SWIM3
ERS

V1.2

Steve Smith
Matthew Nelson

Change History:

Rev1. Initial release.
Rev1.2 Second silicon changes.

Introduction

SWIM3 is the logical extension to SWIM2. By adding several new features to the existing design the intent is to relieve the system processor of most of the babysitting tasks required to talk to the floppy disk. Most importantly it will allow interrupts to remain enabled during disk accesses. Only the minimum amount of extra hardware consistent with this goal is added with the intent that the total design be five to six thousand gates (Verilog based, vendor independent).

It is assumed that the SWIM3 is connected to the host system through a DMA channel. This interface is patterned after the industry standard 765 so that any machines currently designed to use the New Age floppy interface could easily be adapted to SWIM3.

Block Description

The two new tasks required of SWIM3 are to manage the head motion to get to the correct track and to transfer read or write data through the DMA channel once the correct place on the disk is located.

Head positioning

To simplify stepping the head to the desired position, a register is provided (the Step Register) that is loaded with the number of tracks desired to be moved. As with SWIM2 the drive is enabled, direction of stepping set, and the step command address is placed on the phase lines by the system software. When the go_step bit is set SWIM3 will monitor the /step status from the drive and pulse phase3 to cause a step, waiting 80us between steps as required, at the same time decrementing the Step Register. This process is repeated until the Step Register is zero. This generates an interrupt (step_done). Depending on the number of tracks moved and motor speed zones crossed a timeout is required before accessing the drive after stepping. This is left to the software.

Read Control

After the head is properly positioned and the motor is up to speed we must first locate the position within the track. When the action bit is set the SWIM2 circuits begin searching for mark bytes following the bytes of zeros. If found, a new circuit determines if we are reading an ID field. If so, the current head, track, and sector positions are stored in two registers and the last_ID_valid bit is set true if CRC checks good on address mark. (only in mfm mode, in gcr mode the checksum is checked on an address mark) This bit is false on startup or after the next address mark is started to be read. (Software might need to request sector+2 to be sure a complete revolution is not taken) The two position registers are constantly updated while in read mode as long as go is true. An interrupt (if enabled) (ID_read) is generated by each ID read . If we didn't find an ID field the search is restarted. This process will continue forever as long as the go bit is set.

Once valid position data is obtained, software should determine the nearest sector to read or write. The number of this sector is placed in the first_sector register but only after the current position is known (for best performance). On reset this register is set to $FF which means don't match any sector. When SWIM3 reads a sector ID that matches the contents of the first_sector register, it will then look for the data field mark bytes and when found start a DMA read transfer consisting of the number of bytes defined by the mode . In MFM mode the read transfer starts after the mark bytes for 512 bytes. In GCR mode read transfer starts at the sector byte for 704 bytes. After all the data bytes are transfered a number of DMA requests will be requested equal to the value of the Gap register. The purpose of this is to move the DMA pointer in order to leave a track image in memory which later can be re-written to the drive. The gaps will be filled with format bytes that are required to be re-written.

A second register called the sectors_to_xfer register is written by software at the same time the first_sector register is written. At the end of a sector transaction if the sectors_to_xfer register is non-zero it is decremented. This continues until all desired sectors are read . The sectors_done interrupt signals the end of the last complete sector transaction.

Writing

A sector write proceeds just like a read except the DMA transfer starts in the gap between the ID field and the data field. A special protocol for data transmission over the DMA channel uses an escape code $99. When SWIM3 sees a $99 in the data flow the next byte is a command. The following is a table of these commands.

Code	Command

$99	Transfer data $99 (no command)
$A1	Write a $A1 mark byte
$C2	Write a $C2 mark byte
$04	Write both CRC bytes
$0F	Turn off escaping for 512 bytes
$08	End data (terminate dma transfer)

After writting starts data will be written until a $99 $08 is encountered.

Formatting

Formatting and track write as required by GCR with the 2.8M drive are handled by a special case. The phase register is overridden by SWIM3 so that the index mark is being read from the drive. A special bit called format must be set true which will cause SWIM3 to begin DMA write requests once the index pulse is seen. Writing continues just as always until terminated by the $99 $08 command, since escape commanding is active in this mode as well.

Error Handling

SWIM2 generated errors when the FIFO was under or overflowed. Errors could also occur on reads when data from the drive was in some way corrupted. Because the DMA channel has replaced the bus interface the FIFO errors will only occur if the DMA controller fails to service SWIM3 DMA requests. If this happens the pending transaction will terminate and the appropriate FIFO error will be set. A new error has been added for CRC errors. If a CRC error is detected while reading a requested data field the CRC derror bit will be set. Any error will cause a multiple sector request to be terminated. This should be checked when the sectors_done interrupt is received. A CRC error during an address mark read will cause the current sector register not to be updated and the last_id_valid_bit is set false. No transaction will start as the result of reading an address mark that matches the first_sector register if a CRC error (or checksum error in gcr) occurs during the address mark read.

GCR Conversion

The gcr conversion is built into the hardware both on read or write. There are a few special cases when writting. To write any data pattern that is not in the standard conversion table just write what the pattern should be with the high bit set. Example, D5, AA, 3F. Otherwise write the non-encoded value. For example the bit slip bytes, FF,3F,CF,F3,FC,FF, should be written as 3F,BF,1E,34,3C,3F.

Register Description

$0 Data Read or write data to or from FIFO. Selected by DMAack input regardless of values on address bus.

$1 Timer A 1us timer register. Values loaded into this register will decrement at a 1us rate. The first count after the load can occur at any time but all subsequent counts will be 1us apart. For example a load of 2 will timeout in >1us but <2us. The count rate is independent of the setting of the clock divide bit. An interrupt is generated when the count equals zero.

$2 Error Indicates the type of error that has occurred. Cleared on a reset or on read. Only one error can be set at a time. Must be cleared prior to a read or write. If any of these bits are set the error flag in the Handshake register will be set. Errors on reads only function in MFM mode and after the mark byte is found.

Data 0 Underrun FIFO. FIFO empty while writing or full and not read during a read.

Data1 Not used. (Was mark byte read from data register)

Data 2 Overrun FIFO. FIFO written while full in write mode or read while empty in read mode.

Data 3 Not used

Data 4 Not used

Data 5 Not used

Data 6 CRC error on address mark

Data 7 CRC error on data field

$3 Parameter Data The parameter ram has been reduced to two nibbles of data, early time and late time. The data is stored as {late time,early time}. The nominal, no precomp value is $7.

$4 Phase Register Data bits 0-3 represent phase lines 0-3. Data bits 4-

7 are not used.

$5 Setup Register This register sets the various modes of operation. It is reset to all zeros except as noted.

Data bit 0 =0 don't invert wrdata (neg pulses)
=1 invert wrdata (pos pulses)

Data bit 1 =1 Copy protection mode
Causes a change in operation to support copy protection. When set, all data from the next mark byte will be transfered, not stopping until the go bit is set false. The transfered data will consist of two bytes per byte of read data. The first byte is $00 unless it was a mark byte in which case a $80 is transfered. The second byte is the data.

Data bit 2 = 0 MFM mode; =1 GCR mode.

Data bit 3 = 0 normal; =1 clock divided by two.
Note the clock to SWIM2 may be different than SWIM. If supplied a 16M clock SWIM2 will read and write 2,3,4 us MFM cells and 2,4,6 us GCR cells with this bit set to zero. With a 32M clock input SWIM2 will read and write 1,1.5,2 us MFM cells, and should have this bit set to one to generate 2,3,4 us MFM cells and 2,4,6 us GCR cells.

Data bit 4 Disable GCR conversion. If set to a one disables the conversion tables in GCR mode for copy protection.

Data bit 5 0 = Apple data mode; 1 = IBM mode.
(0=NRZ data , 1= RZ data.)

Data bit 6 =0 MFM writes; =1 GCR data writes.

Data bit 7

$7 Handshake Register Read only

Data bit 0 Mark Next byte in FIFO = Mark byte.

Data bit 1	Interupt pending. If =0 at least one interupt bit is set.	
Data bit 2	RDData	Direct read of drive data
Data bit 3	Sense	Direct read of rddata input
Data bit 4	=0	was motor still on
Data bit 5	Error. A bit in the Error register is set.	
Data bit 6	Dat2bytes	FIFO empty in write or full in read when = 1.
Data bit 7	Dat1byte	FIFO has at least one byte in it.

This signal is gated with error in write mode so that if a write error occurs the SWIM will appear empty so to not cause the software to hang.

$6 &$7 Mode Register Write only

Zeros @ $6, ones @ $7
The Mode register is controlled bit by bit by writing to either the write zeros or write ones location with the bits that are being modified set to one. To make bit 0 a zero write 00000001 to location $6, to make it a one write 00000001 to location $7. Reset sets all bits to zero.

Data bit 0	Enable interupts 1 = enabled	
Data bit 1	Enable1	1 = enable drive 1
Data bit 2	Enable2	1 = enable drive 2
Data bit 3	Go	1 = Go active (was action)
Data bit 4	Write	1 = write mode; 0 = read.
Data bit 5	Side select	1 = side 1; 0 = side 0
Data bit 6	Format mode (new)	
Data bit 7	Go_step	1 = Start stepping to desired

track

$6 Read Mode register Read only

Reads contents of mode register.

$8 Interrupt register Read (Cleared by read)

Bit 0	Timer_done	1 = Timer count=0
Bit 1	Step_done	1 = Stepping complete
Bit 2	ID_read	1 = Just read address header
Bit 3	Sectors_done	1 = Just finished reading last byte of a sector(read mode) or just got the end_data command(write_mode) or process terminated by error.
Bit 4	Sense change	1 = (Sense line just changed) Useful for generating an interrupt when a disk is inserted or when waiting for ready/ to signal the drive is ready, etc.

$9 Step register Load this register with the number of steps needed to be moved after the direction is set in the drive and the step address is placed on the phase lines. This register is automatically decremented by the hardware after each step command is sent to the drive and an interrupt (step_done) is generated when the last step is taken. No step commands are sent until the go_step bit is set. An 80us timeout is made between every step. The step_done interrupt is set under this condition as well.

$A Current track register The 7 LSB's of this register holds the last track ID read from the drive. Bit 8 of this register is the last head ID read from the drive. Resets to $ff.

$B Current sector register The 7 LSB's of this register holds the last sector ID read from the drive. This value may be stale as indicated by the Last_ID _valid bit. Bit 8 of this register holds the Last_ID_valid bit. This bit is set after reading a sector id when either the crc is correct or the checksum is ok (gcr). It is cleared during the data field header or when go is turned off. The register resets to $7f.

$C Gap/Format register This register should be loaded with the number of “pad” bytes to be transfered after the data bytes in a multiple sector read operation. When read back it contains the format byte.

$D First sector The first 6 bits of this register should be loaded with the first sector number desired to be read or written. The transfers will start when this sector is found. If bit 6 is set to a one (x1xxxxxx) any sector will match and be transfered. Resets to $ff.

$E Sectors to xfer This register is loaded with the number of sectors desired to be accessed continuously. After each sector the hardware will decrement this value until it reaches zero. The number of untransfered sectors will be retained here after an error has occurred. Resets to $00.

$F Interrupt mask register Masks interrupts in register $8 bit for bit. Setting a one enables each bit, a zero disables a bit.

Pin List

Data [7:0]	Bi-directional	
Addr [3:0]	Input	
AS/	Input	
Dev	Input	
C32M	Input	CPU Side
RD/	Input	
WR/	Input	
DMAreq/	Output	
DMAack/	Input	
Reset/	Input	
IntReq/	Output	

Rddata	Input	
Wrdata	Output	
Enable1	Output	
Enable2	Output	Drive Side
Wreq	Output	
Phase[3:0]	Output	
Hedsel	Output	

Technical Specification

Write cell times***

1uS		0.989**			uS
1.5uS	1.499**			uS	
2uS	1.979*		2.010**		uS
3uS		2.999		uS	
4uS	3.989**		4.021*		uS
6uS		5.999		uS	

* Written with 15.667 MHz clock
** Written with 31.334 MHz clock
***Write times can be added to or subtracted from by setting the write pre-comp register in one clock resolution.

Write pulse width

The write pulse in MFM mode shall be five clock periods long.

Read cell times

	16MHz	32MHz
1uS		.734-1.245 uS
1.5uS		1.277-1.723 us
2uS		1.755-2.266 us
2uS	1.468-2.489 uS	
3uS	2.553-3.447 uS	
4uS	3.510-4.532 uS	
2uS	0.957-2.999 uS	
4uS	3.064-4.979 uS	
6uS	5.042-7.021 uS	

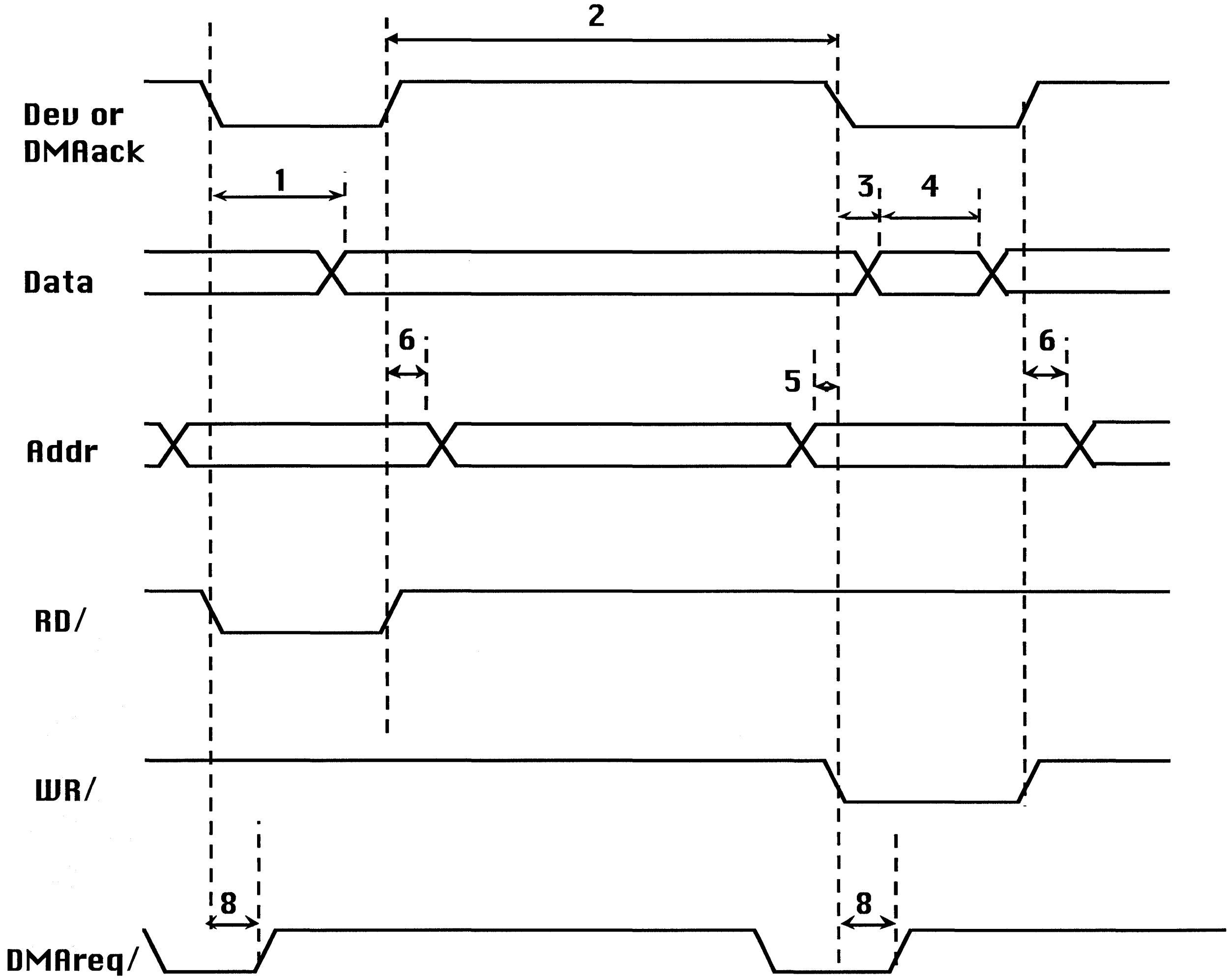
Note: Gaps between adjacent read cell boundries represent areas of uncertainty which may decode as either possible cell.

DC Specification @ Vdd = 4.75 to 5.25V Temp = 0 to 70C

Parameter	Min	Max	Unit
Supply current		50	mA
Input low level		0.8	V
Input high level	2.0		V
Output high level @ I=3.2 mA	2.4		V
Output low level @ I=3.2 mA except the following: Wrreq/ I=10 mA Phase1 I=10 mA Enable1/ or 2/ I=5 mA		0.4	V
Input leakage	-10	10	uA
Output leakage	-10	10	uA
Pullup R Rddata, Sense	5	20	KΩ

AC Specification

Parameter	Min	Max	Unit
Clockin	0	32	MHz
Duty Cycle	40	60	%
Rise and fall	0	10	nS
Clock rise to output (Wrdata,Wrreq/,Dat1byte)	0	25	nS
Dev/ rise to output (Phase,Hedsel,Enabl1/ 2/, Motoen/,3.5Sel/)	0	tbd	nS
Async in to Dev/ fall setup	0	tbd	nS



Read cycle

Write cycle

Timing:

1. Dev or RD/ low to data valid read. 95ns
2. Dev high min. 70ns
3. Data valid after assertion of write 15ns
4. Data valid time 200ns
5. Address setup to Dev low 15ns
6. Addr hold 0ns
8. Dmaack low to req/ invalid 10ns

Pin Description

D0-D7	The bi-directional CPU data bus
A0-A3	Address inputs for register select
RD/	Bus read control input
WR/	Bus write control input
Dev/	Device select input
Q3	OR’ed with Dev/ (input)
Reset/	Hardware reset input
Wrdata	Write data output to disk
Wrreq/	Write enable output to disk
Motoen/	Motor on indication output
Enabl1/	Drive enable output to disk
Enabl2/	Drive enable output to disk
Sense	Readable input used to read disk status
Rddata	Data input from drive
Clockin	Input clock to SWIM2
Phase0-3	Control signals to disk
Hedsel	Head select output to disk
DMAreq	DMA request from SWIM3
DMAack	DMA trsnsfer acknowledge to SWIM3