ST-2900 FDC

Assembly Instructions
and
User's Manual

Sardis Technologies
2261 East 11th Avenue
Vancouver, B.C.
Canada V5N 1Z7

Although much effort has been made to ensure the accuracy of the hardware, software and documentation, David C. Wiens and Sardis Technologies disclaim any and all liability for consequential damages, economic loss, or any other injury arising from or on account of the use of, possession of, defect in, or failure of the supplied material.

This manual last revised September 7, 1984. (and May 14, 1986)

ST-2900 FDC Board Assembly Instructions

[] If you purchased a partially assembled board, skip this section and proceed to the section "Getting Started ...".

[] Read Appendix A and decide on which configuration you want.

[] Carefully check both sides of the board visually for any incomplete etching, foil "bridges" or "breaks".

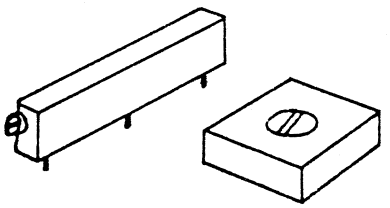
[] Note -- whenever the component markings on the board are not entirely clear, refer to the component layout drawing elsewhere in this manual.

[] Use an ohmmeter between pins 1 and 2 of connector S1 to verify that there are no shorts between ground and +5vdc traces.

[] If you want the drive motors to remain "off" immediately after powerup and/or reset, then cut the trace on the solder side of the board between pin 3 of U1 and the +5 vdc bus, and also cut the trace on the component side of the board between pins 3 and 10 of U1.

[] Install and solder all IC sockets. If they have a notch or indentation at one end, orient them to match the "half circle" marking at the end of each IC outline on the board.

[] Install and solder all resistors. Note that provisions have been made to allow R1 and R2 to be either of two types of packages, although the silk-screen markings only show the placement of the first style. The dotted outlines in the component layout drawing shows the placement of the second style.



[] Measure the resistance between pins 15 and 16 at U1 and adjust R2 until you get a reading of at least 100K ohms.

[] Measure the resistance between pins 7 and 16 at U1 and adjust R1 until you get a reading of at least 80K ohms.

[] Install and solder all capacitors. Be especially careful with C3, C4, C5, C13, C14 -- the "+" marking on the PC board indicates the side where the positive lead should go. A tantalum capacitor installed backwards can explode. The "C3" marking might be partially covered by R2.

[] Install and solder connectors P1 and P2 and P3. If they are a tight fit, use a small screwdriver to force them. Be careful -- the posts of some connectors have very sharp points.

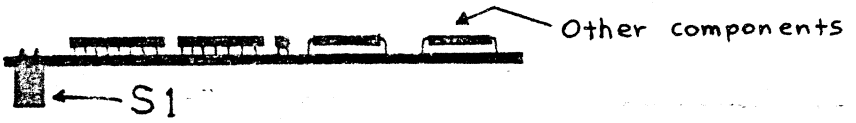
[] Install and solder the 3 pin header strips for jumpers J2, J3, J4, J5, J6, J7, J8, J14.

[] Install and solder the 2 pin header strips for jumpers J1, J9, J10, J11, J12, J13.

[] If you cut the two traces in step 6 above, solder a jumper wire between pins 10 and 16 of U1, and another one between pin 3 of U1 and pin 19 of U3.

[] Install and solder test points TP1 and TP2. These can be made by bending a length of bare wire into a loop:

[] Install and solder connector S1. NOTE -- this connector is inserted into the back side of the board and soldered on the top side of the board. With so many pins, it might take a bit of wiggling to force the connector into place.



[] Install and solder crystal Y1. To be on the safe side, use a piece of electrician's tape to insulate the crystal case from the PC board. Be careful not to overheat the crystal while soldering.

[] Double check your work, especially looking for missed or improperly soldered joints or accidental solder bridges.

[] You can now proceed to the section entitled "Getting Started -- Partially Assembled FDC Board".

Getting Started -- Partially Assembled FDC Board

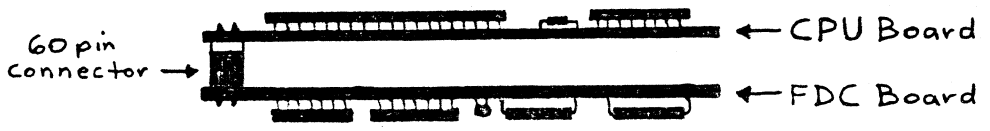
[] NOTE -- static electricity can damage MOS integrated circuits. Even a static electricity charge too weak to feel can cause problems, and the damage, if it merely weakened the chip, might not cause the chip to fail until weeks or months later. While handling any MOS IC's you should ground your body and all tools that will touch the IC leads. Use a 1 Megohm resistor in series between you and ground to protect yourself against dangerous shocks. Also, handle the assembled board with care -- just because the IC's have been installed in their sockets doesn't make them immune to static electricity. Better safe than sorry.

[] Before you proceed with the FDC board, you should have the CPU board up and running and tested.

[] Put shorting blocks on the jumpers. See Appendix B for a description of all jumpers.

[] Install all the IC's into their sockets. Be careful to plug them in the right way. Double check against the component layout drawing.

[] Make sure the power to the CPU board is turned off, then plug the FDC board into the CPU board. Since there are 60 pins in the connector, considerable pressure will be required.



[] Power the system up and verify that the CPU board still works after the addition of the FDC board.

[] Try writing various values (using ST-MON's "M" command) into location $FF01, then reading them back to see if the 1793 chip is working. Do the same for the 6522 chip using location $FF43.

[] Turn the power off. Connect disk drive #0 as per instructions in Appendix D.

[] Turn on the power to the drives and ST-2900 boards.

[] Make sure the ST-2900 FLEX Conversion disk is write protected (has the write protect notch covered), then insert it into drive 0. Execute the "D F" command. The drive motor should turn on, and two seconds later the message "Put FLEX disk in drive 0" should appear on the console. Follow the instructions in the "ST-2900 FLEX Conversion Package" manual.

(use the ST-2900 OS-9 Conversion Boot disk and the "D OC" command instead, if using the OS-9 Conversion package)

Theory of Operation

Most of the FDC board is involved in implementing the floppy disk controller.

The 1793 (U3) handles most of the interfacing between the microprocessor and the disk drives, including issuing the step in / step out signals. The 9229 (U6) chip provides support facilities such as extracting a clock signal from the read data and modifying the write data signal. A few selected pages from SMC's data sheet for the 9229 are included in this manual.

U7 decodes the drive select lines and deselects all drives when the head load signal goes false.

Part of U1 turns the drive motors on whenever the 1793 is accessed (if any of its registers are read), then leaves the motors on until a specified duration after the last access. This duration is adjustable (via R2) from approx. 1 1/2 - 15 seconds.

The other half of U1 is set (via R1) to the length of time it takes for the motors to come up to speed (1/10 - 1 sec.). This signal is used to synthesize the "ready" signal for the 1793 that is "true" only when the drive motors are assumed to be on and up to speed.

Integrated circuits U8, U9, U10 buffer the signals to/from the disk drives.

Integrated circuit U2 (6522) provides two 8 bit parallel ports (unbuffered) plus two 16 bit counter/timers. A few selected pages of Synertek's data sheet are included in this manual.

Trouble-shooting Hints

The first step is to double check the settings of jumpers J1-J14. Refer to the component layout diagram for the orientation of J10/J11/J12.

The second step is to double check and triple check that all components are in the right locations, and that they are oriented properly. Also make sure that the pins of all integrated circuits were correctly inserted into the sockets; sometimes a pin gets bent underneath the IC body.

If everything looks OK so far, carefully inspect all of the solder joints on the board, looking for missed connections, "cold" solder joints, and solder "bridges".

Use a voltmeter to measure the +5 volt (and +12v if used) supplies at each chip -- use the schematic as a guide. Test points TP1 and TP2 provide a convenient connection to ground. The +5 volt supplies should be between +4.75 and +5.25 vdc, the +12 volt supply between +11.4 and +12.6 vdc.

Check that the ribbon cable between connector P3 and the disk drives has been correctly hooked up.

Check the disk drive's spindle motor speed via the built-in strobe disc, and adjust the speed if necessary. Executing ST-MON's "M" command to read location $FF01 will start the motor.

If you still can't get the FDC board working, you will need access to an oscilloscope.

Verify that an 8 MHz clock signal appears at pin 11 of U6, and a 1 MHz clock signal at pin 24 of U3.

If the 6809E and 6522 are both 1 MHz parts that just barely meet their specs, then the 6522 is approx. 30 nsec. too slow in read mode (delay between the rising edge of E and valid data appearing on the data bus is too long). Using a 2 MHz 6522A would solve this problem. However, this is a worst case scenario, and unlikely.

If the 8 MHz clock signal isn't OK, try using a different 74HC14 chip, or try different values (0-33pF) for C6/C7.

Parts List - FDC Board

*	U1	74LS123 Dual Retriggerable Monostable Multivibrator
*	U2	6522 VIA (Versatile Interface Adapter)
*	U3	1793 Floppy disk controller
*	U4	74LS02 Quad 2 input NOR
*	U5	74HC14 CMOS hex inverter Schmitt trigger (do NOT substitute with 74LS14) (avoid SPI brand, prefer Texas Instr.)
*	U6	9229 (Standard Microsystems Corp FDC9229T or FDC9229BT
*	U7	74LS42 BCD to decimal decoder (1 of 10)
*	U8	74367 Hex buffer/driver (3 state) (or 74LS367)
*	U9	7406 Hex inverter buffer/driver (open collector)
*	U10	7407 Hex buffer/driver (open collector)
	Y1	8 MHz crystal (HC-18 package)
t*	R1	100K ohm trimpot
t*	R2	200K ohm trimpot
	R3	10K ohm 1/4 watt resistor
	R4	22K " " " "
	R5	150 " " " "
	R6	1K " " " "
	R7	1K " " " "
	R8	2 Meg " " " "
	R9	10K " " " "
	R10-R13	150 " " " "
	C1,C2	.1uF or .15uF bypass capacitors
*	C3	100uF electrolytic capacitor 6.3vdc
*	C4	22uF " " "
*	C5	47uF " " "
	C6,C7	10pF capacitors
	C8-C12	.1uF or .15uF bypass capacitors
*	C13	10uF tantalum capacitor 6.3vdc
t*	C14	2.2uF " " 16vdc
*		4 - 14 pin IC sockets
*		3 - 16 pin IC sockets
*		1 - 20 pin IC socket
*		2 - 40 pin IC sockets
	P1,P2	14 pin (2 row x 7) .100" center male header with .025" square posts (eg. can be cut from 72 pin strip such as AP Products 929836-01-36-R)
	P3	34 pin (2 row x 17) .100" center male header
*	S1	60 pin (2 row x 30) .100" center female header (eg. can be cut from AP Products 929975-01-36-R)
	J2-J8	3 pin .100" center male header (eg. can be cut from 36 pin strip such as AP Products 929834-01-36-R)
t	J9	2 pin .100" center male header
	J10-J13	2 pin " " " "
	J14	3 pin " " " "

* these components must be oriented as marked on PCB

t see Appendix A for optional configurations

(Note - AP Products are also known as Aptronics. AMP, Molex, etc. also make similar connectors)

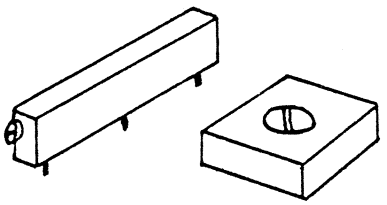
APPENDIX A - Optional Configurations

1) Single Supply 1793

If the 1793 (or equivalent) you are using requires only +5vdc (ie. it doesn't use +12vdc), then you can omit C14 and J9.

2) Trimpot choices

Provisions have been made on the FDC board to allow more than one style of trimpot for R1 and R2:



APPENDIX B - Jumpers

J1 - always shorted for currently supported configurations

J2 - short center pin to "I" pin for current configurations

J3 - short center pin to "I" pin for current configurations

J4 - short center pin to "I" pin to allow 6522 to generate IRQ interrupts. This setting is used by current software.
- short center pin to "F" pin to allow 6522 to generate FIRQ

J5 - short center pin to "N" pin if ALL drives are without head load solenoids
- short center pin to "T" pin if one or more drives have head load solenoids

J6/J7/J8 - selects the desired precompensation delay:

J6	J7	J8	precomp
0	0	0	0 nsec.
0	0	1	125 "
0	1	0	250 "
0	1	1	375 "
1	0	0	500 "
1	0	1	500 "
1	1	0	625 "
1	1	1	625 "

J9 - short if the 1793 chip installed uses the +12vdc supply
- leave open if the 1793 chip is single supply (+5v only)

J10/J11/J12 - determines which line in the 34 conductor ribbon cable is used for drive 3 select. Only one of these three jumpers should be shorted.
J10 = pin 2
J11 = pin 4
J12 = pin 6 (used by most drives)

J13 - when shorted, connects PB6 with PB7 of the 6522. Normally left open

J14 - always short center pin to "S" pin for currently supported configurations

APPENDIX C - Memory Map

When J2 and J3 on the ST-2900 CPU board are connected as recommended:

a) 1793 registers are addressed at $FF00-$FF03 (but occupy all of $FF00-$FF1F)

b) 6522 registers are addressed at $FF40-$FF4F (but occupy all of $FF40-$FF5F) Refer to the 6522 data sheet.

APPENDIX D - Selecting and Connecting Floppy Disk Drives

Selection

Any disk drive that has the standard 34 wire 5 1/4" floppy disk drive interface and runs at 300 rpm should be compatible with the ST-2900 FDC board. The hardware and software will handle 1 to 4 drives, with any mix of single or double sided, with or without head load solenoid, 35, 40 or 80 tracks, and 6 to 30 msec. step rates. This includes the new Shugart SA300 3 1/2" microfloppy. (Depending on the software used, an 80 track drive might not be supported for drive #0.)

However, if two or more drives require write precompensation, they should require approx. the same value of precompensation, as the FDC board can only be set to one non-zero value.

Connection

1) Obtain an OEM manual for the drives to be connected.
2) Using the manuals as a guide, configure each drive for:
   - multiplexed operation
   - head load when drive selected
   - the desired drive number (if any manufacturer numbers the jumpers Drive 1 to Drive 4, renumber them Drive 0 to Drive 3)
3) Remove the terminating resistor package from all drives except for one, which should be the drive at the end of the cable. Be careful when using a mix of models that all signal lines being used are terminated.
4) Obtain a 34 conductor ribbon cable to connect between the drives and the ST-2900 FDC.
5) Connect the cable between P3 on the FDC board and your drive(s). Double check that pin 1 on each male connector connects to pin 1 on the matching female connector.
6) Obtain the power supply cables and connectors for the drive(s).
7) Connect the power supply to the drive(s).
8) If you have the ST-2900 FLEX Conversion Package, familiarize yourself with the DSKSET utility.

APPENDIX E - Suggested Sources of Further Information

1) "'68' Micro Journal", a monthly magazine for 6800/6809/68000 based systems, published by Computer Publishing Inc., 5900 Cassandra Smith, PO Box 849, Hixson, Tennessee, U.S.A. 37343. Current and back issues contain a wealth of program listings and suggestions especially valuable if you are running the FLEX operating system (and also apply to STAR-DOS Level I), as well as the OS-9 operating system. Advertisements for systems and applications programs for the 6809 also appear in the magazine.

2) Frank Hogg Laboratory, The Regency Tower, Suite 215, 770 James St., Syracuse, NY 13203 U.S.A., puts out a "Serious Users Software Catalog" of 6800/6809 software.

3) Magazines specifically for Radio Shack's Colour Computer, such as "Rainbow", "Hot CoCo", "Color Micro Journal", and "Color Computer Magazine", have information that can sometimes be applied to the ST-2900 system.

4) Two sections in Western Digital Corp.'s "Components Handbook" are of interest: the "FD179X-02" data sheet and the "FD179X Application Notes".

5) For additional information on the 6522 VIA check out Synertek's applications note "AN5 - SY6522 Versatile Interface Adapter".

APPENDIX F - Sample Disk Driver Routines

The following samples of code are just that -- samples. They are not intended to implement a complete set of disk drivers. Rather, they are given to show you how to access the various registers in the floppy disk controller, and how to perform simple seeks and sector read and writes. If you want to run the FLEX operating system, we suggest you buy the ST-2900 FLEX Conversion Package and save yourself many hours of programming and debugging.

Equates:

```
COMREG  EQU   $FF00       1793 command register
STATRG  EQU   $FF00       1793 status register
DATREG  EQU   $FF03       1793 data register
SECREG  EQU   $FF02       1793 sector register
TRKREG  EQU   $FF01       1793 track register
DUART   EQU   $FF20       2681 DUART base addres
XXXX    EQU   0           dummy value

BUFADR  RMB   2           contains address of sector buffer
DRIVEN  RMB   1           contains binary drive number (0 - 3)
TRACKN  RMB   1           contains desired track number
SECTORN RMB   1           contains desired sector number
STEPR   RMB   1           contains a step rate code:
                             $00 = 6 msec      $02 = 20 msec
                             $01 = 12 msec     $03 = 30 msec
```

To select a particular drive (0,1,2,3):

```
        LDA   DRIVEN      get drive number
        ASLA
        ASLA
        ASLA
        ASLA
        ASLA
        ASLA
        STA   DUART+15
        COMA
        ANDA  #$C0
        STA   DUART+14
```

To select single density:

```
        LDA   #$20
        STA   DUART+15
```

To select double density:

```
        LDA   #$20
        STA   DUART+14
```

To select side 0:

```
        LDA     #$10
        STA     DUART+15
```

To select side 1:

```
        LDA     #$10
        STA     DUART+14
```

To turn write precompensation off:

```
        LDA     #$08
        STA     DUART+15
```

To turn write precompensation on:

```
        LDA     #$08
        STA     DUART+14
```

To seek to specified track:

```
SEEK    LDA     TRACKN      get track number
        CMPA    TRKREG      already on track?
        BEQ     SK90        .y
        STA     DATREG      .n
        BSR     DELAY       delay
        LDA     #$18
        ORA     STEPR
        STA     COMREG
        BSR     DELAY       delay 56 usec.
SK10    LDB     STATRG      check status
        BITB    #$01        command completed?
        BNE     SK10        .n, check again
        BITB    #$10        error?
SK90    RTS
```

To read a sector (256 bytes) -- some self-modifying code is used to allow double density operation at 1 MHz.

```
READS   BSR     READY       wait until drive motors up to speed
        LDA     SECTORN     get sector number
        STA     SECREG
        BSR     DELAY
        PSHS    DP,CC       save DP register and interrupt masks
        ORCC    #$50        disable interrupts
        LDA     #$FF        set DP register to page $FFXX
        EXG     A,DP
        SETDP   $FF
        LDX     BUFADR      get sector buffer address
        LEAY    256,X       calc. end-of-buffer-plus-1
        STY     RD60+1      <<< self-modifying code
        LDA     #$84        issue read command
```

```
        STA     COMREG
        BSR     DELAY           delay 56 usec.
        LDB     #$02            DRQ bit
RD40    BITB    STATRG          (4) DRQ?  Loop = 24/31/38 cycles
        BEQ     RD40            (3) .n
        LDA     DATREG          (4) .y, read byte
        STA     ,X+             (6)
RD60    CMPX    #XXXX           (4) sector done?  <<< self-modifying code
        BNE     RD40            (3) .n

RD70    LDB     STATRG          command completed?
        BITB    #$01
        BNE     RD70            .n
        PULS    DP,CC           reset DP register and interrupt masks
        SETDP   $00
        BITB    #$1C            errors?
        RTS
```

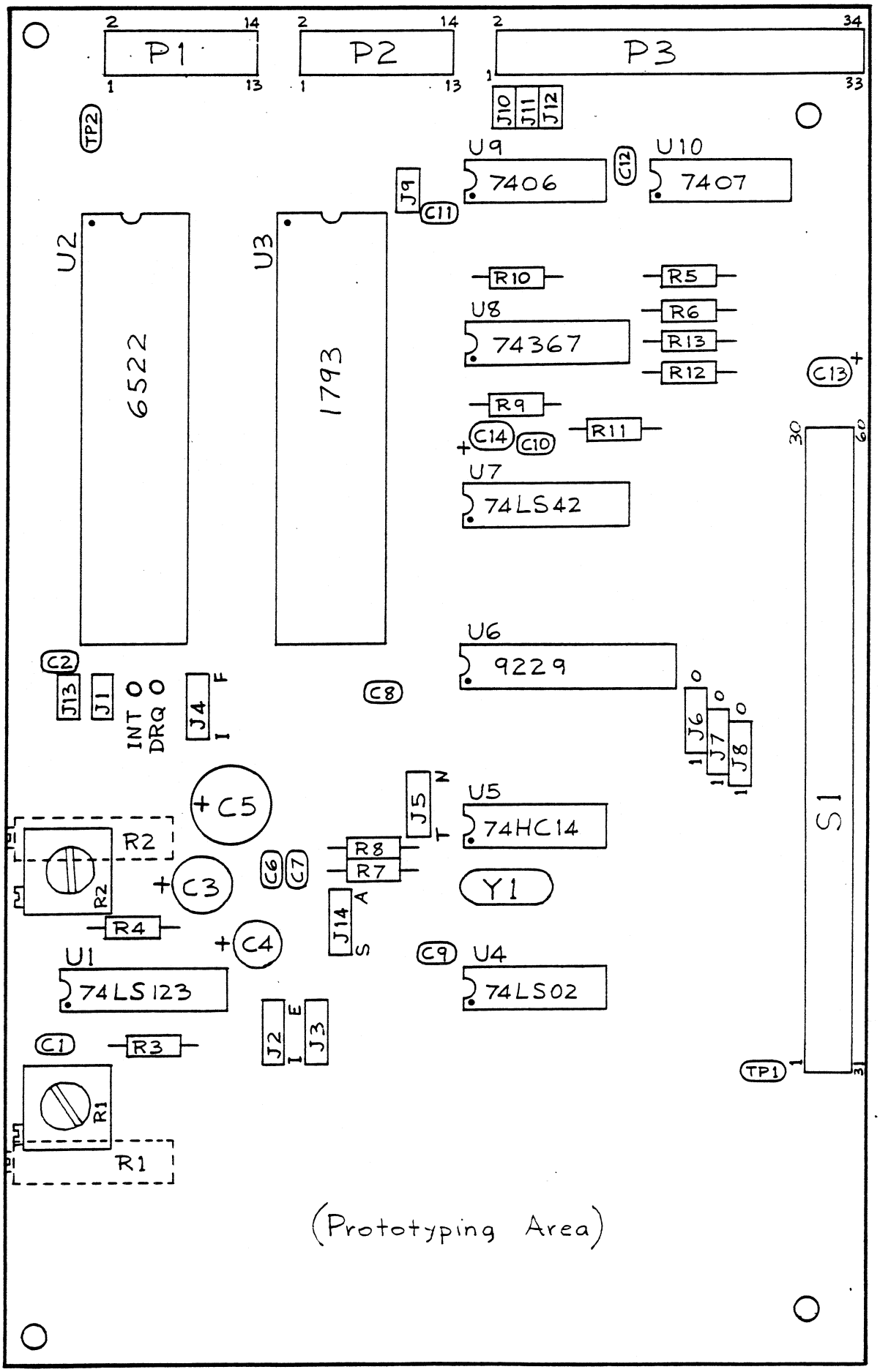
To write a sector (256 bytes) -- some self-modifying code is used to allow double density operation at 1 MHz.

```
WRITES  BSR     READY           wait until drive motors up to speed
        LDA     SECTORN         get sector number
        STA     SECREG
        BSR     DELAY
        PSHS    DP,CC           save DP register and interrupt masks
        ORCC    #$50            disable interrupts
        LDA     #$FF            set DP register to page $FFXX
        EXG     A,DP
        SETDP   $FF
        LDX     BUFADR          get sector buffer address
        LEAY    256,X           calc. end-of-buffer-plus-1
        STY     WR60+1          <<< self-modifying code
        LDA     #$A4            issue write command
        STA     COMREG
        BSR     DELAY           delay 56 usec.

        LDB     #$02            DRQ bit
WR40    LDA     ,X+             (6) get byte in advance
WR45    BITB    STATRG          (4) DRQ?  Loop = 24/31/38 cycles
        BEQ     WR45            (3) .n
        STA     DATREG          (4) .y, write byte
WR60    CMPX    #XXXX           (4) sector done?  <<< self-modifying code
        BNE     WR40            (3) .n

WR70    LDB     STATRG          command completed?
        BITB    #$01
        BNE     WR70            .n
        PULS    DP,CC           reset DP register and interrupt masks
        SETDP   $00
        BITB    #$5C            errors?
        RTS
```

Wait for drive motors to be up to speed:

```
READY   TST     STATRG
        BMI     READY
        RTS                                              .....
```

FDC - Component Layout



Revision B

FDC

Revision B

Apr. 5/84

S1

J14

C14 2.2μF tantalum

C13 10μF tantalum

TP1, TP2

P1

P2

U2 6522

J4

J13

J1

U3 1793

Reset

INT

DRQ

U4

U5

U6 9229

U7 74LS42

U8 74367

U9 74Φ6

U10 74Φ7

74LS123

U1

C3 100μF

C4 22μF

C5 47μF

R1 100K

R2 200K

R3 10K

R4 22K

J2

J3

8MHz

Y1

C6 10pF

C7 10pF

R8 2M

R7 1K

74HC14

74LSΦ2

R9 10K

J6

J7

J8

J9

R13 150Ω

R10 150Ω

R5 150Ω

R12 150Ω

R11 150Ω

R6 1K

J10

J11

J12

P3

(all odd number pins connected to ground)

1-33

spare gates

		Pin number	
		Gnd	+5
U8	74367	8	16
U10	7407	7	14
U9	7406	7	14
U4	74LS02	7	14
U1	74LS123	8	16
U5	74LS14	7	14

Note: C1, C2, C8-12 are .1μF decoupling (bypass) capacitors.

Versatile Interface Adapter (VIA)

Synertek INCORPORATED

SY6522
SY6522A

MICROPROCESSOR PRODUCTS

Preliminary

APRIL 1979

- Two 8-Bit Bidirectional I/O Ports
- Two 16-Bit Programmable Timer/Counters
- Serial Data Port
- Single +5V Power Supply
- TTL Compatible
- CMOS Compatible Peripheral Control Lines
- Expanded "Handshake" Capability Allows Positive Control of Data Transfers Between Processor and Peripheral Devices
- Latched Output and Input Registers
- 1 MHz and 2 MHz Operation

The SY6522 Versatile Interface Adapter (VIA) is a very flexible I/O control device. In addition, this device contains a pair of very powerful 16-bit interval timers, a serial-to-parallel/parallel-to-serial shift register and input data latching on the peripheral ports. Expanded handshaking capability allows control of bi-directional data transfers between VIA's in multiple processor systems.

Control of peripheral devices is handled primarily through two 8-bit bi-directional ports. Each line can be programmed as either an input or an output. Several peripheral I/O lines can be controlled directly from the interval timers for generating programmable frequency square waves or for counting externally generated pulses. To facilitate control of the many powerful features of this chip, an interrupt flag register, an interrupt enable register and a pair of function control registers are provided.

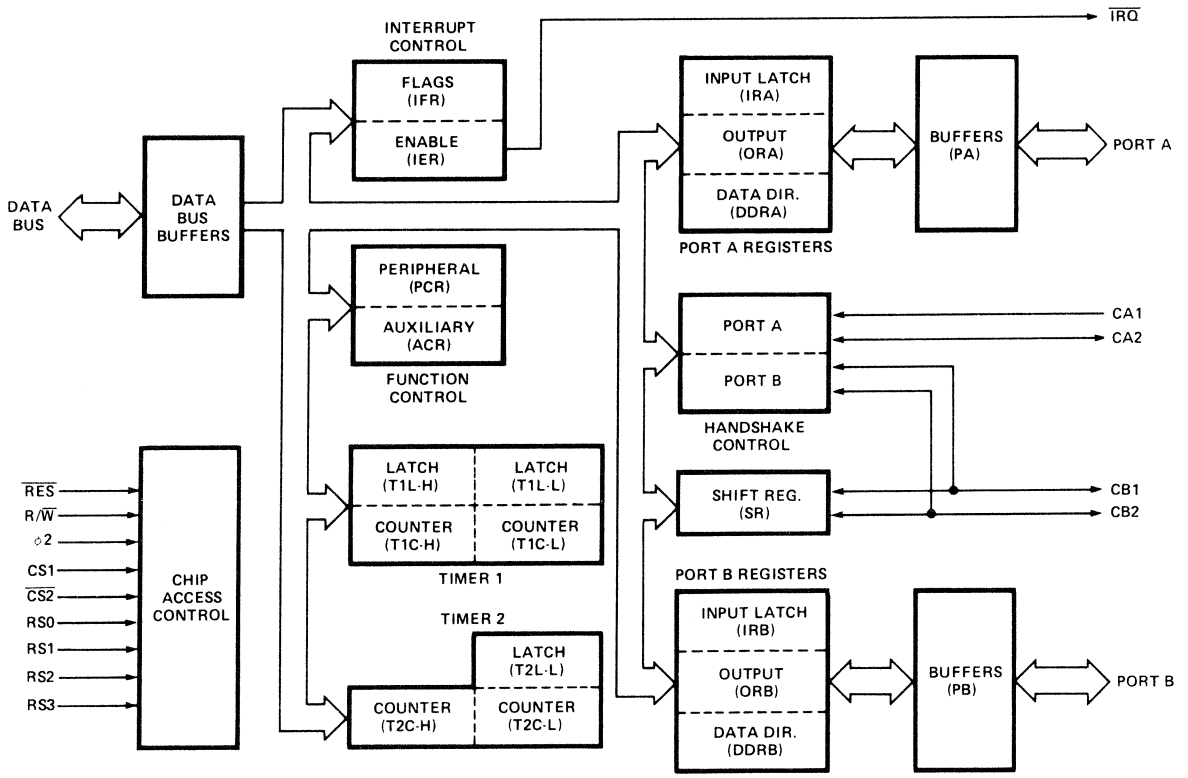


Figure 1. SY6522 Block Diagram

PIN DESCRIPTIONS

$\overline{RES}$ (Reset)

The reset input clears all internal registers to logic 0 (except T1 and T2 latches and counters and the Shift Register). This places all peripheral interface lines in the input state, disables the timers, shift register, etc. and disables interrupting from the chip.

$\phi 2$ (Input Clock)

The input clock is the system $\phi 2$ clock and is used to trigger all data transfers between the system processor and the SY6522.

R/$\overline{W}$ (Read/Write)

The direction of the data transfers between the SY6522 and the system processor is controlled by the R/$\overline{W}$ line. If R/W is low, data will be transferred out of the processor into the selected SY6522 register (write operation). If R/W is high and the chip is selected, data will be transferred out of the SY6522 (read operation).

DB0–DB7 (Data Bus)

The eight bi-directional data bus lines are used to transfer data between the SY6522 and the system processor. During read cycles, the contents of the selected SY6522 register are placed on the data bus lines and transferred into the processor. During write cycles, these lines are high-impedance inputs and data is transferred from the processor into the selected register. When the SY6522 is unselected, the data bus lines are high-impedance.

CS1, $\overline{CS2}$ (Chip Selects)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected SY6522 register will be accessed when CS1 is high and $\overline{CS2}$ is low.

RS0–RS3 (Register Selects)

The four Register Select inputs permit the system processor to select one of the 16 internal registers of the SY6522, as shown in Figure 6.

Register Number	RS Coding				Register Desig.	Description	
	RS3	RS2	RS1	RS0		Write	Read
0	0	0	0	0	ORB/IRB	Output Register "B"	Input Register "B"
1	0	0	0	1	ORA/IRA	Output Register "A"	Input Register "A"
2	0	0	1	0	DDRB	Data Direction Register "B"	
3	0	0	1	1	DDRA	Data Direction Register "A"	
4	0	1	0	0	T1C-L	T1 Low-Order Latches	T1 Low-Order Counter
5	0	1	0	1	T1C-H	T1 High-Order Counter	
6	0	1	1	0	T1L-L	T1 Low-Order Latches	
7	0	1	1	1	T1L-H	T1 High-Order Latches	
8	1	0	0	0	T2C-L	T2 Low-Order Latches	T2 Low-Order Counter
9	1	0	0	1	T2C-H	T2 High-Order Counter	
10	1	0	1	0	SR	Shift Register	
11	1	0	1	1	ACR	Auxiliary Control Register	
12	1	1	0	0	PCR	Peripheral Control Register	
13	1	1	0	1	IFR	Interrupt Flag Register	
14	1	1	1	0	IER	Interrupt Enable Register	
15	1	1	1	1	ORA/IRA	Same as Reg 1 Except No "Handshake"	

Figure 6. SY6522 Internal Register Summary

$\overline{IRQ}$ (Interrupt Request)

The Interrupt Request output goes low whenever an internal interrupt flag is set and the corresponding interrupt enable bit is a logic 1. This output is "open-drain" to allow the interrupt request signal to be "wire-or'ed" with other equivalent signals in the system.

PA0-PA7 (Peripheral A Port)

The Peripheral A port consists of 8 lines which can be individually programmed to act as inputs or outputs under control of a Data Direction Register. The polarity of output pins is controlled by an Output Register and input data may be latched into an internal register under control of the CA1 line. All of these modes of operation are controlled by the system processor through the internal control registers. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Figure 7 illustrates the output circuit.

CA1, CA2 (Peripheral A Control Lines)

The two Peripheral A control lines act as interrupt inputs or as handshake outputs. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. In addition, CA1 controls the latching of data on Peripheral A port input lines. CA1 is a high-impedance input only while CA2 represents one standard TTL load in the input mode. CA2 will drive one standard TTL load in the output mode.

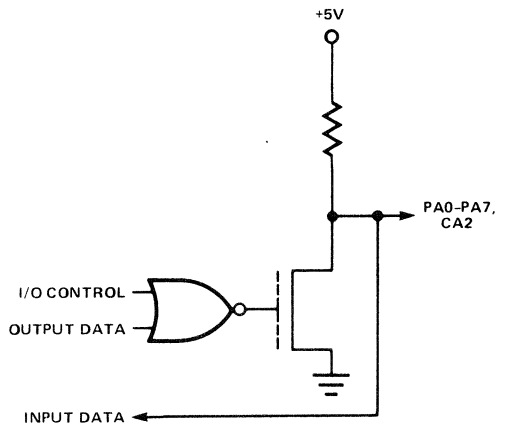


Figure 7. Peripheral A Port Output Circuit

PB0-PB7 (Peripheral B Port)

The Peripheral B port consists of eight bi-directional lines which are controlled by an output register and a data direction register in much the same manner as the PA port. In addition, the polarity of the PB7 output signal can be controlled by one of the interval timers while the second timer can be programmed to count pulses on the PB6 pin. Peripheral B lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. In addition, they are capable of sourcing 1.0mA at 1.5VDC in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 8 is the circuit schematic.

CB1, CB2 (Peripheral B Control Lines)

The Peripheral B control lines act as interrupt inputs or as handshake outputs. As with CA1 and CA2, each line controls an interrupt flag with a corresponding interrupt enable bit. In addition, these lines act as a serial port under control of the Shift Register. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Unlike PB0-PB7, CB1 and CB2 cannot drive Darlington transistor circuits.

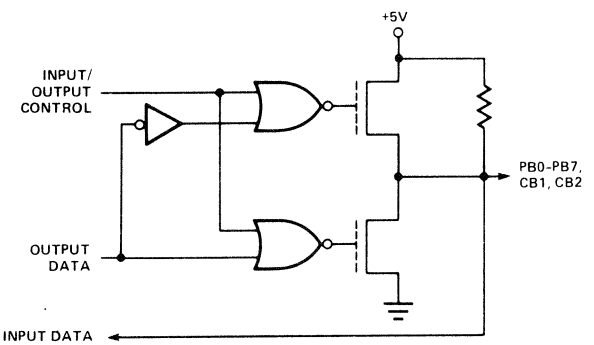


Figure 8. Peripheral B Port Output Circuit

FUNCTIONAL DESCRIPTION

Port A and Port B Operation

Each 8-bit peripheral port has a Data Direction Register (DDRA, DDRB) for specifying whether the peripheral pins are to act as inputs or outputs. A 0 in a bit of the Data Direction Register causes the corresponding peripheral pin to act as an input. A 1 causes the pin to act as an output.

Each peripheral pin is also controlled by a bit in the Output Register (ORA, ORB) and an Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the cor-

responding bit of the Output Register. A 1 in the Output Register causes the output to go high, and a "0" causes the output to go low. Data may be written into Output Register bits corresponding to pins which are programmed as inputs. In this case, however, the output signal is unaffected.

Reading a peripheral port causes the contents of the Input Register (IRA, IRB) to be transferred onto the Data Bus. With input latching disabled, IRA will always reflect the levels on the PA pins. With input latching enabled, IRA will reflect the levels on the PA pins at the time the latching occurred (via CA1).

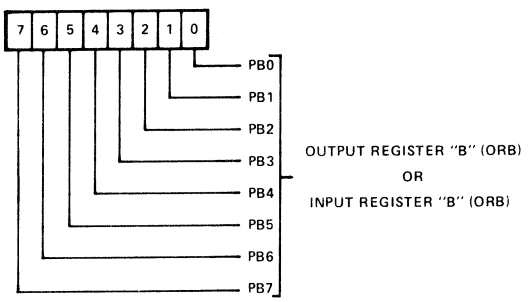
The IRB register operates similar to the IRA register. However, for pins programmed as outputs there is a difference. When reading IRA, the level on the pin determines whether a 0 or a 1 is sensed. When reading IRB, however, the bit stored in the output register, ORB, is the bit sensed. Thus, for outputs which have large loading effects and which pull an output "1" down or which pull an output "0" up, reading IRA may result in reading a "0" when a "1" was actually programmed, and reading a "1" when a "0" was programmed. Reading IRB, on the other hand, will read the "1" or "0" level actually programmed, no matter what the loading on the pin.

Figures 9, 10, and 11 illustrate the formats of the port registers. In addition, the input latching modes are selected by the Auxiliary Control Register (Figure 16.)

Handshake Control of Data Transfers

The SY6522 allows positive control of data transfers between the system processor and peripheral devices through the operation of "handshake" lines. Port A lines (CA1, CA2) handshake data on both a read and a write operation while the Port B lines (CB1, CB2) handshake on a write operation only.

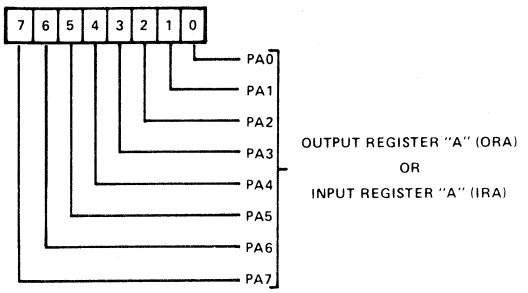
REG 0 – ORB/IRB



Pin Data Direction Selection	WRITE	READ
DDRB = "1" (OUTPUT)	MPU writes Output Level (ORB)	MPU reads output register bit in ORB. Pin level has no affect.
DDRB = "0" (INPUT) (Input latching disabled)	MPU writes into ORB, but no effect on pin level, until DDRB changed.	MPU reads input level on PB pin.
DDRB = "0" (INPUT) (Input latching enabled)		MPU reads IRB bit, which is the level of the PB pin at the time of the last CB1 active transition.

Figure 9. Output Register B (ORB), Input Register B (IRB)

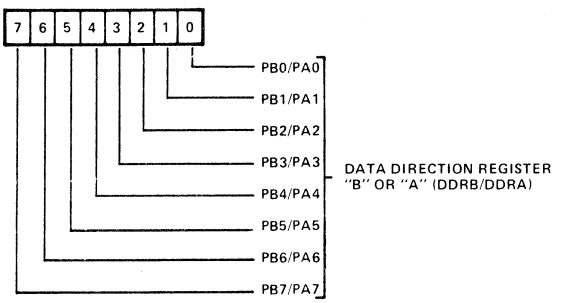
REG 1 – ORA/IRA



Pin Data Direction Selection	WRITE	READ
DDRA = "1" (OUTPUT) (Input latching disabled)	MPU writes Output Level (ORA).	MPU reads level on PA pin
DDRA = "1" (OUTPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition.
DDRA = "0" (INPUT) (Input latching disabled)	MPU writes into ORA, but no effect on pin level, until DDRA changed.	MPU reads level on PA pin.
DDRA = "0" (INPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition.

Figure 10. Output Register A (ORA), Input Register A (IRA)

REG 2 (DDRB) AND REG 3 (DDRA)



"0" ASSOCIATED PB/PA PIN IS AN INPUT (HIGH-IMPEDANCE)

"1" ASSOCIATED PB/PA PIN IS AN OUTPUT, WHOSE LEVEL IS DETERMINED BY ORB/ORA REGISTER BIT.

Figure 11. Data Direction Registers (DDRB, DDRA)

Read Handshake

Positive control of data transfers from peripheral devices into the system processor can be accomplished very effectively using Read Handshaking. In this case, the peripheral device must generate the equivalent of a "Data Ready" signal to the processor signifying that valid data is present on the peripheral port. This signal normally interrupts the processor, which then reads the data, causing generation of a "Data Taken" signal. The peripheral device responds by making new data available. This process continues until the data transfer is complete.

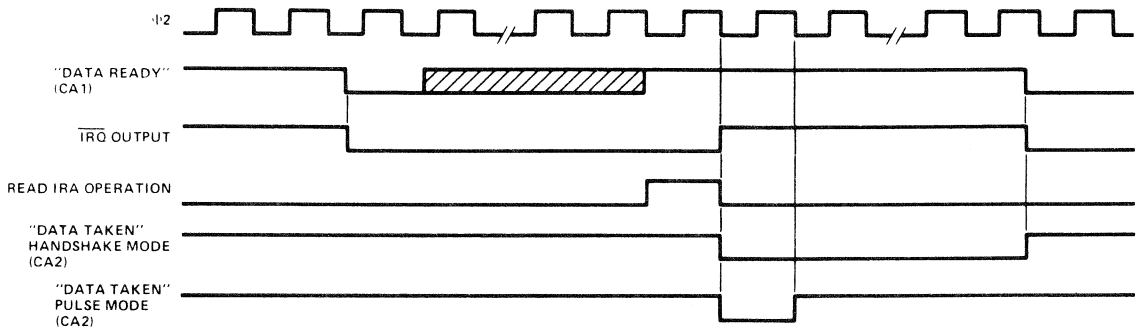


Figure 12. Read Handshake Timing (Port A, Only)

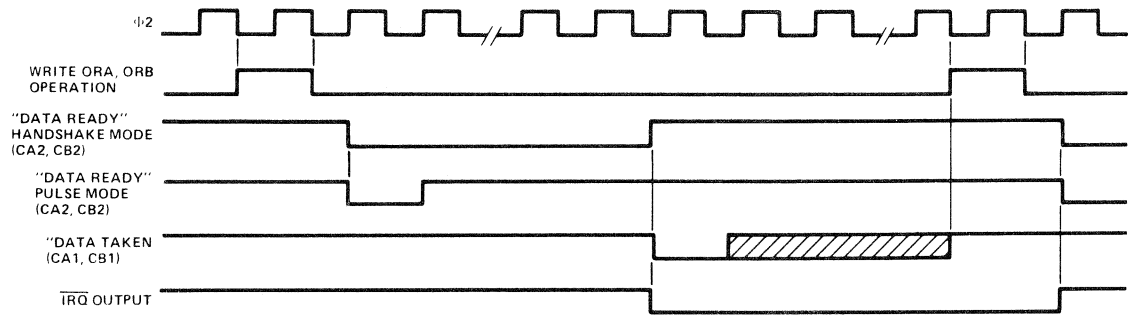


Figure 13. Write Handshake Timing

In the SY6522, automatic "Read" Handshaking is possible on the Peripheral A port only. The CA1 interrupt input pin accepts the "Data Ready" signal and CA2 generates the "Data Taken" signal. The "Data Ready" signal will set an internal flag which may interrupt the processor or which may be polled under program control. The "Data Taken" signal can either be a pulse or a level which is set low by the system processor and is cleared by the "Data Ready" signal. These options are shown in Figure 12 which illustrates the normal Read Handshaking sequence.

Write Handshake

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described for Read Handshaking. However, for Write Handshaking, the SY6522 generates the "Data Ready" signal and the peripheral device must respond with the "Data Taken" signal. This can be accomplished on both the PA port and the PB port on the SY6522. CA2 or CB2 act as a "Data Ready" output in either the handshake mode or pulse mode and CA1 or CB1 accept the "Data Taken" signal from the peripheral device, setting the interrupt flag and cleaning the "Data Ready" output. This sequence is shown in Figure 13.

Selection of operating modes for CA1, CA2, CB1, and CB2 is accomplished by the Peripheral Control Register (Figure 14).

Timer Operation

Interval Timer T1 consists of two 8-bit latches and a 16-bit counter. The latches are used to store data which is to be loaded into the counter. After loading, the counter decrements at $\phi 2$ clock rate. Upon reaching zero, an interrupt flag will be set, and $\overline{IRQ}$ will go low if the interrupt is enabled. The timer will then disable any further interrupts, or will automatically transfer the contents of the latches into the counter and will continue to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin each time it "times-out". Each of these modes is discussed separately below.

The T1 counter is depicted in Figure 15 and the latches in Figure 16.

REG 12 – PERIPHERAL CONTROL REGISTER

7	6	5	4	3	2	1	0

CB2 CONTROL

7	6	5	OPERATION
0	0	0	INPUT-NEGATIVE ACTIVE EDGE
0	0	1	INDEPENDENT INTERRUPT INPUT-NEG EDGE
0	1	0	INPUT-POSITIVE ACTIVE EDGE
0	1	1	INDEPENDENT INTERRUPT INPUT-POS EDGE
1	0	0	HANDSHAKE OUTPUT
1	0	1	PULSE OUTPUT
1	1	0	LOW OUTPUT
1	1	1	HIGH OUTPUT

CB1 INTERRUPT CONTROL

0 = NEGATIVE ACTIVE EDGE
1 = POSITIVE ACTIVE EDGE

CA1 INTERRUPT CONTROL

0 = NEGATIVE ACTIVE EDGE
1 = POSITIVE ACTIVE EDGE

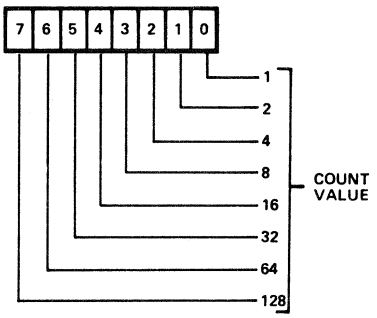
CA2 CONTROL

3	2	1	OPERATION
0	0	0	INPUT-NEGATIVE ACTIVE EDGE
0	0	1	INDEPENDENT INTERRUPT INPUT-NEG EDGE
0	1	0	INPUT-POSITIVE ACTIVE EDGE
0	1	1	INDEPENDENT INTERRUPT INPUT-POS EDGE
1	0	0	HANDSHAKE OUTPUT
1	0	1	PULSE OUTPUT
1	1	0	LOW OUTPUT
1	1	1	HIGH OUTPUT

Figure 14. CA1, CA2, CB1, CB2 Control

Two bits are provided in the Auxiliary Control Register (bits 6 and 7) to allow selection of the T1 operating modes. The four possible modes are depicted in Figure 17.

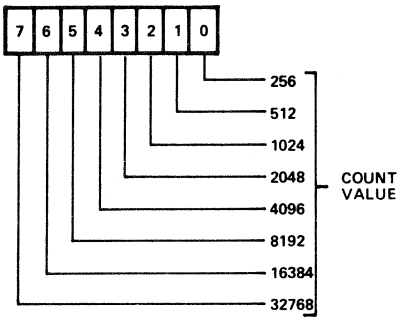
REG 4 – TIMER 1 LOW-ORDER COUNTER



WRITE – 8 BITS LOADED INTO T1 LOW-ORDER LATCHES. LATCH CONTENTS ARE TRANSFERRED INTO LOW-ORDER COUNTER AT THE TIME THE HIGH-ORDER COUNTER IS LOADED (REG 5).

READ – 8 BITS FROM T1 LOW-ORDER COUNTER TRANSFERRED TO MPU. IN ADDITION, T1 INTERRUPT FLAG IS RESET (BIT 6 IN INTERRUPT FLAG REGISTER).

REG 5 – TIMER 1 HIGH-ORDER COUNTER

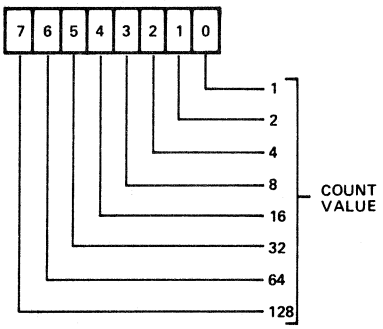


WRITE – 8 BITS LOADED INTO T1 HIGH-ORDER LATCHES. ALSO, AT THIS TIME BOTH HIGH AND LOW-ORDER LATCHES TRANSFERRED INTO T1 COUNTER. T1 INTERRUPT FLAG ALSO IS RESET.

READ – 8 BITS FROM T1 HIGH-ORDER COUNTER TRANSFERRED TO MPU.

Figure 15. T1 Counter Registers

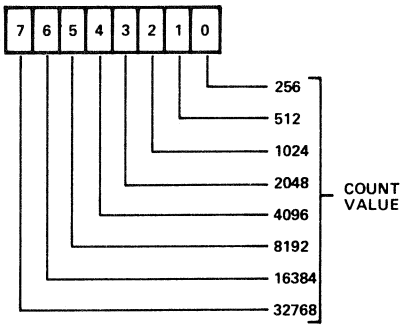
REG 6 – TIMER 1 LOW-ORDER LATCHES



WRITE – 8 BITS LOADED INTO T1 LOW-ORDER LATCHES. THIS OPERATION IS NO DIFFERENT THAT A WRITE INTO REG 4.

READ – 8 BITS FROM T1 LOW-ORDER LATCHES TRANSFERRED TO MPU. UNLIKE REG 4 OPERATION, THIS DOES NOT CAUSE RESET OF T1 INTERRUPT FLAG.

REG 7 – TIMER 1 HIGH-ORDER LATCHES

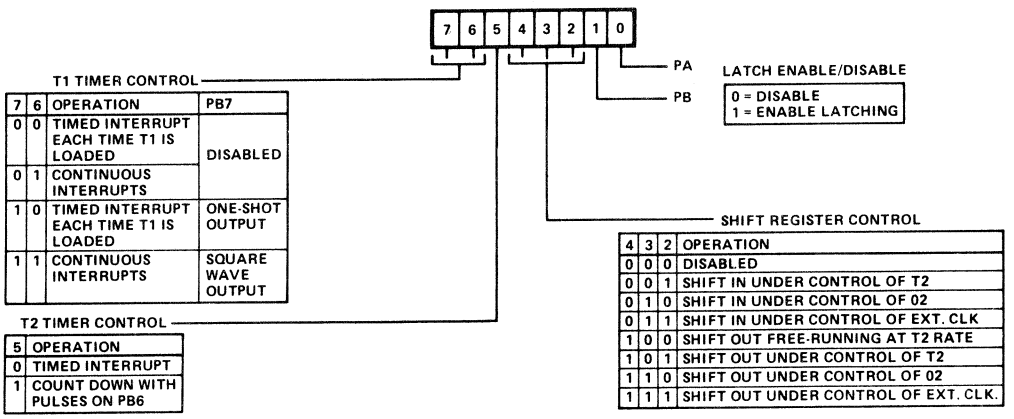


WRITE – 8 BITS LOADED INTO T1 HIGH-ORDER LATCHES. UNLIKE REG 4 OPERATION NO LATCH-TO-COUNTER TRANSFERS TAKE PLACE.

READ – 8 BITS FROM T1 HIGH-ORDER LATCHES TRANSFERRED TO MPU.

Figure 16. T1 Latch Registers

REG 11 – AUXILIARY CONTROL REGISTER



7	6	OPERATION	PB7
0	0	TIMED INTERRUPT EACH TIME T1 IS LOADED	DISABLED
0	1	CONTINUOUS INTERRUPTS	
1	0	TIMED INTERRUPT EACH TIME T1 IS LOADED	ONE-SHOT OUTPUT
1	1	CONTINUOUS INTERRUPTS	SQUARE WAVE OUTPUT

5	OPERATION
0	TIMED INTERRUPT
1	COUNT DOWN WITH PULSES ON PB6

4	3	2	OPERATION
0	0	0	DISABLED
0	0	1	SHIFT IN UNDER CONTROL OF T2
0	1	0	SHIFT IN UNDER CONTROL OF 02
0	1	1	SHIFT IN UNDER CONTROL OF EXT. CLK
1	0	0	SHIFT OUT FREE-RUNNING AT T2 RATE
1	0	1	SHIFT OUT UNDER CONTROL OF T2
1	1	0	SHIFT OUT UNDER CONTROL OF 02
1	1	1	SHIFT OUT UNDER CONTROL OF EXT. CLK.

Figure 17. Auxiliary Control Register

Note: The processor does not write directly into the low order counter (T1C-L). Instead, this half of the counter is loaded automatically from the low order latch when the processor writes into the high order counter. In fact, it may not be necessary to write to the low order counter in some applications since the timing operation is triggered by writing to the high order counter.

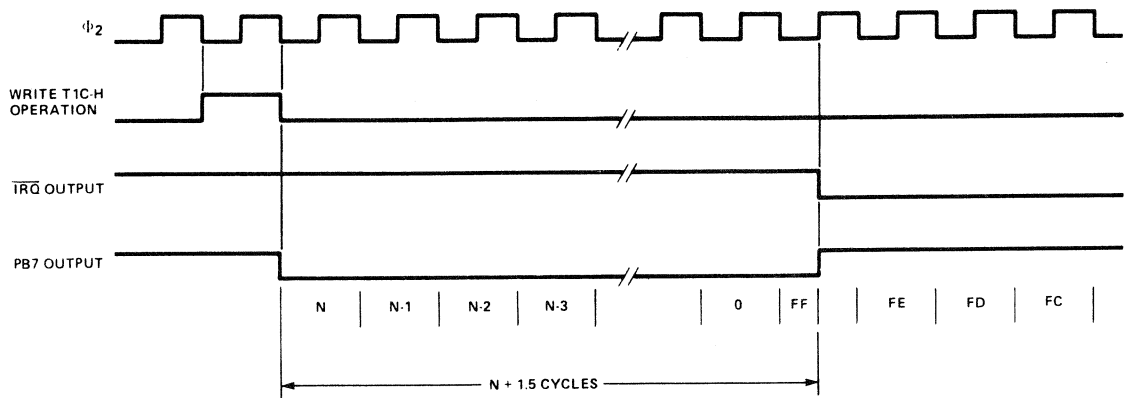


Figure 18. Timer 1 and Timer 2 One-Shot Mode Timing

Timer 1 One-Shot Mode

The interval timer one-shot mode allows generation of a single interrupt for each timer load operation. As with any interval timer, the delay between the "write T1C-H" operation and generation of the processor interrupt is a direct function of the data loaded into the timing counter. In addition to generating a single interrupt, Timer 1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACR7=1) a "write T1C-H" operation will cause PB7 to go low. PB7 will return high when Timer 1 times out. The result is a single programmable width pulse.

In the one-shot mode, writing into the high order latch has no effect on the operation of Timer 1. However, it will be necessary to assure that the low order latch contains the proper data before initiating the count-down with a "write T1C-H" operation. When the processor writes into the high order counter, the T1 interrupt flag will be cleared, the contents of the low order latch will be transferred into the low order counter, and the timer will begin to decrement at system clock rate. If the PB7 output is enabled, this signal will go low on the phase two following the write operation. When the counter reaches zero, the T1 interrupt flag will be set, the $\overline{IRQ}$ pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

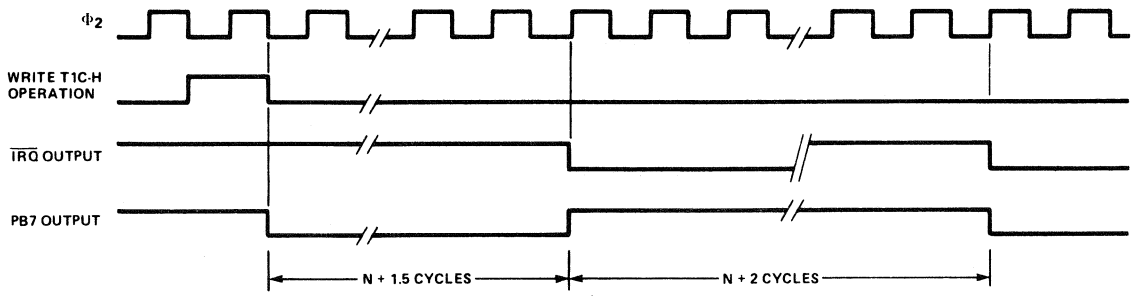
Timing for the SY6522 interval timer one-shot modes is shown in Figure 18.

Timer 1 Free-Run Mode

The most important advantage associated with the latches in T1 is the ability to produce a continuous series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time. This is accomplished in the "free-running" mode.

In the free-running mode, the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter (16 bits) and continues to decrement from there. The interrupt flag can be cleared by writing T1C-H, by reading T1C-L, or by writing directly into the flag as described later. However, it is not necessary to rewrite the timer to enable setting the interrupt flag on the next time-out.

All interval timers in the SY6522 are "re-triggerable". Rewriting the counter will always re-initialize the time-out period. In fact, the time-out can be prevented completely if the processor continues to rewrite the timer before it reaches zero. Timer 1 will operate in this manner if the processor writes into the high order counter (T1C-H). However, by loading the latches only, the processor can access the timer during each down-counting operation without affecting the time-out in process. Instead, the data loaded into the latches will determine the length of the next time-out period. This capability is particularly valuable in the free-running mode with the output enabled. In this mode, the signal on PB7 is inverted and the interrupt flag is set with each time-out. By responding to the interrupts with new data for the latches, the processor can determine the period of the next half cycle during each half cycle of the output signal on PB7. In this manner, very complex waveforms can be generated. Timing for the free-running mode is shown in Figure 19.



Note: A precaution to take in the use of PB7 as the timer output concerns the Data Direction Register contents for PB7. Both DDRB bit 7 and ACR bit 7 must be 1 for PB7 to function as the timer output. If one is 1 and the other is 0, then PB7 functions as a normal output pin, controlled by ORB bit 7.

Figure 19. Timer 1 Free-Run Mode Timing

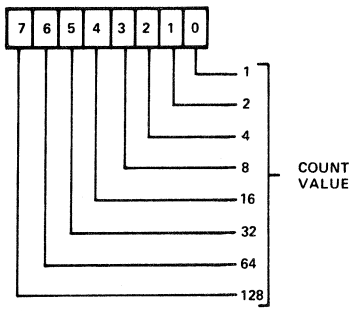
Timer 2 Operation

Timer 2 operates as an interval timer (in the "one-slot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit is provided in the Auxiliary Control Register to select between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter and a read/write high order counter. The counter registers act as a 16-bit counter which decrements at Φ2 rate. Figure 20 illustrates the T2 Counter Registers.

Timer 2 One-Shot Mode

As an interval timer, T2 operates in the "one-shot" mode similar to Timer 1. In this mode, T2 provides a single interrupt for each "write T2C-H" operation. After timing out, the counter will continue to decrement. However, setting of the interrupt flag will be disabled after initial time-out so that it will not be set by the counter continuing to decrement through zero. The processor must rewrite T2C-H to enable setting of the interrupt flag. The interrupt flag is cleared by reading T2C-L or by writing T2C-H. Timing for this operation is shown in Figure 18.

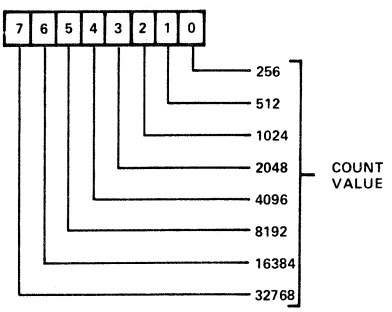
REG 8 – TIMER 2 LOW-ORDER COUNTER



WRITE – 8 BITS LOADED INTO T2 LOW-ORDER LATCHES.

READ – 8 BITS FROM T2 LOW-ORDER COUNTER TRANSFERRED TO MPU. T2 INTERRUPT FLAG IS RESET.

REG 9 – TIMER 2 HIGH-ORDER COUNTER



WRITE – 8 BITS LOADED INTO T2 HIGH-ORDER COUNTER. ALSO, LOW-ORDER LATCHES TRANSFERRED TO LOW-ORDER COUNTER. IN ADDITION, T2 INTERRUPT FLAG IS RESET.

READ – 8 BITS FROM T2 HIGH-ORDER COUNTER TRANSFERRED TO MPU.

Figure 20. T2 Counter Registers

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 serves primarily to count a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into T2. Writing into T2C-H clears the interrupt flag and allows the counter to decrement each time a pulse is applied to PB6. The interrupt flag will be set when T2 reaches zero. At this time the counter will continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on subsequent down-counting operations. Timing for this mode is shown in Figure 21. The pulse must be low on the leading edge of Φ2.

Shift Register Operation

The Shift Register (SR) performs serial data transfers into and out of the CB2 pin under control of an internal modulo-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling external devices.

The control bits which select the various shift register operating modes are located in the Auxiliary Control Register. Figure 22 illustrates the configuration of the SR data bits and the SR control bits of the ACR.

Figures 23 and 24 illustrate the operation of the various shift register modes.

Interrupt Operation

Controlling interrupts within the SY6522 involves three principal operations. These are flagging the interrupts, enabling interrupts and signaling to the processor that an active interrupt exists within the chip. Interrupt flags are set by interrupting conditions which exist within the chip or on inputs to the chip. These flags normally remain set until the interrupt has been serviced. To determine the source of an interrupt, the microprocessor must examine these flags in order from highest to lowest priority. This is accomplished by reading the flag register into the processor accumulator, shifting this register either right or left and then using conditional branch instructions to detect an active interrupt.

Associated with each interrupt flag is an interrupt enable bit. This can be set or cleared by the processor to enable interrupting the processor from the corresponding interrupt flag. If an interrupt flag is set to a logic 1 by an interrupting condition, and the corresponding interrupt enable bit is set to a 1, the Interrupt Request Output ($\overline{IRQ}$) will go low. $\overline{IRQ}$ is an "open-collector" output which can be "wire-or'ed" with other devices in the system to interrupt the processor.

In the SY6522, all the interrupt flags are contained in one register. In addition, bit 7 of this register will be read as a logic 1 when an interrupt exists within the chip. This allows very convenient polling of several devices within a system to locate the source of an interrupt.

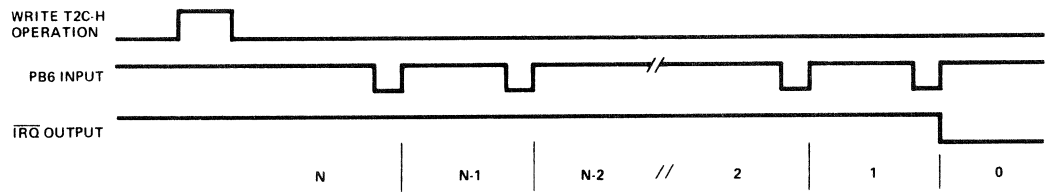
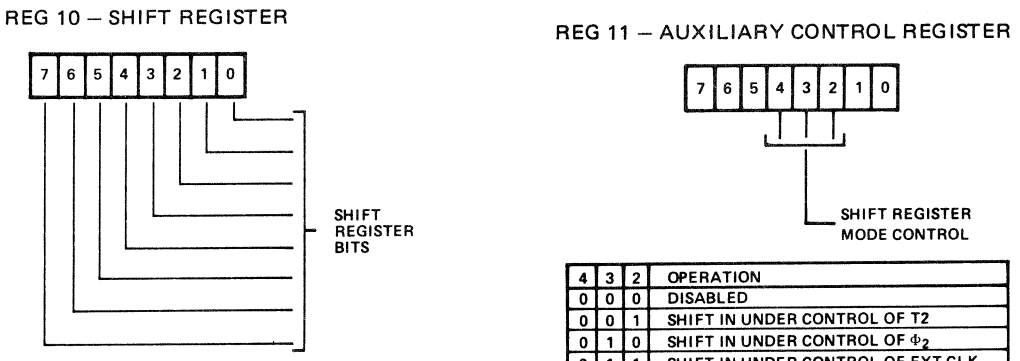


Figure 21. Timer 2 Pulse Counting Mode



4	3	2	OPERATION
0	0	0	DISABLED
0	0	1	SHIFT IN UNDER CONTROL OF T2
0	1	0	SHIFT IN UNDER CONTROL OF Φ_2
0	1	1	SHIFT IN UNDER CONTROL OF EXT CLK
1	0	0	SHIFT OUT FREE-RUNNING AT T2 RATE
1	0	1	SHIFT OUT UNDER CONTROL OF T2
1	1	0	SHIFT OUT UNDER CONTROL OF Φ_2
1	1	1	SHIFT OUT UNDER CONTROL OF EXT CLK

NOTES:
1. WHEN SHIFTING OUT, BIT 7 IS THE FIRST BIT OUT AND SIMULTANEOUSLY IS ROTATED BACK INTO BIT 0.
2. WHEN SHIFTING IN, BITS INITIALLY ENTER BIT 0 AND ARE SHIFTED TOWARDS BIT 7.

Figure 22. SR and ACR Control Bits

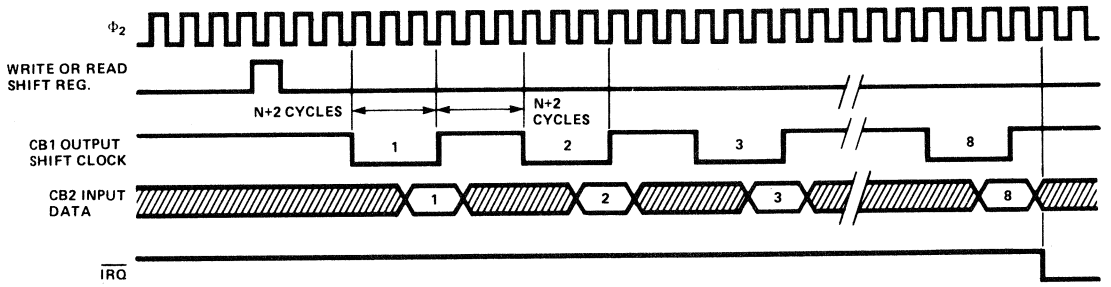
SR Disabled (000)

The 000 mode is used to disable the Shift Register. In this mode the microprocessor can write or read the SR, but the shifting operation is disabled and operation of CB1 and CB2 is controlled by the appropriate bits in the Peripheral Control Register (PCR). In this mode the SR Interrupt Flag is disabled (held to a logic 0).

Shift in Under Control of T2 (001)

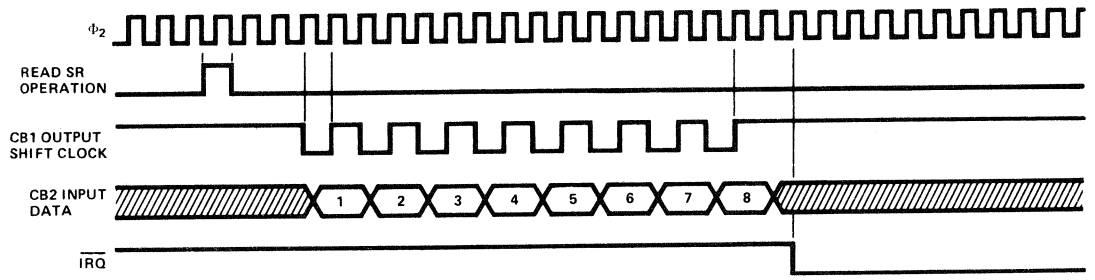
In the 001 mode the shifting rate is controlled by the low order 8 bits of T2. Shift pulses are generated on the CB1 pin to control shifting in external devices. The time between transitions of this output clock is a function of the system clock period and the contents of the low order T2 latch (N).

The shifting operation is triggered by writing or reading the shift register. Data is shifted first into the low order bit of SR and is then shifted into the next higher order bit of the shift register on the negative-going edge of each clock pulse. The input data should change before the positive-going edge of the CB1 clock pulse. This data is shifted into the shift register during the ϕ_2 clock cycle following the positive-going edge of the CB1 clock pulse. After 8 CB1 clock pulses, the shift register interrupt flag will be set and $\overline{IRQ}$ will go low.



Shift in Under Control of ϕ_2 (010)

In mode 010 the shift rate is a direct function of the system clock frequency. CB1 becomes an output which generates shift pulses for controlling external devices. Timer 2 operates as an independent interval timer and has no effect on SR. The shifting operation is triggered by reading or writing the Shift Register. Data is shifted first into bit 0 and is then shifted into the next higher order bit of the shift register on the trailing edge of each ϕ_2 clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.



Shift in Under Control of External CB1 Clock (011)

In mode 011 CB1 becomes an input. This allows an external device to load the shift register at its own pace. The shift register counter will interrupt the processor each time 8 bits have been shifted in. However, the shift register counter does not stop the shifting operation; it acts simply as a pulse counter. Reading or writing the Shift Register resets the Interrupt flag and initializes the SR counter to count another 8 pulses.

Note that the data is shifted during the first system clock cycle following the positive-going edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high.

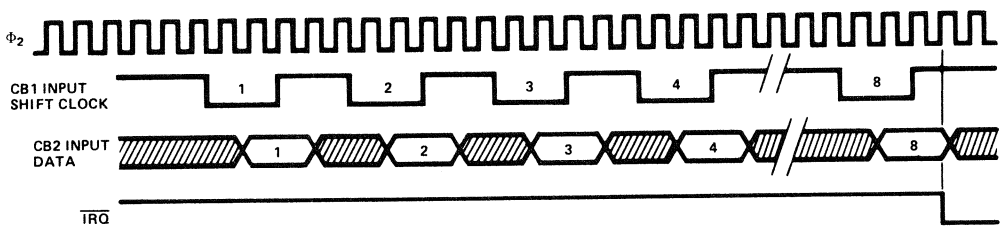
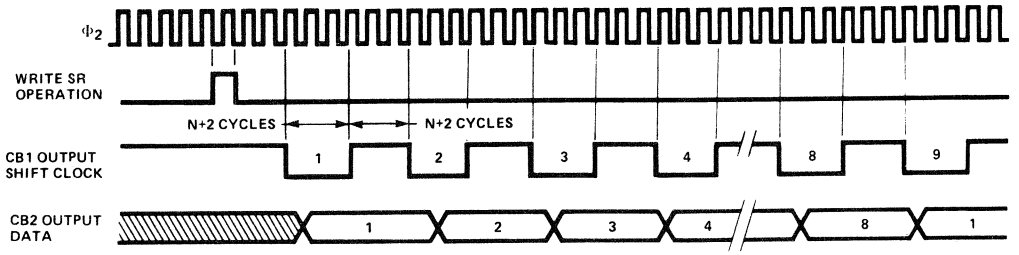


Figure 23. Shift Register Input Modes

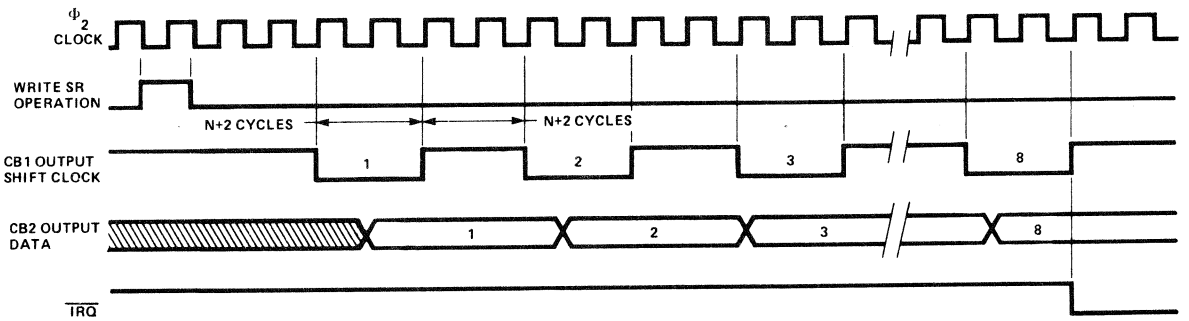
Shift Out Free-Running at T2 Rate (100)

Mode 100 is very similar to mode 101 in which the shifting rate is set by T2. However, in mode 100 the SR Counter does not stop the shifting operation. Since the Shift Register bit 7 (SR7) is recirculated back into bit 0, the 8 bits loaded into the shift register will be clocked onto CB2 repetitively. In this mode the shift register counter is disabled.



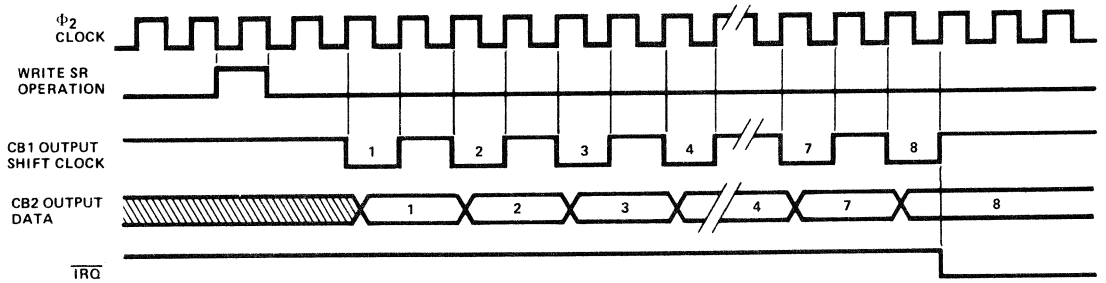
Shift Out Under Control of T2 (101)

In mode 101 the shift rate is controlled by T2 (as in the previous mode). However, with each read or write of the shift register the SR Counter is reset and 8 bits are shifted onto CB2. At the same time, 8 shift pulses are generated on CB1 to control shifting in external devices. After the 8 shift pulses, the shifting is disabled, the SR Interrupt Flag is set and CB2 remains at the last data level.



Shift Out Under Control of ϕ_2 (110)

In mode 110, the shift rate is controlled by the ϕ_2 system clock.



Shift Out Under Control of External CB1 Clock (111)

In mode 111 shifting is controlled by pulses applied to the CB1 pin by an external device. The SR counter sets the SR Interrupt flag each time it counts 8 pulses but it does not disable the shifting function. Each time the microprocessor writes or reads the shift register, the SR Interrupt flag is reset and the SR counter is initialized to begin counting the next 8 shift pulses on pin CB1. After 8 shift pulses, the interrupt flag is set. The microprocessor can then load the shift register with the next byte of data.

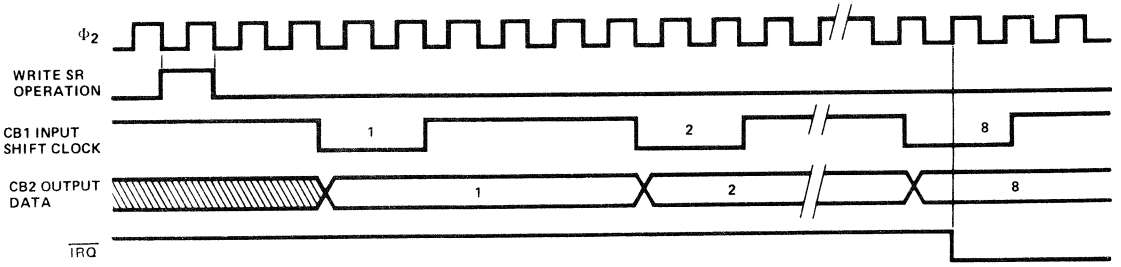


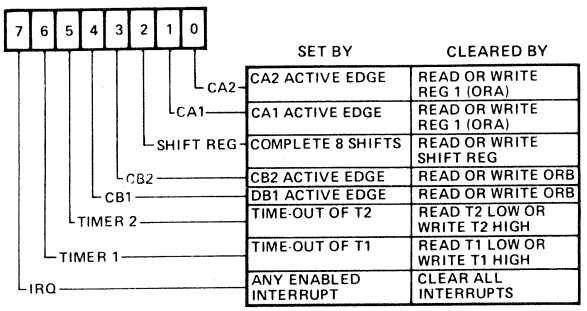
Figure 24. Shift Register Output Modes

The Interrupt Flag Register (IFR) and Interrupt Enable Register (IER) are depicted in Figures 25 and 26, respectively.

The IFR may be read directly by the processor. In addition, individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. When the proper chip select and register signals are applied to the chip, the contents of this register are placed on the data bus. Bit 7 indicates the status of the IRQ output. This bit corresponds to the logic function: IRQ = IFR6 x IER6 + IFR5 x IER5 + IFR4 x IER4 + IFR3 x IER3 + IFR2 x IER2 + IFR1 x IER1 + IFR0 x IER0. Note: X = logic AND, + = Logic OR.

The IFR bit 7 is not a flag. Therefore, this bit is not directly cleared by writing a logic 1 into it. It can only be cleared by clearing all the flags in the register or by disabling all the active interrupts as discussed in the next section.

REG 13 – INTERRUPT FLAG REGISTER



	SET BY	CLEARED BY
CA2	CA2 ACTIVE EDGE	READ OR WRITE REG 1 (ORA)
CA1	CA1 ACTIVE EDGE	READ OR WRITE REG 1 (ORA)
SHIFT REG	COMPLETE 8 SHIFTS	READ OR WRITE SHIFT REG
CB2	CB2 ACTIVE EDGE	READ OR WRITE ORB
CB1	DB1 ACTIVE EDGE	READ OR WRITE ORB
TIMER 2	TIME-OUT OF T2	READ T2 LOW OR WRITE T2 HIGH
TIMER 1	TIME-OUT OF T1	READ T1 LOW OR WRITE T1 HIGH
IRQ	ANY ENABLED INTERRUPT	CLEAR ALL INTERRUPTS

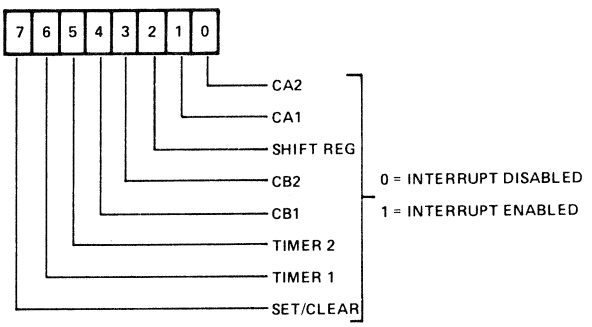
Figure 25. Interrupt Flag Register (IFR)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register. The system processor can set or clear selected bits in this register to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to address 1110 (IER address). If bit 7 of the data placed on the system data bus during this write operation is a 0, each 1 in bits 6 through 0 clears the corresponding bit in the Interrupt Enable Register. For each zero in bits 6 through 0, the corresponding bit is unaffected.

Setting selected bits in the Interrupt Enable Register is accomplished by writing to the same address with bit 7 in the data word set to a logic 1. In this case, each 1 in bits 6 through 0 will set the corresponding bit. For each zero, the corresponding bit will be unaffected. This individual control of the setting and clearing operations allows very convenient control of the interrupts during system operation.

In addition to setting and clearing IER bits, the processor can read the contents of this register by placing the proper address on the register select and chip select inputs with the R/W line high. Bit 7 will be read as a logic 1.

REG 14 – INTERRUPT ENABLE REGISTER



NOTES:
1. IF BIT 7 IS A "0", THEN EACH "1" IN BITS 0 - 6 DISABLES THE CORRESPONDING INTERRUPT.
2. IF BIT 7 IS A "1", THEN EACH "1" IN BITS 0 - 6 ENABLES THE CORRESPONDING INTERRUPT.
3. IF A READ OF THIS REGISTER IS DONE, BIT 7 WILL BE "1" AND ALL OTHER BITS WILL REFLECT THEIR ENABLE/DISABLE STATE.

Figure 26. Interrupt Enable Register (IER)

STANDARD MICROSYSTEMS CORPORATION

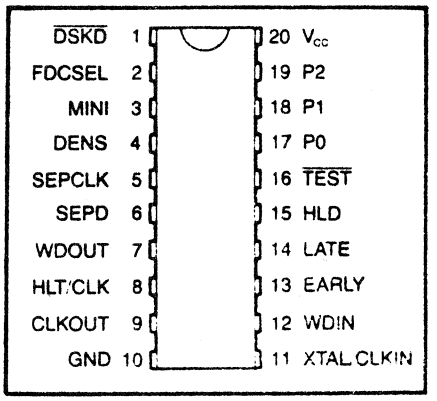
FDC 9229
FDC 9229B
FDC 9229T
FDC 9229BT

FLOPPY DISK INTERFACE CIRCUIT

FEATURES

- Digital Data Separator
  - Performs complete data separation function for floppy disk drives
  - Separates FM and MFM encoded data
  - No critical adjustments necessary
  - 5¼" and 8" compatible
- Variable Write Precompensation
- Internal Crystal Oscillator Circuit
- Track-Selectable Write Precompensation
- Retriggerable Head-Load Timer
- Compatible with the FDC 179X, 765, and other standard Floppy Disk Controllers
- COPLAMOS® n-channel MOS Technology
- Single +5 Volt Supply
- TTL Compatible

PIN CONFIGURATION

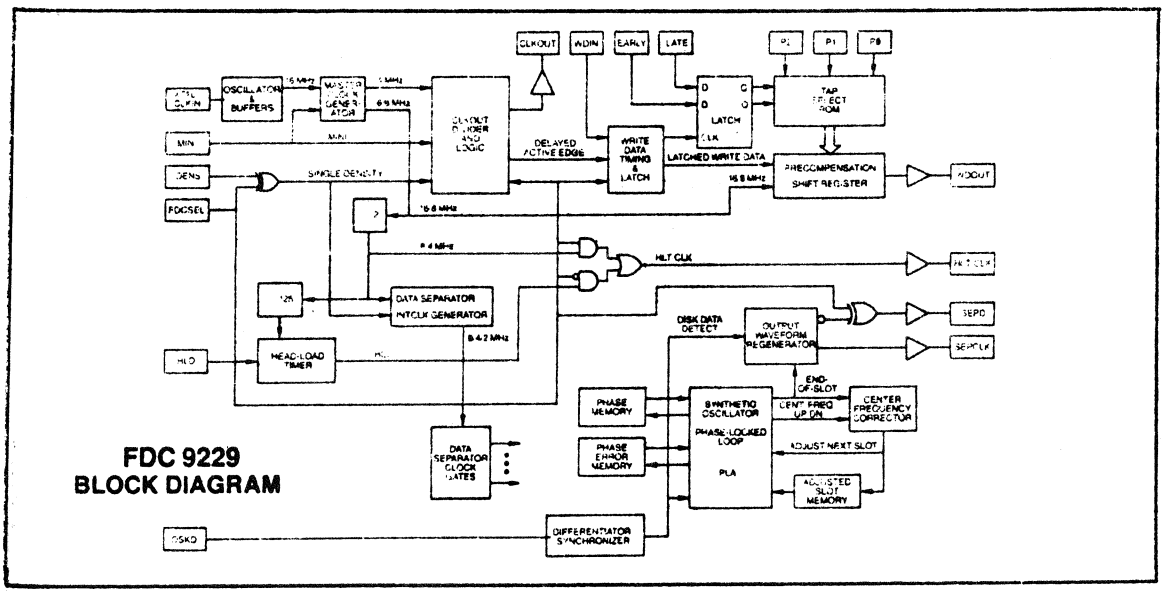


SECTION V

FUNCTIONAL DESCRIPTION

The FDC 9229/B is an MOS integrated circuit designed to complement either the 179X or 765 (8272) type of floppy disk controller chip. It incorporates a digital data separator, write precompensation logic, and a head-load timer in one 0.3-inch wide 20-pin package. A single pin will configure the chip to work with either the 179X or 765 type of controller. The FDC 9229/B provides a number of different dynamically selected precompensation values so that different values may be used when writing to the inner and outer tracks of the floppy disk drive. The FDC 9229/B operates from a +5V supply and simply requires that a 16 or 8 MHz crystal or TTL-level clock be connected to the XTAL/CLKIN pin. All inputs and outputs are TTL compatible.

The FDC 9229 is available in four versions: The FDC 9229/T are intended for 5¼" disks and the FDC 9229B/T for 5¼" and 8" disks. The FDC 9229/B have an internal crystal oscillator circuit; the FDC 9229T/BT require an external clock.



FDC 9229 BLOCK DIAGRAM

DESCRIPTION OF PIN FUNCTIONS

PIN NO.	SYMBOL	I/O	DESCRIPTION
1	$\overline{\text{DSKD}}$	I	This input is the raw read data received from the drive. (This input is active low.)
2	FDCSEL	I	This input signal, when low, programs the FDC 9229/B for a 179X type of LSI controller. When FDCSEL is high, the FDC 9229/B is programmed for a 765 (8272) type of controller. (See fig. 4.)
3	MINI	I	The state of this input determines whether the FDC 9229/B is configured to support 8" or 5¼" floppy disk drive interfaces. It is used in conjunction with the DENS input to prescale the clock for the data separator. The state of this input also alters the CLKOUT frequency, the precompensation value, the head load delay time (when in 179X mode) and the HLT/CLK frequency (when in 765 mode). (See figs. 2, 3, and 4.)
4	DENS	I	The state of this input determines whether the FDC 9229/B is configured to support single density (FM) or double density (MFM) floppy disk drive interfaces. It is used in conjunction with the MINI input to prescale the clock for the data separator. The state of this input also alters the CLKOUT frequency when in the 765 mode. (See figs. 2, 3, and 4.)
5	SEPCLK	O	A square-wave window clock signal output derived from the $\overline{\text{DSKD}}$ input.
6	SEPD	O	This output is the regenerated data pulse derived from the raw data input ($\overline{\text{DSKD}}$). This signal may be either active low or active high as determined by FDCSEL (pin 2).
7	WDOUT	O	The precompensated WRITE DATA stream to the drive.
8	HLT/CLK	O	When in the 765 mode (FDCSEL high), this output is the master clock to the floppy disk controller. When in the 179X mode, this signal goes high after the head load delay has occured following the HLD input going high. This output is retriggerable. (See fig. 3.)
9	CLKOUT	O	This signal is the write clock to the floppy disk controller. Its frequency is determined by the state of the MINI, DENS, and FDCSEL input pins. (See fig. 3.)
10	GND		Ground
11	XTAL/CLKIN	I	This input is for direct connection to a 16 MHz or 8 MHz crystal (FDC 9229/B only). The other pin of the crystal is grounded. XTAL CLKIN may alternatively be connected to a single-phase TTL-level clock. The FDC 9229T and BT require an external TTL-level clock.
12	WDIN	I	The write data stream from the floppy disk controller.
13	EARLY	I	When this input is high, the current WRITE DATA pulse will be written early to the disk.
14	LATE	I	When this input is high, the current WRITE DATA pulse will be written late to the disk. When both EARLY and LATE are low, the current WRITE DATA pulse will be written at the nominal position.
15	HLD	I	This input is only used in 179X mode. A high level at this input causes a high level on the HLT/CLK output after the specified head-load time delay has elapsed. The delay is selected by the state of the MINI output. (See fig. 3.)
16	$\overline{\text{TEST}}$	I	This input (when low) decreases the head-load time delay and initializes the data separator. This pin is for test purposes only. This input has an internal pull-up resistor and should be tied high or disconnected for normal operation.
17	P0	I	P2-P0 select the amount of precompensation applied to the write data. (See fig. 2.)
18	P1	I	
19	P2	I	
20	V_{CC}		+5 VOLT SUPPLY

OPERATION

ta Separator

e XTAL/CLKIN input clock is internally divided by the FDC 29/B to provide an internal clock. The division ratio is ected by the FDCSEL, MINI and DENS inputs depend- on the type of drive used. (See fig. 1.)

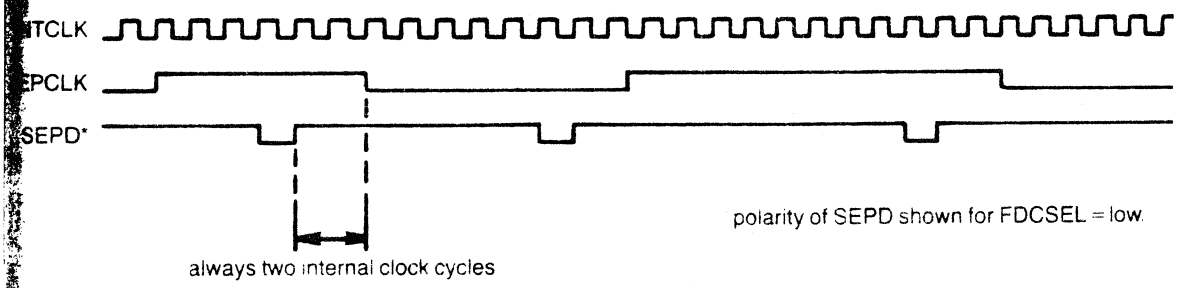
e FDC 9229/B detects the leading (negative) edges of e disk data pulses and adjusts the phase of the internal ck to provide the SEPCLK output.

parate short- and long-term timing correctors assure curate clock separation.

e SEPCLK frequency is nominally 1/16 the internal clock quency. Depending on the internal timing correction, the ation of any SEPCLK half-cycle may vary from a nomi- l of 8 to a minimum of 6 and a maximum of 11 internal ck cycles.

INPUTS			DIVISOR
FDCSEL	DENS	MINI	f(XTAL/CLKIN)/f(INTCLK)
0	0	0	2
0	0	1	4
0	1	0	4
0	1	1	8
1	0	0	4
1	0	1	8
1	1	0	2
1	1	1	4

FIG. 1



SECTION V

recompensation

he desired precompensation delay is determined by the tate of the P0, P1 and P2 inputs of the FDC 9229/B as per g. 2. Logic levels present on these pins may be changed ynamically as long as the inputs are stable during the time he floppy disk controller is writing to the drive and the inputs eet the minimum setup time with respect to the write data om the floppy disk controller.

MINI	P2	P1	P0	PRECOMP VALUE
0	0	0	0	0 ns
0	0	0	1	62.5 ns
0	0	1	0	125 ns
0	0	1	1	187.5 ns
0	1	0	0	250 ns
0	1	0	1	250 ns
0	1	1	0	312.5 ns
0	1	1	1	312.5 ns
1	0	0	0	0 ns
1	0	0	1	125 ns
1	0	1	0	250 ns
1	0	1	1	375 ns
1	1	0	0	500 ns
1	1	0	1	500 ns
1	1	1	0	625 ns
1	1	1	1	625 ns

NOTE: All values shown are obtained with a 16 MHz reference clock. Multiply pre-comp values by two for 8 MHz operation.

FIG. 2 WRITE PRECOMPENSATION VALUE SELECTION

Head Load Timer

The head load time delay is either 40 ms or 80 ms, depending on the state of MINI. (See fig. 3.) The purpose of this delay is to ensure that the head has enough time to engage properly. The head load timer is only used in the 179X mode; it is non-functional in the 765 mode.

The FDC 179X initiates the loading of the floppy disk drive head by setting HLD high. The controller then waits the programmed amount of time until the HLT signal from the FDC 9229/B goes high before starting a read or write operation.

INPUTS			OUTPUTS	
FDCSEL	DENS	MINI	CLKOUT	HLT/CLK
0	0	0	2 MHz	40 ms*
0	0	1	1 MHz	80 ms*
0	1	0	2 MHz	40 ms*
0	1	1	1 MHz	80 ms*
1	0	0	500 KHz	8 MHz
1	0	1	250 KHz	4 MHz
1	1	0	1 MHz	8 MHz
1	1	1	500 KHz	4 MHz

NOTE: All values shown are obtained with a 16 MHz reference clock. Divide all frequencies and multiply all periods by two for 8 MHz operation.

*May be mask programmed at factory to any value from 1 to 512 ms in 15.625 μs increments (MINI low) or 1 to 1024 ms in 31.25 μs increments (MINI high).

FIG. 3 CLOCK AND HEAD LOAD TIME DELAY SELECTION

INPUTS			FLOPPY DISK	FLOPPY DISK	FLOPPY DISK
FDCSEL	DENS	MINI	DRIVE TYPE	DRIVE DENSITY	CONTROLLER TYPE
0	0	0	8" DRIVE	DOUBLE	179X
0	0	1	5¼" DRIVE	DOUBLE	179X
0	1	0	8" DRIVE	SINGLE	179X
0	1	1	5¼" DRIVE	SINGLE	179X
1	0	0	8" DRIVE	SINGLE	765 (8272)
1	0	1	5¼" DRIVE	SINGLE	765 (8272)
1	1	0	8" DRIVE	DOUBLE	765 (8272)
1	1	1	5¼" DRIVE	DOUBLE	765 (8272)

FIG. 4 FLOPPY DISK DRIVE AND CONTROLLER SELECTION

XIMUM GUARANTEED RATINGS*

perating Temperature Range	0°C to +70°C
torage Temperature Range	−55° to +150°C
ead Temperature (soldering, 10 sec.)	+300°C
ositive Voltage on any I/O Pin, with respect to ground	+8.0V
egative Voltage on any I/O Pin, with respect to ground	−0.3V
ower Dissipation	0.75W

resses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the vice at these or at any other condition above those indicated in the operational sections of this specification is not implied.

TE: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not exceeded or device failure can result. Some power supplies exhibit voltage spikes or "glitches" on their outputs when the AC power switched on and off. In addition, voltage transients on the AC power line may appear on the DC output.

ECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V ± 5%)

PARAMETER	MIN	TYP	MAX	UNIT	CONDITIONS
DC CHARACTERISTICS					
INPUT VOLTAGE					
Low Level V_{IL}	−0.3		0.8	V	Except XTAL/CLKIN
High Level V_{IH}	2.0		(V_{CC})	V	
XTAL/CLKIN INPUT VOLTAGE					
AC Amplitude	1.0			V_{P-P}	XTAL/CLKIN only, input is
Instantaneous voltage	−0.3		(V_{CC})	V	AC-coupled.
OUTPUT VOLTAGE					
Low Level V_{OL}			0.4	V	I_{OL} = 1.6 mA except HLT/CLK
					I_{OL} = 0.4 mA, HLT/CLK only
High Level V_{OH}	2.4			V	I_{OH} = −100 µA except HLT/CLK
					I_{OH} = −400 µA, HLT/CLK only
POWER SUPPLY CURRENT					
I_{CC}			100	mA	
INPUT LEAKAGE CURRENT					
I_{IL}			10	µA	V_{IN} = 0 to V_{CC}
INPUT CAPACITANCE			10	pF	Except CLKIN
C_{IN}			25	pF	CLKIN only

ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V ± 5%

PARAMETER	MIN	TYP	MAX	UNIT	CONDITIONS
AC ELECTRICAL CHARACTERISTICS	(All times assume XTAL/CLKIN = 16 MHz unless otherwise specified)				
XTAL/CLKIN frequency	3.95	16	16.2	MHz	FDC 9229B
	3.95	8	8.1	MHz	FDC 9229
XTAL/CLKIN DUTY CYCLE	25		75	%	
t_{CLKIN}	465	500	515	ns	FDCSEL = low; MINI = high.
	215	250	265	ns	FDCSEL = low; MINI = low.
	90	125	140	ns	FDCSEL = high.
t_{WDO}	280	312.5	350	ns	
t_D	50		400	ns	
t_{WREC}	0		400	ns	
t_{WDE}		562.5		ns	
t_{WPR}		precomp value			See fig. 2.
t_{WDL}		2 x precomp value			See fig. 2.
t_R	1.0			µs	

CRYSTAL SPECIFICATIONS

Frequency (8" Disk Drive) 16 MHz, at Cut
(5¼" Disk Drive) 8 MHz, at Cut
Holder Preferred HC-18/U
Frequency and stability tolerance ±.05% from 0°C to 70°C
Series Resistance 50 ohm max

AC TIMING CHARACTERISTICS