Hybrid Technology Ltd 273 The Science Park Cambridge CB4 4WE *Telephone:* Cambridge (0223) 420360

6/9/89 Issue 1

Music 2000 Interface - Direct Access

This note gives the basic information required to drive the Music 2000 Interface directly, that is, from user programs rather than AMPLE.

registers

The Music 2000 Interface is mapped into page &FC (IO page 'Fred' of the 1Mhz bus). All addresses are fully decoded so each register appears only once. The unit contains three independent ACIA devices, type 6850.

address	device	port	read function	write function
&FC00-&FC07	none	none	none	none
&FC08	ACIA 1	OUT 1	status	control
&FC09	ACIA 1	OUT 1	none	transmit data
&FC0A	ACIA 2	OUT 2	status	control
&FC0B	ACIA 2	OUT 2	none	transmit data
&FC0C	ACIA 3	OUT 3, IN	status	control
&FC0D	ACIA 3	OUT 3, IN	received data	transmit data
&FC0E,&FC0F	reserved	reserved	reserved	reserved
&FC10-&FCFF	none	none	none	none

Note: in other 1Mhz bus devices, inc. Music 5000 and Music 3000:

&FCFF	page register	none	extended addressing

bus connections

signal name	use
data	see 'registers' above
address, _PGFC, R/_W	see 'registers' above
_IRQ	from all ACIA interrupt outputs
$\overline{E}$	divided by 2, then to all ACIA clock inputs
_NMI, _PGFD, _RST, AN	not used

initialisation

For normal applications, each ACIA must be initialised before use by writing the following two values to its control register, one after the other:

binary	hex	function
0000 0011	03	set reset mode
0001 0101	15	set operational mode, and no parity, stop bit, counter divide by 16

This sets-up the ACIA for MIDI-compatible operation, including division by 16 of the 500KHz clock derived from the 1MHz bus clock.

operation

The standard methods for driving ACIA's may be used. Recommended methods include:

1 Simple transmission

The program simply stores the byte to be transmitted in the transmit data register of the ACIA, from where the ACIA will immediately send it over a period of 400us. Program timing must ensure that no further stores to the transmit data register take place during this period.

2 Polled transmission

The program repeatedly examines the ACIA's 'transmit register empty' flag (status register bit 1, hex mask 02) until it is 1 (indicating that transmission of the last byte has completed), and only then does it store the byte in the transmit data register, from where the ACIA sends it automatically.

This is the recommended method for non-demanding applications.

3 Interrupt-driven transmission

The program enables the ACIA's 'transmit data register empty' interrupt via its control register (see the 6850 data sheet for details). When this interrupt occurs, the program transfers the next byte from its own buffer to the transmit data register, unless the buffer is empty in which case it disables the ACIA interrupt (again via the control register) until a byte is inserted into the buffer.

4 Polled reception

The program repeatedly examines the ACIA's 'receive data register full' flag (status register bit 0, hex mask 01) until it is 1 (indicating that a byte has been received), and then reads the the byte from the receive data register. The program should also check the status register for evidence of faulty reception.

If the program fails to read a received byte before another is received, the first will be lost, so this method is only recommended for very undemanding applications.

5 Interrupt-driven reception

The program enables the ACIA's 'receive data register full' interrupt via its control register (see the 6850 data sheet for details). When this interrupt occurs, the program reads the byte from the receive data register. The program should also check the status register for evidence of faulty reception.

It has been found that, due to the limited speed of the BBC computer processor and its use under interrupt by the operating system, even this method cannot give 100% reliable operation in normal applications. This suggests that 100% reliable MIDI reception is not possible on the BBC computer, at least while the operating system is running.
