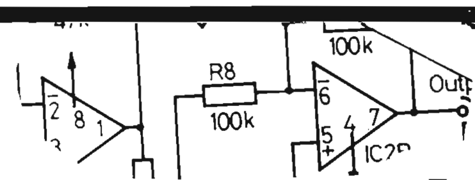
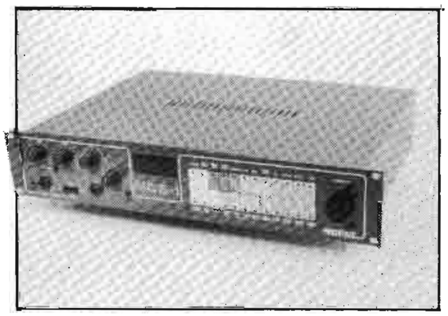


Powertran MCS1

Part 1: Playing with Time

This issue sees the start of a series of articles describing the design and construction of the MIDI Controlled Sampler, E&MM's most advanced project ever. To kick things off, we present an in-depth analysis of the effects the MCS1 can generate and how they are used, plus an insight into the workings of digital audio.

MCS1 Design: *Tim Orr, R Monkhouse, and Paul Bird*
Editorial Presentation: *Tim Orr*



Put simply, the MCS1 is a digital sampling unit, though it is in fact capable of a great deal more than that description would suggest. Any sound can be stored within the unit's memory and played back *via* either a MIDI instrument or a one-volt-per-octave keyboard. The controlling keyboard determines the pitch of the reconstructed signal, thus making it possible for the user to make music from a single natural sound: pitch bend and vibrato effects can also be added as required.

A sophisticated looping system is used to turn sound endings into a sustained loop of infinite duration, and this allows the controlling keyboard to be played without the risk of 'running out' of sound due to lack of memory space. Sampled sounds can be stored on floppy disk for later retrieval.

Recordings can be made in free-running or auto-triggered modes, and on replay the sounds can be gated or triggered. Gated operation produces a sound output – including any loops – for as long as notes on the keyboard are depressed, while triggered operation requires only a start signal.

Sound-sampling is only one aspect of the MCS1's potential, however.

The machine can also be used as a conventional digital delay line. It can be used to generate all the usual time delay effects such as phasing, flanging, vibrato, ADT and echo, and the theory and application of these effects will be discussed later. The available delay times range from a few milliseconds to tens of seconds, and other features include bypass, repeat, and infinite freeze functions. Memory size and sample speed are both continuously variable, whie a pair of tracking filters takes care of anti-aliasing and recovery filter considerations, a software power clear ensures a quiet power-up, and a click-track is provided to aid timing during long sequences.

The MCS1 front panel incorporates 24 illuminated push-switches and a continuous rotary encoder to modify parameters, a four-digit 0.6" LED display indicating parameter information. Controls for level, repeat, mix and tune are also provided, along with a level indicator.

MCS1 memory size is variable from eight bytes to 64Kbytes: storage time at a 32kHz sampling rate is two seconds, while at 8kHz, the time is eight seconds, the longest possible replay time (for special effects) being 32 seconds. Eight-bit companding analogue-to-digital and digital-to-analogue converters are employed in the sampler's design, and the unit has an overall dynamic range of 72dB, audio bandwidth being variable from 12kHz to 300Hz. There are internal four-pole tracking filters for anti-aliasing and recovery, and a programmable wide-range sweep generator, generated in software.

Control range using a MIDI keyboard is five octaves, and using a one-volt-per-octave instrument, two octaves, though transposition can be employed in the latter case to provide a range of a further five octaves.

Effects

Before going on to how the MCS1 itself works and how it can be built, it's worth devoting some time to how the various time delay effects it produces work out in theory. Figure 2 shows how these effects vary in their delay time.

Many of the effects commonly heard on today's modern music records are generated by manipulating natural sounds through a time delay unit. When a time delay is short, its effects are observed as frequency coloration, but longer delays move the effect out of the frequency domain and into the time domain. What follows is a guide to how these effects work.

Most **phasing** devices use an analogue all-pass phase-shift circuit to generate a mobile comb filter, the number of notches commonly being between two and six. The same mobile comb filter can be simulated with a time delay unit such as the MCS1, the notch spacing being equal to the reciprocal of the delay time. Figure 3 shows how notch spacing expands and contracts with varying time delays, but what it doesn't show is that subjectively, phasing is quite a subtle effect, adding depth to the output of an electric guitar, for example.

Flanging was first heard on records in the late sixties and was initially generated by extending recording tape around the 'flange' of the play head (using something like a pencil), thereby producing a varying time delay which was then mixed with the undelayed signal. This technique produces a comb filter with lots of notches, which move in the frequency domain as the time delay is altered.

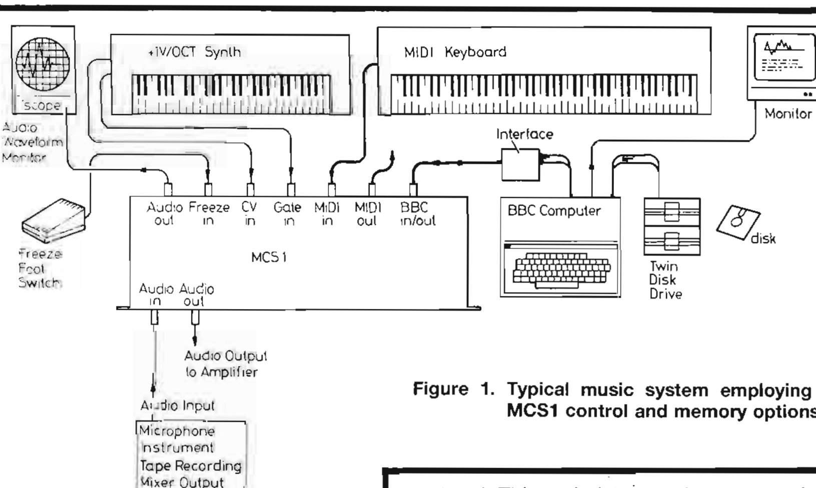


Figure 1. Typical music system employing all MCS1 control and memory options.

This effect – illustrated in Figure 4 – can be simulated relatively easily with a delay line, and by adding feedback, the filter response can be made excessively peaky, which gives modern-day flanging its characteristic tubular or 'drain-pipe' sound coloration.

By varying the time delay with a sinewave modulator, a natural sound becomes frequency modulated, and modulation frequencies between 3Hz and 8Hz produce a standard **vibrato** effect: see Figure 5. Increasing the modulation depth causes something of an extreme effect, as does increasing the modulation frequency.

Short time delays in the order of 5–50mS can be used to simulate the addition of another instrument to the one being fed into the delay

BUILD IT

line. This effect comes off because when several instruments are being played together, perfect timing between their players is something of a rarity, to put it mildly. Since sound travels at about one-foot-per-millisecond, two instruments separated by a distance of ten feet may well be out of time by 10mS, though that delay is unlikely to be constant.

These effects are known in modern technological parlance as **ADT** (auto double tracking) and **chorus**, and delay lines can simulate this natural phenomenon quite easily – see Figure 6.

Natural **reverberation**, on the other hand, is a very complex phenomenon (Figure 7). Thousands of separate time delays and reflections conspire to achieve the final sound, but it is still possible to simulate their effects, even though the hardware required to do this is also complex. However, a simple reverberator can be constructed using a single time delay: using this method, the reverb is highly coloured and is often used on human speech to simulate a metallic or robotic quality.

Time delays greater than 50mS can be heard as distinct echoes. A short single **echo** is often used in modern music to provide a sharp 'slapback' sound, while longer echo times can be used with a little repeat to simulate Alpine echoes, for example.

More recently, time delays of the order of a few seconds have been widely used to build up sequences of rhythm tracks. To achieve this effect, the delay line's repeat control is kept fully on so that the inputted sound takes several trips around the loop before it disappears. On the MCS1, a click-track gives the user an audible indication of the loop's length, while a freeze function inhibits further writing to the memory so that the stored sound(s) can repeat indefinitely without any degradation in signal quality. The replay rate can then be altered to create dramatic pitch-shifting effects.

The effects described above can be applied to all common input signals and are available on a large number of digital delay lines from various manufacturers. However, the MCS1 adds a further dimension to effect manipulation by allowing the pitch of the unit's output to be controlled by an electronic keyboard.

The stored sound is transposed up or down in pitch by varying the replay rate, though the lower the replay pitch, the longer it takes for the sound to be reproduced. When a key is pressed on the controlling keyboard, reading starts from the beginning of memory and continues until either the key is released or the memory is exhausted, whichever is sooner. However, the MCS1 incorporates a looping facility that enables a continuous loop to be constructed at the end of a sound, giving it sustain for as long as a note on the keyboard is depressed. This method is widely employed by designers of electronic percussion units whose budget does not allow them to use ten EPROMs to store the sound of a cymbal! Figure 8 gives a graphic illustration of the looping process.

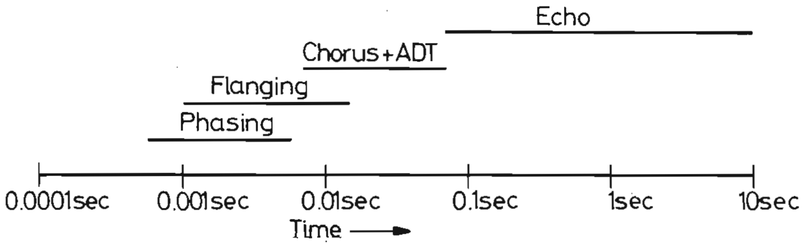


Figure 2. Some popular time delay effects.

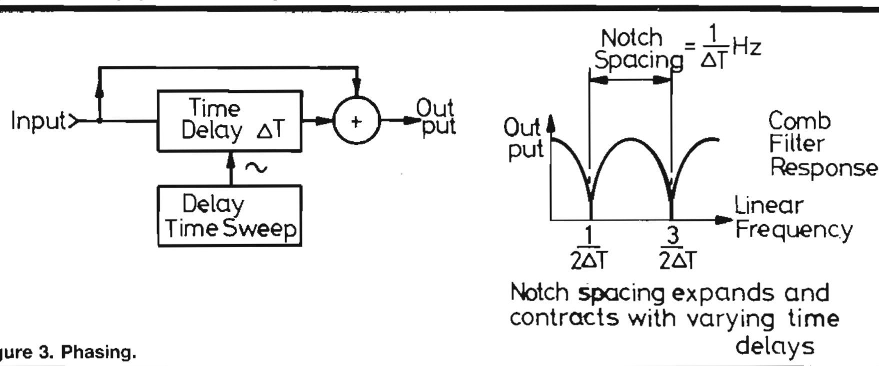


Figure 3. Phasing.

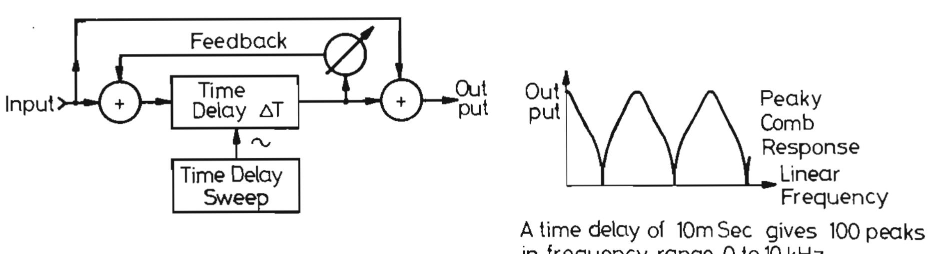


Figure 4. Flanging.

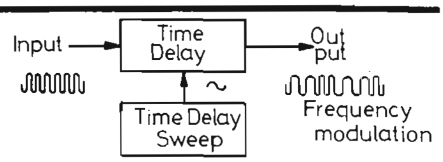


Figure 5. Vibrato.

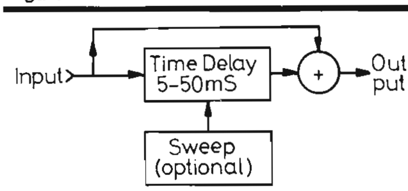


Figure 6. ADT and Chorus.

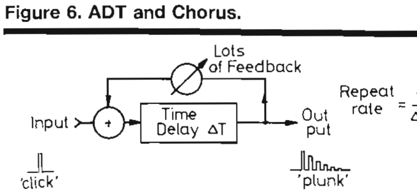


Figure 7. Reverb.

Digital Audio

In order for any audio signal to be stored in digital memory or held on a magnetic storage medium (eg. a floppy disk), it must first be converted into binary code. Once the signal has been digitally encoded, it can be stored and manipulated without any of the risks associated with analogue storage methods: noise does not accumulate, and the distortion level remains constant when the audio data is transferred from one unit to another.

A typical digital audio system is shown in Figure 9, where the sample and hold unit is used to freeze the input signal so that the ADC can perform a conversion on a static signal. The low-pass filter removes out-of-band frequency components after both stages of conversion have taken place. The audio signal is converted into a stream of digital information by the ADC (analogue-to-digital converter) and is then re-converted into an audio signal by the DAC (digital-to-analogue converter, simple isn't it?)

How do these converters work?

Well, think of the ADC as a sort of rapid digital voltmeter that measures the magnitude of the input voltage at regular time intervals. Each time it completes a measurement (the process is known as 'performing a conversion') it outputs a binary word representing the magnitude of the input voltage at that point in time. If the binary word is eight bits wide, the converter is capable of resolving the input voltage into 256 (2 to the power of 8) individual levels. The resolution of an ADC is proportional to the size of the binary word it produces, so that, for example, a 12-bit system has a resolution of one part in 4096 and a 16-bit system has a resolution of one part in 65, 536.

The DAC is then used to convert the binary words back into an analogue voltage, and because the voltage is directly proportional to the magnitude of the binary code, the bit size of the DAC determines the number of separate voltage levels. However, the DAC's output is only a 'square wave' approximation to the original analogue input (which can move up and down smoothly) and this effect is known as quantisation – the digital equivalent of distortion, shown in Figure 10. Its effect can be reduced by increasing the bit size of the system as a whole, but as is the way of things, this invariably increases the system cost.

When we digitise an audio signal, we sample it at regular intervals of time, and in doing this, we define the time-varying shape of the signal as a series of points. By joining up these points (this is accomplished by the DAC) we can reconstruct the original signal. But how often should the signal be sampled, and does the sampling rate affect the system bandwidth? The answers to these questions lie in sampling theory. This states that a sinusoid defined by only two samples is recoverable, which in turn implies that the system bandwidth can be as much as half the sample frequency. In practice, however, the bandwidth is usually limited to about one-third of the sample frequency, due to filter limitations.

If a signal is sampled at a frequency of less

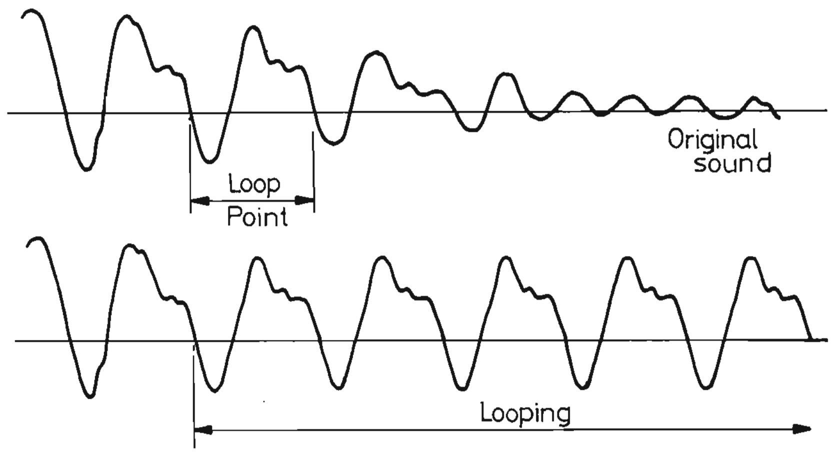


Figure 8. Sustaining a sound by turning a single cycle into a loop.

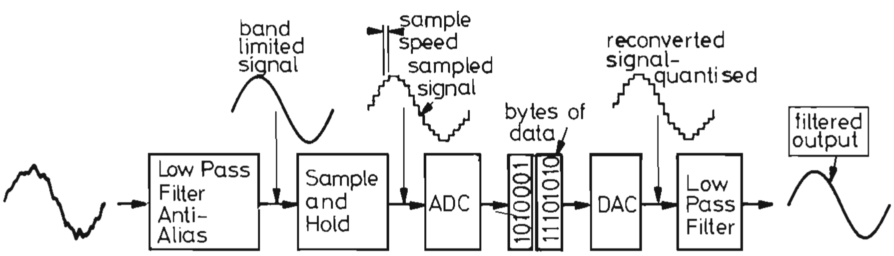


Figure 9. Digital Audio.

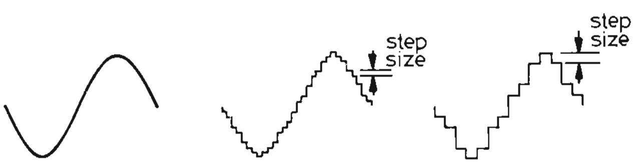


Figure 10. Quantisation distortion.

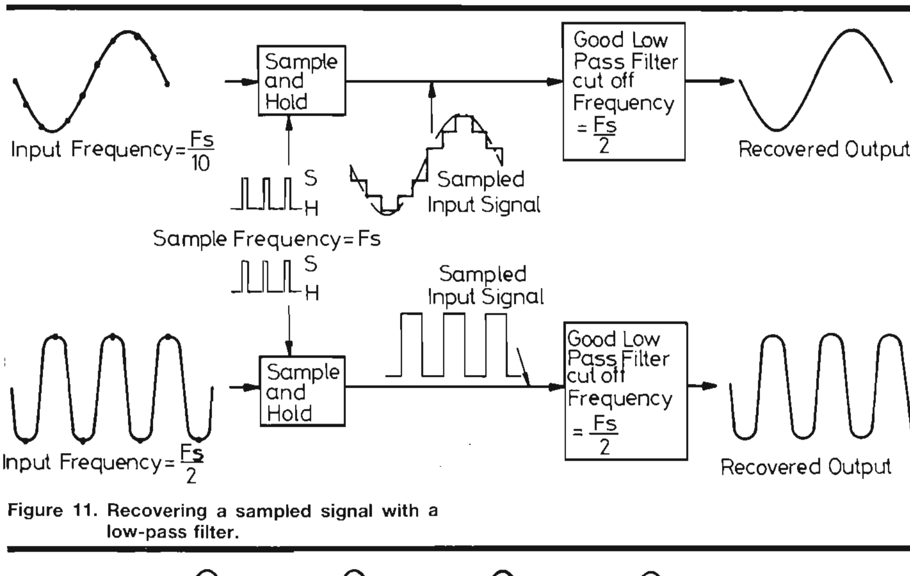


Figure 11. Recovering a sampled signal with a low-pass filter.

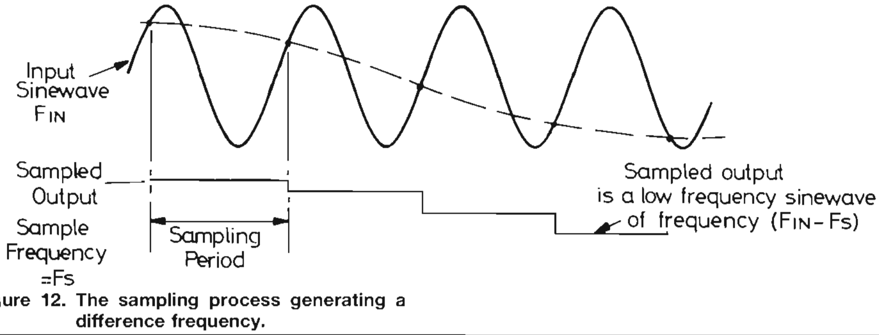


Figure 12. The sampling process generating a difference frequency.

than twice the signal bandwidth, there's a chance of frequency domain distortion – or 'aliasing' – taking place. Consider the sinewave being sampled at a frequency less than its own, as in Figure 12. The sampling process is meant to result in the original signal being recovered, but what is generated instead is a difference frequency. The resultant sound is something akin to ring modulation or detuned SSB reception, ie. very disturbing when applied to complex signals such as music or speech.

However, aliasing can be prevented by bandlimiting the input signal to one-half of the sample frequency using a good low-pass filter, which merely removes the signals that would cause aliasing products.

There's always a temptation to slow down the conversion frequency so that long time delays can be obtained from a fixed memory size, so it has become a practical necessity to incorporate a tracking low-pass filter in the design of some delay line produts. The low-pass filter that precedes the ADC is known as the anti-aliasing filter.

In the first graph in Figure 13, the shaped area represents the power density spectrum of a typical audio signal, while the second drawing shows the same spectrum sampled at a frequency of Fs. Note that the lower sideband has an inverted spectrum and that the sideband pairs repeat at integer multiples of Fs. In the third diagram, the sample frequency has been reduced to 2 × FA, and the lower sideband is close to the audio base band: as a result, the system is on the verge of generating aliasing components. Finally, the fourth graph shows what happens when the sample frequency is reduced below 2 × FA. The low-pass filter is now allowing frequency components which generate aliasing signals to pass through, and aliasing begins.

Quantisation Noise

This phenomenon is caused by the inability of digital components to reproduce an arbitrary analogue signal accurately: a smooth analogue signal is presented to the ADC at the start of the process, but a crunchy output signal is reproduced by the DAC at the end of it. We can use distortion measuring techniques, normally used to measure THD (total harmonic distortion) in linear amplifiers, to examine quantisation noise. Figure 14 demonstrates the effect of quantisation.

Generally speaking, quantisation distortion has the spectral properties of noise. Because there is no simple integer relationship between the input signal and the sample frequency, the quantisation distortion bears no simple relationship to the input signal and therefore sounds noise-like.

Now, if a 1kHz sinewave were sampled 20 times faster at 20kHz, the resulting output would contain no quantisation noise as such, because the quantisation distortion would always be in the same place on each cycle of the sinewave and would therefore be heard as harmonic distortion.

A linear converter has quantisation levels at fixed, linearly-spaced intervals, and the best signal-to-quantisation noise ratio for a linear converter is given by the formula S/QN = (N × 6)dB, where N is the bit size of the converter. Thus, an eight-bit converter has quantisation noise 48dB below the maximum signal level. When the maximum signal level is only a quarter of the maximum, the ratio is poorer by 12dB, falling to 36dB, and if the input signal is so small that only the LSB (least significant bit) of the code is changing, then the S-to-QN ratio is only 6dB! However, when the input signal is so small that no bits are changing at all, then

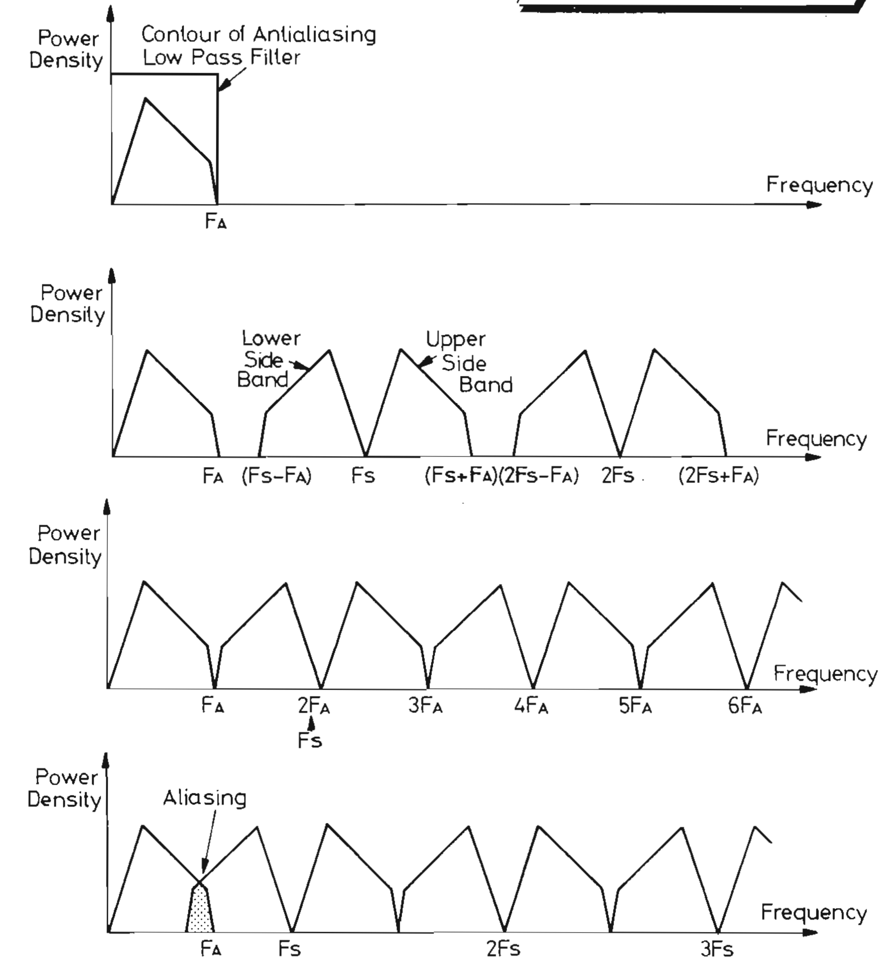


Figure 13. Aliasing.

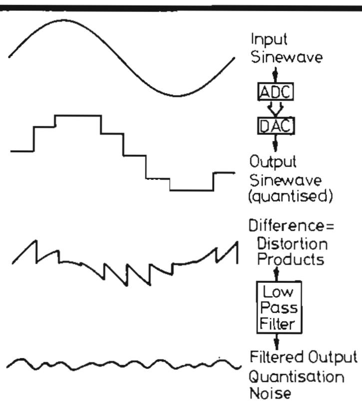


Figure 14. Quantisation noise.

no quantisation noise is generated, thus proving that the process is a distortion mechanism rather than a noise one.

Dynamic range is the ratio of the biggest signal level divided by the smallest the converters can handle. For linear converters, the dynamic range is represented by the S-to-QN ratio, or 48dB for an eight-bit system. If the quantisation levels are logarithmically spaced, ie. with small step sizes for low signal levels and large step sizes for higher ones, a somewhat larger dynamic range can be obtained. This is illustrated in Figure 15.

The DAC88 used in the MCS1 has a dynamic range of 72dB, and the log law is used to compress the signal at the ADC and expand it again at the DAC. For obvious reasons, this form of converter is known as a companding device, and is well suited to handling natural sounds such as speech and music, which require a large dynamic range if they are to be reproduced with any realism. However, it should be remembered that whenever an improvement is made to the dynamic range, quantisation distortion is invariably made worse.

Some of the effects of the degradation caused by quantisation distortion can however be masked by using a frequency pre-emphasis before the ADC and an equal (but opposite) de-emphasis after the DAC. This principle is shown in Figure 16.

How does it work? Well, because the spectral energy of sound rolls off with increasing frequency, it's possible to add high-frequency lift without running into any clipping problems, though if the sounds being processed are a little on the bright side, the system has to be run at a lower operating level.

That about wraps up the theory side of digital audio and the effects that can be generated by digital delay lines: I hope what we've discussed has cleared up a few grey areas that might have existed in some people's minds, as well as providing some 'food for thought' for budding designers and constructors. Next month, we'll take a closer look at the MCS1's circuit operation. ■

Pricing and availability details of the MCS1 will be announced in a forthcoming issue of E&MM, but in the meantime, you can get further information from the suppliers, Powertran Cybernetics, at Portway Industrial Estate, Andover, Hants, SP10 3EM.

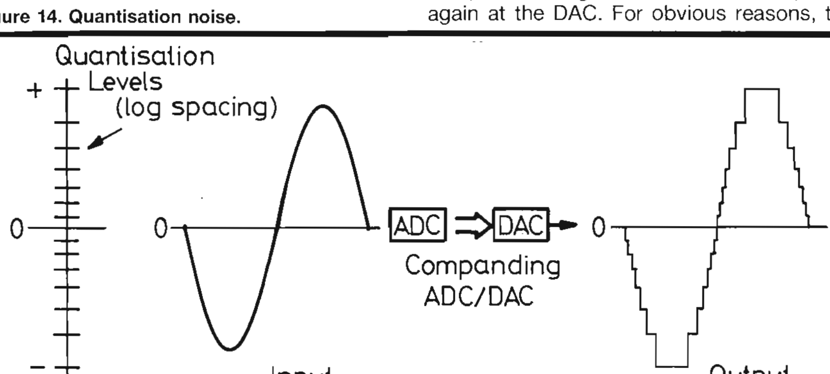


Figure 15. Companding.

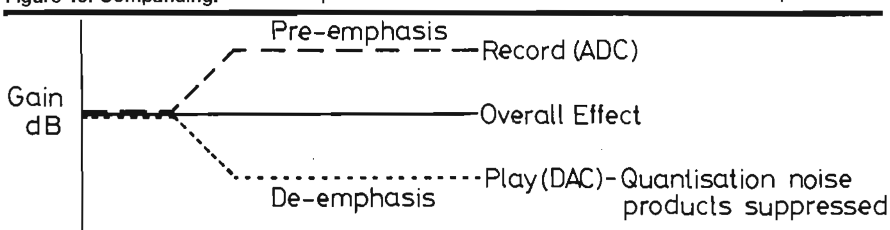


Figure 16. Pre-emphasis noise reduction.

BUILD IT

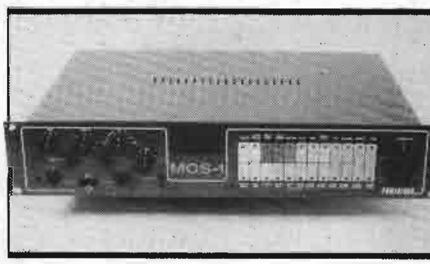
Powertran MCS1

Part 2: The Circuit

E&MM's most advanced project ever is a digital delay line and sound sampler in one unit. The second part of our constructional series includes the circuits behind the MCS1's operation.

As explained last month, the Powertran MCS1 is a digital sampling unit that can also function as a high-specification digital delay line. The pitch of the sound stored within the MCS1's memory can be controlled *via* either MIDI or one-volt-per-octave keyboards, and the kit itself, although complex, represents a significant cost saving over any other commercially-available ready-built product.

The following four pages contain the circuit diagrams that lie at the heart of the MCS1. The first of these incorporates the master clock generator from which the whole delay line is driven, the second includes the components used for memory address functions, the third includes the microprocessor, MIDI and BBC B computer connections, while the fourth contains the filtering and companding stages.

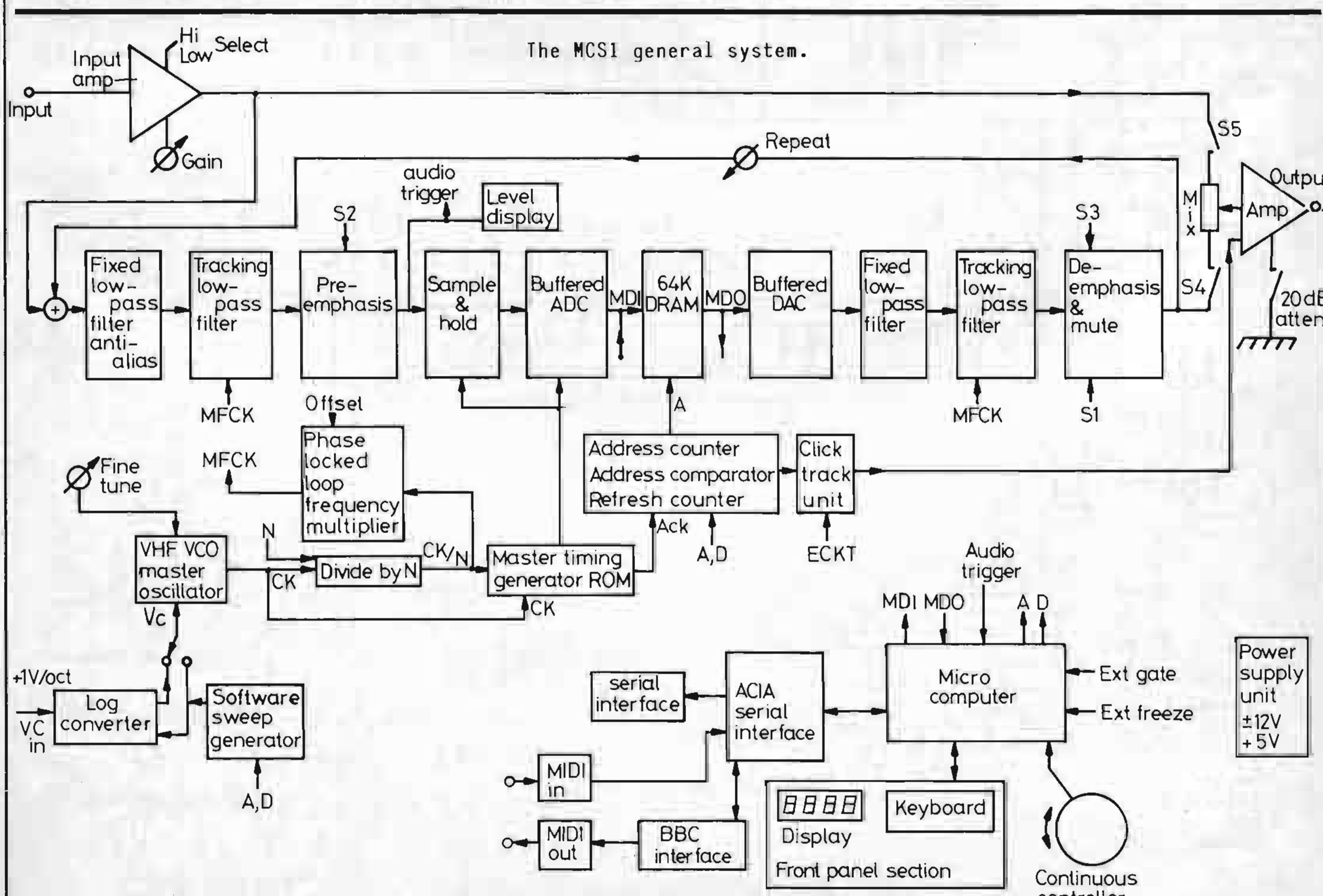


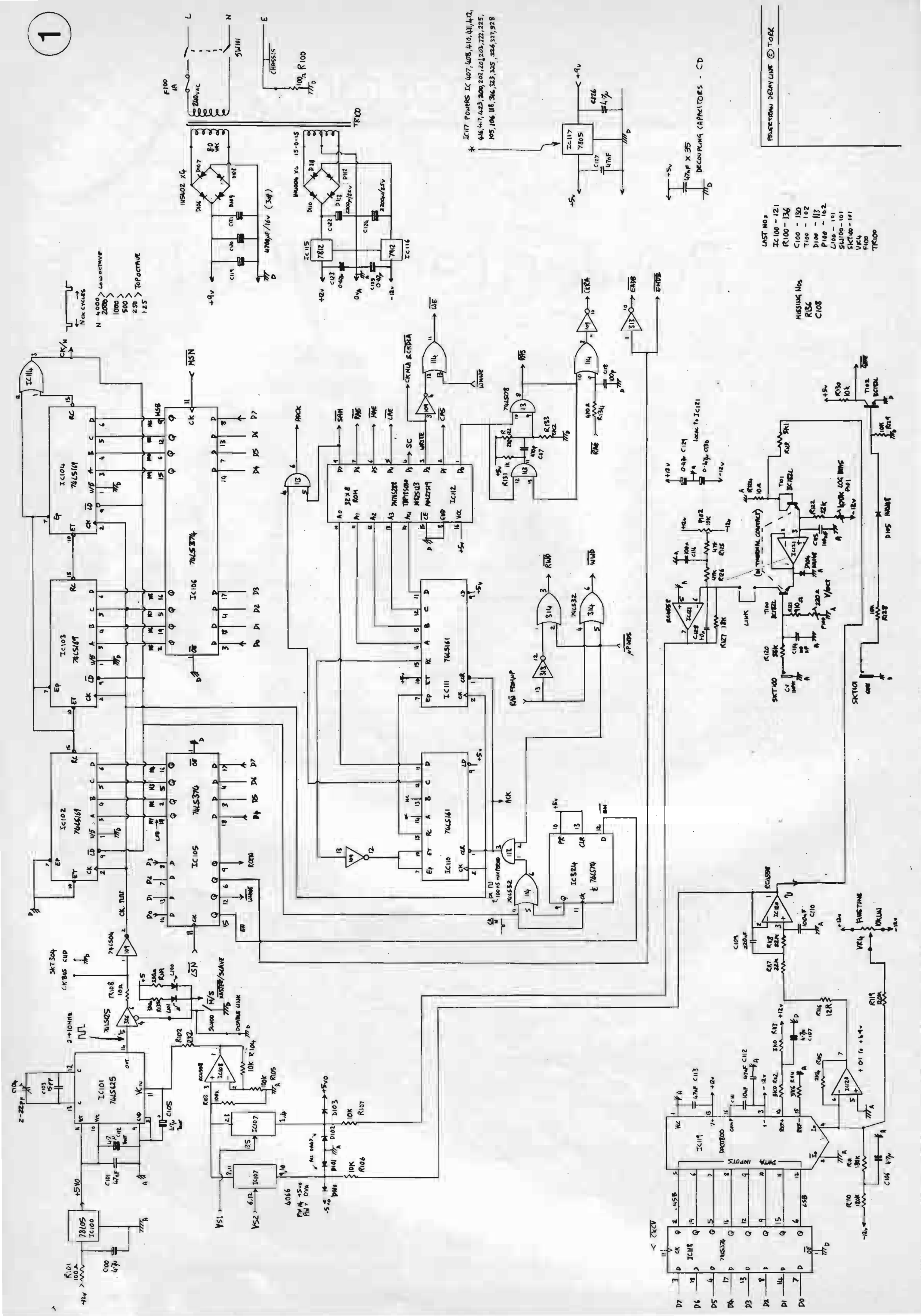
For the less technically-orientated, the table on this page is a block diagram that displays the MCS1's operating system in what are more or less layman's terms.

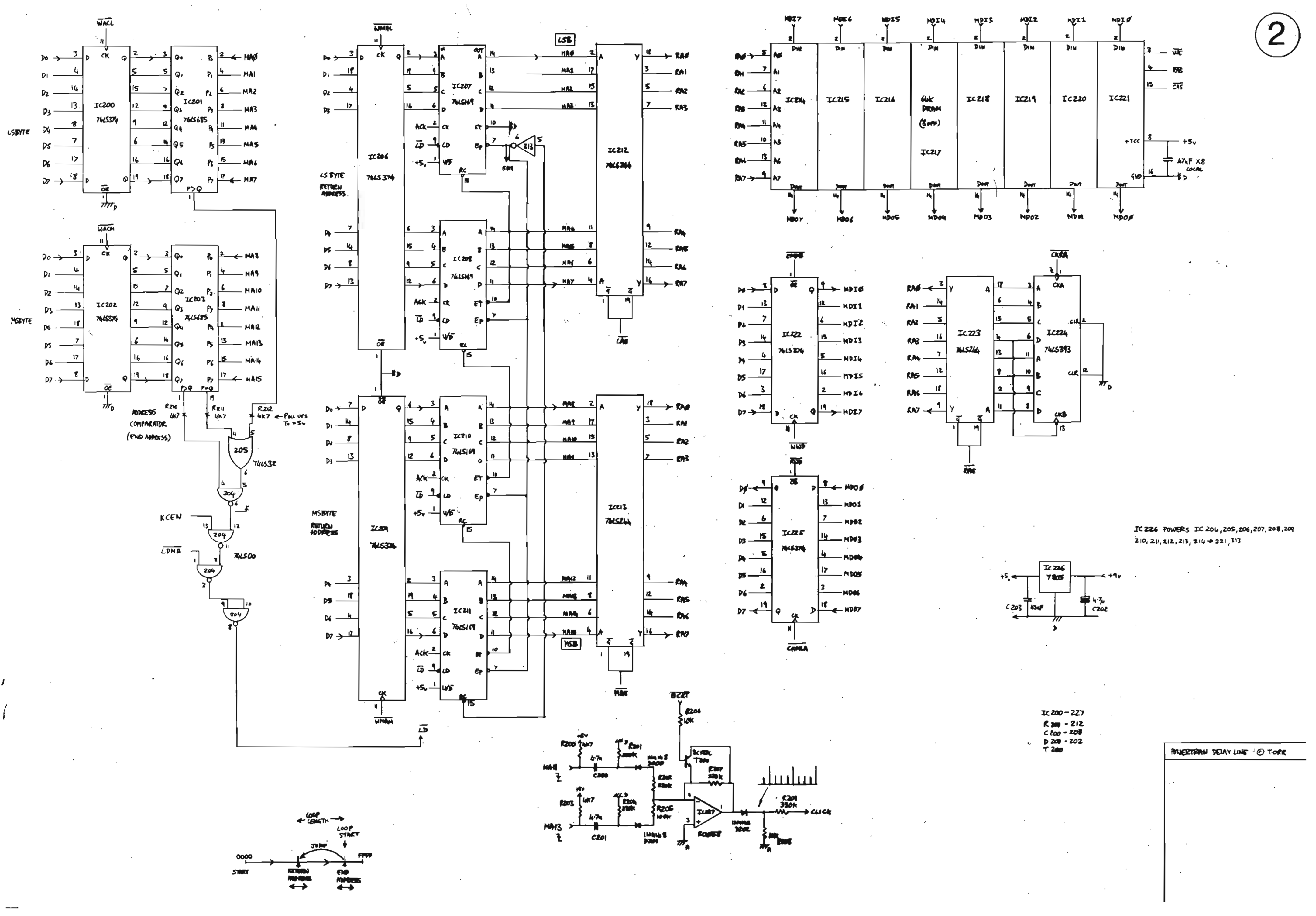
Next month will see a full circuit description related to the diagrams printed on the following pages, so whatever you do, don't lose this issue before the next E&MM appears on the bookstalls!

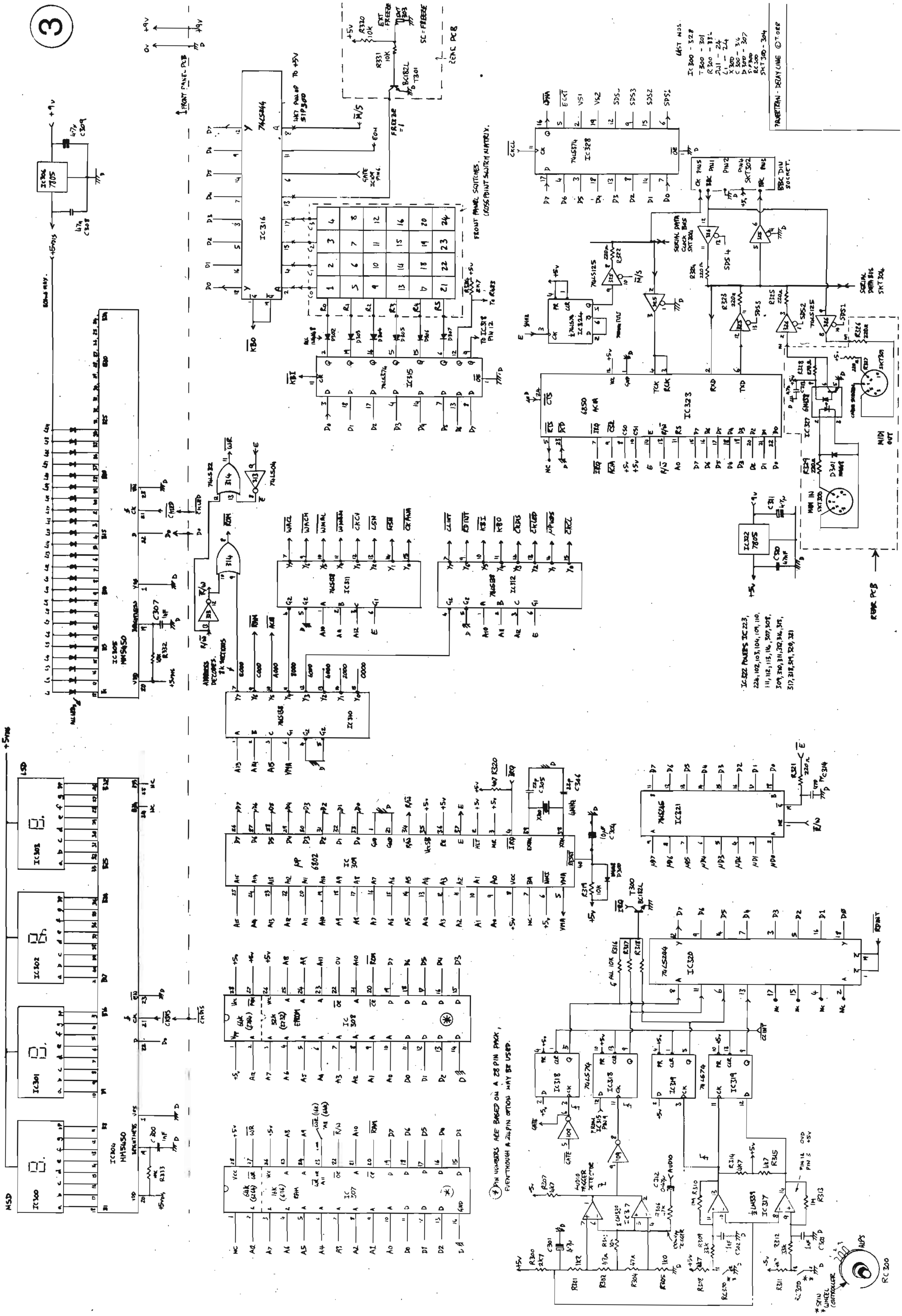
Meanwhile, selling prices for the MCS1 in both kit and ready-built form have now been fixed at £499 and £699 respectively, to which VAT must be added at the current rate. Orders are now being taken by the manufacturers, Powertran Cybernetics Ltd, Portway Industrial Estate, Andover, Hants, SP10 3EM. ☎ (0264) 64455. ■

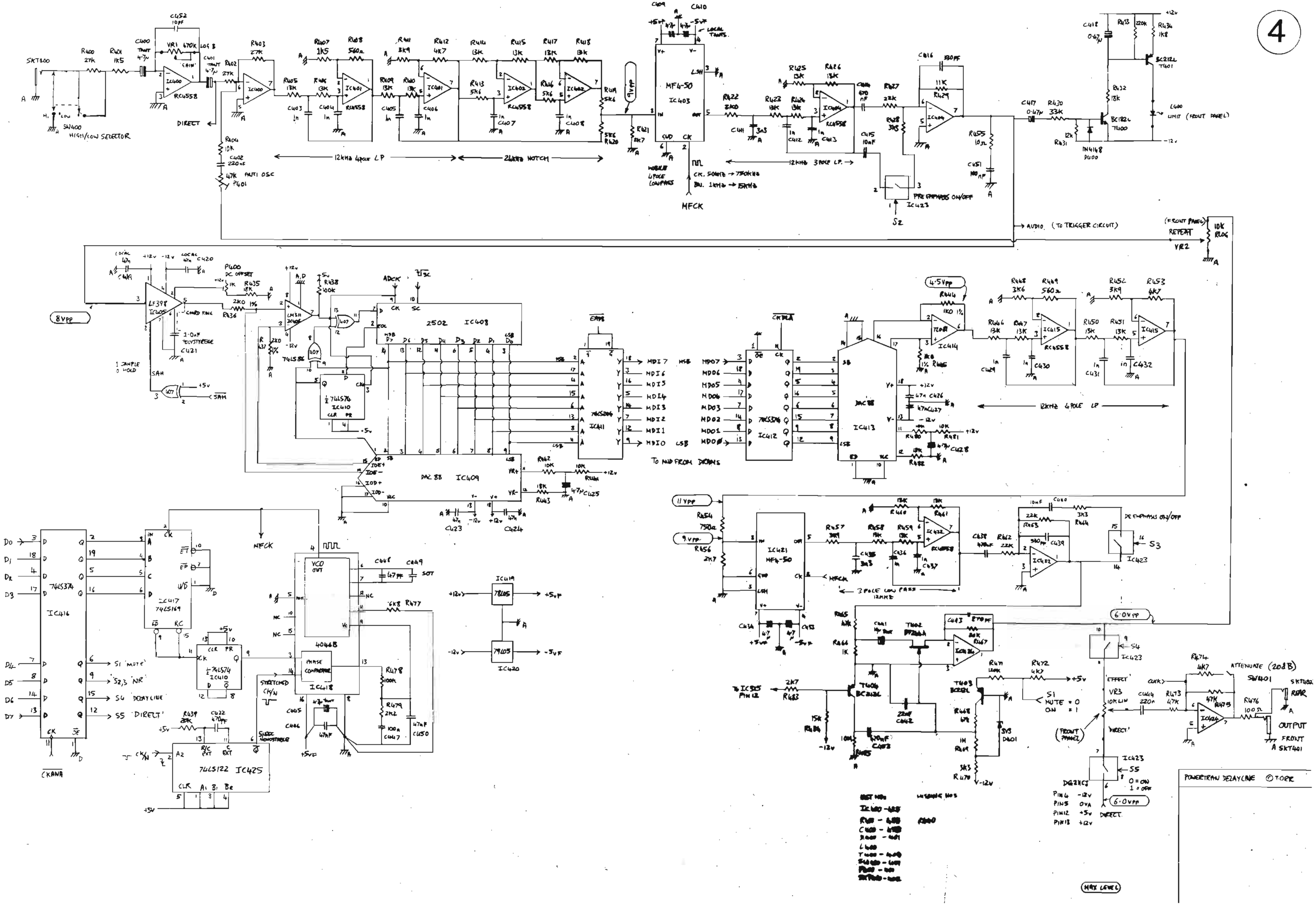
The MCS1 general system.

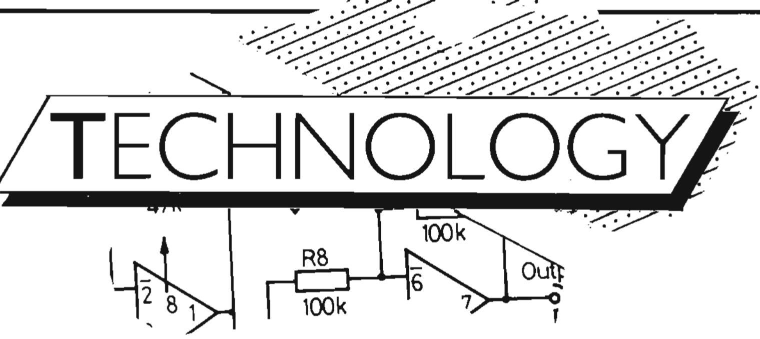



1
POLYSTROM DELAY LINE © TORK
DECOUPLING CAPACITORS - CD
LAST NOs
MISSING NOs
IC101 74LS625
IC102 74LS169
IC103 74LS169
IC104 74LS169
IC105 74LS374
IC106 74LS374
IC110 74LS161
IC111 74LS161
IC112
IC118 74LS374
IC119 DAC0800
IC100 78L05
IC115 7812
IC116 7912
IC117 7805
FINE TUNE
DATA INPUTS
MASTER/SLAVE
DISABLE LINK
LOW OCTAVE
TOP OCTAVE


2
IC226 POWERS IC 204, 205, 206, 207, 208, 209
210, 211, 212, 213, 214 → 221, 313
IC200 – 227
R 200 – 212
C 200 – 203
D 200 – 202
T 200
ADDRESS COMPARATOR (END ADDRESS)
LS BYTE RETURN ADDRESS
MS BYTE RETURN ADDRESS
LSBYTE
MSBYTE
LOOP LENGTH
LOOP START
JUMP
START
RETURN ADDRESS
END ADDRESS
KCEN
CLICK
PULL UPS TO +5v
47nF X8 LOCAL




SKT400
SW400 HIGH/LOW SELECTOR
DIRECT
12kHz 4pole LP
24kHz NOTCH
ANTI OSC
MF4-50
IC403
MOBILE 4POLE LOWPASS
CK. 50kHz → 750kHz
BW. 1kHz → 15kHz
MFCK
12kHz 3POLE LP.
PRE EMPHASIS ON/OFF
IC423
S2
LED LIMIT (FRONT PANEL)
AUDIO (TO TRIGGER CIRCUIT)
(FRONT PANEL) REPEAT
VR2
8Vpp
LF398
IC405
1 SAMPLE 0 HOLD
SAH
DC OFFSET
POLYSTYRENE
GUARD RING
2502 IC408
74LS74
IC410
DAC 88 IC409
74LS244
IC411
74LS374
IC412
IC413
TO AND FROM DRAMS
4.5Vpp
12kHz 4POLE LP
11Vpp
9Vpp
IC421
MF4-50
3 POLE LOW PASS 12kHz
DE EMPHASIS ON/OFF
S3
6.0Vpp
74LS374
IC416
74LS169
IC417
VCO OUT
4046B
PHASE COMPARATOR
IC418
STRETCHED CK/N
5μSEC MONOSTABLE
74LS122 IC425
IC419 78L05
IC420 79L05
TO IC505 PIN 12
S1 MUTE = 0 ON = 1
S1 'MUTE'
S2,3 'NR'
S4 DELAYLINE
S5 'DIRECT'
'EFFECT'
'DIRECT'
(FRONT PANEL)
VR3 10K LIN
ATTENUATE (20dB)
SW401
SKT402 REAR
OUTPUT FRONT
SKT401
0 = ON 1 = OFF
DG211CJ
PIN4 -12V
PIN5 0VA
PIN12 +5V
PIN13 +12V
MAX LEVEL
POWERTRAN DELAYLINE ©TORC



Powertran MCS1

Part 3: How it Works

Last month saw the MIDI Controlled Sampler's circuit diagrams reproduced in full. Now we take a look at what those circuits add up to in real life. *Tim Orr*

To re-cap for those who might have missed the first two parts of this series, the Powertran MCS1 is a digital delay unit that can also act as a sound-sampling device. This means that in addition to providing all the commonly used time delay effects such as chorus, echo, flanging and so on, the MCS is also able to sample externally-generated sounds digitally and store them in memory. The pitch of the stored sound can then be altered *via* either a MIDI keyboard or a one-volt-per-octave one.

The description that follows refers to the four circuit diagrams we published in E&MM November, so it's important you have that issue in front of you, otherwise you won't understand a word of it (some of you probably won't understand a word of it anyway, but . . .). Although the reproduction of some parts of the circuit wasn't quite as good as we would have liked, you shouldn't experience too many problems in identifying which bit goes where.

With luck, this description should assist newcomers to the world of electronics design in getting acquainted with some of the principles behind the MCS1, as well as giving experienced readers a detailed insight into how well-tried design conventions have been applied carefully to make the Powertran unit a high-performance machine whose cost-effectiveness is second to none.

Diagram 4

I know it sounds strange, but let's kick things off with a look at the last of the four drawings published in November.

IC 400 is a variable-gain preamplifier whose input sensitivity can be selected for high-level or low-level operation from a low-impedance microphone. The RC4558 is a low-noise op-amp that's used at various points throughout the MCS1's design.

IC401 forms a four-pole low-pass filter followed by IC402, which forms a notch. The combination of these two creates an elliptic filter (Figure 1) with a 12kHz low-pass response and a notch at 24kHz: this filter stage precedes the mobile tracking filter, IC403. This is a switched-capacitor four-pole low-pass filter presented in an eight-pin package and manufactured by National Semiconductor. The break frequency is determined by dividing the input clock frequency by 50 (ie. MFCK/50). The package is used in this circuit as an anti-aliasing filter (see Part 1, E&MM October, for an explanation of aliasing and what can be done to eliminate it), but seeing as it is itself a sampled-data device, it needs an anti-aliasing filter of its own, albeit at a frequency 50 times higher. This explains why the fixed active filter precedes the mobile one.

A further low-pass filter, IC404, is used to provide further filtering and to remove any high-frequency clock breakthrough from the mobile filter. The MF4-50 (IC403) is claimed to have a dynamic range of 80dB. In other words, it can pass a peak signal of 2.8V rms and the residual noise, band-limited to 20kHz, will be 0.282mV rms. I measured it and made it 79dB, which for a noise measurement is *very* close. Anyway, the second half of IC404 is the pre-emphasis circuit, and this can be switched on to give treble lift, thereby masking quantisation noise (again, see Part 1 for details) at the de-emphasis stage. The complete anti-aliasing filter stage is shown in Figure 2.

The signal is now ready to be converted into digital data by the analogue-to-digital converter.

IC405 is a sample-and-hold device, and this freezes the analogue signal long enough for a conversion to be performed. The ADC itself comprises IC406 (a fast voltage comparator), IC408 (a successive approximation register, or SAR for short), IC409 (a companding DAC set in encode mode) and IC407 and 410, steering logic.

How does the converter operate? Well, the SAR is given a command called SC (for Start Conversion) and a clock signal, ADCK. The SAR then performs a series of tests on the frozen analogue input, and these are as follows.

First, it tests the MSB (Most Significant Bit) of the code to a 1 and all the others to zero: the DAC output can then be compared to see if it's bigger or smaller than the analogue input. If the DAC output is smaller (ie. less positive) than the analogue input, the MSB is stored as a 1. The next bit of the code can then be tested by setting it to a 1, and the whole process is repeated. In fact, all the bits are tested in this way, the results being stored by the SAR.

As this process continues, so the DAC output successively approximates towards the magnitude of the analogue input. After eight tests, the conversion is complete, and the DAC output is then equal to the analogue input, ± ½ LSB. As far as time is concerned, the whole conversion process takes nine ADCK periods, and when it's completed, the output data becomes stable and can be written into the memory.

Data is read from the memory by latching it into IC412 and then feeding it into IC413, a companding DAC set into encode mode. The DAC output can be seen at the output of IC414.

IC415 forms a four-pole low-pass filter, used to recover the analogue signal from the 'crunchy' DAC output. The signal is then filtered by another mobile four-pole low-pass filter, IC421. Some replay rates are very slow (perhaps as low as 2kHz), so the mobile filter has to track at least an octave below this if the 2kHz sampling rate is not to be too noticeable. Again, the mobile filter is followed by a fixed low-pass one, IC422: the second part of this is used to provide the de-emphasis circuit.

T402, T403 and IC424 form a simple voltage-controlled attenuator used in the MCS1 as a mute circuit, the first-mentioned being a junction FET (field-effect transistor). When the voltage on the gate is about −3V, the FET is turned off, and the channel resistance is about 50Mohms, the attenuation through the circuit in this mode being about 90dB. By comparison, when the gate voltage is 0V, the FET is turned on and has a channel resistance of only 400ohms. This attenuates the analogue signal as seen at the positive end of C441 to about ±60mVp, and therefore allows distortion operation through the FET. See Figure 3. The last stage of IC424 is an output amplifier driven by the mix between direct and delayed signals.

The MFCK signal that drives the switched-capacitor filters is generated by IC416, 417, 418 and 410. This is a phase-locked loop (PLL) which multiplies the system conversion frequency (CK/N) by a number between 8 and 32 – this is the filter offset. The break frequency of the mobile filters is given by the equation:

$$Fb = \frac{CK/N \times Z \text{ Hz}}{50}$$

where Fb is the break frequency and Z the filter offset.

Circuit operation is as follows. The CK/N pulse is fed into the phase comparator of the phase-locked loop, while the output of the PLL VCO is used to clock a down counter, IC417. This counter counts down to zero, whereupon the RC output causes it to load in a four-bit code. The counter is then loaded with this code and proceeds to count down to zero again: thus a programmed division is performed. The RC output is divided by 2 by IC410, which generates the square-wave output fed into the other half of the phase comparator. The feedback loop is now complete.

The PLL VCO will adjust itself to be equal to CK/N multiplied by the total loop division number, the latch IC416 being used to store this number. Occasionally, the PLL will be commanded to generate an output frequency in excess of 700kHz, but will not be able to do this simply because its own VCO cannot exceed this frequency. And seeing as the MF4 mobile filters have a clock frequency maximum of 1MHz, this enforced limitation is actually desirable.

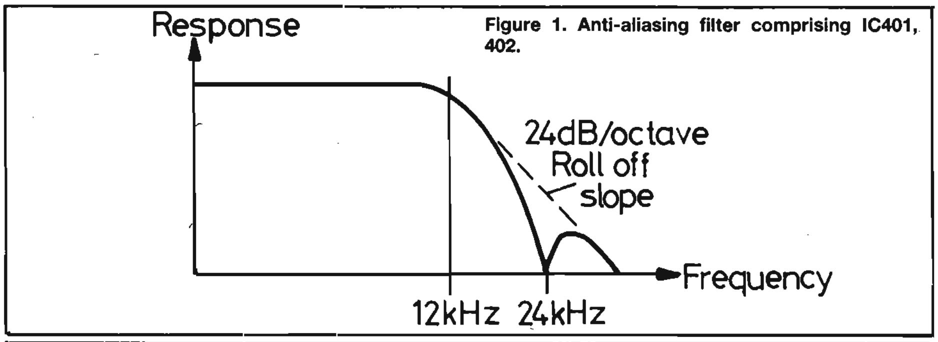


Figure 1. Anti-aliasing filter comprising IC401, 402.

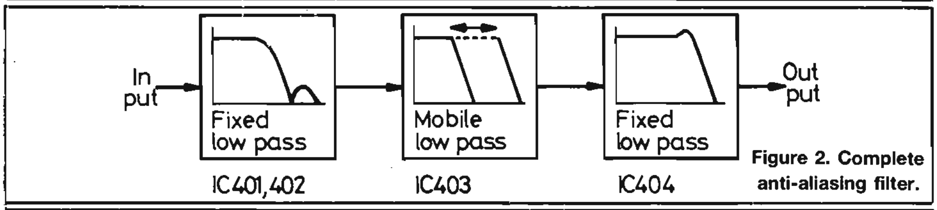


Figure 2. Complete anti-aliasing filter.

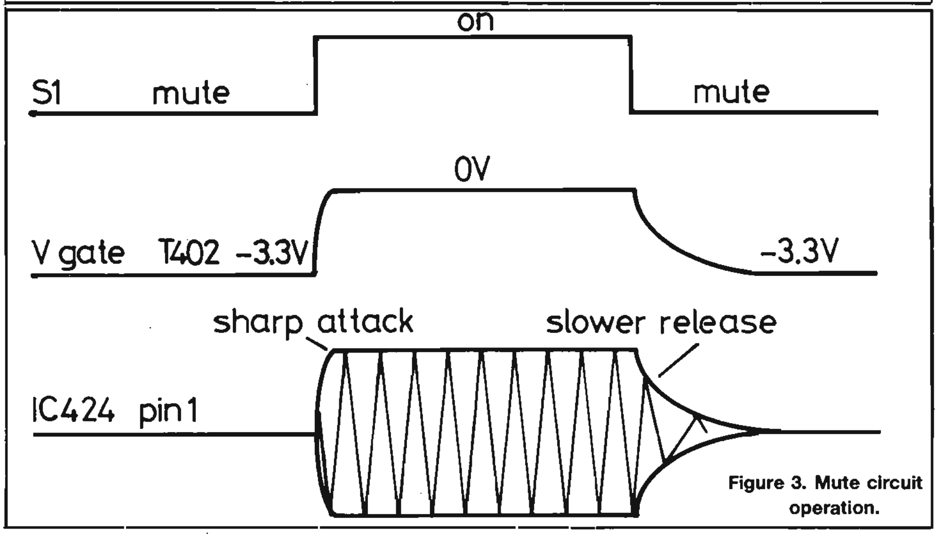


Figure 3. Mute circuit operation.

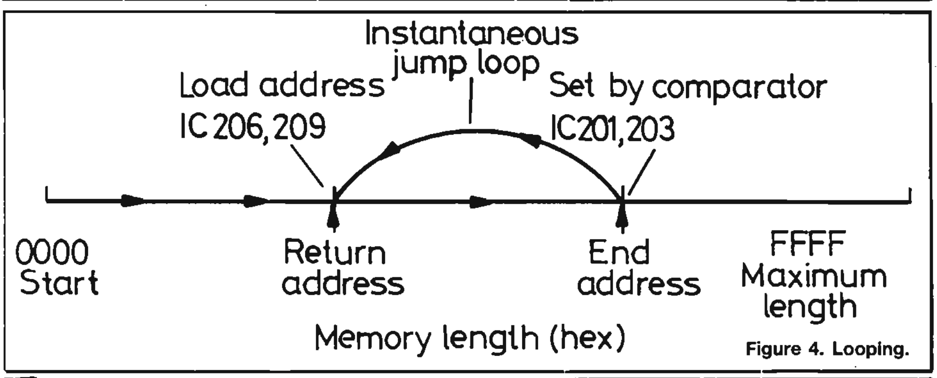


Figure 4. Looping.

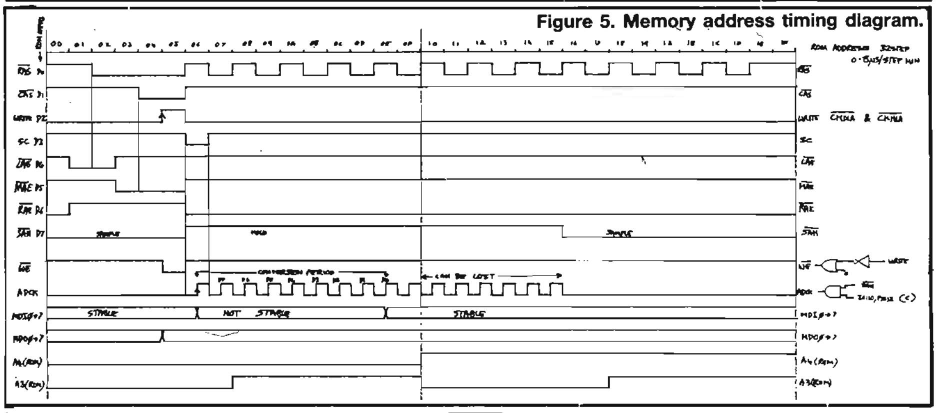


Figure 5. Memory address timing diagram.

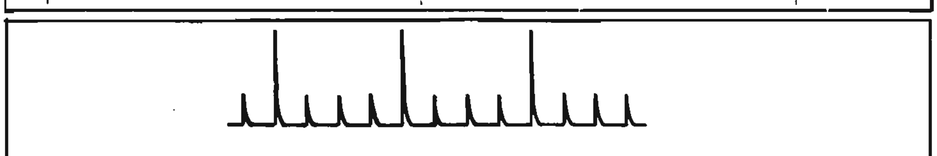


Figure 6. Click-track signal.

Diagram 2

The MCS1 memory is 64Kbits long, and therefore requires a 16-bit address counter in the shape of IC207, 208, 210 and 211. These counters are used to count through memory locations for both Record and Play functions. They can be set to any 16-bit address simply by being loaded with the data stored in latches IC206 and 209. This data is H0000 for the Delay Line mode and front panel-selectable in Edit mode.

A 16-bit comparator (IC201, 203) compares the memory address with the data held in latches IC200 and 202, and when the memory address is greater than or equal to the address stored in the latches, it generates a load pulse for the memory address counters. IC200 and 202 hold the memory end address and IC 206 and 209 the memory return address. See Figure 6.

The logic hardware performs all the looping functions, with the MCS1's microprocessor merely setting up the two 16-bit parameters. Note that the circuit allows these parameters to be any number between the start and finish of the memory – this makes both very short memory lengths and continuously variable loop lengths possible.

IC212 and 213 are used to multiplex the memory address into the MCS1's DRAMs. First, IC212 is enabled, and the bottom eight bits of the memory address are entered into the DRAMs as the ROW address. Next, IC213 is enabled and the top eight bits of the memory address are entered as the DRAM's COLUMN address (see Figure 5).

The DRAMs are actually 64K dynamic devices and are refreshed by performing ROW reads. This is why the fastest moving part of the memory address (the LSB end) is used to select the ROWs. In fact, many of the refresh requirements are generated by the natural reading process. The refresh time as quoted by manufacturers is usually between 2 and 4 milliseconds: that is, the refresh electronics should perform a dummy read on each row every 2–4mS. If for some reason this doesn't happen, there's a possibility that the contents of the memory will be corrupted.

I've tested the MCS1 DRAMs myself and found they actually needed a refresh every 25 seconds, whereas the makers specify 4 milliseconds. They might from time to time get a duff memory cell that actually discharges itself in that time, but personally I have my doubts . . .

Anyway, to make absolutely certain that no refresh problems are encountered, a high-speed refresh counter (IC224) and buffer (IC223) are used to perform the dummy reads.

Data can be transferred from the DRAMs to the microprocessor data bus *via* two latches, IC222 and 225. These routes are used to transfer the memory data to and from the external floppy disk, and also to clean out the memory on powering-up the MCS1. Just to illustrate how important this function is, imagine getting 10 seconds of digital junk blasting out of the audio output every time you turn the MCS1 on.

Moving on, a click-track (IC227) uses the MSBs of the memory address counter to contrive a metronome beat, and a graphic illustration of this is shown in Figure 6.

Diagram 1

The whole delay line (but not the microprocessor) is driven from a master clock generator IC101, which runs at frequencies between 2.5MHz and 10MHz. The oscillator is voltage-controlled, and can therefore be controlled by a voltage from any conventional one-volt-per-octave music synthesiser. Obviously, this voltage has to be converted into an exponential

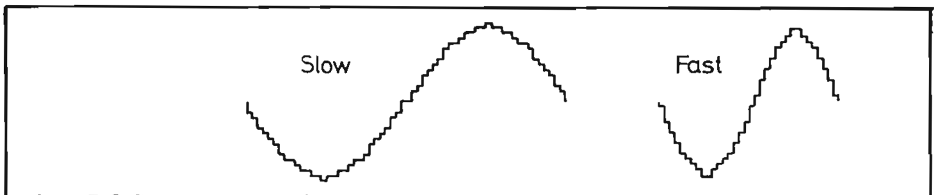


Figure 7. Software-generated sinewaves using a counter-plus-lookup routine.

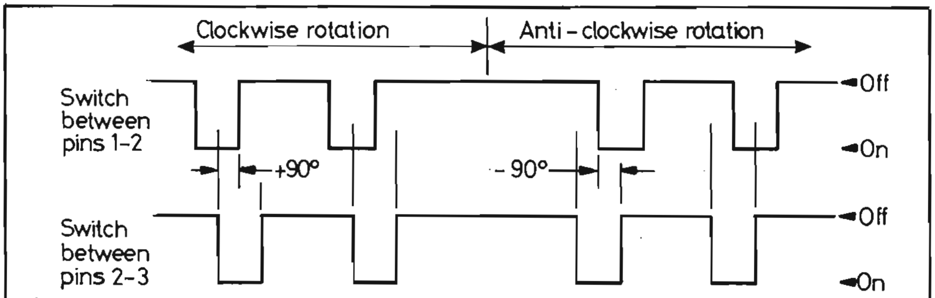


Figure 8. Spin-wheel controller.

signal by a simple log converter, and this is made up from IC121, T100 and T101.

As we mentioned in Part 1, a two-octave range can be obtained using the voltage control input, but the MCS1 can also add a large musical transposition to the output signal.

The voltage control is a Play mode function, and a two-way switch (IC107) selects either one-volt-per-octave or microprocessor control of the master clock generator. By using a latch (IC118), a DAC (IC119), and a low-pass filter (IC120), the microprocessor can generate sinusoidal sweep and pitch-bend control voltages.

The sinewaves are generated in software with a counter and lookup table. A number is generated by the sweep frequency controller (a panel function, this) which is added to the counter at regular intervals of time. The counter is used as a pointer to read the magnitude of the sinewave in the lookup table: as the pointer moves through the table, a sinewave is generated and turned into a voltage by the DAC. The number added to the counter determines the sinewave frequency – if the number is small, the pointer will take a long time to travel through the lookup table, and so on. The filter is used to smooth out the crunchy shape of the sinewave generated by the DAC, though it really isn't all that bad in the first place. If you want a visual representation of the software-generation of sinewaves, look no further than Figure 7. Incidentally, amplitude control of the sinewave is also performed in software, by a multiply routine.

The control voltage electronics as a whole is actually rather difficult to align, while the master high-frequency oscillator contains a non-linearity which can cause detuning at low frequencies. The best musical results are obtained by optimising the log circuit for operation over the top octave range, because if you try to align it over the full keyboard range, tuning errors are simply inevitable. The MCS1's two-octave CV range is still useful for effects purposes, while CV devices can be used to drive the sampler over a six-octave range if you can get your hands on one of the analogue-to-MIDI converter units that are either available now or are soon to become so. You'll also need a MIDI keyboard, of course.

The master clock generator is fed into a divide-by-N down counter comprising IC102–104. The value of N is stored by latches IC105 and IC106. The counters count downwards to zero, and when they reach zero, an RC pulse is generated by IC104, and this loads the counters with the value of N. They then count down to zero again, so as you can see, they don't lead a particularly interesting existence.

The CK/N signal generated by this timing process is the sample rate of the system as a whole. Thus, by changing the value of N (N is a 12-bit word), the sample rate can be modified directly. In fact, this mechanism is used to vary the sample rate continuously *via* the control on the MCS1 front panel.

Additionally, and as a result of having a lookup table of values of N that result in a musical distribution of CK/N frequencies, it's possible for the microprocessor directly to control the pitch of the output signal in semitone steps. And yes, this control information can be obtained by decoding MIDI pitch data.

Now, while a source of MIDI codes can be used to generate the Play pitch for the MCS1, the method of dividing a master frequency by N to generate equally-tempered tuning is not without its problems. For one thing, N has to be an integer, and while this results in good pitch resolution for low notes, things aren't quite so consistent further up the scale. The overall resolution can be improved by increasing not only the size of N but also the frequency of the master oscillator: the MCS1 uses a maximum N value of 4096 and a maximum frequency of 10MHz.

On the basis of that 4096 figure, the first (lowest) octave has 2048 values of N at its disposal to define its 12 semitone frequencies, but the fifth (highest) octave has only 128. In other words, tuning resolution is still rather better at low frequencies than it is at high ones . . .

Moving still further ahead, ADC and read-write timing is generated by two counters (IC110, 111) and a bipolar ROM, IC112 (again, see Figure 5). If the CK/N period is long, extra refresh counts are generated by the RAS and RAE signals, ORed together by IC114.

Diagram 3

The microprocessor at the heart of the MCS1's design is a 6802, shown on the drawing as IC309. Strange though it may seem, this device generally has nothing much to do. It scans all the panel controls, loads up all the control latches and display registers, and then waits patiently for something to happen.

The reason for this inactivity? Simply that most of the MCS1's functions do not depend on the active intervention of the microprocessor for them to operate smoothly. The 6802's busy time is when it's generating a software sinewave or loading from or saving to floppy disk. A data bus buffer (IC321) has been used because the microprocessor would otherwise be unable to support loading on the data bus. The program is held in an EPROM, IC308.

A static RAM (IC307) is used as a scratch pad memory for items such as filter offsets and current values of N, while IC310–312 generate all the address decodes for the memory-mapped devices.

All the MCS1's front panel display details are handled by two chips, IC304 and 305. These are 34-bit long-shift registers with parallel outputs that can drive LED displays directly. Display data is entered serially as a 34-bit-long chunk, and the IC does the rest. No seven-segment decoding takes place inside the IC because the segments that have to be switched on are controlled directly by the input data stream. The four-digit display and the illuminated panel switches are driven by these ICs.

The pitch-controlling keyboard is scanned by IC315 and 316. The former pulls one row at a time low while the latter reads the keyboard in four-bit nibbles: any key depression is detected as a low voltage, and the six rows and read sequentially. IC316 also reads some of the other system signals.

IC328 is a control latch, the outputs of which are used to enable various functions. Serial data is handled by the ACIA (IC323), and transmission and reception between the ACIA, the BBC Micro (a future update), and MIDI is performed by enabling various tristate buffers (IC235, 326). The MIDI In signal is coupled to the MCS1 *via* an optoisolator IC327: this helps prevent the ground loops and other unwanted hiccups that so often occur while equipment is being interconnected.

The Gate, Audio and spin-wheel controller signals all generate a hardware interrupt (IC317, 318, 319, 320 and T300). When an external Gate signal occurs, it clocks flip-flop IC318, setting the Q output (pin 5) to a 1. This turns on T300, which in turn generates an interrupt. The microprocessor services this interrupt by reading the contents of bus buffer IC320. It discovers it was the Gate that caused the interrupt (the Gate signal is also available for reading at IC316), and takes appropriate action by generating a clear interrupt (CLINT) signal which sets the flip-flop back to a zero. The remaining two interrupts are similar.

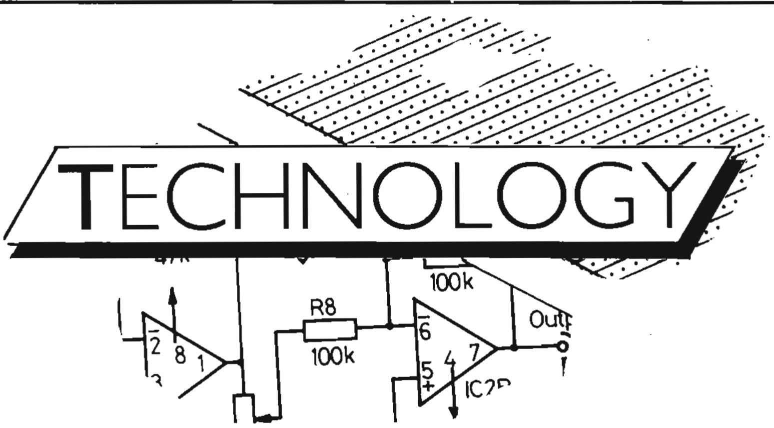
The Audio signal is fed into a voltage comparator (IC317), and when the voltage at its input pins exceeds ±50mV, the interrupt is set: this circuit is used to trigger the start of a recording at the beginning of memory.

The spin-wheel controller is an important part of the circuit because it is the manual interface between the user and the MCS1's internal control parameters. The controller (Figure 8) is a rotary switch with 50 positions per revolution: there are two switch contacts 90° out of phase with each other, and the rotation range is a full 360°.

Inside the MCS1, IC319 is used to detect the controller's rotation and the direction of that rotation. One switch output is used to clock a D-type flip-flop, the other provides the data input. If the rotation is clockwise, the Q output is set to a 1, and if it's anti-clockwise, a zero. The individual switch pulses generate an interrupt and need to be cleared if the next event is to be recognised.

The MCS1 software provides for three control sensitivities for the spin-wheel, these being Fine, Medium and Coarse. Just think, if you only got 50 'clicks' per revolution of the wheel in reality, it would take a grand total of 1310 turns of the controller to travel the full length of the MCS1's memory address. ■

The MCS1 retails at £499 plus VAT as a complete kit of parts, £699 plus VAT as a ready-built unit. Further information from Powertran Cybernetics, Portway Industrial Estate, Andover, Hants. ☎ (0264) 64455.



Powertran MCS1

Part 4: Testing, Testing

If you're buying an MCS in kit form, you'll need to know how to test it to make sure your unit's working as it should do once it's built. Here's how to do just that. *Tim Orr*

Now that the previous articles in this series have convinced you that the MCS1 is perhaps the greatest technological advance in musical history and you've built one from a kit, the last thing you want is for the unit to work incorrectly or, Heaven forbid, not at all.

But first things first. Make sure you inspect all the PCBs thoroughly, checking that all the components are in their correct places, that all joints are properly soldered, and that there are no open or short circuits caused by solder splashes, for example. The experienced constructors amongst you will probably undertake all these tasks as a matter of course anyway, but it's worth remembering that these all-too-common faults are easier to spot at the inspection stage prior to final assembly.

Whatever you do, don't fit the ICs yet. Make the wire connections between the PCBs and clean the solder connections: the MCS1 is best tested outside its box, as shown in Figure 1.

Powering Up

With no socketed ICs fitted, power up the unit. Be on your guard for smoke, small fires and minor explosions. Yes, I know it sounds alarmist, but the chances of a power supply smoothing capacitor that's been fitted back to front blowing up in front of you are unsettlingly high.

Using a digital voltmeter or a scope, test the power supply rails for their correct operating values, which are as follows. Unregulated inputs for the IC115 and 116 are +21V and −21V respectively, while that for IC117, 226 and 322 is +8.7V. Ripple (on load) for unregulated rails should be 300mVpp for IC115 & 116 and 700mVpp for IC117, 226 and 322. Current consumption on load should be 150mA for 12V rails and 1.8A for the 5V one, while other voltage regulators worth checking out are IC100, 306, 419 and 420.

In their present condition, the voltage regulators shouldn't even be warm: shorts across the rails don't usually destroy these regulators, as they incorporate their own thermal shutdown mechanism.

Next, turn the power off, fit all the op-amps and retest the ±12V rails, having already turned the power back on, of course. Turn it off again, and insert the logic chips in lots of ten at a time. Retest the 5V rails and continue this procedure until all the ICs have been inserted.

Your MCS1 is now fully powered and should operate without generating too much in the way of heat. The voltage regulators (with heatsinks), the microprocessor and the PSU power diodes may well be warm or even hot, but most of the remaining components should be only slightly warm. Retest the power supply rails: this is *vital* for the simple reason that the MCS1's electronics will not operate unless the power supply rails are as they should be.

The machine should now be capable of a safe power-up, and if everything else is OK, your MCS1 will operate first time. One encouraging sign of intelligent life is the processor going through a start-up sequence, during which it flashes all the display LEDs on the MCS1's front panel. If by some unlucky chance this doesn't happen on power-up, you've hit trouble, and a further visual inspection is called for. Typical faults are inter-track short circuits, ICs not being where they should be or fitted backwards, IC pins being folded over underneath the chip instead of going into the IC sockets, non-soldered pins, broken tracks, all the LEDs being inserted backwards or the displays being installed upside down. And if you think some of these eventualities are unlikely – not to say mildly amusing – don't laugh until you're *sure* you haven't made any of them yourself: it's very easy to do.

Now, even if your unit seems to come to life straight away on first power-up, it's still advisable to test everything. Thoroughly.

Using a DVM or a scope, test for correct power on all the ICs: refer to the power pin chart shown.

Microprocessor System

Ideally, it's this section that should be tested first. If you're working with an oscilloscope, things won't be quite as easy as you'd probably like, but there are a few simple tests you can employ to locate faults, most of which involve connecting the scope to an IC pin or two and having a look-see at the results.

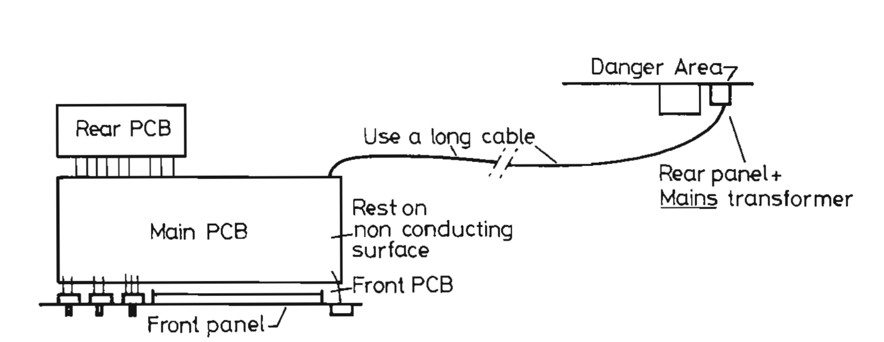
Have a look initially at pins 38 and 39 on IC309, the crystal oscillator. Figure 2 illustrates its correct operation, while Figure 3 does the same for the E signal on pin 37.

The data and address buses should all be busy for normal operation. Note that the data bus is buffered by IC321. Both the reset signal and the IRQ should be high. One thing that mustn't escape your attention is to look out for illegal logic levels on the data and address buses. Levels of between 1V and 1.5V are generally a sign that something is amiss – a bus clash caused by a short between two logic signals, for instance.

Next, check out the CE signal on the EPROM (IC308, pin 20). This should be relatively active, indicating that the software is running. ICs 310 to 312 generate a range of address codes, many of which are continually active, so if a particular machine function doesn't operate, it may well be the result of a decoder fault.

Now we come to the MIDI keyboard side of things. If nothing happens when you play a note on the keyboard, it could be because the

Figure 1. MCS1 Test layout.



Power pin chart.	0v	+5v	+12v	−12v
6N138	5	8	–	–
MM5460	1	20	–	–
4066B	7	14	–	–
4046B	8	16	–	–
LF398	–	–	1	4
LM311	–	–	8	4
TL081CP	–	–	7	4
RC4558	–	–	8	4
MF4-50	4(−5V)	7	–	–
LM339	12	3	–	–
DAC0800	–	–	13	3
ROM (IC112)	8	16	–	–
HM4864* (DRAM)	16	8	–	–
HM6116**	14	26,28	–	–
2764	14	1,26, 27,28	–	–
6802	1, 21	8	–	–
6850	1	12	–	–
2502	6	16	–	–
DAC88	–	–	18	13
DG211	5	12	13	4
74LS00	7	14	–	–
74LS04	7	14	–	–
74LS08	7	14	–	–
74LS32	7	14	–	–
74LS74	7	14	–	–
74LS86	7	14	–	–
74LS122	7	14	–	–
74LS125	7	14	–	–
74LS138	8	16	–	–
74LS161	8	16	–	–
74LS169	8	16	–	–
74LS244	10	20	–	–
74LS246	10	20	–	–
74LS374	10	20	–	–
74LS393	7	14	–	–
74LS625	1,9	10,16	–	–
74LS685	10	20	–	–

*Note: reverse of normal convention.
**Based on 28-pin layout.

keyboard itself is not being scanned. Have a look at IC315: the Q output pins generate active low row scans. Take a look also at the KBI signal, which clocks the latch. One device certainly worthy of your attention in this department is IC316, the tristate buffer. Check its A inputs first of all, then look at pin 2 and check that switches 1, 5, 9, 13, 17 and 21 all pull this point low when pressed (low is only 0.8V in this case). Don't forget to test the other three column pins, too, and look for the KBO signal.

IC324 is used to divide the 1MHz clock signal by two, and pin 5 should be a 500kHz square wave – look for this signal at IC325, pin 8. Note that the M/S signal must be low for normal operation.

Finally, test the MIDI opto-isolator by injecting either a MIDI or a TTL signal into SKT300. This signal should be repeated at IC327, pin 6, and just in case you're stuck for a source of the latter kind, Figure 4 shows a simple circuit for a DIY TTL drive simulator.

Onward ever onward, in this case to the MCS1's spin-wheel controller. Have a look at IC317, pins 13 and 14: the two outputs should be 90° out of phase with each other when the controller is spun. Now spin the wheel in the other direction and check the phase reversal.

Figure 2.

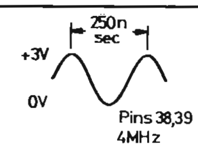


Figure 3.

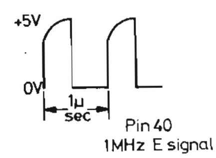


Figure 4. DIY TTL drive simulator.

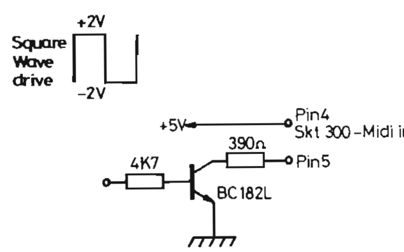
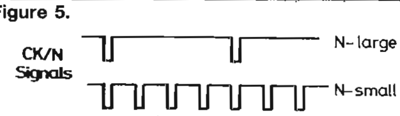


Figure 5.



Moving to the audio trigger, you should inject a high-level signal into the audio input and take a look at IC317, pins 1 and 2: you should see a TTL signal of the same frequency as the audio one. Check to see that ICs 318 and 319 are cleared regularly by the CLINT signal.

Lastly in this section, we come to the LED display. Each time a controller function is altered, the display should be updated: check that the CKDIS signal is active at IC304, pin 21.

Panel Controls

Just because a component is in full view of everybody and not tucked away inside the MCS1's box, doesn't mean to say it shouldn't be checked for correct operation. This is really quite a straightforward procedure, and I'll run through each control in turn.

To kick things off, press the Delay Line and Voice Mode switches: the LEDs should follow the switch selections. In Delay Line mode, the Freeze and Click Track should both have an independent toggle action. Select Sample Speed – the display can now be varied *via* the controller. Now select RAM Size and try out the Coarse, Medium and Fine sensitivities; the Bypass has a toggle action.

Press Record in Voice Mode. The LED will flash until the system receives an audio trigger, and it will then remain on until the MCS1's maximum recording time has been exhausted. Once this has happened, the Play LED should illuminate if Play is pressed or if an external gate signal occurs. Once again, the Gate Trig and MIDI CV switches both have independent toggle actions – make sure yours have.

The Pitch Shift is a controller function, and since only one controller function can be selected at a time, the currently selected function takes over from that previously in operation. Loop Start and Loop Length are two further examples of controller functions.

It probably won't have escaped your attention that the MCS1's alphanumeric display is four digits long, while some parameters (such as memory address) are actually five digits in length. In these instances, the display simply loses the last digit, so that a memory address of 65535 (the maximum, incidentally) is represented on the front panel as simply 6553.

Moving back to the controller functions, Filter Shift, Sweep Range and Sweep Speed are all examples of these, with display ranges of 0–12, 0–100, and 1–100 respectively. Finally, both the Sweep On/Off and NR (noise reduction) switches are intended to have toggle actions.

Back to the Hardware . . .

. . . and the master clock generator (IC101), to be precise. Adjust C104 for an oscillation frequency of 2–5MHz, and have a look at pin 14 (C104 will be aligned later), making certain that the M/S signal is low. The next step is to follow the CK signal through to IC109, pin 2, and then on to the same pin of ICs 102, 103 and 104: this is the CK/N generator circuit. If you select Sample Speed mode, you'll be able to vary the value of N *via* the spin-wheel controller. Have a look at the CK/N signal (IC104, pin 9) and vary the value of N: Figure 5 shows what effect this variation should have. It's worth noting that this circuit should work fine at frequencies of up to 13MHz, but that above this speed, delays in the counters will cause the divide-by-N circuit to crash . . . Fortunately, normal MCS1 operation avoids going this high.

Let's turn our attention to the timing generator circuit, ICs 110–112. Sync from A4 (IC112, pin 14), and check that all the timing signals are as shown in the timing diagram reproduced

Figure 6.

Figure 7.

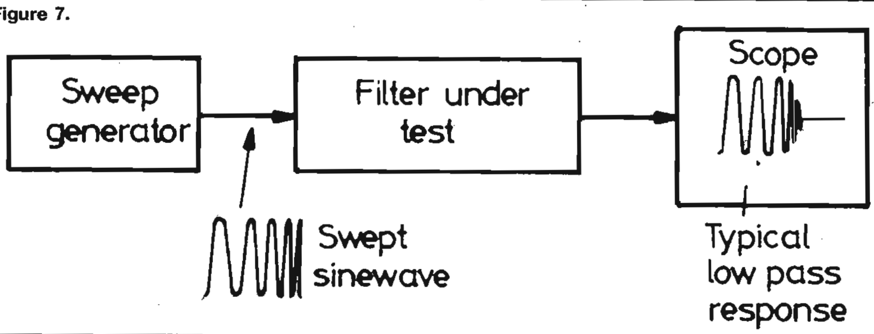


Figure 8(a).

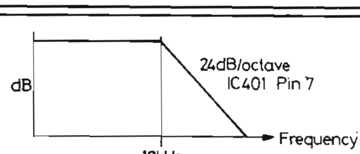


Figure 8(b).

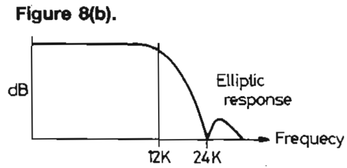
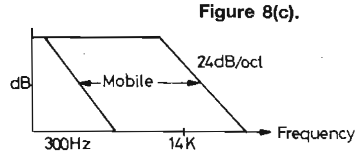


Figure 8(c).



in E&MM December 83. Note that if N is large, all 32 portions of ROM are used as a result of extra refresh periods. As N is reduced, the extended refresh is curtailed.

Now for the software sweep. Set the Sweep Range and Sweep Speed to 100 and activate the Sweep On/Off function. Take a look at IC120, pin 7, and you should see a crunchy sinewave, but a filtered sinewave should be produced at pin 1 (Figure 6 illustrates the difference between these two). Try various Range and Speed settings. CKCV clocks the latch that stores the sinewave data that feeds the DAC (*very poetic – **Ed***).

The log converter circuit (IC121, T100, and T101) converts the input from a one-volt-per-octave keyboard into a log control voltage capable of controlling the master clock generator's frequency (see later for alignment notes on this). DC test points are at pins 1 and 3 of IC121, and should both be −0.6V.

To test the memory address counter (ICs 207, 208, 210 and 211) select the highest sampling speed and the largest RAM size possible. Once you've gone into Delay Line mode, the CK/N signal should be the same as the ACK signal, and the counters will all count up. Look at MA0 throught to MA15, a 16-stage binary counter.

The end and return address circuits are best tested by recording a sound in Voice mode. You may decide to record something along the lines of the classic phrase 'One, Two, Testing' – then again, you may not. Once you've recorded your speech, set the Loop Length to 0 and the Loop Start to 6553. Press Play and your words will be replayed as if by magic (or something).

Try taking the Loop Start value from 6553 down to 0 – less of the speech will be heard each time Play is activated. Now increase the Loop Length with Loop Start set at 6553 – a loop of increasing length should be audible. Lastly, try the effect of implementing Fine, Medium and Coarse sensitivities.

The 16-bit words generated by the two looping functions are held in latches. The Loop Start (end address) is held in IC200 and 202, which form one 16-bit input for the address comparator, IC201/3. Varying the Loop Start point obviously varies the data held in these latches. The Loop Length function is a computed return address, so that the end address minus the Loop Length equals the return address: this is held in IC206, 209. When the end address is reached, the counters are loaded with the return address. Varying the Loop Length alters the data held at the Q outputs of IC206, 209.

Buffers IC212 and 213 multiplex the memory address into DRAMs. Check the control signals LAE and MAE: again, refer to the timing diagram. IC224 is the refresh counter, so check CKRA and RAE against the timing diagram, too.

ICs 214 to 221 are the DRAM memory. To test this, go into Delay Line mode and inject an audio signal: look for multiplexed memory addresses on A0–A7 on the DRAMs. The data inputs (DIN) are driven directly from the ADC – check to see that they are all busy.

The data outputs are usually tristate, but are occasionally active. Check the RAS, CAS and WE signals against the timing diagram. Pressing Freeze at this point should result in the WE signal going high, preventing any further writes into the memory from taking place. The external Freeze at the rear of the MCS1 should have a similar effect.

Our last port of call for this section is the unit's built-in Click Track. To test the operation of this, set RAM Size to maximum and select Click Track. The click signal itself is generated at pin 1 of IC227: switch the Click Track off and T100 should sort out the signal.

Audio Section

If you have access to a sinewave sweep generator, the MCS1's active filter responses can be analysed very quickly, as Figure 7 shows.

The first thing to do is to inject an input signal into the MCS1. Test sensitivity selector SW400 by looking at pins 1 and 7 of IC400 – the signal should be unfiltered. However, as we saw in E&MM December, the MCS1 incorporates several filter stages, and some of these are illustrated in Figure 8. Pin 7 of IC401 is a 12kHz four-pole lowpass response (a), pin 8 of IC403 is a 12kHz elliptic response with a notch at 24kHz (b), while IC403 is the mobile filter as a whole (c). If the phase-locked loop is working, a filtered output should be visible at pin 5. Look at pin 2 to see the MFCK signal – this is a square wave with a frequency range of 15kHz to 700kHz, and the break frequency in Figure 8(c) is MFCK divided by 50. Varying the Filter Shift should move the break frequency of the mobile filter over a ±1 octave range. Maximum signal output level is 8Vpp. IC404, pin 1 is another 12kHz lowpass response, incidentally.

Set the Filter Shift to 12 and select the high sampling rate: the MCS1's audio bandwidth is now at its maximum. Take at look at IC404, pin 7, the noise reduction (pre-emphasis) circuit. With NR on, the circuit adds a treble lift to the signal. Beware of clipping when testing this circuit, and use a 3Vpp signal.

To test the MCS1's sample & hold device (IC405), inject a 1kHz sinewave, vary the Sample Rate control, then take a look at pin 5. Figure 9 gives some idea of the sort of waveshapes you can expect to see.

The next checkpoint is IC408, where pins 9 and 10 generate two signals (ADCK and SC) that must be checked against the timing diagram.

The ADC performs eight tests, going from the MSB to the LSB, and these data bits can be seen stabilising as the analogue-to-digital conversion proceeds, if you know where to look. I'll give you a clue – it's IC409, pins 2 to 9.

Moving logically from the ADC to the DAC, the latter can be tested by following this procedure. Select Delay Line mode, set RAM Size to 0, Freeze and Sweep off, and select the highest sampling speed. Use a 1kHz sinewave input. This signal should be converted by the ADC into binary code, which is stored in memory and subsequently converted back into an analogue signal by the DAC, ICs 412 to 414. Have a look at pin 6 of IC414, then at pin 7 of IC415 – Figure 10 shows two typical outputs. In fact, the DAC's output is further filtered by the mobile filter IC421 and fixed lowpass filter IC422 (pin 7) in turn.

Now have a look at IC422, pin 1. This is the de-emphasis circuit, and as already implied, can be used to make a sound appear bright simply by recording it with noise reduction switched in and playing it back with NR switched out, so that you're recording with a treble lift and playing back with a flat response.

Turning now to the Mute circuit, this is made up of IC424, T402 and 403. Select Voice mode, record a tone and then Play it. When the Play switch is released, the output signal is attenuated by about 90dB and is effectively off (Figure 11).

Our last place of interest on this whistle-stop audio section tour takes us back to the MCS1's front panel. Go into Delay Line mode, set the Pan control to centre and input a microphone signal: the MCS1's output will be a direct signal with echo. Press the Bypass switch, and the echo should disappear, leaving just the direct signal.

To check that the Click Track is working as it should be, select maximum memory size and activate the click signal. Assuming the metronome is now audible, you're free to lower the memory size, at which point a corresponding shortfall in the Click Track sequence's audibility will take place.

The phase locked loop is made up of ICs 410, 416, 417 and 418. If you select Filter Shift and vary it from 0 to 12, this will alter the division number loaded into IC417 by IC416. Sync from the CK/N input, pin 14 of IC418: pin 3 should be a square wave at the same frequency, while pin 4 will be a square wave with a frequency of CK/N × the PLL division ratio. To check this out, look at pin 4 and alter the Filter shift value.

The PLL is designed to limit at about 700kHz, whereupon it is no longer in lock.

Alignment Procedure

Anti-oscillation (P401)

Select Delay Line mode. Use a short memory length and turn the Repeat control up to maximum. Adjust P401 so that echoes are not *quite* self-sustaining. Experiment with other Filter Shift values, different Sample Speeds and with NR on and off. Readjust P401 so that it's as stable as possible with all configurations.

PLL SOT Capacitor (C449)

As already mentioned, the maximum frequency of the PLL is typically 700kHz. Set the Filter Shift to 12 and the Sample Speed to maximum. If the MFCK frequency exceeds 1MHz, add extra capacitor C449 (22pF) to reduce said frequency. However, this is not considered a very likely eventuality . . .

Sample & Hold DC Offset (P400)

The sample & hold device (IC405) has a DC offset that varies with the sample frequency. If recording with a frequency sweep is undertaken, this effect can move the quiescent DC offset past quantisation levels, and this process often generates small puffs of noise as it happens.

The solution? Select Delay Line mode with no audio input, no NR, and a short delay time. Look at IC414 pin 6 and listen to the delayed audio output. Vary P400, and you'll notice that as you do so, the sample & hold DC offset moves and the above-mentioned aural effects make themselves apparent. Adjust P400 for the most quiet position, which should be equidistant between two quantisation levels. Turn on the Sweep and select a 1Hz full-range sinewave. If this causes some further noise generation, readjust P400 accordingly.

Log Converter (P100, 101, 102 & C104)

This circuit has already been discussed and enables an analogue synthesiser to control the pitch of the MCS1's sound output. The normal control range is two octaves, but the MCS1 can add a further five octaves of pitch transposition *via* its CK/N divider circuit. Unfortunately, the log converter is not the easiest device in the world to set up, but the best method is described below and outlined in Figure 12.

First off, select CV Mode: this allows external voltage control of the master clock generator, IC101. Turn off the Sweep.

Our first job is to determine the voltage/transfer frequency function of the VCO (IC101). Figure 13 shows a typical graph for this function, but obviously the exact plot will vary from unit to unit, which is why your MCS1 came with a blank graph pad (don't tell me you used it as a bin-liner by mistake . . .). In order to plot your own function, adopt the following procedure. Measure the control voltage at IC108, pin 1, against the output frequency at pin 14 of IC101. Adjust P101 and P102 so that the external CV can move the voltage at IC108 fully over the 0V to +5V range. Set this voltage

Figure 9.

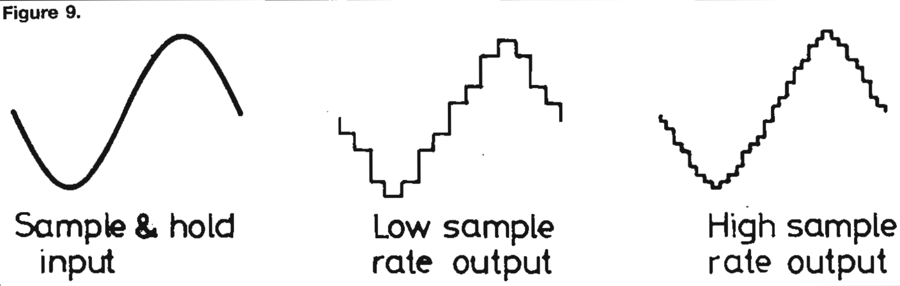


Figure 10.

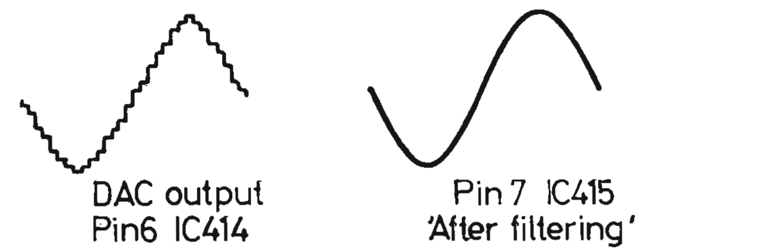


Figure 11.

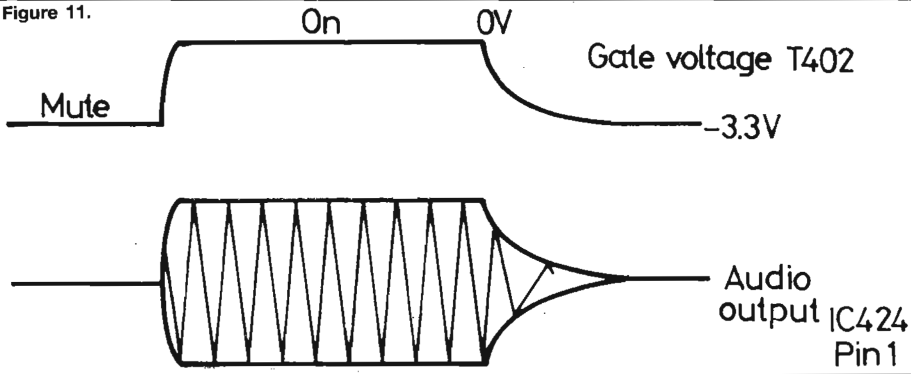


Figure 12. Typical log converter test set-up.

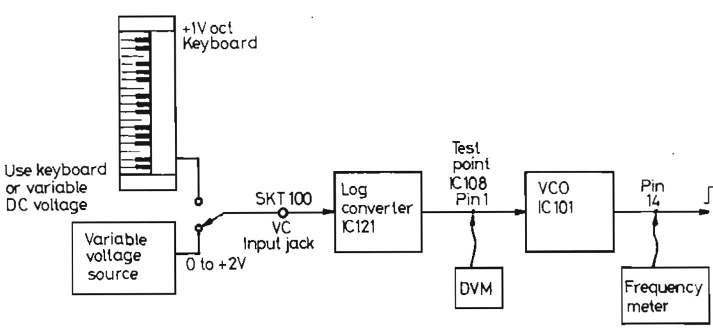
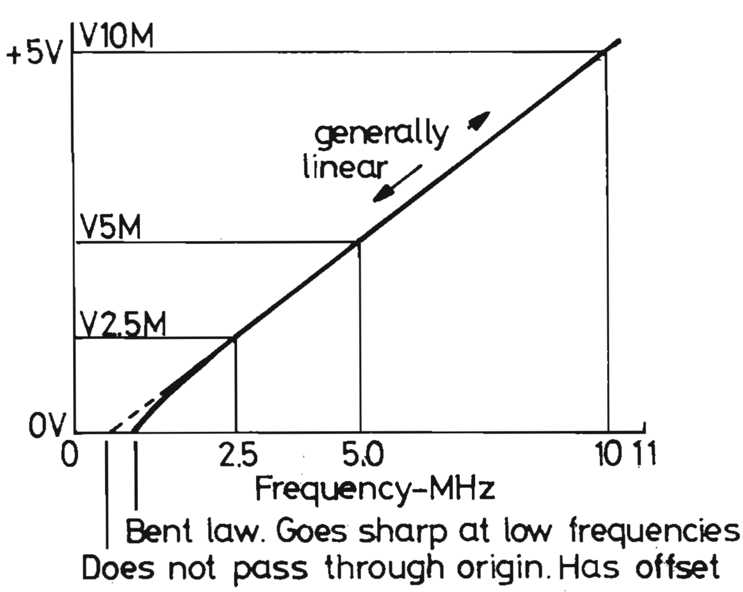


Figure 13. IC101 voltage/frequency transfer function.



▷ up to be +5V, and adjust C104 for an output at 11MHz. Now plot the transfer function on the graph pad, using 0.5V steps going from 0V to +5V.

Measure the voltages needed to generate frequencies of 10MHz, 5MHz and 2.5MHz (otherwise known as V10M, V5M, and V2.5M respectively). This is the two-octave range over which the MCS1 will operate in CV mode.

P100 is the volts/octave preset that adjusts the tuning of the keyboard voltage span. Log transistor T100 *must* generate a current that doubles for every volt increase in the input control voltage.

The next step is to cut the link between T100 and IC121 and insert a current meter to bridge the gap. Set the meter to a 0–2000μA range. For an input of 0V, set P101 to give an approximate current reading of 75μA and P100 to a central position. Make a note of the measured currents for input voltages of 0V, +1V, +2V (those CVs will generate the required two-octave swing). In order for everything to work properly, the current levels must form a ratio of 1:2:4, or in other words, a musical log law of an octave increase per volt. If the ratio is less than two per volt, rotate P100 clockwise; if it's more, rotate it anti-clockwise. Continue to adjust P100 in this way until the ratio is exactly 2.00: typical final currents should be in the order of 75, 150, and 300μA. Once these levels have been attained, it makes sense to increase the chances of P100 remaining in its proper position by daubing it with something indelible and instantly recognisable – a blob of Tipp-Ex fluid should do the trick. Once that's done, you can remove the current meter and solder the link back into position.

However, that's not the end of the procedure, because the log bias preset, P101, must also be aligned for this aspect of the MCS1's performance to be properly exploited. Using Figure 13 in conjunction with your own function plot, subtract V2.5M from V10M to give you the voltage needed for a two-octave shift (we'll give it the theoretical value of V2oct). Once this has been calculated, apply two external voltages, one of 0V and one of +2V. Adjust P101 so that the *voltage change* at pin 1 of IC108 is equal to V2oct for the input two-volt change. Clockwise rotation of P101 increases the size of the voltage change, anti-clockwise rotation decreases it. Beware of the fact that P102 may also have to be adjusted during this process so as to maintain the DC position on the graph.

Once you've calibrated P101, set the external control voltage to 0V and adjust the linear offset correction (*via* P101) to give a measured voltage at IC108's pin 1 of V2.5M. If you now apply an external CV of +2V, this voltage measurement should change to V10M.

If, by some miracle, this is precisely what happens, then both P101 and P102 are aligned, and the Tipp-Ex must be brought into action again. The log converter is now working and full aligned.

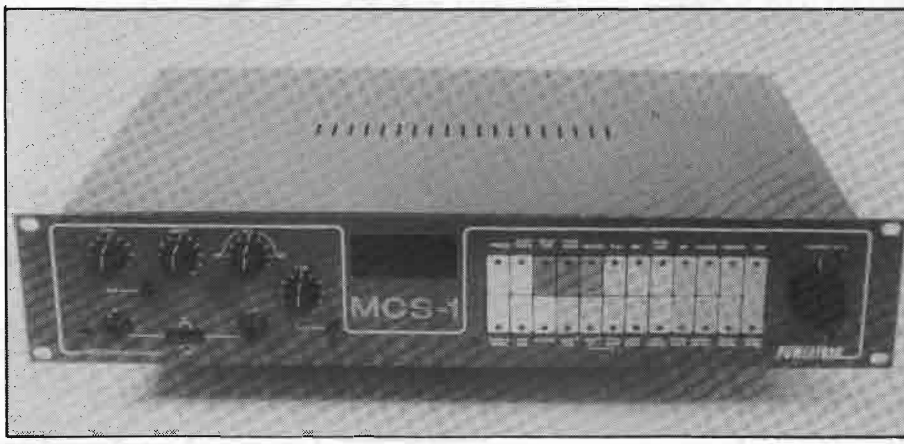
Just to make sure the theory works out in practice, try sampling a pitched sound (in Voice mode) and replaying it *via* a one-volt-per-octave keyboard in CV mode. Don't forget to make the Gate connections between synth and MCS1: the latter's gate-on is +2V or more, while gate-off is 0V or negative (the Play LED illuminates when the former condition is prevalent).

The recorded sound can now be played and pitched over a two-octave range, and pitch transposed over a five-octave range *via* the Pitch Shift function. However, as I mentioned in December, the tuning in this mode is not entirely perfect, due mainly to the bent VCO transfer function. If you have the option of using the now almost ubiquitous MIDI as a control source, then do so – the results are undoubtedly superior. ■

Powertran MCS1

Part 5: Tidying Up

We conclude our coverage of the build-it-yourself MIDI Controlled Sampler with mechanical assembly details and a quick run-down of the unit's front panel controls. *Tim Orr*



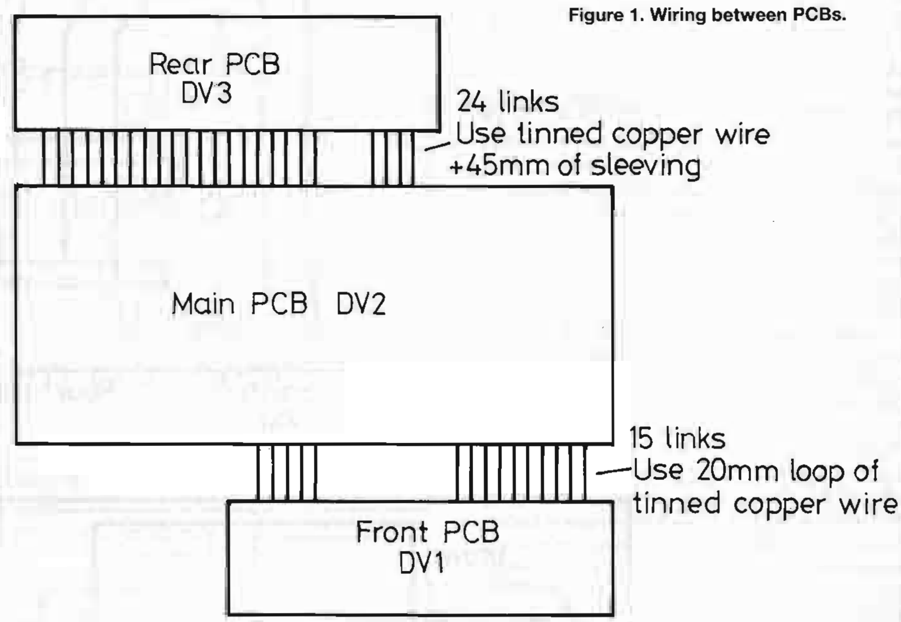
We've now covered most aspects of the MCS1's design, performance and construction, but one element of the last-mentioned topic that hasn't yet been dealt with is the mechanical assembly which, though not in itself particularly demanding, has to be done with care if you're to have a unit that works as it should do and, just as important to many, looks the part as well.

E&MM's over-worked Technical Illustrator, Len Huxter, has done such a good job on these particular drawings that, to a large extent, the diagrams speak for themselves, and no additional explanation is necessary. Figure 1 is a case in point. This shows a plan view of the interboard wiring that has to be undertaken before the various PCBs within the MCS1 will get to be on speaking terms with each other: the picture says it all.

The front panel wiring is a little bit more involved, as a quick glance at Figure 2 would suggest. The problem here is that what was originally a multicolour (one colour for each connecting wire) illustration is now a decidedly monochrome one, so it isn't perhaps as easy to read as it could be. Note also that VR3, labelled on the diagram as Pan, carries the name Mix on production Samplers: the change in nomenclature is due simply to a change of heart over how the pot's function could best be described, as the panel description on the pages that follow should explain.

The transfer of Figures 3(a) and 3(b), which together show the mechanics of how to wire up the unit's power supply, from original colour sketch to printable diagram also posed problems, in that the two drawings you see here form one super-diagram in the author's original. The split is a logical one, though, with (a) showing the connections between the transformer and the PCB and (b) illustrating the wiring between transformer and the MCS1's back panel. With luck, the printed results should be even clearer than they need to be.

Figure 1. Wiring between PCBs.



Assembly in Detail

Figures 4 and 5 are detail illustrations of mounting procedures for the front panel and main PCB (DV2) respectively. These are pretty self-explanatory, but the drawings that follow aren't quite so easy. Figure 6, for instance, is a view of the inside of the front panel, with the display filter as the focus of attention: don't forget to peel off the protective covers from both sides of the filter before you attempt to fit it, and once you've done that, the device should be installed with its matt side pointing outwards and its gloss side in, with a 3mm gap left at the top of the panel. Trim off any excess double-sided tape with a sharp knife to complete the job.

The installation of the spin-wheel controller (RC300) poses similar difficulties, though like the display there's only one of it, so things aren't nearly as serious as they might be. Figure 7 is obviously a help here, but there are a few more points worth making concerning the exact sequence of operations. The first task is to pack out the inside of the controller with a surplus jack-socket washer, after which the wheel can be trial-fitted to the front panel. Mark the controller's shaft as close to the panel as possible once it's fitted, remove the controller and cut to the appropriate length - you can now fit it for real. One last mechanical detail is the fitting of four rubber feet to the underside of the MCS1 cabinet, with a further two beneath the main PCB: Figure 8 shows the approximate positioning of the latter.

The Controls

Our last port of call is an explanation of the MCS1's controls and connections – what they do and why they're there. We'll start, logically enough, with the Audio In connection, which is made *via* a standard quarter-inch jack socket. A pushbutton is used to select low (1K5) or high (28K5) impedance levels, which should cover most operational eventualities. Audio Out is similarly connected, its socket being echoed (no pun intended) on the rear panel – another pushbutton selects zero or 20dB attenuation.

The four rotary pots at the left of the Sampler's front panel are easily explained. Level is a continuous gain control operating on the input signal, Repeat controls the amount of internal feedback around the delay line and hence the number of repeats produced, Mix alters the proportion of direct signal in relation to the output of the delay line, and Tune is a fine overall pitch control. The Controller on the panel's extreme right is a bit more complicated, as it's used to alter the value of whichever parameter has been selected. It has a 360-degree rotation, and is capable of incrementing or decrementing up to 50 steps

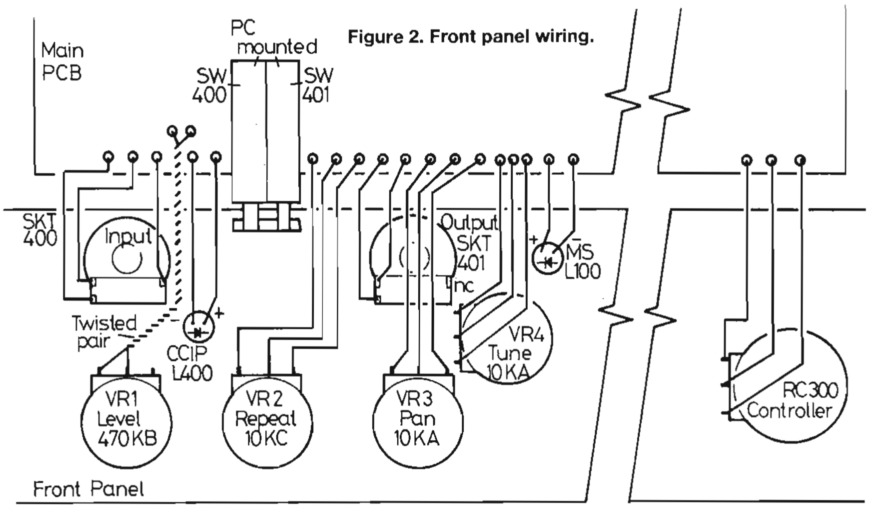


Figure 2. Front panel wiring.

▷ per revolution, depending on the parameter in question. Just left of centre on the panel layout is a four-digit (0.56", seven-segment LED) display, and this shows both parameter values and other information of importance to the user.

The remainder of the MCS1's front panel is occupied by no fewer than 24 selector switches, and I can think of no better way of explaining these than going through each of them one by one.

The Freeze button is what you use when you want to prevent any further data being written into the unit's memory, so that any sound already in there is frozen. The switch operates only in Delay Line mode and works in conjunction with the Freeze input on the rear panel. Like all the selector switches, it incorporates its own LED, and this is used in Voice mode to indicate when a recording is in progress. The light remains off while the recording is actually taking place, but illuminates once the sound has been successfully stored in memory.

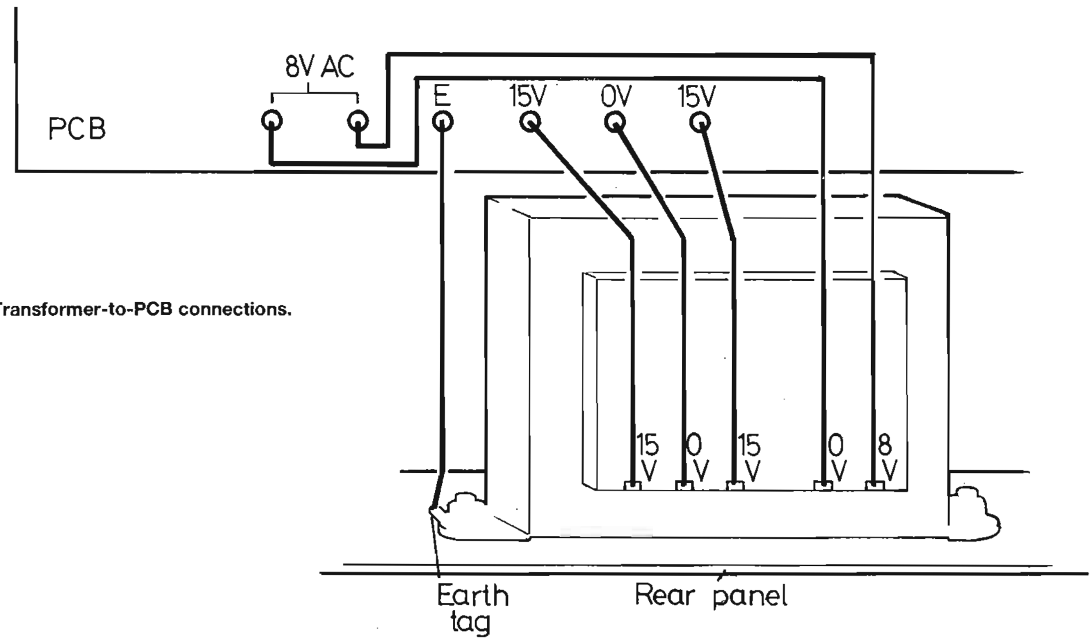


Figure 3(a). Transformer-to-PCB connections.

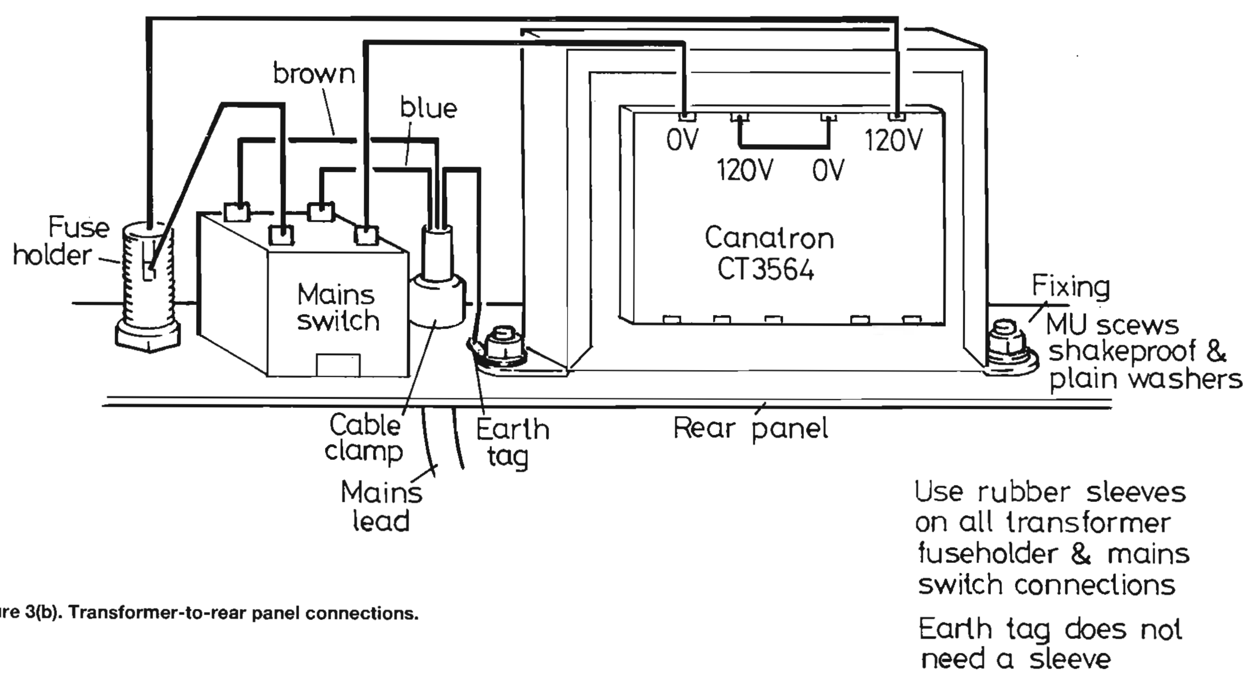


Figure 3(b). Transformer-to-rear panel connections.

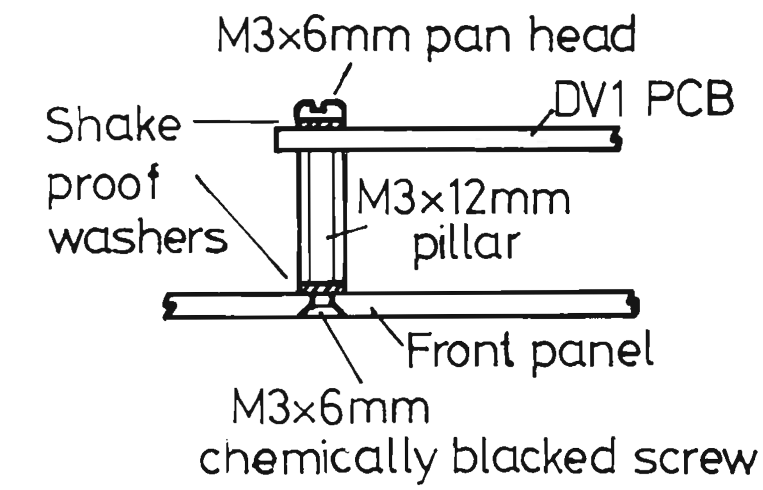


Figure 4. Front panel mounting details.

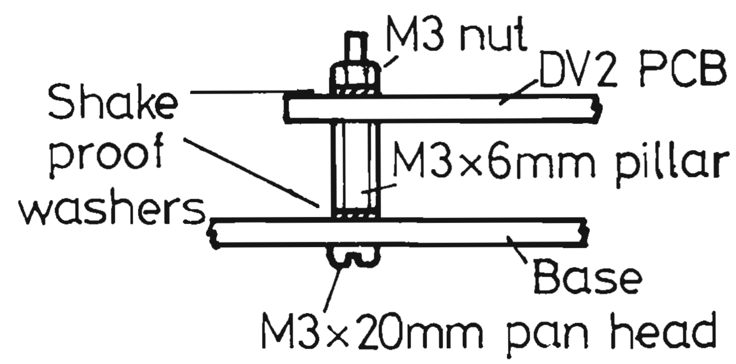


Figure 5. Main PCB mounting details.

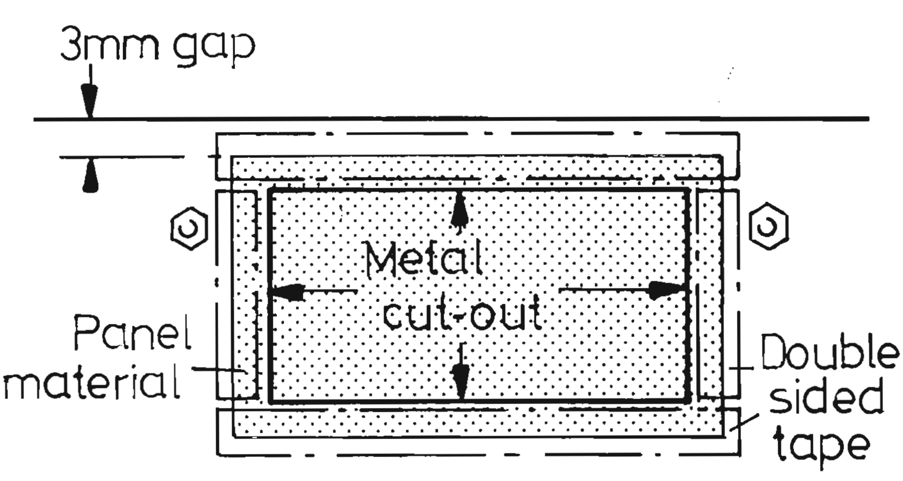


Figure 6. Display filter fitting (see text).

Moving to the right along the front panel, Click Track selects the unit's built-in metronome, capable of superimposing 16 beats onto the audio output for every complete trip around a full memory length, though obviously the shorter that length, the smaller the number of clicks the circuit will produce.

The two blue switches that follow are used to select Delay Line and Voice modes respectively. In the former, the MCS1 reads and writes from memory continuously, thereby generating a continuous cycle of sounds/ echoes. As we've just seen, these sounds can be frozen or, should you so desire, transferred to Voice mode operation, in which their pitch and duration can be played from a connected keyboard. The red Record button comes into play in Voice mode, and its operation is worth describing in some detail. Once your finger (or whatever part of the body you happen to be using) has made contact with the button, the associated LED starts to flash at a rate of about four times a second, indicating that the MCS1 is ready to begin recording. Said recording commences at the onset either of an audio trigger or the Record button being pressed for a second time, and it's at this point that the Record LED turns continuously on and the Freeze LED continuously off. And when the recording is complete, the status of these indicators is reversed and the stored sound can be replayed *via* either a keyboard or the Play switch that lives next-door, as it were.

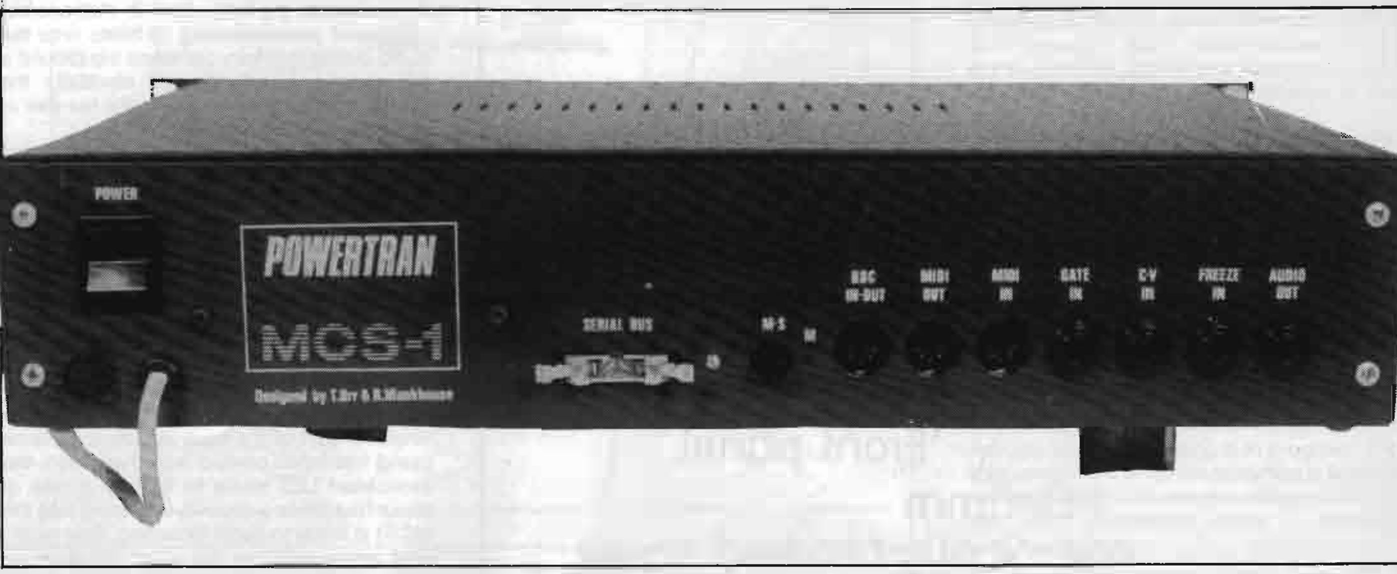
Future Options

The BBC and Down Load pushbuttons are both concerned with add-on hardware options that'll be available in the near future. The former is so called because a soon-to-be-available interface will allow MCS1 sounds to be stored within the memory of a BBC Micro, and the pushbutton will be used to access this facility. Moving on, the NR switch is already fully operational and is used to activate the MCS1's built-in noise reduction circuitry, which brings about a useful audible removal of otherwise bothersome quantisation noise.

The Coarse, Medium and Fine switches are used to select the sensitivity of the spin wheel controller, something that will obviously vary depending on the parameter whose value is being modified. The operation of the Sample Speed selector isn't quite as straightforward as its name might suggest. On the MCS1, the speed is expressed as a variable number with a value between 128 and 4095, which represents the value needed to program the internal divider chain. Thus, contrary to all expectations, 128 is a fast sample speed and 4095 a rather slower one. Whichever speed you select, it's indicated on the LED panel display, so you always know precisely where you stand.

Like the Sample Speed selector, the RAM Size button functions in Delay Line mode, and in this case the display shows the parameter as a number between zero and 6553 (actually, 65,536, but the display can only show a maximum of four digits, so you'll have to use your imagination a little). And once you've discovered what the current RAM size is, you can modify it using the Controller. Another somewhat self-evident selector is Bypass, which simply routes the input signal to the audio output, thereby ignoring the antics of the MCS1's processing entirely.

If you intend connecting the Sampler to a controlling keyboard, then a working knowledge of the Gate* Trig and MIDI* CV selectors is essential. Both induce Voice mode functions, and as its name might imply, the first switch is used to select whether the sound stored in memory is gated (* = LED on) or



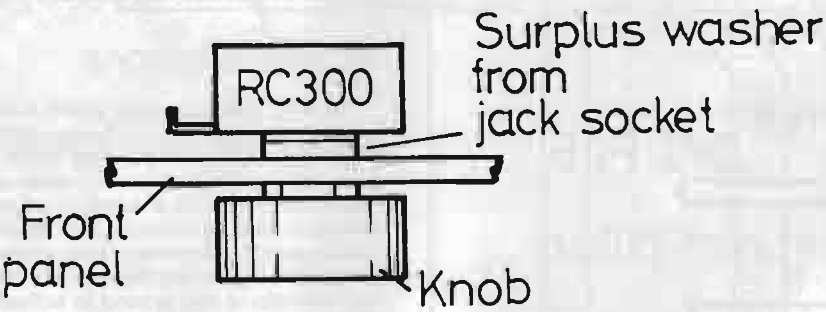


Figure 7. Controller fitting (see text).

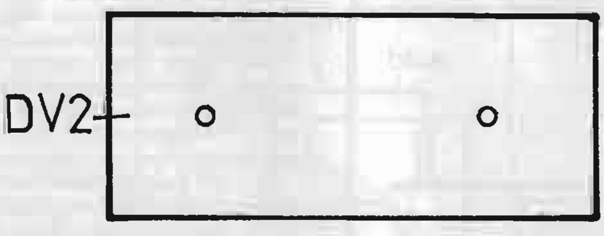


Figure 8. Outline of rubber feet positioning on main PCB.

triggered. Two gate modes are available, one that turns off abruptly and a second that has a half-second decay tail on its end. The display shows S for a short decay time and L for a long one. An unilluminated LED tells you the sound will be triggered, which means that although it's started by the gate signal, it'll play itself automatically to the end of the memory. For the non-synthetically literate, a gated signal can be put through a looping process, but a triggered one cannot. The second switch selects whether MIDI (LED on) or CV connections will be used to relay pitch information to the MCS1. In fact, pitch, sound duration, pitch-bend and vibrato are all controllable *via* MIDI, the note(s) played being indicated in MIDI code on the LED display. On switch-on the MCS1's MIDI channel number will default to 01, but all you do to alter this to any of 16 channels is hold the selector switch down and rotate the Controller accordingly.

We've already seen that the sample speed of the unit can be varied continuously as a value of N from 128 to 4095, but there is in fact an alternative way of changing that speed. The Pitch Shift button lets that same value be altered in semitone increments, from 12 (low end) to 80 (high end), and again, it's the spin wheel controller that's used to do the value-changing. Looping is quickly and simply achieved on the MCS1, using the Loop Start and Loop Length selectors. The former defines the point in memory where the loop begins, while the latter determines the 'jump-back' length. If you're foolish enough to set a loop length greater than the loop start (or to put it in layman's terms, you ask a loop to jump back to a position that's actually in front of its starting point), the display will show all its decimal points lit up at once, and the unit will automatically reset the jump-back address to zero.

The four remaining selectors (yes, the end is well and truly in sight) operate in conjunction with the MCS1's internal tracking filters and sweep oscillator. Not surprisingly, Filter Offset allows said filters to be offset by an octave less or an octave more than the system sample speed. The chosen figure is displayed by the LED network as a number between zero and 12. Sweep On-Off is used to, er, turn the sweep on and off, and Sweep Range acts as a level control for the oscillator's sinewave modulation. This is shown on the display as a figure between zero and 100, while the value set by the Sweep Speed pushbutton (in conjunction with the good ol' Controller, of course) is displayed as a figure between 1 and 100.

Rear Panel

In addition to the somewhat insignificant Power switch, the MCS1's *derrière* houses a fair complement of connecting sockets, a couple of which – the Audio Out and Freeze In jacks – have already been mentioned. As for the rest, the CV and Gate In connections are also quarter-inch jacks: the latter carries a high TTL (+4V) signal to generate a gate command, and can be used either in conjunction with the CV socket for control *via* a one volt per octave keyboard, or on its own as a trigger input from a drum unit of some description.

Five-pin DIN connectors take care of MIDI In and Out (note that the latter does not function as a MIDI Thru connection) and the BBC In-Out socket that connects the MCS1 to a BBC Micro *via* the custom-designed interface unit. Finally, the Master-Slave and Serial Bus multi-way connecting points are both intended for use with future options, so you needn't worry about them for the time being. ■