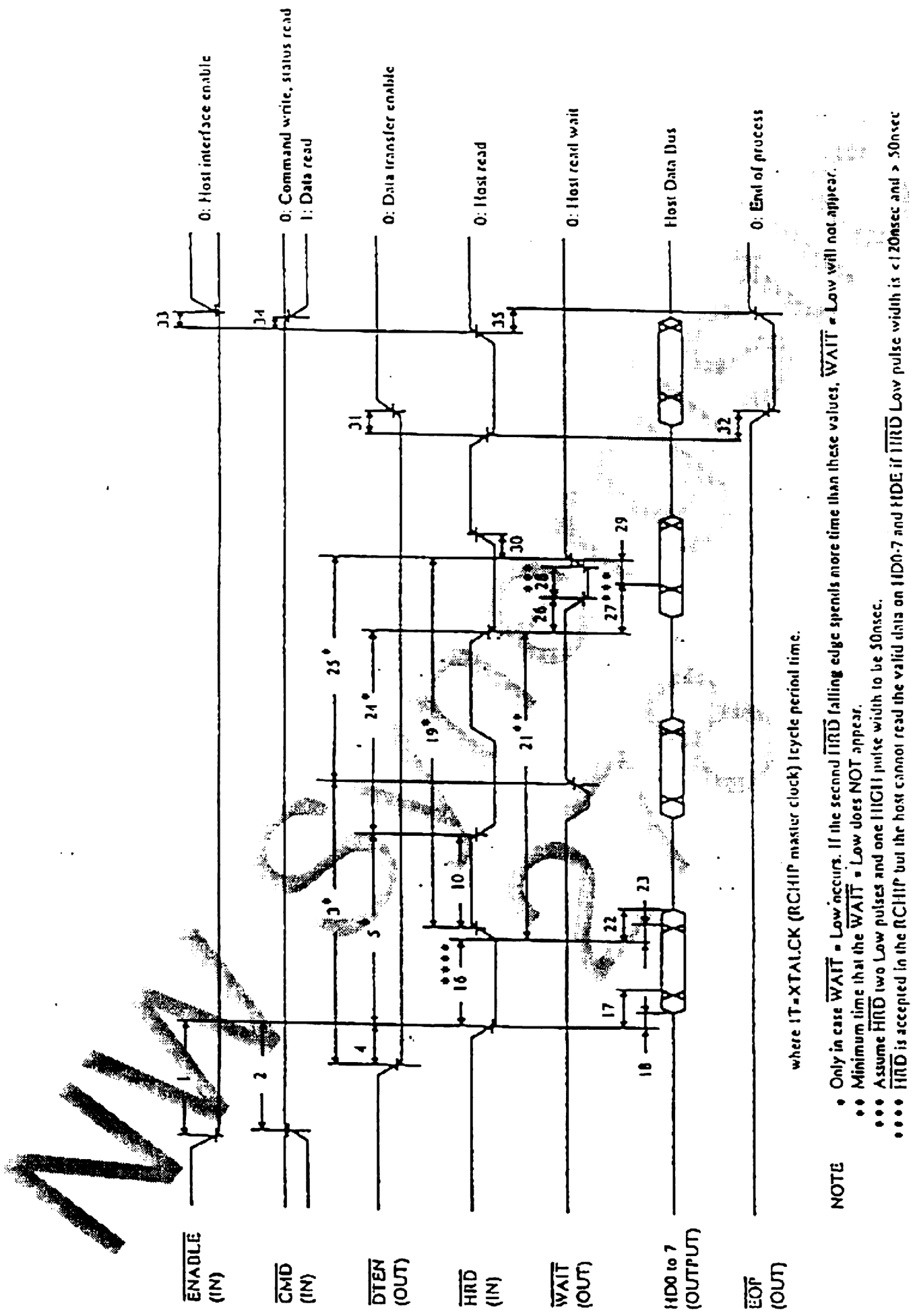


Figure 39. LC8950/LC8951 Data Read with WAIT Control

LC89510

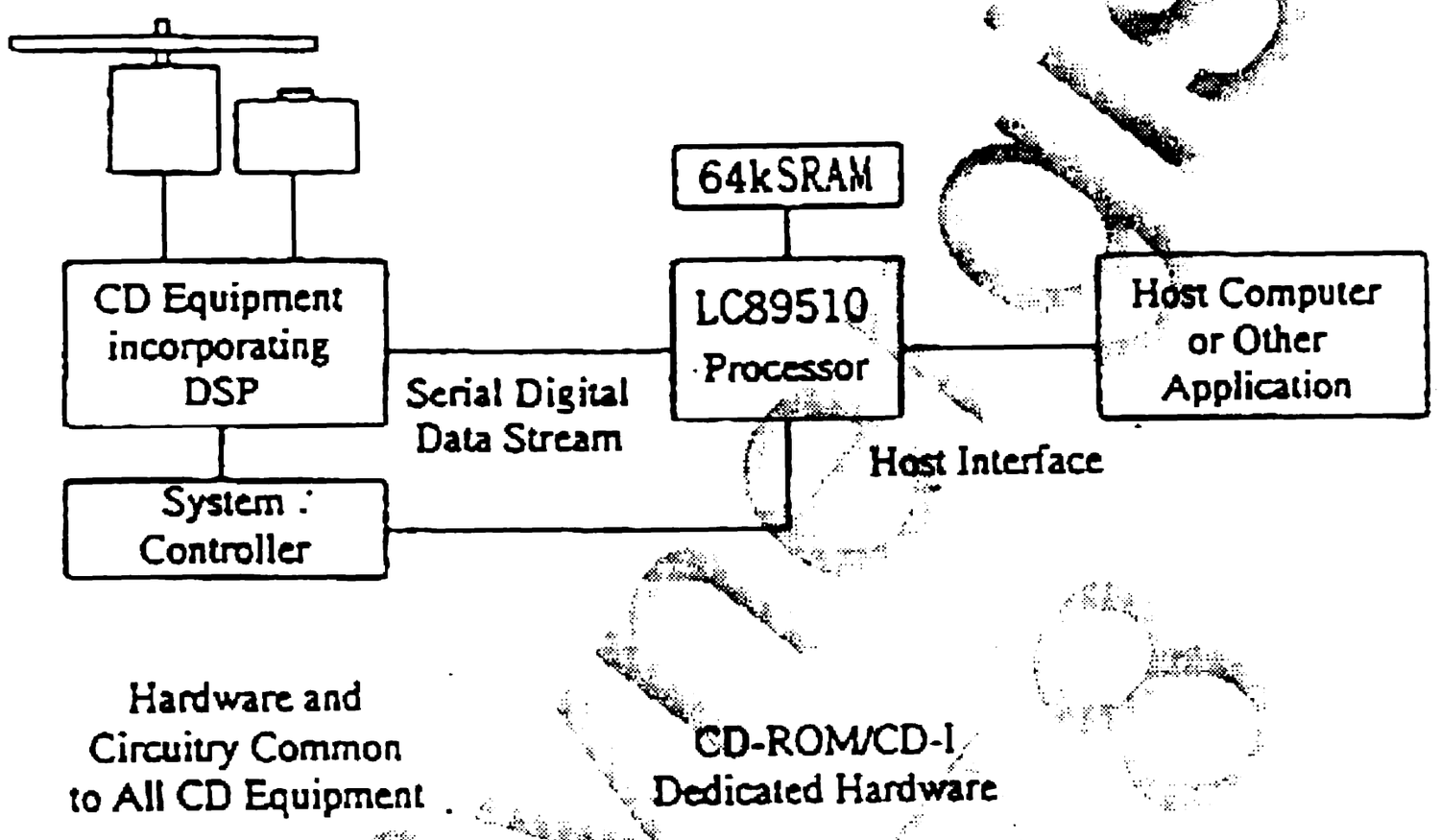
Error Correctt ion LSI for CD-ROM/CD-I

1. Overview

The LC89510 is an error correction LSI for CD-ROM/CD-I.
It is an advanced pin-compatible version of the LC8951.

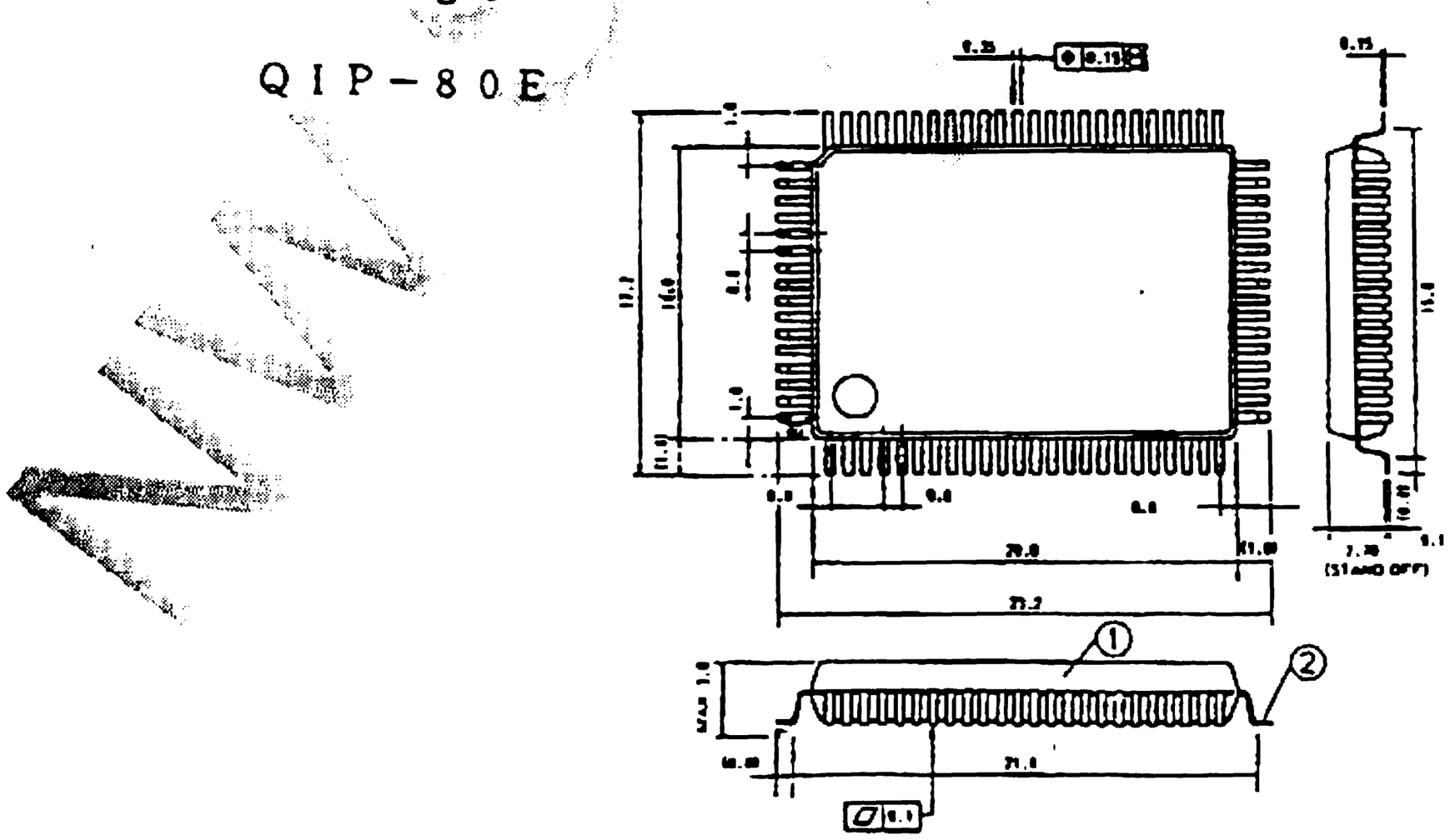
2. Function & Features

*On chip 12byte status FIFO
*On chip SRAM(8kbit) for erasure correction



3. Package

QIP-80E



PINOUTS

Pin Numbers, Names and Functions

Pin #	Pin Name	I/O	Function
1	GND	—	Ground
2	RA6	O	Data buffer RAM and erasure flag RAM address outputs
3	RA7	O	
4	RA8	O	
5	RA9	O	
6	RA10	O	
7	RA11	O	
8	RA12	O	
9	RA13	O	
10	RA14	O	
11	RA15	O	
12	$\overline{RWE}$	O	RAM write enable
13	GND	—	Ground
14	$\overline{ROE}$	O	RAM output enable
15	ERA	I/O	Erasure flag RAM data input/output
16	IO8	I/O	Data buffer RAM input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
17	IO7	I/O	
18	IO6	I/O	
19	IO5	I/O	
20	IO4	I/O	
21	IO3	I/O	
22	IO2	I/O	
23	IO1	I/O	
24	GND	—	Ground
25	XTALCK	I	Clock input
26	XTAL	O	Clock output
27	TEST1	I	Test inputs. Normally held LOW.
28	TEST2	I	
29	CSEL	I	Serial data clock phase select
30	LMSEL	I	Serial data byte order control (LSB/MSB first)

Pin #	Pin Name	I/O	Function
31	VDD	—	Positive power supply connection
32	LRCK	I	44.1 KHz left & right channel separator strobe
33	SDATA	I	Serial data input
34	BCK	I	Buffer input clock
35	C4LR	I	C2 error flag pointer strobe
36	C2PO	I	C2 flag pointer
37	MCK	O	Master clock output
38	D0	I/O	Controller data input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
39	D1	I/O	
40	D2	I/O	
41	GND	—	Ground
42	D3	I/O	Controller data input/output. (LC8951 has 20 kΩ pull-up resistors on-chip.)
43	D4	I/O	
44	D5	I/O	
45	D6	I/O	
46	D7	I/O	
47	RS	I	Register select
48	$\overline{RD}$	I	Controller data read
49	$\overline{WR}$	I	Controller data write
50	$\overline{CS}$	I	Controller chip select
51	$\overline{INT}$	O	Controller interrupt
52	GND	—	Ground
53	$\overline{RESET}$	I	LC8950 chip reset
54	$\overline{ENABLE}$	I	Host interface enable
55	$\overline{HWR}$	I	Host data write input
56	$\overline{HRD}$	I	Host data read input
57	$\overline{CMD}$	I	Host command/data select
58	$\overline{WAIT}$	O	Data transfer WAIT signal/DRQ signal
59	$\overline{DTEN}$	O	Data enable output
60	$\overline{STEN}$	O	Status enable output

Pin #	Pin Name	I/O	Function
61	$\overline{EOP}$	O	End-of-process flag output
62	$\overline{DOUT}$ ($\overline{RCS}$)	O	Host computer data buffer control output (LC8351 RAM chip select)
63	$\overline{HDE}$	O	Host data erasure flag tristate output
64	GND	—	Ground
65	HD7	I/O	Host data input/output
66	HD6	I/O	
67	HD5	I/O	
68	HD4	I/O	
69	HD3	I/O	
70	HD2	I/O	
71	HD1	I/O	
72	HD0	I/O	
73	VDD	—	Positive power supply connection
74	$\overline{SELDRQ}$	I	Data transfer mode select (WAIT control/DRQ control)
75	RA0	O	Data buffer RAM and erasure flag RAM address outputs
76	RA1	O	
77	RA2	O	
78	RA3	O	
79	RA4	O	
80	RA5	O	

Additional Registers Description

[R14] CTRL2 (Control2)

ERAMSL(Erasure RAM select)

0 : On-chip RAM used

1 : On-chip RAM not used

(0 is set at RESET)

STENCTL(STEN CONTROL)

0 : ZSTEN output LOW at that first STATS data is set by micon.

1 : ZSTEN output LOW at that STENTRG register is set to 0.

STENTRG(STEN TRIGGER)

0 : ZSTEN output LOW at that STENTRG register is set to 0.

This register is set to 1 automatically at that HOST-CPU read last byte.

This register is valid at STENCTL=1.

= WRITE =

RS	AR	No.	記号	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0	----	---	AR					A_3	A_2	A_1	A_0
1	0000	R0	SBOUT	msb	~	~	~	~	~	~	lsb
	0001	R1	IFCTRL	CMDIEN	DTEIEN	DECIEN	/CMDBK	/DTWAI	/STWAI	DOUTEN	SOUTEN
	0010	R2	DBCL	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
	0011	R3	DBCH					B_{11}	B_{10}	B_9	B_8
	0100	R4	DACL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	0101	R5	DACH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	0110	R6	DTTRG								
	0111	R7	DTACK								
	1000	R8	WAL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1001	R9	WAH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1010	R10	CTRL0	DECEN		EDIRQ	AUTORQ	ERAMRQ	WRRQ	ORQ	PRQ
	1011	R11	CTRL1	SYIEN	SYDEN	DSCREN	COWREN	MODRQ	FORMRQ	MSCKRQ	SHDREN
	1100	R12	PTL	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	1101	R13	PTH	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
	1110	R14	CTRL2						ERAMSL	STENCTL	STENTRG
	1111	R15	RESET								

note : ☐ Don't care