Sect. 3

PROGRAM

ADVANCE INFORMATION

Contents

Contents (Cont'd.)

General

1. The program for DUET is a fixed set of instructions which directs the microprocessor through a step-by-step execution of a complete cycle every period of up to 36mS. The basic program of processing fades is contained on the plug-in program card. Programming for various options (VDU, floppy disc unit, riggers control) is contained on the card which interfaces that facility with the system.

2. The object of the PROGRAM section is to provide the broad details of program organisation. For absolute detail of what occurs at every address the program listing should be consulted. This is provided under a separate cover. Programming is a manufacturing process and once the equipment is commissioned it is invariable.

Hexadecimal numbering

3. A hexadecimal system of numbering (i.e. counting up to 16) has been adopted for DUET. This system is mainly concerned with specifying addresses.

4. There are 16 address lines A0 - A15 which can be considered as four groups or four lines. In binary notation each four lines can count up to 16 ; in hexadecimal notation each four lines can be represented by one digit. Hexadecimal counting follows a simple alpha-numeric sequence progressing from 0 - 9 and continuing from A - F. 16 binary address lines are thus expressed as four hexadecimal digits (Table 1).

Table 1 Hexadecimal conversion example

Address areas	Multiples of 1K						1K area addressing									
Address bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Decimal	0				7				15				15			
Hexadecimal	0				7				F				F			

5. In comparing hexadecimal numbering 1K areas of addressing (those areas of 1024 addressable bytes of data most commonly handled by memory devices) it is the second hexadecimal digit which counts $\frac{1}{4}$K areas. Multiples of four; 0,4,8,C, in this second position therefore designate a 1K area. For example the 0K address area extends from 0000 to 03FF and the 1K address area from 0400 to 07FF. The example shown in Table 1 is thus the highest address in the 1K area.

Parameters

6. There are several parameters within which the program is constructed. These are :

(a) Address capacity
(b) Data word length
(c) Program instructions
(d) Memory capacity
(e) Microprocessor capabilities
(f) Peripheral interface adaptor (PIA) capabilities

Address capacity

7. Address capacity depends upon an address length of 16 bits. This gives an address capacity of 2^{16}, i.e. 65,536 (or 64 1K areas). This capacity is more than adequate.

Data word length

8. The greater the data word length the greater the accuracy achieved in a complex calculation. The data word length is 8-bit, which is accurate enough for the calculations performed by DUET. The eight bits of data at a particular address is known as a byte.

Program instructions

9. The program instructions consist of data words contained at addressable locations and are preset during the commissioning phase. Programable read-only memory hardware is used. This cannot be overwritten in use and can only be read out non destructively. 8192 (8K) addressable locations are allocated for general calculations and 2048 (2K) are allocated for each peripheral device interfaced.

Memory capacity

10. There are two sections of memory. One is built into the motherboard hardware and is used to provide 2K capacity of temporary storage for the processing of calculations. For example, partial products can be stored for subsequent retrieval.

11. The other section of memory is plug-in card mounted and can be between 4K and 24K bytes, see OPTIONS section. This is strictly an operator tool, used to store particular lighting states for recall at a later time. The capacity installed has a direct bearing on the number of lighting cues the system can handle.

Microprocessor capabilities

12. The microprocessor contains three 16-bit registers and three 8-bit registers of which are used in the execution of the DUET program. These are :

(a) the program counter, 16-bits, which point to the next program instruction addre ss
(b) the stack pointer, 16-bits, which contain the address of the next available memory location in the stack. This stack, located in RAM, is used to contain the return address for subroutines and also the original processor state whilst servicing interrupts.
(c) the index register, 16-bits, which is used to store the memory address for index mode addressing
(d) accumulators A and B, 8-bits each, are used to hold operands and results of arithmetic calculations
(e) condition code register, 8-bits (0-7) with bits 6 and 7 at logic 1. This indicates the results of an arithmetic calculation as follows :

Bit 0 - carry (from bit 7 at the accumulators)
Bit 1 - overflow
Bit 2 - zero
Bit 3 - negative
Bit 5 - half - carry (from bit 3)
Additionally, bit 4 is the interrupt mask bit
0 = IRQ and 1 = NMI

Peripheral interface adaptor usage

13. The peripheral interface adaptor (PIA) uses its peripheral interface bus A to output the current channel number being processed and its peripheral bus B as individual input / output flags (controllers). Bits B1 to B5 of this highway are outputs which control fade processing and LED mimic displays, and bit 7 is an input used in the analogue to digital conversions of staticising fader and lighting levels. Both highways interface (via internal control and direction registers) with the data bus D0 to D7 and hence with the program.

Address allocation

14. The various parts of the DUET system are all allocated areas of address within the program. The data released and manipulated within these address areas performs the task required of that Sub system. Address areas are allocated in blocks of 1K (or multiples of 1K) and are shown as in Table 2. The following paragraphs describe what occurs in these areas.

Table 2 Address allocation

Address Area (K)	Address count (Hexadecimal)	Function
64 ↓ 56	E000 - FFFF	Program
55 ↓ 40	A000 - DFFF	Spare
39 ↓ 16	4000 - 9FFF	Memory
15,14	3800 - 3FFF	Spare
13,12	3000 - 37FF	VDU interface
11 ↓ 8	2000 - 2FFF	Spare
7,6	1800 - 1FFF	Serial link interface
5,4	1000 - 17FF	Spare
3	0C00 - 0FFF	Contacts / Mimics
2	0800 - 0BFF	Fade processing
1,0	0000 - 07FF	RAM for general MPU use

General purpose RAM

15. On the motherboard is a 2K group of 8-bit random access memory which is used by the microprocessor in carrying out the basic program. It is addressed within the OK and 1K areas. Data areas contain :

(a) channel stores (A,B,T, and S etc.), 9 per channel.
(b) Stack data. This is used to hold the return address for sub-routines and also the original processor state whilst servicing interrupts.
(c) Temporary data (flags, partial products etc.)

Fade processing

16. Fade processing occurs within the 2K area of microprocessor addressing. It follows a channel-by-channel sequence every cycle, the implementation of which is described in the ELECTRICAL section. Fader levels are digitised and put into RAM every cycle. They are then withdrawn, along with the channel ABT and S store data (para 14), for the hardware to perform the fade processing calculations. 16 addresses, 0800 to 080F, call up the steps of the fade processing calculation.

Contact and mimics

17. Each microprocessor cycle the contacts of the control switches, together with wheel movement, are scanned and the display mimics are refreshed. With the exception of the programming switch, an 8-way dual in line device behind the front panel, all controls and mimics are front-panel mounted operator sellection or information.

18. Addresses 0C00 to 0C07 are used together with read and write instructions to provide a read matrix and a write matrix each of eight addresses of eight data bits. This is shown in Table 3. Each control and mimic function is allocated a space within its matrix which represents its state. Allocation of the address / bit to the contacts / mimics is arbitary. The matrix represents the method of interfacing the front panel controls and indications with the program.

Plug-in cards

19. Each plug-in card has its own specific function within the system and operates at the addresses shown in Table 2.

Program routine

20. The program has a fixed series of operations which it can go through each cycle. A large number of these procedures will not be applicable (e.g. the control selections which are not made, the peripheral unit which does not require program time). During each cycle the program determines which particular sub routines are required; those that are are actioned, those that are not are bypassed.

21. Program routine shown in Table 3 can be considered as consisting of :

(a) cycle processing
(b) action routines
(c) peripheral unit routines (VDU etc.)
(d) miscellaneous routines (power-up, interrupt, etc.)

Cycle processing

22. Cycle processing has top priority. It commences with a system generated interrupt and can break in on action routines or the low priority and background routines which service the peripheral units. The system interrupt which starts the cycle processing occurs at 4mS intervals. Most routines are in a sequence started by every 8th interrupt, but contact scanning needs to be actioned more frequently and occurs every 3rd interrupt. Cycle processing consists of the following steps :

(a) The contacts of the control settings are scanned every third system interrupt
(b) If a timed fade is in progress, the T dial is updated every 8th interrupt. Subsequent actions follow this in the sequence (b) to (h)
(c) Digital to analogue levels are output
(d) Fader settings are digitised, A/B and T/S fades are checked for completion or change and operator commanded action routines are set up

(e) When rec and flash timers have been updated the mimic display is refreshed. The timers are used to flash warnings of illegal operation of record or channel / memory selection

(f) The routine for the riggers control unit is examined, and if applicable is executed

(g) Channel processing is performed to determine the lighting state applicable to each channel

(h) When channel processing is complete, IRQ, set up for the duration of channel processing, is cleared and the interrupt mask is removed

(i) The cycle, now being completed, the program moves on to check the action routines.

Action routines

23. Action routines are program subroutines which are established to accomodate selection or adjustment of the front panel controls. Action routine requirements are established during cycle processing once the contact settings have been scanned. They are implemented in a sequence defined by an action routine distributor.

24. There are separate action routines for the following functions :

(a) the distributor which designates the next routine to be done
(b) MEMORY push selection
(c) REC LINK push selection
(d) digit 0-9 selection
(e) + or - push selection
(f) F or . push selection
(g) +1 or -1 push selection
(h) CLEAR selection
(j) view push M,A,B,T selection
(k) Timed fader lever or mins/secs switch
(l) Timed fader lever conversion
(m) Time fade parameters
(n) Setting of the timer LEDS
(p) ↑↓, ↴(A), ↴(B) push selection
(q) REC ABT and REC Σ push selection
(r) Store select A,B,T push selection
(s) SEQ push selection
(t) Actioning of selected channels
(u) GROUP push selection
(v) THRU push selection

(w) CHANNEL push selection
(x) RET push selection
(y) Display of A.B.T. or
(z) Implementation of wheel movement
(aa) Specification of channel capacity (preset)
(ab) Specification of memory size (preset)
(ac) Riggers action routine
(ad) @ push selection
(ae) . (with @) push selection
(af) + or - (with @) push selection
(ag) 0-9 and F (with @) push selection

Peripheral unit routines

25. Beneath the priority level of the action routines comes "low priority" and "background". It is within these levels that programs for the DUET peripheral units - the VDU, floppy disc unit or the riggers unit - are actioned. These units have family tree structures of their own which consist of main and subroutines orgainsed by the program contained on the interface card appropriate to the partivular peripheral unit.

26. A VDU, if installed, uses up the remaining program time every cycle ; serial devices (floppy disc unit, riggers control) obtain program time by generating an interrupt request. This is because the VDU is a synchronous device and needs its display constantly refreshed, while the assynchronous nature of the other peripheral units means they only need occasional access to the main program.

Miscellaneous routines

27. There are a few routines which can be considered under this heading, notably the initialisation processes which are carried out at the time of power-up and the handling of an interrupt request.

Detailed program description

28. The paragraphs which follow cover main cycle processing routines to a fairly full level and action routines to an outline level. Accompanying flow charts are coded both on an overall level and on a step-by-step level to tie in with the computer print-out. Low priority and background routines are not covered.

Timed fade progress (AA)

29. Timed fade progress is the first cycle routine to be carried out. At the start of the cycle a byte of all 1's is output to TP1 on the motherboard as a reference. Every 8th system interrupt causes the timed fade progress routine to start.

30. First, the routine must establish if a timed fade is in progress or if a timed fade was comple ted by the last cycle(B2, B3). Either of these decisions can exit from the routine without the need of processes described in the following paragraphs.

31. The process of a timed fade involves the setting up of a timed fade dial (TF DIAL) which represents, by a count from 0 to 255, the progress of a fade from its starting point (S store) to its terminating point (T store). The routine monitors and updates this dial every cycle by a factor (DL FACT) dependant upon cycle time (32mS) and the fade time (TIME) in $\frac{1}{8}$ths of a second. A record is kept of excess time each cycle by an accumulator (TIMAC) which is also updated every cycle. The accumulator can provide fine adjustment to the dial (or in the case of a slow fade when DLFACT$<$1 each cycle, it can be the sole dial adjuster every few cycles.

32. Processing of the time accumulator every cycle occurs at steps C3, E3, F2, F3, and, assuming that the fade dial does not overflow to signal the end of the fade, the following functions occur :

(a) The time accumulator (TIMAC) has the incremental value (TMINC) added to it (C3)

(b) The result has time accumulator lower boundary factor (TALOW) subtracted from it (E2). Ideally the two (TMINC and TALOW) should cancel, leaving the accumulator with nothing to carry over to the next cycle. If, however, the minimum cycle time of 32mS is exceeded TMINC will exceed TALOW and the difference will be left in the accumulator to carry on to the next cycle

33. The accumulated time is calculated in 1/256 x $\frac{1}{8}$ sec units and is compared (F2) with the fade time (TIME) which is calculated in $\frac{1}{8}$ sec units. Each cycle that the accumulator is the lower value this comparrison will cause an exit from the routine. When sufficient time has accumulated to equal or exceed the fade time figure then the accumulator has the fade time figure subtracted from it. The fade dial is incremented by one and a second comparison of the accumulator and fade time figures causes an exit.

34. The overflow of the fade dial indicates the end of the fade. This can occur as a result of the initial updating (C2, D2) or the dial incrementation (F3, G3). The fade dial is set to all 1's (H4) and the routine is exited. On the next cycle, when the all 1's state is detected (B3), the fade-in-progress flag is cleared (B4) and the timed fade is complete.

35. Two bytes of fade-time information (FDTIME) are used as flags and reference data. The lower byte provides a fader lever position reference 0-255. The higer byte is set as follows :-

Bit 7	1 = minutes, 0 = seconds
6	1 = running, 0 = seconds
5	1 = fade in progress
4	1 = level mode
3-0	Not used

Both these bytes are the result of action routines, (EE) resulting from front panel control adjustment.

Scan contacts and output D/A levels (AB)

36. The purpose of this routine is twofold :

(a) Front panel control selections and adjustments are scanned and any change since the last scan is noted. A record of selection setting is kept in order to set up and run appropriate action routines.

(b) The digital stores representing A/B UP, A/B DN and T fader settings, together with their complements are output as also are a "full" digital reference and the meter level. This is the first step in establishing fade processing analogue registers (Sect. 2, Chap 2).

37. The two actions outlined above are interleaved in an eight step process, although contact scanning is done more frequently than D/A outputting. This is because selections are latched by the program and normal cycle timing could cause them to be missed. Contact scanning is actioned separately every 3rd interrupt while D/A outputting is interleaved with it on the main loop timing, nominally every 8th interrupt. A main loop flag is used to prevent a double scan of contacts when the 3rd and 8th interrupts con-incide.

38. Flow chart AB which illustrates this routine is supported by sub-routine charts ABA which illustrates contact scanning and ABM which illustrates D/A output sub-routine for meter information.

39. The first seven of the eight steps are the same : D/A output is simply a matter of releasing software controlled register contents to the hardware and contact scanning goes through its regular sub routine. On the eighth step the meter D/A is output (separate sub-routine chart ABM) and the wheel movement is scanned and updated (shown on main routine chart).

Contact scanning (ABA)

40. A byte of contact scan information (Table 3) is obtained from the hardware (B2) and is compared with the current program-held data for the same byte (C2). Should a change have been made the routine takes action, otherwise it proceeds to set up the next byte comparison.

41. If a change has been made, the new byte of information is temporarily held, the changed bits are noted and a counter is cleared and then started up (D3). Each increment of the counter is compared with the list of changed bits, the counter progressing until it reaches a changed bit (E3). When this occurs the remainder of the list is temporarily stored while the action routine number is calculated from the byte number and twice the count number (F4). Should the changed bit represent a new selection the calculated number must be incremented by one to set up the associated action routine (G4, H5), but should a de-selection have been made the calculated number will not be incremented by one and the action routine will not occur.

42. When the appropriate action has been applied to the action routine the remaining bits are retrieved and tested until the byte has had all its bits tested. (K4, L3). Then the new byte is stored in program for the next cycle (M3) and the pointers are updated to guide the next testing towards the suceeding byte. If this subroutine has been occuring as part of the main program loop (8th interrupt) it now exits to interleave a D/A output before testing the next byte of contacts (P3). If it is a 3rd interrupt contact scan routine, it reverts to the top of the sub-routine ABA and checks the next byte right away (Q3). In this case the exit, when all bytes have been checked, is to the routine which was interrupted.

Meter level output (ABM)

43. After seven D/A stores have been output and seven bytes of contact information scanned, the 8th D/A action occurs; This is the meter output and it involves its own sub-routine.

44. Provided that a valid channel number has beeen specified (B1) and not channel Ø (C1) then the sub routine gets underway. Firstly it must be ascertained which of the playback stores (A, B, T,) or stage store, holds the level to be metered (D1). Then assuming that the memory view psuh is not being pressed (E1) the address of that channel is calculated and the contents of the store transferred to it (F3). The level is obtained (G3) and output to the meter (L1).

45. If the memory view push is being pressed then (unless the memory number is Ø) (F1), the memory address will already contain the store contents. The address is calculated and the level of the contents obtained (G1). Assuming that the level is not zero (H1) then an interpolation factor is added in bit 1 to stretch the (normally) 6-bit memory to 8 bits (J1). This is then output to the meter (L1).

46. An invalid channel or memory number (B1, C1, F1) causes any previous channel level being metered to be cleared and a zero reading to occur.

Wheel movement (AB)

47. The final step of the main AB routine is to assess wheel movement. The last (8th) byte of contact data represents wheel movement, up or down, every microprocessor cycle. This is produced by hardware on the processor motherboard (Sect. 2 Chap 2) and deposited on the data highway at the end of the contact scan sequence .

48. If the wheel has moved since the last test (K2) and if level mode is established (L3) (by the use of @, flash or the wheel itself) then wheel updating action is taken by action routine EQ. If the level mode is not established, the byte is checked to see if it is within acceptable limits of system jitter (M4) (one data bit of"movement") before setting up the wheel movement action routine. Level mode is established for the next cycle and the routine exits to routine AC.

Digitise fader settings (AC)

49. As a preliminary process before channel processing, the settings, of the manual crossfaders and the time fader need to be digitised. This is achieved by an eight-bit approximation technique in conjuction with the hardware on the motherboard (Sect. 2 Chap 2).

50. Fader settings are digitised in the sequence ABUP, ABDN and T. Each time the digitisation process is required the main routine (AC) branches into a subroutine (AC1) identified by a code specific to the fader being digitised. The routine is entered from routine AB).

51. When ABUP (B2) and ABDN (C2) fader settings have been digitised (para 53) three possibilities must be considered : is a sequence of crossfades required as selected by the SEQ push (D2), has there been fader movement since the last digitisation (D1) and has the latest crossfade concluded, i.e. are the ABUP and ABFN stores equal to the destination store (DSSTOR) (E1). If the answer to all considerations is yes, then the appropriate action routine for the ↴ (A) or ↴ (B) is set up. Otherwise this step (F1) is bypassed.

52. Next the setting of the T fader is digitised (para 53) (F2) and when this is done it is checked to see if a movement freater than one bit occurred (G2). If so then the new value is stored (G3) and the action routine for ↑↓ selection is set up (H3). If not G3 and H3 are bypassed. The routine then exits.

53. Within the main routine the digitising sub routine occurs three times. Each is identical but is referenced by identification codes \$42, \$44 and \$ 46 for ABUP, ABDN and T respectively. These codes, initially loaded in accumulator A of the MPU, are transferred to peripheral register B of the PIA (C4) for the duration of the digitisation. The data sampling gate is opened in readiness for the inputing of the appropriate fader A-D level.

54. Next (D4), MPU accumulator A is cleared and accumulator B is loaded with \$ 80 i.e. 1 in position 7 and 0's in positions 6 - 0. Bit 7 is the shift bit. The value set in PIA PRB (the fader code) is held (E4).

55. The subroutine now goes into its 8-bit comparison phase. The following actions occur :

(a) The shift bit is accumulator B is transferred to accumulator A which us used as the shift register. (F4)
(b) The appropriate bit, A to D converted by the hardware, is loaded into a fade processing A-D register.(G4)
(c) A comparison is organised (H4) between the two registers. Should the shift register value exceed that of the fade processing register value it is necessary to subtract the current shift bit from the shift register (J4). If the fade processing register has the greater value, step J4 is bypassed.
(d) The shift bit is now moved up one position in the accumulator B (K4) and the subroutine is ready to process the next bit of D-A input data. When the 8th bit has been processed (L4) the sub routine is complete and goes back to the main routine.

Refresh mimics (AD)

56. Updating of front panel mimic displays is done on a once-per-cycle basis by main cycle routine AD. This includes the internal illumination of push buttons together with numerical and LED indicators.

57. The routine goes through a sequence of four operations :

(a) record push illuminations, the duration of whose illumination is controlled by a timer

(b) the warning blinker, which goes through an on/off timed routine

(c) calculation and output of the fade progress dial figure

(d) output of mimic data to all other pushes.

58. The routine is entered from routine AC. When a record push is selected, a successful recording is acknowledged by a short illumination of the appropriate push. A timer denotes the length of the illumination. The first thing the routine ascertain is if the timer is running or not (B2). If it is running it is decremented by one (C2) and if this terminates the count (D2) the mimic display drive bits are cleared (E2) and the mimic flag set to indicate the change of state of these particular mimics.

59. The blink timer runs continuously giving an on and off output which is always available for warnings. Its output is controlled by the hardware (Sect. 2 Chap 2). It is necessary to update this timer every cycle (G2) and if the timer has completed an ON or OFF count (H2) to reset it and invert its output (J2). In this case the mimic flag is set.

60. The timed fade dial progresses from 0 to 256 over the duration of a timed fade. Fade progress is registered on the front panel numitron display as single digit; 0 to 9. It is therefore necessary to divide the timed fade dial by 25.6 to obtain the particular single figure digit to be output at any time (L2).

61. Finally, if mimic flags are set by relevant action routines (M2), the appropriate mimics are refreshed (N2) and the routine exits to the riggers routine AH.

(Paras 62 - 66 inc. omitted (for description of the riggers routine AH) — page [illegible]

Channel processing (AE)

66. Allocated to each channel in designated RAM locations are nine stores associated with channel processing. These are :

(a) The A,B,T and S stores which provide a depositary for the selected crossfade or timed fade levels selected of a particular channel

(b) A flag store, each bit of which can contain a specific instruction

(c) A return store which temporarily holds channel levels when the wheel is engaged so that a push of the RET button can re-instate the original level

(d) A stage store which holds the level output to the dimmers for any particular channel. Associated with the store are two other stores, direction and average stores, used in the damping sub-routine.

67. Each cycle the channel processing goes through a complete sequence, starting at the top channel and ending when the first has been actioned. ABT and S store level are output to the hardware (Sect. 2 Chap 2) and the GM (or stage) level calculated there. Stage level for a group of 10 consecutive channels is digitised and stored in RAM, this process being organised in ascending order, (i.e. ch.1-10 one cycle, channels 11-20 or the next, etc.). The stage level of the latest channel selected is also digitised so that it is available as a meter output if required. Considerations of group selection and wheel movement are also included in channel processing.

68. The routine is entered when the mimic refresh routine is completed. The first process (B1) is to set up the group routine. This varies in accordance with wheel movement, up or down, and in which store the channels are being controlled. The pointers are then set up. For most cycles the reqular A/D will be an automatic process, until the regular advance of the channels exceeds the number of channels installed. The end of channel processing routine (Para 75) then restarts the process. The channel number register in the PIA is loaded with the top channel number, less 1 (so that, at the end of the routine, channel 1 will be signified by an empty register).

69. The routine is now set up and succeeding processes recur for every channel. Channel levels in ABT and S are output to hardware as is the flag store (C1) and the fade processing calculation is performed there in analogue operation. A sample gate (bit 7 of PIA PRB) is opened to accept the calculation result for those 10 channels whose stage level is to be digitised this cycle.

70. If the stage level of a particular channel is to be digitised (D1) and if it is one of the group of 10 (E2) then, before digitisation occurs the A/D flag (bit 6) is cleared and entered into the flag store ten channels higher (F3). This re-allocation of the A/D flag is not wanted if the digitisation is for the meter, as flagged by bit 0, and is not within the group of 10.

71. Digitisation of the stage level (G2,H2,J2,K3,L2) is the same process as the digitisation of the fader settings, part of routine AC (para49). There are some minor organisational differences between the two routines, AC and AE, but the technique is the same.

72. When the digitisation has been completed (L2) channel processing goes into a damping sub routine AEJ which culminates in the stage level stores of the 10 channels and/or the meter being refreshed. This sub routine is designed to increase stability when a level approaches the threshold of a percentage step and prevent flicker from one figure to another on the VDU display. As well as being applied here, it is also applied to the T fader level and to the A/B fader levels when these are output to the VDU. Details of sub routine AEJ are not given as it is a refinement rather than an essential step.

73. If the flag bit 5 is set this signifies that a channel is under wheel or keyboard control (N1). Flash or group fade can then be performed on the channel. A/B faders must be designated up or down by flag bit 4 (P2). Subsequent wheel movement is then either added to or subtracted from the level in store (Q2,Q4). A check is kept of the stores reaching full or zero (R2,R4) and when overflow or underflow occurs it is no longer possible to retain the selected balance with other channels and return to the relative starting points by wheel. The only way to do this is to use the return push which releases the return store contents back to the store it was memorising.

74. A check is kept on the number of the channel being processed (T1). The routine repeats until the channel number register in the PIA is empty and the routine then exits to the next routine.

End of Channel processing (AF)

75. As described in the previous paragraphs, channel processing is done by the hardware with 10 stage level stores in RAM being refreshed every cycle. Considering a 48 channel system it will thus take five cycles to update the stage stores for all channels; for a 60 channel system, 6 cycles, etc. This routine (AF) organises a restart to the sequence however many cycles may be needed to refresh all stage stores.

76. A dummy channel is allocated immediately above the top channel number and the addresses of a further nine are never used in the program. When channel processing has completed for a particular cycle the A/D flag bit of the top channel is inspected (C2) except when, on the penultimate cycle of a sequence, an A/D flag in the top channel signifies that the next cycle will be the last of the sequence.

77. The dummy channel has no part in the channel processing routine and does not have its A/D flag cleared in the same way valid channels do. When it contains an A/D flag and the top channel does not, then this specifies that the final batch of channels have had their stage stores updated. This routine then comes into effect and sets the pointers for the next cycle to the first 10 channels.

78. The flag bit is cleared in the dummy channel and the pointer set up for the seventh store (flag store) of channel 10 to insert an A/D bit there (D3,E3). A decrement of 9 to the pointer indicates the flag store of channel 9 and an A/D bit is inserted there. The process continue until the first 10 channels all have A/D flags inserted. The routine is then exited.

Program continuation (AG)

79. At the end of channel processing the main loop functions have all been completed. A check must now be made to determine what (if any) process was underway when an 8th interrupt caused the main loop routine to start.

80. A record is kept by an action routine flag (ARFLAG) of the interrupted routine. The flag bits 0 and 1 are checked and have the following meanings :

(a) An empty flag register indicates nothing was happening prior to main loop functions. This can only be the case immediately following power-up. If this is the case the routine exits to EA, the action routine distributor.

(b) A count of 1 in the flag register indicates that an action routine was interrupted. In this case the exit from this routine is directly to the interrupted routine which is continued from the position it was interrupted.

(c) Counts of 2 or 3 in the flag register respectively indicates that a low priority or background routine was interrupted. In these cases status of the interrupted routine system is preserved by storing the linkages until normal system prioity allows them to be continued. In either of these cases this routine exits to EA.

Action routines

81. Action routines are brought about by changes of the equipment state (mainly front panel selections) and are the instruments by which the program is modified. They are listed in para 24 ; the succeeding paragraphs give a brief resume of each one.

Action routine distributor (EA)

82. The purpose of the distributor is to find the next required action routine and put it in hand. It checks if any actions are held in buffer, and if so, the value and the pointer are retrieved. The action routine flag is set to show an action is being done (flag 0 bit). If no action routines are required and flag bit 3 is set then the program proceeds to the interrupted low priority routine, otherwise it goes to the low priority distibutor routine (BA).

Memory push selection (EBA)

83. The action routine for MEMORY selection organises the following :

(a) the @ action routine module is entered and @ is deselected

(b) mimics are cleared for channel, group, thru' and rec. link

(c) mimic is set for memory

(d) the correct number is set up for display by a sub-section of the view push (MABT) routine (para 93).

Rec Link push selection (EBB)

84. REC LINK can only be selected in the memory mode and in the absence of a memory warning, otherwise the routine exits to the action routine distributor. The mimic bit is set or cleared as appropriate and the memory address obtained. Depending on whether the REC LINK facility has been or is still to be set memory numbers are either adjusted to the required sequence or await this action (which is dependent upon the record key switch being made). The correct number is set up for display by a sub-section of the view push (MABT) routine (para 93).

Digit 0-9 selection (ECA)

85. This routine first checks if the @ module (EZ) is in use before proceeding. If a GROUP selection is current this is cleared. The address of the current keyboard number, and whether in memory or channel, is obtained. If in memory mode, the memory warning is cleared, and the way cleared for the keyboard entry.

86. The number is entered by first masking off the 100's, multiplying any previous selection by 10 and then adding in the new digit. The entry is then checked for validity, be it channel or memory. If it is a valid entry then the routine completes via a sub-section of the view push (MABT) action routine (para 93) which displays the correct number. If invalid, error and used flags are set (and memory warning if it is an invali memory entry) before the routine exits.

+ or - push selection (ECB)

87. This routine first checks if the @ module (EZ) is in use, then whether REC LINK has been selected. If neither of these situations apply the routine continues. When + or - selection is part of a channel instructi the previously selected channel number and selection of CHANNEL / GROUP / THRU (as applicable) must be actioned. Then, whether in CHANNEL or MEMORY a sub-section of the view push (MABT) action routine is entered to display the + or - symbols.

F or . push selection (ECC/ECE)

88. These selections are only used in the @ module (EZ) (para 119).

+1 or -1 push selection (ECD)

89. Firstly, @ module processing is de-selected, then the address of the keyboard number last selected is obtained and the relevant MEMORY or CHANNEL selection retained. If the numeric portion of the last keyboard selection in store is zero, this means that it has been cleared, and must be retrieved from a special store which holds it against this eventiality. The number is incremented or decremented and then output to be checked for validity in routine ECA (para 86).

Clear selection (ECF)

90. The CLEAR key is used to cancel a keyboard channel/memory selection. First step in the routine is to de-select @ module processing (EZ), if applicable. Then the keyboard number is obtained and cleared. If operating in the channel mode, any group flags in the channel stores are cancelled and the correct number is set up for display by a sub-section of the view push (MABT) routine (para 93).

View push (MABT) selection (ED)

91. Each of the four selections has its own entry to the action routine when its own particular identifying code is loaded. Two or more view selections can be made simultaneously, this routine causes the latest selection to be viewed. Its code is placed on a view stack and the appropriate number is moved to a display buffer. The number is then converted to BCD before being output via a display buffer. The 'mimic changed' flag is set.

92. A de-selection of a view push puts the following sequence of events in hand. The position of the code in the view push stack is found and then it is cleared. The stack is then "shuffled" to determine if any other view push is being held selected, even though the latest selection has been released. If so, its number is converted to BCD and output as before, but if not then the view pushes no longer require to output a selection.

93. The remaining stages of this routine determine which number is to be displayed in the absence of any view push selection. It is the point at which other action routines enter this routine when a numerical display is required. Any one of four numbers be output. These are :

(a) channel number

(b) memory number

(c) group number (when CHANNEL and GROUP selected)

(d) sequence number (when MEMORY and REC LINK selected).

The number is converted to BCD and output as before, and the 'mimic changed' flag is set.

Timed Fader lever or mins/secs switch (EE)

94. Action routine EE is a top routine which includes EEA, EEB and EEC as sub routines. It first ascertains if a fade is in progress or not. If not it exits but if so then it branches to subroutine EEC to set the appropriate timed fade LED. If it is a fade from memory the action routine exits but if it is under channel control then the timed fade lever and switch need to be loaded and the sub routines EEA and EEB are entered. These calculate and load the new time parameters used by the timed fade progress routine AA. The action routine now exits.

Timed fader lever conversion (EEA)

95. The purpose of this sub routine is to convert the fader lever position into terms of time by a six part approximation. The calculation only occurs if the timed fader lever is set to a position recognisably above zero, and branches into sub routine EEAX to do so.

96. The calculation is performed using a two-part table, FDTAB. Entrie 0-5 are the top limits in each range and entries 6-11 are the related constants for each equation. The result is in $\frac{1}{8}$ths of a second. If the fade-time switch is set to minutes then the result is multiplied by 60 as the final step of routine EEA. The routine now exits to EEB.

Timed fade parameters (EEB)

97. The parameters used by the timed fade progress routine (AA) in the main cycle processing loop are set up by this action routine. It operates following the selection of a timed fade (EFC), and keeps quotients used by the main loop operation up to date. It also controls the fade-running flag and the interrupt mask. The sub-routine exits to the main action routine EE.

Setting of timer LEDS (EEC)

98. If in the memory mode and the timed fade lever is not set to the stored time then both LEDS are cleared. If in the channel mode or the memory mode with the timed fader lever registering the stored time then the appropriate LED mimics are set, minutes, seconds or (if the lever is zero) both.

↓↑ , ↴ (A) and ↴(B) push selections (EF)

99. Action routine EF is a particularly long routine encompassing EFA, EFB and EFC. Its purpose is to transfer the memory called up on the keyboard or reached in sequence into the playback stores for T, A or B. A flag word indicates which action ; 0 = ↓↑, 1 = ↴ (A) and FF = ↴(B).

100. The routine only occurs for a valid memory ; if a memory warning is current it exits to the action routine distributor EA. For an A or B transfer the crossfade parameters introduced completely replace the exiting contents unless the memory number has a + or a - prefix, in whih case the memory is added (highest or latest takes precedence) or subtratced (common channels to zero). The pushes illuminate after a successful transfer and remain lit until a blank state (by 'clear') is transferred. Then the store is cleared and the push extinguished.

101. For a T transfer the new parameters are fed into the next store in the timed plyback and starts a dipless crossfade. This takes as long as the time set on the lever and goes from the current state of the playback to the new state. This results in the complete replacement of the current state by the new state unless the memory transfer was prefixed by a + or a -, in which case the new memory will be added (highest or latest takes precedence) or subtracted from the current state during the fade. If a new memory is transferred whilst a fade is running the current fade is halted and the fade to the new state is started from this point. The push illuminates after a successful transfer and remains lit until clear is selected.

Rec ABT and RecΣ push selection (EG)

102. The first step in this action routine is to set up an offset in channel store for the store to be recorded and then modify RAM 1 with the offset to pick up the appropriate store level. Then, if the record keyswitch is ON, no memory warning is current and the memory number is valid, the routine proceeds to obtain the memory address and move the appropriate store to memory. If the mins/secs switch is set to other than the centre (memory) position the time delay must be obtained from the lever + switch setting. The appropriate mimic is lit for 2 seconds to indicate the transfer has been successful. A mimic changed flag indicates this and a 'memory number used' flag is also set before the routine exits to the action routine distributor.

Store select A,B,T push selection (EH)

103. First an offset is set up for the appropriate A,B or T store. If the store is selected for display, the new contents are calculated, double precision, under an interrupt mask. If in the level mode, return stores ar set up by a stage of action routine EN and the @ mode of operation is de-selected. Then the mimic for the selection is set and the mimic-changed flag set. The routine exits back to EA.

Seq push selection (EI)

104. This routine causes the SEQ push mimic bit to be inverted and when the MEMORY mode is selected causes the memory number to be incremented by 1 after a transfer. If a sequence link number is recorded for the memory transferred, then the memory number automatically jumps to the link number. The routine also causes an automatic take action into the inactive side of the manual playback at the completion of a crossfade.

Actioning of selected channels (EJ)

105. Channel mode must be selected otherwise the routine exits immediately. If THRU is selected then all channels within a particular memory are systemmatically searched out and sub routine EJB applied for each. If neither of the above selections are made a single channel action occurs in subroutine EJA (provided that the channel is unused). When one of these three alternatives have occured the level mode is cleared and the routine exits.

106. Subroutine EJA/EJB is one routine with two entry points. EJA is the complete routine of preamble and channel execution. EJB aviods the preamble. Thus for a sequence of channels (as selected by THRU) EJA occurs for the first channel, but EJB is sufficient for all subsequent channels. Channel-used flags are set for selected channels.

Group push selection (EK)

107. This routine depends on CHANNEL being selected, otherwise it exits immediately. First the memorised group number is cleared, then the system channel number is investigated. If it has no used flag and its contents are above zero, then its sign is held by the memorised group number while the routi ne branches to action routine EJ to action the selecte channels. The group mimic is set and the mimic changed flag is set before the routine exits to set up the correct number for display by a sub section of the view push (M.A.B.T.) routine (para 93).

Thru push selection (EL)

108. This routine depends upon CHANNEL being selected and upon the channel number being unused and having contents greater than zero inserted at this time, otherwise it exits immediately. The system channel number then becomes the thru channel number and a thru flag is set. Then the thru mimic is set and the routine exits to set up the correct number for display by a sub section of the view push (M.A.B.T.) routine (para 93).

Channel push selection (EM)

109. If MEMORY is currently selected at the start of this routine, it (and REC LINK, if applicable) is cleared and its mimic reset. The mimic is set for channel and the routine exits to set up the correct number for display by a sub section of the view push (M.A.B.T.) routine (para 93).

Ret push selection (EN)

110. This routine has two parts. One part sets up and keeps a record of levels of particular channels when these are varied from the original settir This is done by a jump from the routine varying the channel level, the operation of the wheel (routine EQ). It then returns to that action routine The other part of the routine is brought about when the RET is actioned. The routine is then entered from the action routine distributor (EA), reinstates the return levels into channel stores and exits to EA.

Display of A.B.T or Σ (EO)

111. This routine has two parts. The first part is entered from the action routine distributor (EA) when a selection has been made. The display switches are a bistable pair, so the mimic settings must be changed over, the mimic-changed flag set, the second part of the routine carried out before exiting to EA.

112. The second part is a sub routine which is entered from the first part and also from the action routine controlling the level of store or output. In this sub routine the address of the channel is calculated and a display store set up. If an output display is required a special A/D flag must be raised to output that channel. The stage or store offset is added into the display store and the channel address is saved. The sub routine then exits back to the main routine or back to the action routine from which it entered.

Implementation of wheel movement (EQ)

113. This action routine is subsidiary to the actioning of selected channels (EJ) which has a prerequisite that CHANNEL has been selected. Each cycle wheel movement is added into a special contact scan (AB) word, CTWEEL. This is derived from a calculation performed by the hardware. If the wheel movement word contains a significant ammount, the action routine gets underway.

114. If in the level mode, CTWEEL is picked up and cleared then converted to an unsigned 8-bit value and direction and movement flags are set. This is all done under an interrupt mask. If not in the level mode the return stores are set up, the return action routine EN is actioned, CTWEEL and the wheel accumulator (WLACC) are cleared and the level mode flag is set for the next cycle. If there are significant contents in WLACC then the return mimic is set, but if not it is cleared. The routine exits to set up the current number for display by a sub section of the view-push (M.A.B.T.) routine (para 93).

Specification of channel capacity (ER) and memory size (ES)

115. These action routines introduce the preset parameters of the system into the program. The channel capcity is defined by the setting of bits 3-7 of the program switch and placed upon CTPROG (see table 3). Memory size is defined by the capacity of memory installed. In normal operation neither of these routines will be actioned ; only at times of power-up or system

change do they become effective.

Riggers action routine (ETA and ETB)

116.

@ push selection (EZ)

117. Selection of the @ key brings about an action routine "module" in which ., + or -, 0-9 and F are also actioned when selected as a subsidiary command to @. The @ module must be entered by a special routine (FA). A code word (MCTRL) contains a bit for each module, each bit being represented by a defined name, e.g. MCTLAT for the @ module.

118. The purpose of the module is to control the setting level by use of the @ key on the keyboard. Five flags in a flag word (XXATFG) are used. These are :

Bit 0	decimal point flag
Bit 1	sign flag
Bit 2	flag 1
Bit 3	flag 2
Bit 4	0 = + ; 1 = - in conjunction with Bit 1

Decimal point flag, when set, implies that the next digit is units (N.B. F - Key invalid). The sign flag, when set, implies that the value is relative to the present level, rather than absolute. Flag 1 when set implies that the next digit is a 10's digit. Flag 2 when set implies that the @ processing is active. Input to the module is either 10's or units depending upon whether the @ mimic is set for full or half illumination

119. Entries to the @ module routine are from the previously discussed action routines ECA, ECB, ECC and ECE (para 85 et seq.). Each has its own point of entry, as also do selection and de-selection of @. Exit can either be back to the sub routine from which the entry was made or to the action routine distributor (EA).

Miscellaneous routines

Power up (GA)

120. At the time of power up the system needs to be reminded of its parameters and to be set to an initial state with any residual commands or actions cancelled. First the channel capacity and memory size action routines (ER, ES) are performed. Then the action routine addresses are loaded into the action stack and the view push stack cleared. The RAM routines, the PIA and the stack pointer are set up and a mimic changed flag set. When complete the power-up routine exits to the top of the main cycle, routine AA.

Interrupt handler (JA)

121. Interrupts are hardware generated, either via the PIA on a regular 4mS basis, or from the ACIA when a floppy disc or a riggers unit facility is included in the system. In the latter case interrupts are of an intermittent nature.

122. The first step is to determine which type of interrupt it is, PIA or ACIA, then to read that device, and in so doing, to reset the interrupt request. For a PIA interrupt each 3rd and each 8th is extracted to initiate contact scanning (within routine AB) or the complete chain of cycle processing routines (AA to AH) respectively. PIA interrupts other than these cause a return to the interrupted routine. For an ACIA interrupt the action routine for either a floppy disc or riggers facility is set up as appropriate, before this routine exits to the routine which it interrupted.

Other routines

123. Other routines not described can be summarised as :

(a) BA - low priority and background routine distributor
(b) C prefix - VDU routines
(c) Y prefix - mathematical calculations and conversions.

Memory address tags

124. The following tags are used within the program. They are chosen to be easily memorised, once introduced. Coverage of these tags is by routine function.

Timed fade

125.	(a)	TIMAC	Time accumulator, double precision in two bytes
	(b)	FDTIME	Time for fade, coded in two bytes. The higher byte represents : Bit 7 - 1 = mins, 0 = secs Bit 6 - 1 = stopped. 0 = running Bit 5 - 1 = fade in progress Bit 4 - 1 = in level mode Bit 3-0 Not used The lower byte represents fader level position 0-255.
	(c)	TAINC	Time accumulator cycle increment
	(d)	TAUPP	Time accumulator upper boundary, double precision in two bytes
	(e)	TALOW	Time accumulator lower boundary
	(f)	TFDIAL	Fade progress dial
	(g)	DLFACT	Dial cycle incremental factor

Memory numbers

126.	(a)	XSYMNO	System memory number
	(b)	AMEMNO	A-store memory number
	(c)	BMEMNO	B-store memory number
	(d)	TMEMNO	T-store memory number
	(e)	XSYSEQ	Sequence number
	(f)	XSYGRP	Memorised group number
	(g)	XHIMNO	Hidden memory number
	(h)	XHISEQ	Hidden sequence number
	(i)	XMEM	Number of cues

All but (i) use two bytes.

Channel numbers (sockets)

127. (a) XSYSOC System channel number

(b) XHISOC Hidden channel number

(c) XTHSOC Thru channel number

(d) CHEND End address of current channel store

(e) WLACC Wheel movement from position in return store

All the above use two bytes

(f) XWAYS Number of channels

(g) CHSTOR 1080 channel dependant stores. This consists of up to 120 channels each requiring nine stores in per channel sequence of T, B, S, A, return, stage, flag, direction and averager, (the latter two used on a damping routine).

Action routines

128. XACTS Action stack, used as a circular buffer, first in, first out

VPSTAK View push stack, last in, first out

* XACTE End of action stack

XACT Current output address in action stack

XACTL Current input address in action stack

VPSTP View push stack pointer

XMEM Start address of current cue

XWK General action routine work area.