MICROPROCESSOR-CONTROLLED LIGHTING SYSTEM

This final article describes the overall operation and performance of the prototype lighting system. Details are given of the operating system, equalization table, and the hardware required to set-up the control desk's processing and recording modes.

by John D. H. White and Nigel M. Allinson*

Before discussing the operating software used in the lighting system, its relevance is best understood by considering the layout of a typical control desk, as shown in Fig. 1, and how such a desk is operated. The desired lighting pattern is set on the channel faders (presets), and this pattern is stored in the processor-system memory by pressing the 'record' button associated with a particular master fader, or 'master preset'. This pattern will be recalled and sent to the dimmer modules whenever its associated master preset is not at zero. Assuming for the moment that only one master preset is at a non-zero setting at any one time, any other master preset may now be used and another lighting pattern set in the same manner. Hence, a complete lighting pattern may be stored for each master preset.

There are two ways in which these stored patterns may be controlled using the master presets.

– Scaling – the equivalent of analogue control-desk processing – in which each preset level is multiplied by the master preset level and the resulting signals sent to the dimmer channels. Relative levels of the channels are maintained at all times.

– And stepping, where the master preset level is compared with the stored preset levels and the lesser of the two levels used for output. This type of processing is used to build up a lighting pattern, i.e., all dimmer outputs rise according to the level of the master preset and then stop at their predetermined levels. In an analogue control desk, this type of processing would require very complex circuits.

By using more than one master preset at a time, lighting patterns can be gradually changed from one stored pattern to another. As the operating program endlessly polls all the faders and record buttons, any lighting pattern produced by a combination of master and channel presets may be recorded by simply pressing the appropriate master-preset record button.

Operating Software

The operating program and the 'look-up', or equalization table, are contained in just over 1/2Kbyte of p.r.o.m. The requirements for r.a.m. depend on the overall size of the control desk. Around 256 bytes are required for the operating program and $N(M+1)$ bytes for lighting pattern storage, where N is the number of channel presets and M the number of master presets. Except for the largest of systems, 2Kbytes of r.a.m. will suffice. Organization of the data structures is shown in Fig. 2. The present memory stores the lighting-pattern preset levels associated with each master preset. The output-buffer memory is used to store the required lighting pattern temporarily, before the levels are converted to output signals for the dimmer modules, using the equalization table.

The equalization table performs two important functions. Firstly, the scaling process entails the multiplication of numerous channel and master preset levels. Without an external multiplier unit, most microprocessors carry out multiplication relatively slowly (some recent microprocessors, such as the 6809 and 9995 have such a multiplier internally). The multiplication problem could have

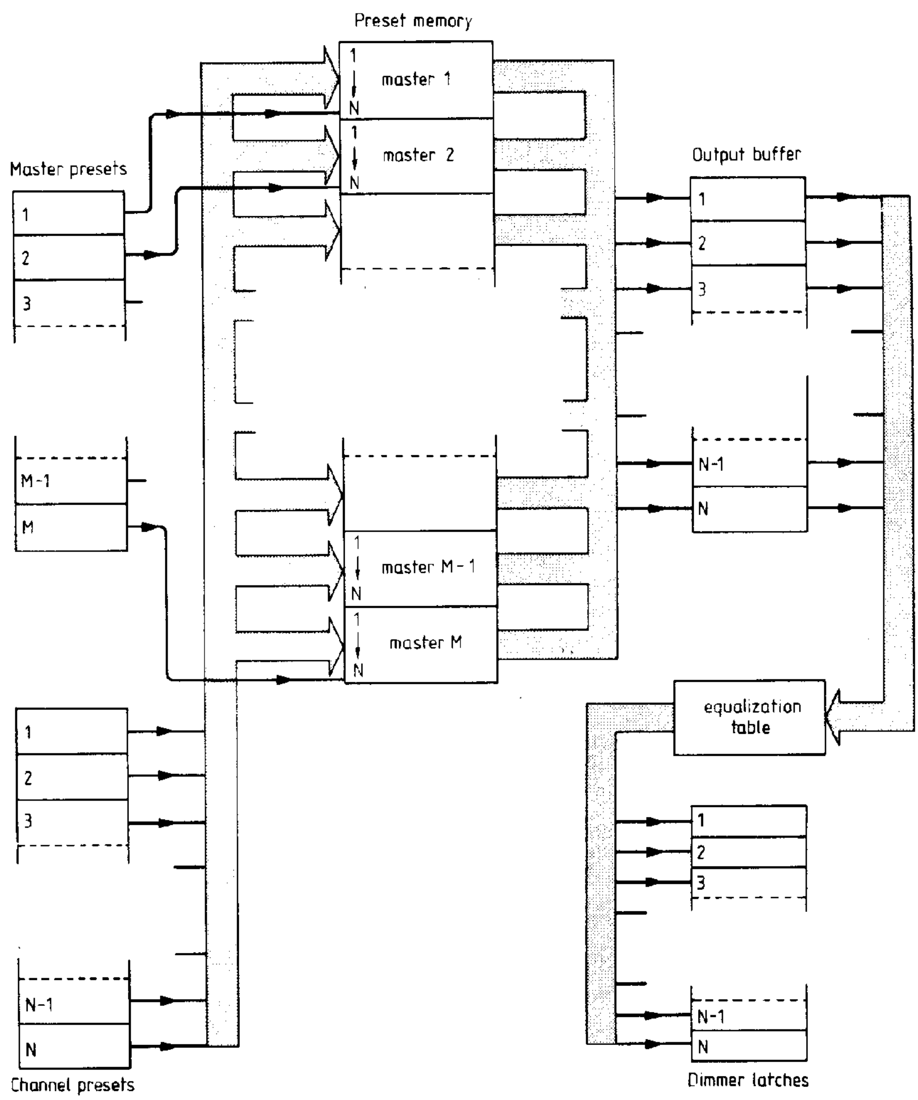
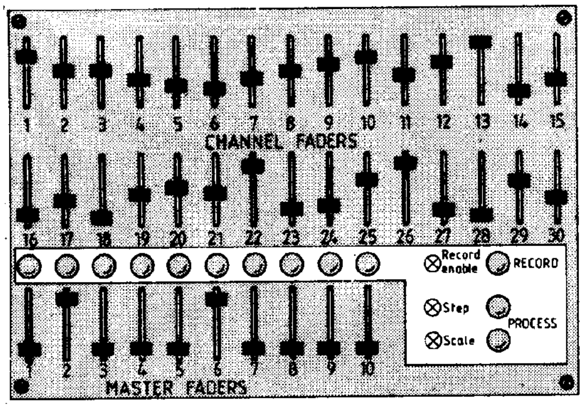


Fig. 2. Data structures used in the control-desk software. M is the number of master presets, and N is the number of channel presets.

* University of Keele



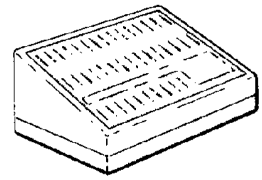


Fig. 1. Layout of a 30-channel/10-master control desk.

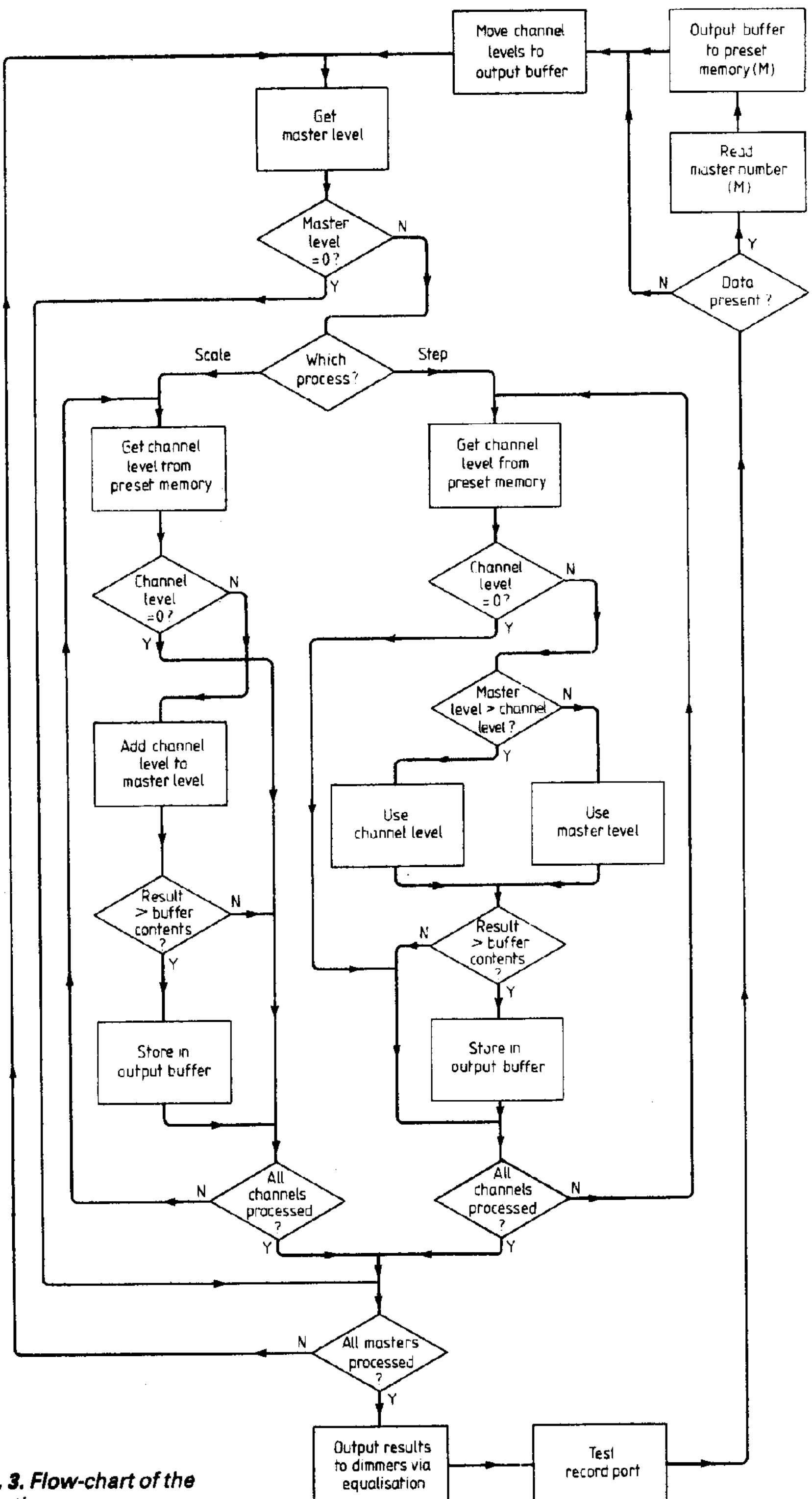


Fig. 3. Flow-chart of the operting program.

been solved by using a logarithmic a-to-d converter, but in this case, logarithmic-law faders were used together with a look-up table containing base 2 antilogarithms for the 256 possible levels – hence, multiplications become simple additions.

Secondly, the table provides compensation for the non-linear relationship between the fader position and the subjective brightness of the lamps, mentioned in the first article. This code transformation is fairly difficult to formulate, and will be of more general interest than the code transformation used in the prototype system which combines both this subjective brightness compensation and the antilogarithm conversion, so this coding is given in Table 1.

The operating program is not listed because it is specific to the processor used and consists of only eight short sub-routines and three core-routines for lighting pattern recording and processing. However, using the flow-chart of Fig. 3, it should be possible to program most microcomputer systems to provide the facilities described. The program tests data present on the data bus to decide whether scaling/stepping processing, or recording mode is required. The hardware needed for this is described in the next section. Note that, to reduce processing time to a minimum, there are a number of conditional branches dependant on channel or master levels being zero.

Process/record select circuits

The operating program must test whether stepping/scaling processing, or pattern recording is required. This could be achieved by connecting the control desk's record and process keys, through some form of keyboard encoding, to a programmable i/o device (such as the 8155/6). However, since mapped-memory techniques are used for all other data input and output, a single i/o port can be connected directly to the data bus which is enabled when the IO/$\overline{M}$ status line goes high. Figure. 4 shows the process/record-select circuit. When the 'record enable' key is pressed, the octal encoder (74148) is enabled and its output will stay high until a master-preset record key is pressed. The three RS flip-flops connected to the octal encoder are reset, and hence the 4-bit binary counter (74163) is enabled. The counter outputs are connected, through a 4-to-16-line demultiplexer (74154), to sixteen cross-lines in the master-preset 'record' key-matrix. When a key is pressed, at least one of the encoder's outputs goes low and disables the counter. The three-state buffer is enabled when either $\overline{E}$, W/$\overline{R}$ or M/$\overline{IO}$ is low, and the input data is transferred to the processor data bus. Also, the four inputs to the NAND gate (1/2 7420) are high, and on the next rising edge of the system enable, $\overline{E}$, a '0' is clocked out of the D-type flip-flop and the four RS flip-flops are reset. The next $\overline{E}$ pulse will enable the system again. The 0 input of the octal encoder is not used, as a low level on this input will cause all three outputs to be high (i.e., equivalent to no key being pressed).

Table 1: Code conversion for subjective brightness correction: I is input, in hexadecimal form, O is output, also in hexadecimal form and L is relative luminous intensity.

I	O	L	1F	3D	224	I	O	L	5F	60	160	I	O	L	9F	7C	97
			20	3E	223				60	60	160				A0	7D	95
00	00	255	21	3F	222	40	51	191	61	61	158	80	6E	129	A1	7D	95
01	12	254	22	3F	222	41	51	191	62	61	158	81	6F	126	A2	7E	93
02	17	253	23	40	220	42	52	189	63	62	156	82	6F	126	A3	7E	93
03	1B	252	24	41	219	43	52	189	64	62	156	83	70	124	A4	7F	91
04	1E	251	25	41	219	44	53	187	65	63	154	84	70	124	A5	7F	91
05	20	250	26	42	217	45	54	186	66	63	154	85	71	122	A6	80	89
06	22	249	27	43	216	46	54	186	67	63	154	86	71	122	A7	80	89
07	24	248	28	43	216	47	55	184	68	64	151	87	71	122	A8	80	89
08	26	247	29	44	214	48	55	184	69	64	151	88	72	119	A9	81	86
09	27	246	2A	45	213	49	55	184	6A	65	149	89	72	119	AA	81	86
0A	29	245	2B	45	213	4A	56	181	6B	65	149	8A	73	117	AB	82	84
0B	2A	244	2C	46	211	4B	56	181	6C	66	147	8B	73	117	AC	82	84
0C	2B	243	2D	46	211	4C	57	179	6D	66	147	8C	74	115	AD	83	82
0D	2D	242	2E	47	209	4D	57	179	6E	67	145	8D	74	115	AE	83	82
0E	2E	241	2F	48	208	4E	58	177	6F	67	145	8E	75	113	AF	84	80
0F	2F	240	30	48	208	4F	58	177	70	67	145	8F	75	113	B0	84	80
10	30	239	31	49	206	50	59	175	71	68	142	90	75	113	B1	85	78
11	31	238	32	49	206	51	69	175	72	68	142	91	76	110	B2	85	78
12	32	237	33	4A	204	52	5A	173	73	69	140	92	76	110	B3	86	76
13	33	236	34	4B	203	53	5A	173	74	69	140	93	77	108	B4	86	76
14	34	235	35	4B	203	54	5B	171	75	6A	138	94	77	108	B5	87	74
15	35	234	36	4C	201	55	5B	171	76	6A	138	95	78	106	B6	87	74
16	36	233	37	4C	201	56	5C	169	77	6A	138	96	78	106	B7	88	72
17	37	232	38	4D	199	57	5C	169	78	6B	135	97	79	104	B8	88	72
18	38	231	39	4D	199	58	5D	167	79	6B	135	98	79	104	B9	89	70
19	38	231	3A	4E	197	59	5D	167	7A	6C	133	99	7A	102	BA	89	70
1A	39	229	3B	4E	197	5A	5E	165	7B	6C	133	9A	7A	102	BB	8A	68
1B	3A	228	3C	4F	195	5B	5E	165	7C	6D	131	9B	7A	102	BC	8B	67
1C	3B	227	3D	4F	195	5C	5E	165	7D	6D	131	9C	7B	99	BD	8B	67
1D	3C	226	3E	50	193	5D	5F	162	7E	6E	129	9D	7B	99	BE	8C	65
1E	3C	226	3F	50	193	5E	5F	162	7F	6E	129	9E	7C	97	BF	8C	65

Hence up to seven modules may be used, each with 16 master presets.

Processing-mode keys are simply latched by an RS flip-flop and connected to D_7 of the data bus. Unlike the record-key data, they do not form a destructive read circuit.

Conclusion

For the 8085A microprocessor used and with a system clock of 3MHz, Fig. 5 shows the experimentally determined relationship between processing speed (that is, the time taken for all the output channels to be updated) and the number of master presets in use at any one time. Assuming that the minimum acceptable update frequency is 20Hz, a 60-channel desk will operate fast enough, provided that less than about 20 master presets are in use. This limitation will not usually effect operation as the maximum number of master presets in use at one time is normally from four to eight.

There are a number of ways in which this prototype system could be extended. In most stage applications, the control desk and dimmers are remote from each other, and the system described would require an expensive 40-way connecting cable. Some form of high-speed serial interface, with high noise immunity, would be of greater practical use. Because the

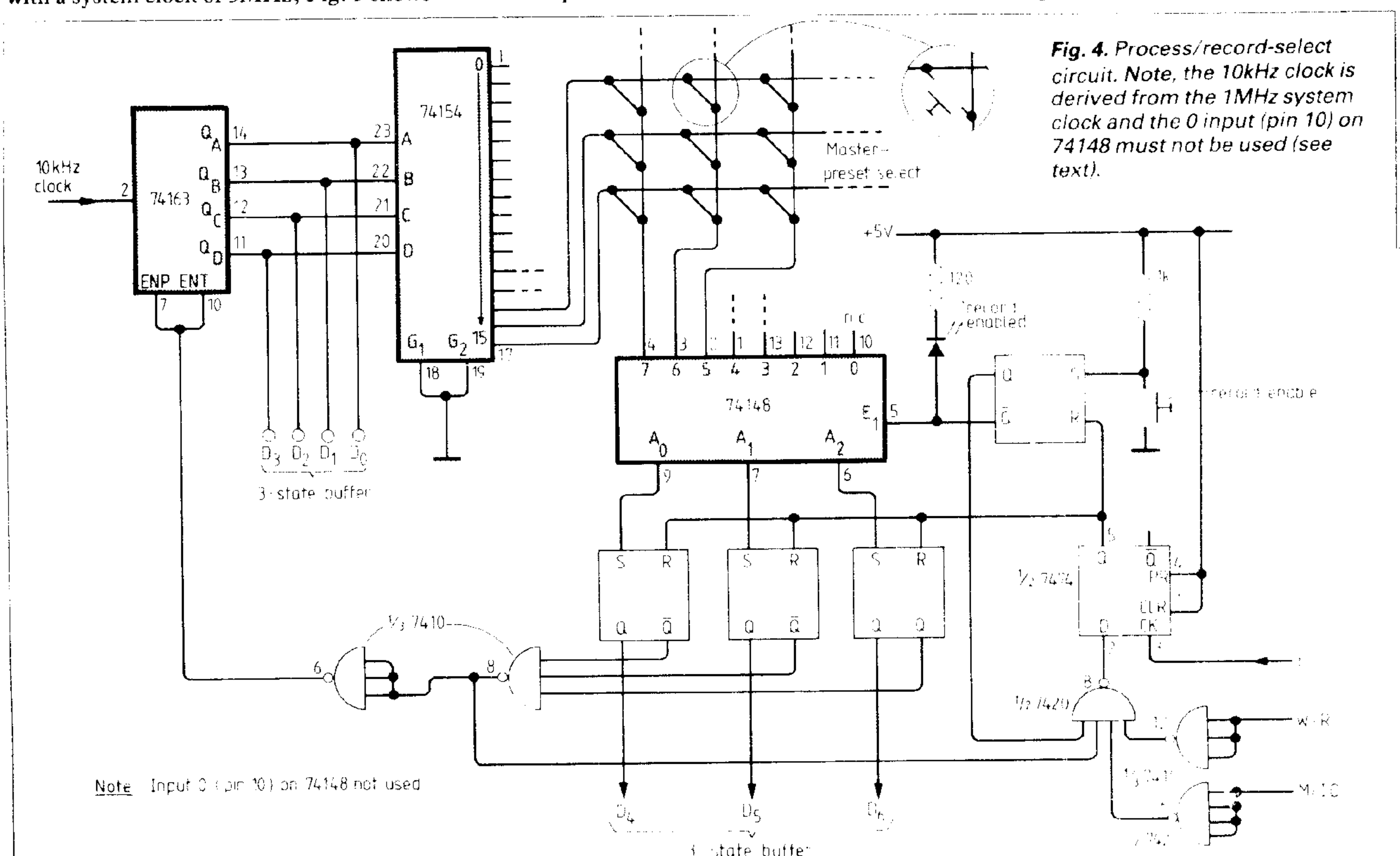
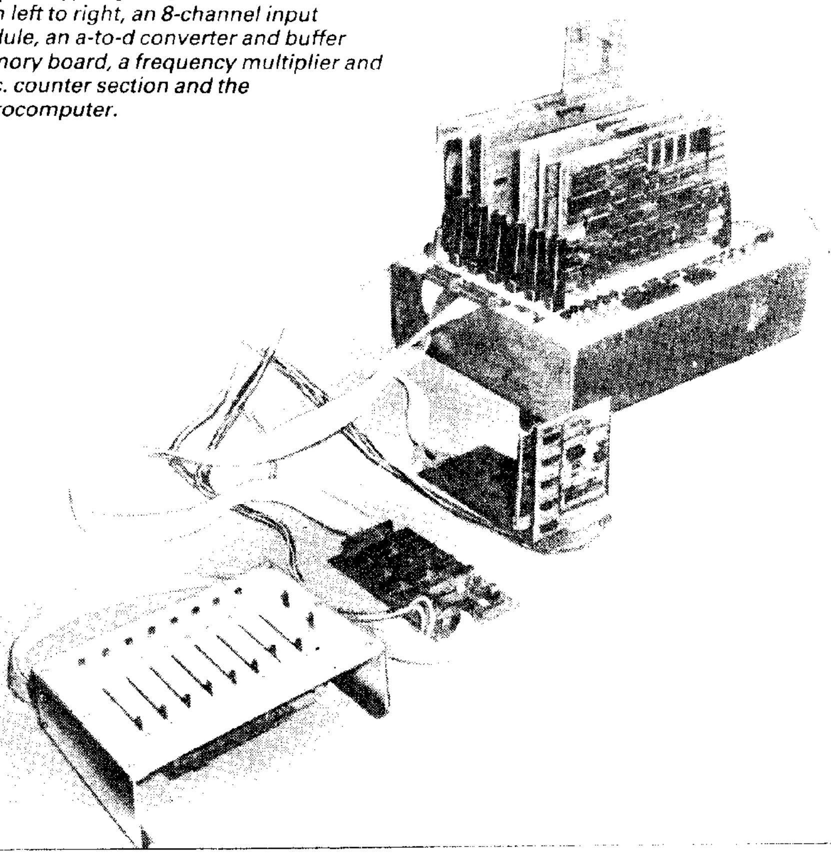


Fig. 4. Process/record-select circuit. Note, the 10kHz clock is derived from the 1MHz system clock and the 0 input (pin 10) on 74148 must not be used (see text).

I	O	L			
			DF	A1	32
			E0	A1	32
C0	8D	63	E1	A2	30
C1	8D	63	E2	A3	29
C2	8E	61	E3	A4	28
C3	8E	61	E4	A5	27
C4	8F	59	E5	A6	26
C5	8F	59	E6	A7	25
C6	90	57	E7	A8	24
C7	91	56	E8	A9	23
C8	91	56	E9	AA	22
C9	92	54	EA	AB	21
CA	92	54	EB	AC	20
CB	93	52	EC	AD	19
CC	94	51	ED	AE	18
CD	94	51	EE	AF	17
CE	95	49	EF	B0	16
CF	95	49	F0	B1	15
D0	96	47	F1	B3	14
D1	97	46	F2	B4	13
D2	97	46	F3	B6	12
D3	98	44	F4	B7	11
D4	99	43	F5	B9	10
D5	99	43	F6	BB	9
D6	9A	41	F7	BD	8
D7	9B	40	F8	BF	7
D8	9B	40	F9	C1	6
D9	9C	38	FA	C4	5
DA	9D	37	FB	C7	4
DB	9E	36	FC	CA	3
DC	9E	36	FD	CF	2
DD	9F	34	FE	D6	1
DE	A0	33	FF	FF	0

The prototype lighting control system with, from left to right, an 8-channel input module, an a-to-d converter and buffer memory board, a frequency multiplier and sync. counter section and the microcomputer.



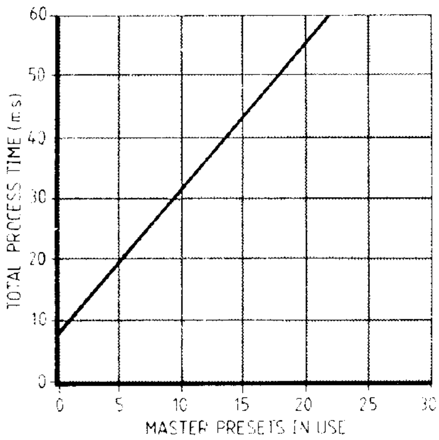
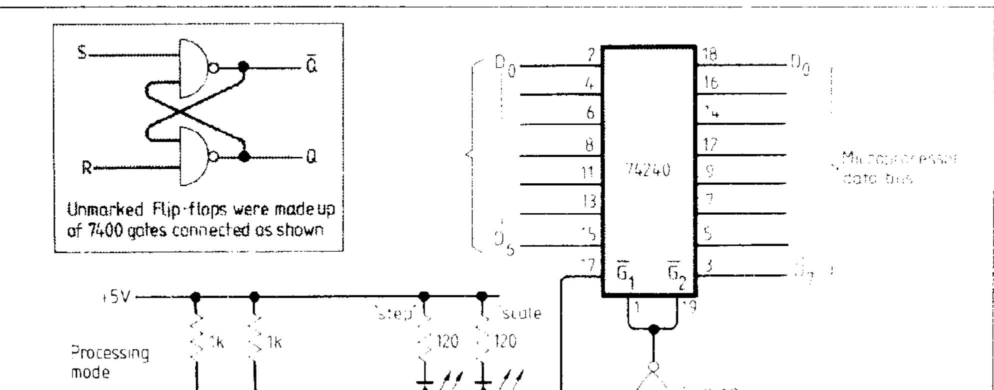


Fig. 5. Experimentally determined process time, i.e., time for updating all dimmers, for a 60-channel control desk.

major cost of most installations is in the dimmer modules, the replacement of an analogue control desk with a digital one would most effectively be achieved by providing a low-cost interface between the digital output data and the existing 0-10V-direct voltage-controlled dimmers. More permanent lighting pattern storage could be easily achieved by providing either battery back-up for some of the r.a.m., or a tape interface. In exacting situations where the colour, or hue, of lights has to be maintained as their luminosity changes, the software could be extended, and lighting sets with three primary coloured gels used.

Estimates of the cost of a digital control desk compared with a conventional analogue desk suggest that the former solution is the cheaper alternative for systems with more than 40 channels and 20 master presets. The fixed cost element of the microprocessor system is offset by the absence of a diode-matrix board and the large reduction in the number of faders.

The authors are grateful for the use of the microprocessor development facilities provided in the final-year Electronics laboratory at the University of Keele, and the technical help provided by B. W. Cornes and F. J. T. Greasley.



From page 46

Computation

The calculations illustrated may be carried out with the aid of a pocket calculator. Access to a computer is clearly advantageous and Fortran programs have been previously published [3] for the procedures required. For convenience, four simple Basic programs are presented on page 46 for the pole and zero calculations constituting the most difficult part of the computation. These programs may be readily incorporated into a complete digital filter design package for a desk-top computer. The programs have been developed and listed on a Hewlett Packard HP 85 microcomputer which allows several statements per line, separated by @. The command DISP may have to be replaced by PRINT in may Basic implementations.

Bibliographical references have had to be held over and will be included in a third part of this series.