FLOPPY DISK/SASI CONTROLLER BOARD

Instruction Manual and Functional Specification

P. QUINTON
AMERSHAM COMPUTER CENTRE 02403 22307.
18 WOODSIDE RD
AMERSSHAM
BUCKS HP7 0BH

0494 791010

Gemini Microcomputers
Oakfield Corner,
Sycamore Road,
Amersham, Bucks HP6 5EQ

Telephone 02403 28321

Telex 837788

Issue 1

For Part No GM829

18-1-83

Introduction

The Gemini GM829 is an 80-BUS and Nasbus compatible combined Floppy Disk Controller (FDC) and Shugart Associates Systems Interface (SASI) board for use with Gemini MultiBoard and Nascom systems. The board has been designed to allow it to be used with up to four mixed 5.25" and 8" drives. These may be single or double sided, ordinary or double track density drives, in single or double density formats. Switching between single and double density, and between 5.25" and 8" drives, is under software control. The board also has variable write precompensation and phase locked loop data recovery circuitry. The controller I.C. used is the Western Digital 1797.

In addition to floppy disk drive support the GM829 board offers a SASI interface for the connection of a hard disk (Winchester) drive sub-system. The Gemini GM835 series has been designed to plug directly into the GM829's SASI socket.

The GM829 board occupies a group of 8 x Z80 I/O ports, and these ports may be located in one of two positions, allowing up to two boards to be used in a single system. When used with 5.25" drives this board is completely software compatible with the earlier Gemini GM809 board.

Commissioning

Carefully unpack your GM829 and examine it for any mechanical damage. In the event of any damage please inform your dealer immediately.

Your GM829 will have been shipped to you fully tested and working. Dependant on the type of disk drives being used all that may be required is for the board to be plugged in. However, please take time to read through this manual as it may prove useful.

When plugging the GM829 into the bus, please take care. Excessive force should not be required. Any difficulty which may be encountered will, in all probability, be due to the keyway of the edge connector not fitting accurately into the slot in the edge of the board. Please ensure that the board is plugged in with the edge connector going in first and the correct way round. It is not possible to plug the board in the incorrect way because of the keyway.

The floppy disk drives are connected to the GM829 via the 34 way (PL2, for 5.25" drives) and 50 way (PL3, for 8" drives) connectors on the board, and the SASI hard disk sub-system via the second 50 way (PL1) connector. The pin 1 end of each connector on the board is identified by a '1' on the PCB and an arrow head or other marking on the connector itself. Ensure that the cable is connected with the correct orientation, or damage may occur to the board and drives.

There are several linking options on the GM829. These have been set during manufacture and full details are provided in the section on links, to allow the user to configure the board to his own requirements.

80-BUS & MultiBoard are trademarks of Gemini Microcomputers Limited.
Nasbus is a trademark of Lucas Logic Ltd. (Nascom Microcomputers Division).

Links.

There are seven linking options on GM829. All link options are set at manufacture by PCB tracks. However, if alternative settings are required, then the PCB tracks linking the pads should be cut and new links should be made by soldering wire from one pad to the other. Please take great care as an incorrect link may cause damage and a poor connection will result in an unreliable system.

Link 1. (LK1). 5.25" Drive type.

There are two alternative 'standards' for the pin out of the 34 way connector (PL2) for 5.25" drives and the LK1 link block is used to select between them. LK1 is placed above IC3.

If pin 6 of the drive connector is the fourth drive select line (ISLT3), then B should be linked to C. Alternatively if pin 34 of the drive connector is the fourth drive select line (DS4), then C should be connected to D. In the latter case pin 6 will be the drive ready line (RDY), and so B and A should also be connected.

For example:	Pertec FD 250 drives	-	link B to C
	Micropolis 1015 drives	-	link A to B, and C to D

Also see link 3.

Link 2. (LK2). 1797 clock.

This link is used to provide the necessary clock to the 1797 FDC chip via a software controlled 'divide by 2' stage.

LK2 is placed between IC6 and IC11. A 2 MHz clock, derived from the 80-BUS system clock, is required at point B. At point C is the system clock itself, and at point A the system clock divided by two. This link therefore should be set as follows:

A - B for 4 MHz CPU clock
C - B for 2 MHz CPU clock

For systems with different CPU clock speeds, see link 6 below.

Link 3. (LK3). Ready line connection.

This link should be inserted when pin 6 of the drive connector is the ready line (RDY), or if 8" drives are being used. This link is placed between IC7 and IC8 and should be present for Micropolis 1015 type drives and 8" drives.

Link 4. (LK4). Write Precompensation selection.

This link area is used to enable/disable the write precompensation circuitry. Different options may be selected for both 5.25" and 8" drives. Each of these may be set to disable the write precompensation circuitry, although this would only normally be done to test the board.

Alternatively write precompensation may be provided on all tracks, or only on tracks beyond track 43 on either drive. This latter option should normally be chosen for 8" drives, and for 5.25" drives with more than 40 tracks per side.

LK4 is positioned below IC7, and link options are shown below. (Note: NC=no connection.) The write precompensation circuitry is only enabled when the system software selects double density operation.

	5.25" drive .	8" drive .
No write precomp.	D-C	B-C
Write precomp. track 43+	D-A	B-A
Write precomp. all tracks	NC to D	NC to B

Link 5. (LK5). Z80 Port addresses.

This link selects one of the two blocks of Z80 port addresses that this board may occupy. LK5 is positioned below IC23. Linking A to C locates the board at Ports CO-C7, linking A to B locates the board at ports EO-E7. The latter is the normal configuration.

Link 6. (LK6). Clock selection.

This link determines the source of the clock on the GM829, and is positioned below IC26. If B is linked to C, the clock will be taken from the bus AUXCLK line. If B is linked to A, the clock will be taken from the main system clock line, and this is the normal position. If the host system is running at 2 or 4 MHz, then the clock signal required by the 1797 FDC can be provided by appropriate selection at link 2. However, if a different system clock is used, then a 4 MHz clock should be provided on the AUXCLK bus line, and B should be linked to C. For full details of implementing the AUXCLK line, please see the manual for the bus master.

Link 7. (LK7). SASI line 32 select.

Link 7 is used for SASI devices requiring the /ATN signal on line 32 of the interface. LK7 is located on the GM829 above IC2. The signal is present if the link is fitted. It is not required for the Gemini GM835 hard disk sub-systems.

Connectors.

The GM829 is provided with a 34 way insulation displacement connector (PL2). This is the standard type of connector used with 5.25" drives. It is also fitted with two 50 way insulation displacement connectors, PL1 and PL3. PL3 is the standard type of connector used with 8" drives, and PL1 is that used by SASI devices. The pin 1 end of each connector on the board is identified by a '1' on the PCB and an arrow head or other marking on the connector itself. Ensure that the cable is connected with the correct orientation, or damage may occur to the board and drives.

Potentiometers.

There are 3 multiturn potentiometers on GM829. These have been set at manufacture for 5.25" drives and 250 nS write precompensation. If alternative adjustments are required, these should be made with reference to a 1797 data sheet and the GM829 circuit diagrams. DO NOT adjust the potentiometers unless the board is to be used with alternative drives as this may negate any warranty claim.

VR3 is set (while no data is being transferred) for 1.45V +/- 0.05V at Pin 2 of the 74LS629 VCO (IC 13) (TP5).

VR1 is adjusted to provide a 4 MHz (+0% -5%) clock signal at Pin 7 of the 74LS629 VCO (IC 13), or a 2 MHz (+0% -5%) clock signal on TP2, provided the GM829 is software selected to 5.25" operation.

VR2 is used to set the required write precompensation timing (Nominally 250nS). It should be adjusted whilst formatting in double density for the desired effect at Pin 6 of the WD 1691 FSL (IC 17)(TP1). Trigger a 'scope from IC17 pin 1 (TP3) and adjust the delay between the start of the 'early' and 'late' pulses to twice the desired precompensation value.

Notes on 8" operation.

Bit 5 of the floppy control port (Port C4 or E4) is used to control the switching of a number of signals on the GM829. These correspond to the differences normally necessary to switch operation between 5.25" and 8" drives, and all references in this manual correspond to these. However, there may be occasions when it is useful to switch these signals for alternative purposes, and the following information may prove useful.

Bit 5 Port C4/E4	0	1
1797 clock =	1 MHz	2 MHz
1691 VCO =	2 MHz	4 MHz
1797 READY line generated by -	Motor-on AND Head loaded AND Drive RDY	Drive Ready ONLY
Write Precomp. determined by -	Connection to 'd' on LK4 (see above)	Connection to 'b' on LK4 (see above)

Note that it is possible to write code to service the 1797 controller in seven of the permutations of:

5.25" or 8" drives
single or double density modes
Z80 CPU clock of 2 or 4 MHz

The illegal permutation is to use 8" drives in double density mode if the Z80 CPU is only running at 2 MHz.

Nascom 2 modification.

In the Nasbus specification the bus reset pulse (line 14) was specified as 10 uS. In the 80-BUS specification this has been extended to 50 uS, as chips in the 179X family require a reset pulse of this duration. In practise this has not been found to cause a problem. However, Nascom 2 owners using the GM829 who are concerned by this should substitute a 10 nF capacitor for C1 (previously 1nF) on their Nascom 2.

Preferred Drives.

The GM829 design has been tested for use with the following drives in both single and double density modes:

Shugart SA200	-	5.25", single sided, 48 TPI, 40 track
Pertec FD250	-	5.25", double sided, 48 TPI, 35 track
Teac FD-55E/55F	-	5.25", single/double sided, 96 TPI, 80 track
Micropolis 1015F5/F6	-	5.25", single/double sided, 96 TPI, 80 track

Pertec FD650	-	8", double sided
Shugart SA800/850	-	8", single/double sided
Tandon TM848-1/-2	-	8", single/double sided

The Micropolis drives are the recommended drives for use with the board, and software support is currently based on these. Alternative drives may be used, but it must be pointed out that although other drives may be electrically compatible, there are often small differences in timing that would require modification to any software written specifically for the above drives.

CP/M 2.2 is available configured for use with GM829 and Pertec FD250 or Micropolis 1015 drives for:
(a) Nascom 1 or 2 with Nascom screen or Gemini GM812 IVC board (80x25).
(b) Gemini MultiBoard System (GM811 CPU or GM813 CPU/RAM).

For Nascom owners Polydos is available as an alternative operating system to CP/M for use with the GM829 and Pertec FD250 or Micropolis 1015 drives.

Software for other drive types may become available and any Gemini dealer should be able to keep you informed of further developments.

Drive Address Selection

Up to four floppy disk drives may be connected to the GM829, and these may be either 5.25", 8", or a combination of both. Note that all drives of the same size should be of the same type. The 34 and 50 way floppy drive connectors (PL2 & 3) are basically connected in parallel, and so should be considered as being on the same connector as regards selecting drive addresses.

GM829 Specification

1. Electrically and physically compatible with 80-BUS and Nasbus.

2. Operates with 5.25" and 8" drives.

3. Operates up to four drives. Drives of each size (5.25" and 8") must be of the same type.

4. Operates with single or double density, single or double sided drives.

5. Operates with 2 MHz or 4 MHz CPU clock. (Software timing may be critical.)

6. Continuously variable write precompensation in double density mode.

7. Single/double density selection under software control.

8. 5.25"/8" selection under software control.

9. Phase locked loop data recovery circuitry.

10. Utilises Western Digital FD1797 controller.

11. Incorporates hardware delays (a) Head load - 45 mS nominal.
    (b) Motor on - 1 sec.nominal.

12. Motor time out approximately 10 seconds after last write to drive select latch.

13. Two boards may be used in the same system, allowing up to 8 floppy drives.

14. The board may occupy Z80 read/write ports C0H-C7H or E0H-E7H (i.e. 8 ports) thus:

C0-C3	or	E0-E3	: 1797 controller
C4	or	E4	: Write - drive and density select Read - flag read
C5	or	E5	: Write - SASI control Read - SASI status
C6	or	E6	: SASI data
C7	or	E7	: Decoded but unused

15. Port C4/E4

Bits	Write	Read
0	Select drive 0	INTRQ from 1797
1	" " 1	NOT READY *
2	" " 2	0V
3	" " 3	0V
4	Select density	0V
5	Select 5.25"/8"	0V
6	Not used	0V
7	Not used	DRQ from 1797

Connector Details

PL1 - 50-way Connector

Line Number	SASI Signal
2	Data 0
4	Data 1
6	Data 2
8	Data 3
10	Data 4
12	Data 5
14	Data 6
16	Data 7
32	/ATN
36	/BUSY
38	/ACK
40	/RST
42	/MSG
44	/SEL
46	C/D
48	/REQ
50	I/O

All odd pins (1-49) are grounded.
Line 34 is implemented by inserting LK7

PL2 - 34-way Connector

Line Number	Pertec FD250 signal	Micropolis 1015 signal	GM829 implementation
6	ISLT3	/RDY	* Drive select D/Ready line
8	IINXP	/SECP	Index Pulse
10	ISTLO	/DS1	)
12	ISTL1	/DS2	)Drive select lines A-C
14	ISLT2	/DS3	)
16	IDEN	/MTRN	Drive motor enable
18	IDIR	/DIRN	Direction select
20	ISTP	/STEP	Step
22	IWDA	/WDA	Write data
24	IWGT	/WRT	Write gate
26	ITRKO	/TRKO	Track 0
28	IWPT	/WPT	Write protect
30	IRDA	/RDA	Read data
32	ISSLT	/HSLT	Side select
34	Spare	/DS4	* Unused/Drive select D

All odd pins (1-33) are grounded.
* The function of lines 6 & 34 are selected by link 1.

PL3 - 50-way connector

Line Number	Pertec FD650 signal	GM829 implementation
2	IHCS	Head current switch
14	ISSLT	Side select
20	IINXP	Index pulse
22	IDRDY	Drive ready
26	ISLTO	)
28	ISLT1	)Drive select lines
30	ISLT2	)
32	ISLT3	)
34	IDIR	Direction select
36	ISTP	Step
38	IWDA	Composite write data
40	IWGT	Write gate
42	ITRKO	Track 0
44	IWPTD	Write protect
48	IRDA	Composite read data

All odd pins (1-49) are grounded.

GM809 and Eight Inch Drives

by D. Parkinson

Every once in a while I get an enquiry (or statement!) passed to me about the possiblity (or impossibility!) of using the Gemini GM809 disk controller card (for which I have written various BIOS's) with double density 8" drives. Rather than publish a few lines of code showing how it can be done I've presented it in the form of a short tutorial in the hope that people other than those with 8" drives and GM809 will find it useful.

Problem: How to code a routine to handle the data transfer for the Western Digital 1797 disk controller and 8" double density drives? (In the following discussion it is assumed that the target system has a Z80A running at 4MHz with no Wait states. However the same arguments apply to a 2MHz system and double density 5.25" drives, but all the times should be doubled).

It will be useful, though not essential, if the following items are to hand:

GM809 circuit diagram.
1797 Data sheet.
Z80 Data sheet.

An 8" DD disk has a serial data transfer rate of 250kbits/sec which results in a byte transfer to/from the controller every 16us (64 T-states with a 4MHz clock). Therefore it would appear that the data transfer routine must be able to move data at this rate. Can this be achieved?

Once the 1797 controller has been given a Read or Write instruction the controlling program then has to transfer the data to/from the disk under the control of the DRQ (Data ReQuest) signal from the 1797. This DRQ signal appears externally (on pin 38) of the 1797, and also internally as bit 1 of the status byte that can be read from the 1797. When the data transfer is complete (or the 1797 gives up because of an error condition), the INTRQ signal, (INTerrupt ReQuest), on pin 39 comes on. (Also bit 0, the "Busy" flag, gets reset in the internal status register). With GM809 the external DRQ and INTRQ signals are connected to bits 7 and 0 of another input port, and so may be read directly as well as via the bits in the 1797's internal status register.

Shown below is a first attempt at coding the data transfer routine for the Read command. All it has to do is transfer a byte every time a DRQ (Data ReQuest) occurs, and exit when an IRQ (Interrupt ReQuest) occurs. Note that it uses the externally read status bits.

```
        code                                          T states
        ====                                          ========
        ...set up all the registers...
LOOP:   IN      A,(STATUS)      ; Read DRQ/IRQ bits    - 11
        AND     81H             ; Mask DRQ and IRQ     -  7
        JR      Z,LOOP          ; Loop if no request   -  7 / 12
        JP      P,IRQ           ; Exit on IRQ          - 10
        IN      A,(DATA)        ; Read the data byte   - 11
        LD      (HL),A          ; Store in memory buffer-  7
        INC     HL              ; Bump the address     -  6
        JR      LOOP            ; Get next byte        - 12
IRQ:    ..........
                         Total T states (no wait for DRQ)  =73
                         Total T states (one wait for DRQ) =73+37=110
```

The AND 81H masks out the DRQ and INTRQ status bits. If neither is set the Z flag gets set and so the routine waits. Once one or other bit is set it will fall through to the following bit of code. At this point there is no need to do a further test to find out which bit was set as the AND instruction will have also set the sign bit in the flag register to indicate the state of bit 7. Thus if it is Positive (i.e. bit 7 was 0), then bit 0 must have been set to create the NZ result of the AND.

Turning now to the execution timing of the code: It can be seen that even the fastest loop is outside the time limit that has to met, so the code - simple as it is - needs revising to make the loop tighter. Setting the mask byte of 81H in a register and performing the AND with that register will save 3 T states, and making the last JR a JP will save a further 2, but that only reduces the loop to 68 T states, still 4 over the target. The "Exit on IRQ" JP in the middle of the code is a bit of a time waster as it only serves to terminate the loop and contributes nothing to the data transfer itself, and so could be moved with some benefit.

With some re-organisation the code could now look like this:

```
        ....
        LD      B,81H             ; Set the mask
        JR      LOOP              ; Skip
DRQ:    IN      A,(DATA)          ; Read the DATA byte    - 11
        LD      (HL),A            ; Store                 -  7
        INC     HL                ; Bump address          -  6
LOOP:   IN      A,(STATUS)        ; Wait for DRQ/IRQ      - 11
        AND     B                 ; Mask/test             -  4
        JR      Z,LOOP            ; Wait if neither       -  7 / 12
        JP      M,DRQ             ; Jump if DRQ           - 10
        ....                      ; Else done
                      Total states ( no wait for DRQ) = 56
                      Total states ( one wait for DRQ)= 56 + 27 = 83
```

The re-organisation has effectively removed one JP from the critical loop (at the expense of an extra JR at the start to get into the loop), and has improved the timing figures as a result. Going straight through is now well under the 64 T-state limit, but the extra 27 T-states that occur when there is a wait for the DRQ are slightly worrying. A more detailed cosideration is required to see if they matter.

Assume that a DRQ has appeared towards the end of the execution of the IN instruction that is checking the status port, and has just been missed. Now the number of T states that occur until the data byte is actually read is: 4+12+11+4+7+10+11 = 59 - so that byte will be read before the controller assembles the next byte (which comes after 64 T states). When the label LOOP is next reached the DRQ bit will be set (because it has taken >64 T-sates to get back there) so the next data transfer loop will be a short one (56 T states) and some time will be caught up. This will continue until the routine overshoots and has to wait again. Anyway it all seems possible.

The code for a Write transfer is similar:-

```
DRQ:    LD      A,(HL)
        OUT     (DATA),A
        .......
```

So the time has come to code the routines up and to try them out.

Disaster! They don't work! Why not?

When all else fails read the data sheet. For those readers who like a challenge read no further until you've found the answer in the 1797 data sheet.....

In true text book style I leave it as an exercise to the reader to work out the timing of the above sections of code!

One other point for budding Bios writers to bear in mind - The most useful (and accurate) part of the 1797 data sheet are the various flow charts of command execution. For example the text for Read/Write commands goes...

..Upon receipt of the Type II command, the busy status Bit is set. If the E flag = 1 (this is the normal case) HLD is made active and HLT is sampled after a 15ms delay......

Referring to the flow charts it can be seen that after Busy is set the "Ready" line is checked, and only if it is true will the 1797 continue with the command, otherwise it sets IRQ and terminates. If there has been no disk activity for a while there will be no Ready signal as the controller will have unloaded the heads and as a result deselected the drive. Thus, in order to get the Read/Write commands to execute, the software has to ensure that the drive is selected. (My software checks the Ready line before all Read/Writes, and issues a Seek command to the current track if it is found to be false. This activates HLD without actually stepping the heads at all, and ensures that the drive is selected and the Ready line represents the true state of the drive. All type I commands, of which SEEK is one, execute irrespective of the state of the Ready line - see the flowcharts).

Finally a note about hardware mods. These are covered in the manual supplied with the card, but don't forget that as well as changing a few straps some components need changing. The frequency of the VCO has to be doubled so capacitor C5 (100pF) should be changed to 50pF. The loop filter components (C4 and R12) also need to be changed. Early versions of the manual escaped with a misprint, C4 should be changed to 0.33uF not 33uF! One thing not noted in the manual is the fact that the "Motor on" and "Motor start-up delay" monostables are still connected through to the Ready line. These connections are not required for 8" drives and their presence is an irritant. Rather than butchering your board to get rid of them I suggest that you just disconnect one end of resistor R5. This has the effect of making the "motor on" time infinite, and the signals from the monostables will no longer interfere with the Ready signal from the selected drive.

Please note I have not written a full double-density Bios for GM809 and 8" drives - so don't try ringing up for a copy! (Anyway I've moved recently). In writing your own I suggest you follow the guide lines in the CP/M alteration guide.

Start with 8" single density (The only standard format there is!). Most 8" disks come ready formatted to this standard so you needn't worry about a format program initially. In your current environment write and test a keyboard routine and a screen routine. Next write and debug a disk read routine. If you're using the CP/M distribution disk at this point ensure it is write protected! Expand this routine to "GETSYS" (see the CP/M alteration guide). Now do the write part (using a scratch disk!) and work up to "PUTSYS". Then put the various routines together to form your Bios. Next use GETSYS to read in the system tracks of the distribution disk. Copy your Bios (and Boot) (suitably assembled) over the distribution Bios (and Boot) and use PUTSYS to write the system back to a fresh disk. Then, provided you've made no boobs, you're away, and can go on to refine the software in a disk-based environment.

WOULD YOU BELIEVE IT?

Dear Dealer, The fault on my Nascom occurs as follows. As you will see, the display is stable with some degree of flickery as the CPU works. Hitting reset, some characters change, but some stay exactly the same all the time. Now and then the computer enters the HALT state.

Besides this, the machine is dead.

Anon.