MC-8

MicroComposer

—INSTRUCTION MANUAL—

Roland

CONTENTS

1. Introduction

> If you are in a hurry to begin experimenting, read Section 3 (Connections, p. 5) and then try the experimental programs in Section 5 beginning on page 54.

The logic used in the operation of the MicroComposer is consistent. You will find that once you start using the MicroComposer, it is not anywhere near as difficult to use as a cursory glance through this manual might seem to indicate.

The material in this manual is arranged more or less in the order it would be needed to encode the score, program the MicroComposer, edit the program, and produce the finished music. A certain amount of repetition and cross referencing has been used so that this order may be varied if desired, and to make review easier.

One thing you should know before you begin experimenting: If all of the display LED's start flashing on and off, this indicates that the ERROR function has been activated. This is discussed in Section 17, but it can often be corrected by re-punching the correct button sequence; otherwise, for the time being, you can start all over from the beginning by turning off the POWER switch for a second or so to clear all the memories.

2. What is the ROLAND MC-8 MicroComposer?

What every electronic music studio needs is a device that can be programmed to control the production of sound for recording each voice line. The ideal device would have the following:

1. a method of storing information for all of the variables in a sound.
2. a synchronous function for multi-channel recording.
3. a high capacity for recording full compositions without break.
4. provisions for changing and editing any of the information during and after loading.
5. the load mode must have some form of monitoring provision to help in the programming.
6. quick access time during the recording process so the device will play passages faster than humanly possible.
7. the device must be able to add all emotional qualities to the music as desired by the musician.

The answer . . . . . .

The ROLAND MC-8 MicroComposer is the answer to all the above. And more.

In the simplest terms, you might say that the MicroComposer is a sofisticated digital sequencer; it is controlled by a microprocessor (the 8080A). To say digital sequencer, though, is like trying to compare an abacus to a programmable pocket calculator.

The MicroComposer is designed to operate in conjunction with a professional studio type synthesizer such as the ROLAND System 700. It is also compatible with the ROLAND System 100 synthesizer. And, of course, it is compatible with other makes of synthesizer, too.

The MicroComposer is not limited to the production of music in the studio. It can be used in live performance, too. It can be used in any situation where you need a set of preprogrammed, timed pulses, and/or preprogrammed, timed voltage levels, as for example, in stage lighting.

Once the MicroComposer is programmed to your satisfaction, it will produce a perfect recording the first time through without mistake. Programming is as simple as adding a column of figures with a calculator. Or, if you can tap out a melody with one finger, no matter how slowly, you can use the synthesizer keyboard controller to load pitch information. If you prefer, you can load gate information also, so that the MicroComposer will play back any sequence exactly as you play it on the keyboard controller. Editing and correcting mistakes can be done simply at any time during the loading process.

And more . . . . . .

Using the eight channel output feature, all the voice lines in an average popular song can be stored in the memory; when the START button is pushed, the finished composition is produced in its entirety and can be recorded in one operation on the final master tape without the need for a multichannel tape recorder. In a multichannel recording system, using a one or two channel output from the MicroComposer will give enough memory capacity for recording music such as major symphonic works. Using the multichannel output of the MicroComposer, you can increase the number of first generation voices it is possible to record on the multichannel master tape.

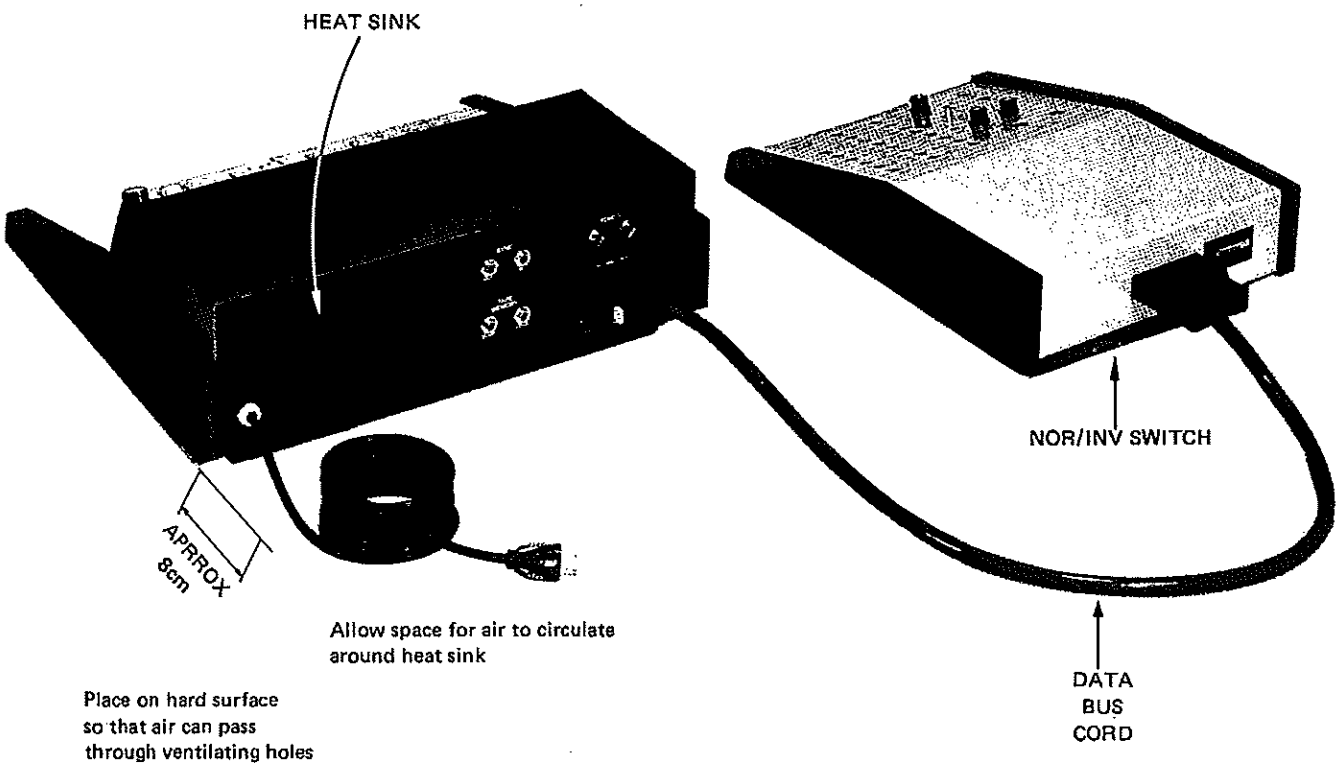
The digital information in the memories can be loaded intact directly onto tape for permanent storage by using a cassette tape recorder, and reloaded back into the MicroComposer memories for later use or editing.

The TIMER displays the elapsed time for the music. In commercial work such as for radio and television where timing is essential, setting the TEMPO control while checking the TIMER display allows you to adjust the playing time of the composition to within one tenth of a second of the desired time length.

And more. Read on . . . . . .

3. Connections

The MicroComposer and the Interface are connected as shown below.



Use of the jacks on the rear panel is explained on the following pages:

CAUTION:
Always turn off POWER switch before connecting or disconnecting data bus cord.

Be sure to set the NOR/INV switch correctly.

NOR (normal)	plus gate
INV (invert)	inverted (minus) gate

The Interface contains the D/A (digital-to-analog) converters which convert the digital output of the MicroComposer to control voltages that the synthesizer can use.

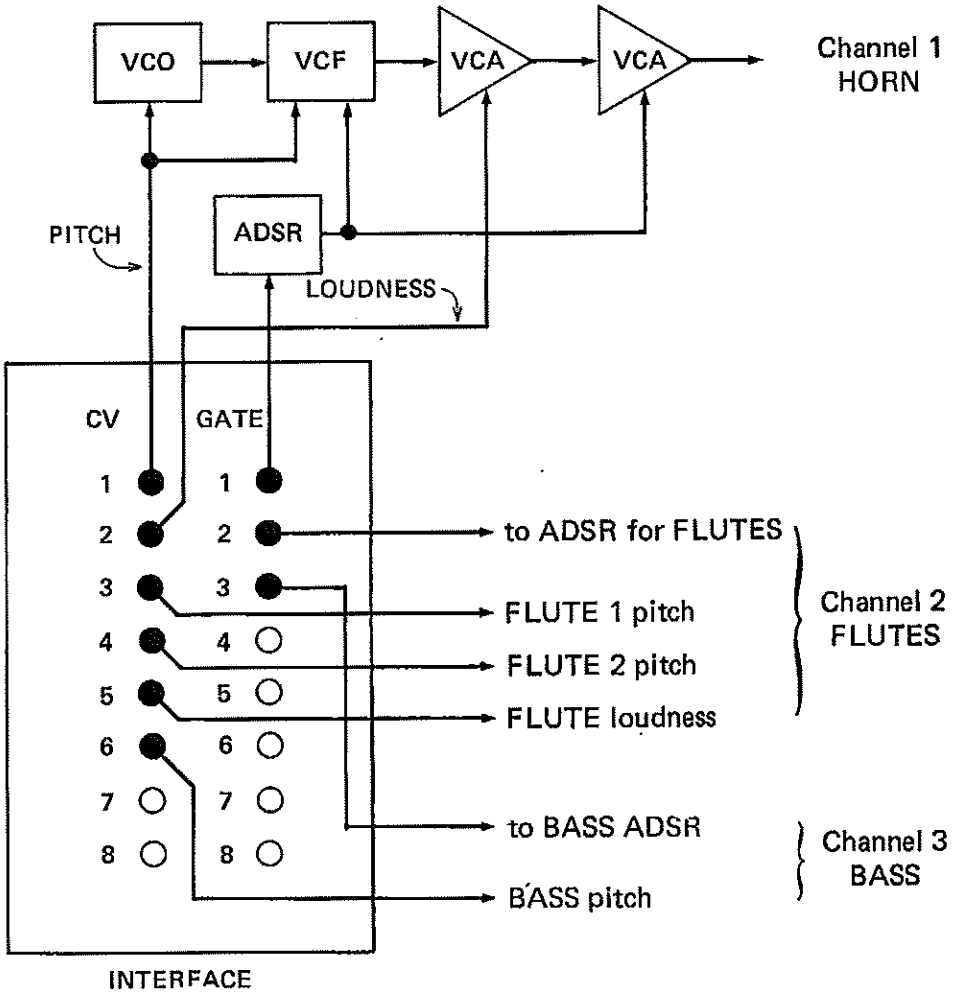
Outputs

The MicroComposer provides eight control voltage outputs and eight gate outputs, as well as a six bit multiplex output with a special seventh bit set aside for portamento control (multiplex is explained fully in Section 11).

The gate outputs are completely independent of each other and are referred to as channels. Each of the CV outputs may be assigned to any desired channel. Generally, the MicroComposer Program Sheets will clearly designate channel assignments for the CV outputs. An example is shown below:

CHANNEL		1. HORN						2. FLUTES						3. BASS					
MEASURE	STEP	VCO CV1	S	G	VCA CV2			VCO CV3	VCO CV4	S	G	VCA CV5		VCO CV6	S	G			
1	1																		
	2																		

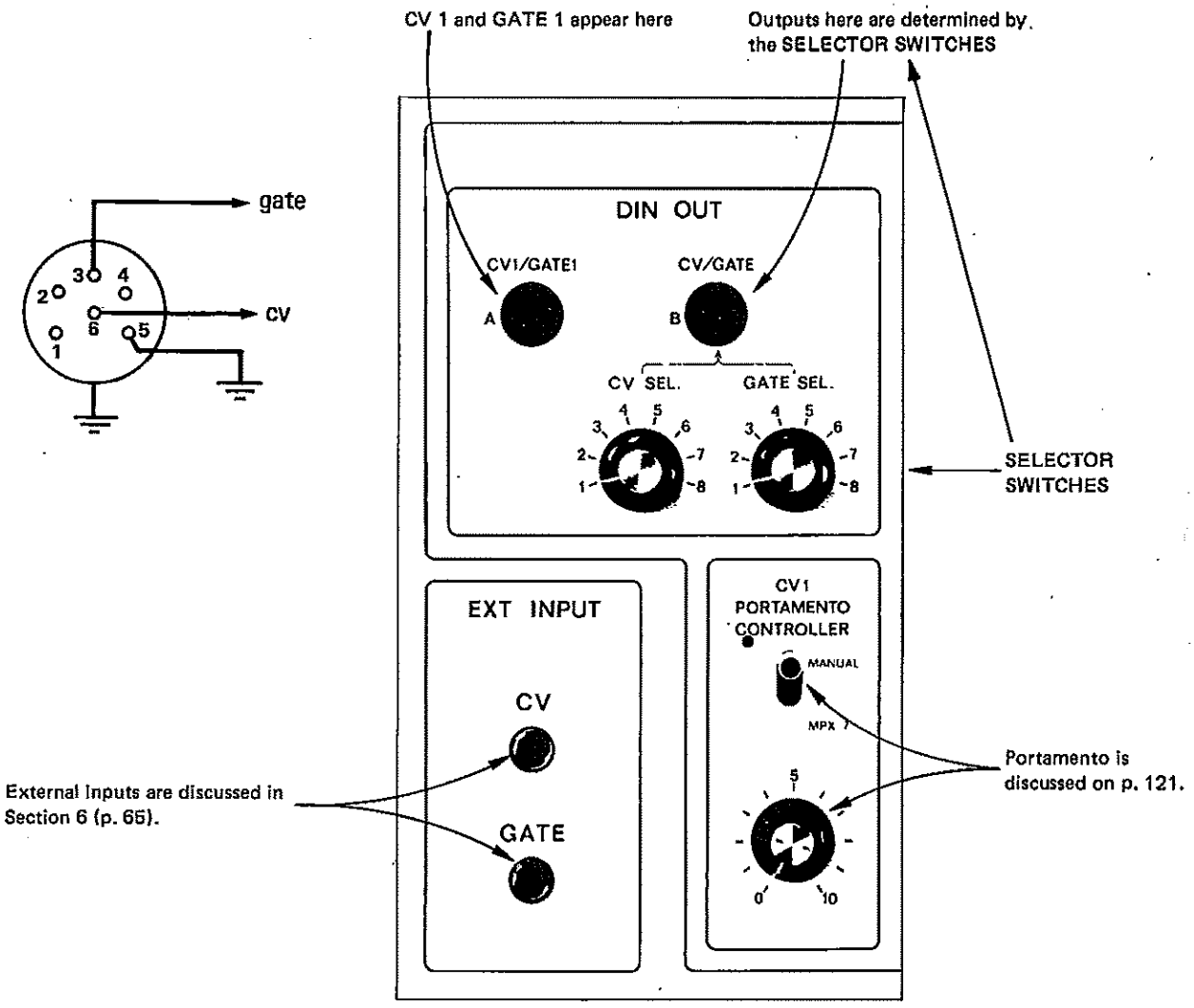
The Connections:



Each channel must contain one "S" (STEP TIME for note time valves) column as shown. In this example, the synthesizer is to produce four voices divided into three channels. Channel 2 contains two flutes of different pitches but controlled by the same gate pulse train. Note that two of the CV outputs are to be connected to VCA's for control of loudness. Control of the loudness of Channel 3 (in this example) is either not necessary, or it is to be controlled by one of the other CV outputs.

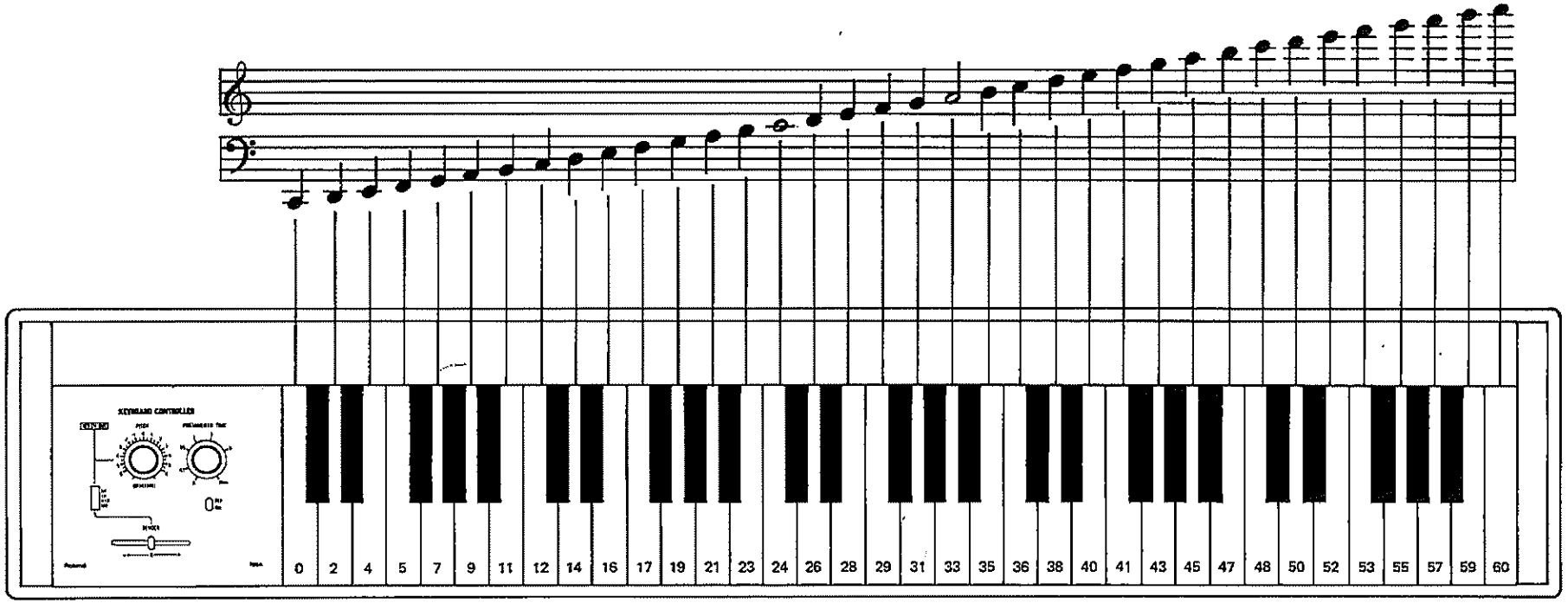
The DIN plugs

The DIN plugs are provided for the convenience of owners of ROLAND synthesizers equipped with DIN plugs for CV/GATE output. The drawing below shows the pin assignment if you would like to use the DIN plugs with other synthesizers.



Pitch

The pitches shown below are valid when the ROLAND System 700 VCO's are set at 8'.



NOTE: Use VCO tuning controls for pitches above and below the range shown.

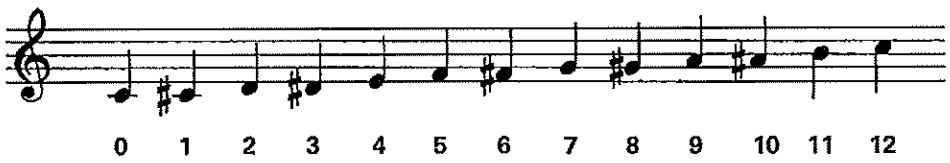
Section 15 discusses the production of other scale systems (such as quarter tones, etc.)

4. Writing the Program

The music must be translated into a language (numbers) that the MicroComposer can understand. These numbers are then written onto the program sheets, then loaded into the MicroComposer memories.

Pitch

If we let middle C equal "0", then:

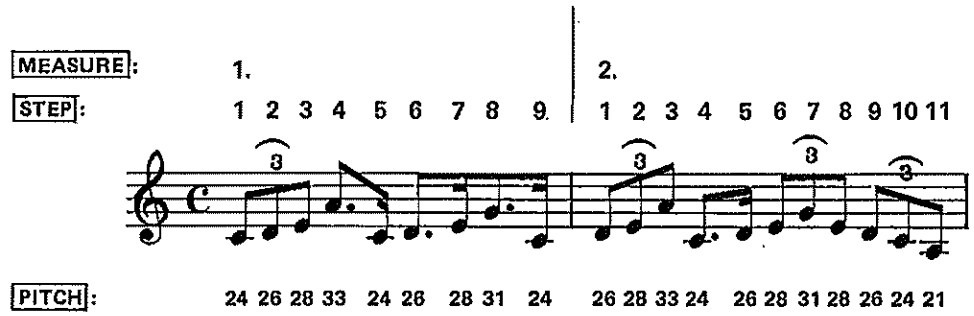


"0" can be made to equal any pitch, but there are distinct advantages to setting up and using only one standard. In this manual we use a standard based on a five octave keyboard controller as shown on the opposite page. With constant use of the same standard, you will begin to memorize the numbers and can soon do away with having to constantly refer to a keyboard diagram. Also, with the same standard, all VCO's may be tuned to unison except when pitches below "0" are required, or when programming transposing parts.

Each of the eight CV memories will accept values from 0 to 127 for a total range of over ten octaves (from 0 to slightly over +10 volts). (This wide range is not necessary for efficient pitch control but will prove useful for control of other synthesizer functions).

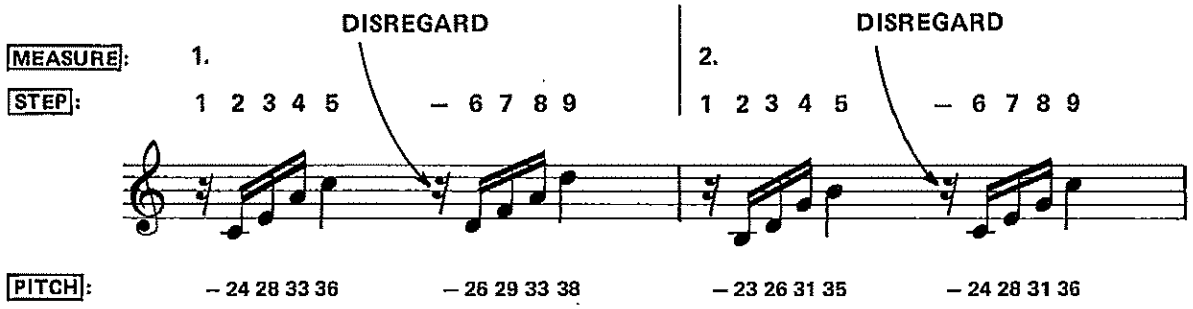
Generally, when encoding music, each note in each measure is numbered consecutively:

PROGRAM 1:



A rest or rests occuring between the bar line and the first note in a measure should be considered as one step; all other rests may be disregarded:

PROGRAM 2:



PROGRAM 1:

MM ♩ = 150 TIME BASE ♩ = 12

CHANNEL		1.				
MEASURE	STEP	VCO CV1	S	G		
1	1	24	4	2		
	2	26	4	2		
	3	28	4	2		
	4	33	9	7		
	5	24	3	2		
	6	26	9	7		
	7	28	3	2		
	8	31	9	7		
	9	24	3	2		
2	1	26	4	2		
	2	28	4	2		
	3	33	4	2		
	4	24	9	7		
	5	26	3	2		
	6	28	4	|		
	7	31	|	|		
	8	28	|	|		
	9	26	|	|		
	10	24	|	|		
	11	21	4	2		

PROGRAM 2:

MM ♩ = 208 TIME BASE ♩ = 32

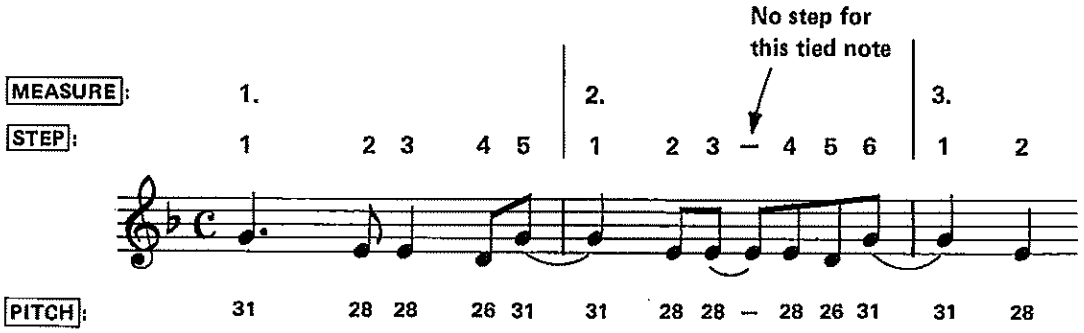
CHANNEL		1.				
MEASURE	STEP	VCO CV1	S	G		
1	1	(24)	8	0		
	2	24	|	6		
	3	28	|	6		
	4	33	8	6		
	5	36	40	30		
	6	26	8	6		
	7	29	8	6		
	8	33	8	6		
	9	38	32	30		
2	1	(38)	8	0		
	2	23	|	6		
	3	26	|	6		
	4	31	8	6		
	5	35	40	30		
	6	24	8	6		
	7	28	8	6		
	8	31	8	6		
	9	36	32	30		

Pitch of first note which will sound.

Data from previous step is repeated here (see p. 21).

Tied notes may be considered as one. Ties across bar lines should be considered as separate notes.

PROGRAM 3:



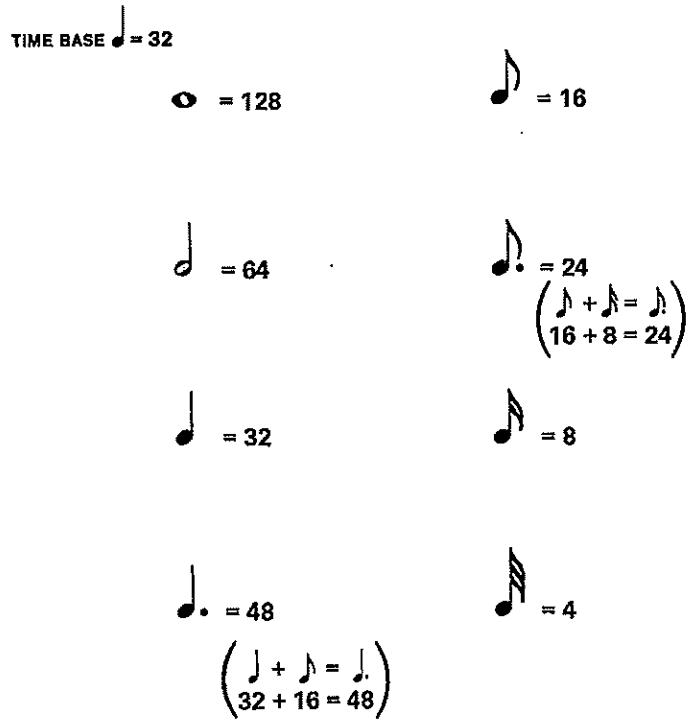
MM ♩ = 126 TIME BASE ♩ = 32

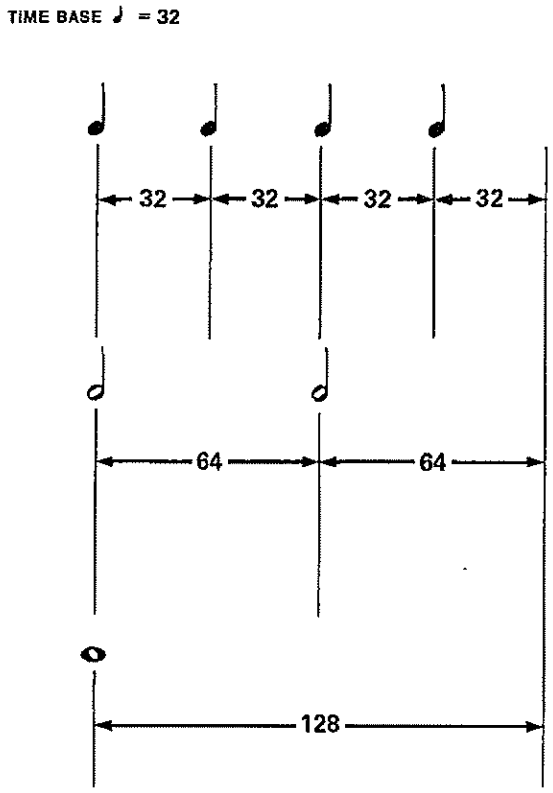
CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1	31	48	42				
	2	28	16	14				
	3	28	32	28				
	4	26	16	14				
	5	31	16	16				
2	1	31	32	30				
	2	28	16	14				
	3	28	32	30				
	4	28	16	14				
	5	26	16	14				
	6	31	16	16				
3	1	31	32	30				
	2	28	32	30				

Time Values

The STEP TIME and GATE TIME memories are used for storing note time value data and for programming rests, and together they produce the gate pulse train for control of the synthesizer envelope generator.

STEP TIME determines the total time duration for each step in the program. The STEP TIME memory has a capacity of 256 increments for each step in the program which are numbered from 1 to 256. If we let a quarter note equal 32 increments, then:

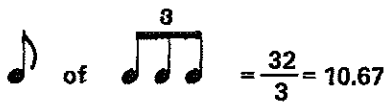




The actual length of time required for each increment will depend on the programmed tempo (see Section 14). In the above example, since a whole note equals 128, the smallest time increment we could program (a "1") would be equal to a 1/128 note. Also, the longest time value we could program into one step would be equal to two whole notes (128 + 128 = 256). For longer duration (such as tied notes) more than one step would have to be used as explained later in this section.

Compound rhythms can be handled in one of two different ways. If, as above, TIME BASE ♩ = 32, then:

TIME BASE ♩ = 32



Since the memories are not designed to handle fractions, two of the notes could be assigned the value of 11 and the remaining note 10 for a total of 32.

TIME BASE ♩ = 32

10 + 11 + 11 = 32

Or all of the notes could be assigned the value of 11 for a total of 33 for that beat.

TIME BASE ♩ = 32

11 + 11 + 11 = 33

The continual addition of an extra increment for a series of triplets will soon throw the count off. To be safe, the extra increments should be "stolen" from a nearby note (even if that note is not in the same measure):

TIME BASE ♩ = 32

For more precise division, the increment value assigned to a quarter note (TIME BASE) would have to be divisible by 3.

TIME BASE ♩ = 36

12 + 12 + 12 = 36

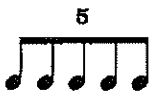
For exact precision in music with both triplets and ordinary eighth notes, the TIME BASE would have to be divisible by both 2 and 3.

If TIME BASE ♩ = 24, then  = 8 + 8 + 8 and  = 12 + 12

or:

If TIME BASE ♩ = 36, then  = 12 + 12 + 12 and  = 18 + 18

The same process works for other compounds:

If TIME BASE ♩ = 20, then  = 8 + 8 + 8 + 8 + 8 = 40

and:

Gate Times and Rests

The information in the GATE TIME memory determines the gate time (equivalent to the length of time the key is depressed) for each step. Like STEP TIME, the GATE TIME memory has 256 increments for each step, but these are numbered from 0 to 255. The inclusion of the 0 allows programming of rests (no gate pulse), but makes the total number of increments one less than STEP TIME.

STEP TIME: 1 to 256

GATE TIME: 0 to 255

Loading a 0 in GATE TIME for a given step will produce no gate pulse for that step. The result will be a rest or group of rests whose total time value equals the value stored in the STEP TIME memory for that step.

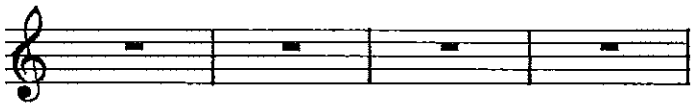
If TIME BASE ♩ = 32, then:

						
STEP TIME:	128	64	32	16	8	
GATE TIME:	0	0	0	0	0	etc.

PROGRAM 4:

TIME BASE ♩ = 32

MEASURE:	1.	2.	3.	4.
STEP:	1	1	1	1



STEP TIME:	128	128	128	128
GATE TIME:	0	0	0	0

MM ♩ = 100 TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1	0	128	0				
2	1	↓	↓	↓				
3	1	↓	↓	↓				
4	1	0	128	0				
		↑						

STEP TIME
GATE TIME

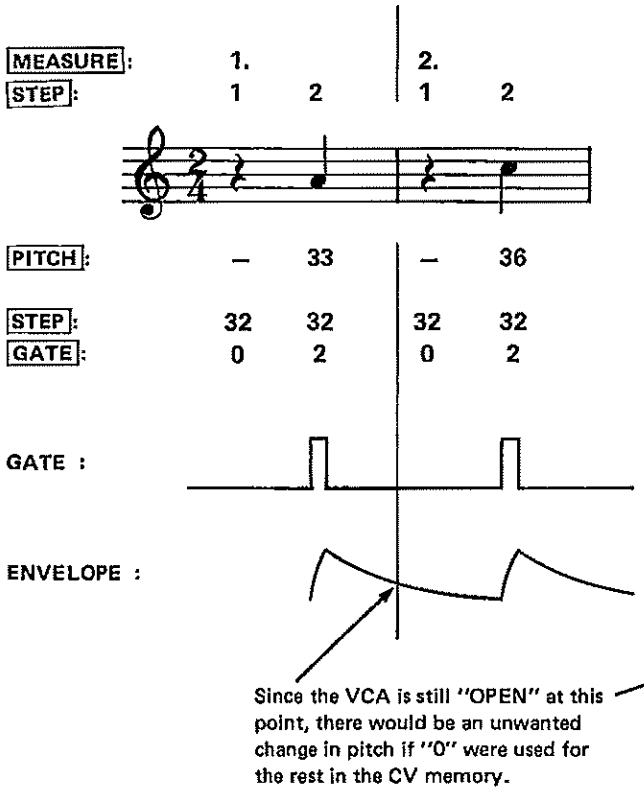
Any number may be entered here, but some number must be entered.

All steps in the program must contain CV data, including rests. Since rests usually produce no sound, any value may be used. The following, however, may prove to be good rules to follow.

If the first step in the program is a rest, use the CV value for the first pitch which will sound. For all others, use the same data as was used for the previous step. This latter practice will prevent an unwanted pitch change due to long envelope release times which extend into the following step.

PROGRAM 5:

TIME BASE ♩ = 32



Using "33" here will save a little time when it comes to loading the program. If the program is to be run in the CYCLE (repeating) mode, use the same value here as was used in the last step in the program.

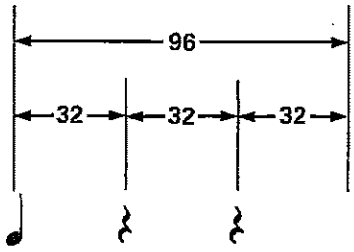
MM ♩ = 216 TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	CV1	S	G				
1	1	33	32	0				
	2	33		2				
2	1	33		0				
	2	36	32	2				

Since the VCA is still "OPEN" at this point, there would be an unwanted change in pitch if "0" were used for the rest in the CV memory.

Normally, rests are programmed with the notes they follow:

TIME BASE ♩ = 32



TIME VALUE: 32 + 32 + 32 = 96

STEP TIME: 96
GATE TIME: 32

PROGRAM 6:

TIME BASE ♩ = 32

MEASURE:	1.					2.				3.				4.	
STEP:	1				2	1	2	3	4	1		2		1	2
TIME VALUE:	32	32	32	16	16	32	32	32	32	32	32	32	32	32	64
STEP TIME:	112				16	32	32	32	32	64		64		32	64
GATE TIME:	32				14	30	30	30	30	32		32		0	64

32 + 32 + 32 + 16 = 112

These values will depend on ADSR settings and the particular phrasing desired.

32 + 32 = 64

MM ♩ = 250

TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1		112	32				
	2		16	14				
2	1		32	30				
	2		|	|				
	3		|	|				
	4		32	30				
3	1		64	32				
	2		64	32				
4	1		32	0				
	2		64	64				

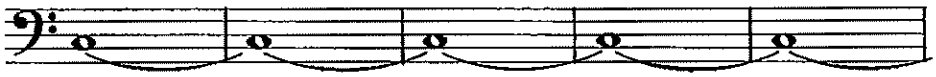
If you try this program remember that some pitch values must be loaded here.

Loading GATE TIME with the same value (or more) as STEP TIME for a given step will result in a gate pulse which is continuous with the gate pulse in the next step. This can be used for notes longer than the 255 increments available or for legatissimo passages.

PROGRAM 7:

TIME BASE ♩ = 32

MEASURE:	1.		2.		3.		4.		5.
STEP:	1		1		1		1		1
STEP TIME:	128		128		128		128		128
GATE TIME:	128		128		128		128		128



MM ♩ = 200 TIME BASE ♩ = 32

CHANNEL		1						
MEASURE	STEP	VCO CV1	S	G				
1	1	12	128	128				
2	1	↓	↓	↓				
3	1							
4	1							
5	1							

PROGRAM 8:

TIME BASE ♩ = 32

MEASURE:	1.			2.		
STEP:	1	2	3	1	2	
PITCH	36	33	34	36	29	
STEP TIME:	64	48	16	64	64	
GATE TIME:	64	48	16	64	(60)	

Being the end of the phrase, this value should be shorter than 64 so the phrase is separated from the next one.

MM ♩ = 200 TIME BASE ♩ = 32

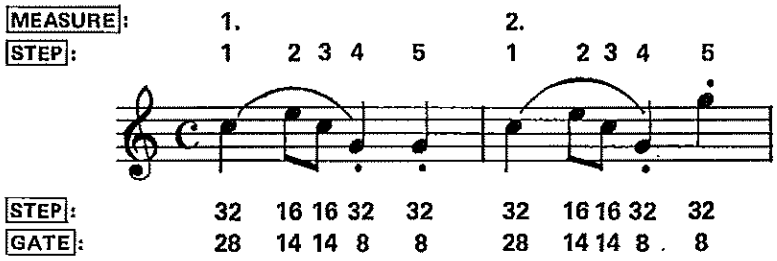
CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1	36	64	(64)				
	2	33	48	|				
	3	34	16	|				
2	1	36	64	(64)				
	2	29	64	60				

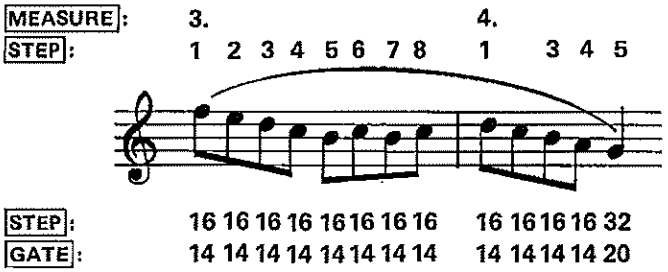
In this case it is much simpler to load 64 in these steps of the GATE TIME memory rather than going to the trouble of writing in the different values. The result is the same: one gate pulse for the whole passage.

Other values loaded into the GATE TIME memory will usually depend on personal interpretation of phrasing and articulation and can be determined by experimentation.

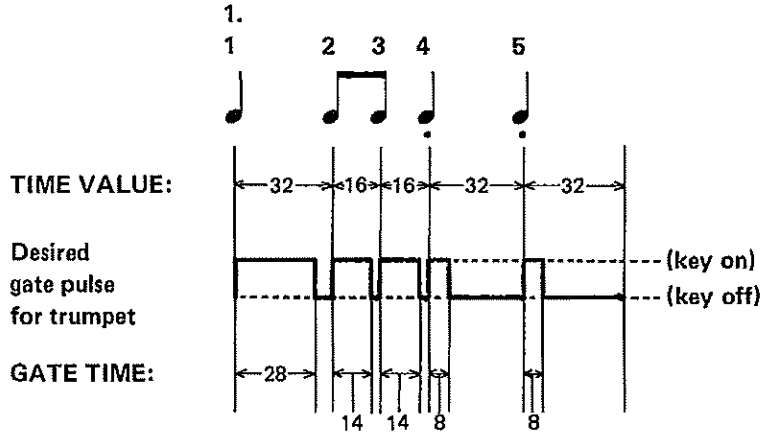
PROGRAM 9:

TIME BASE ♩ = 32





Timing for measure 1:



MM ♩ = 250 TIME BASE ♩ = 32

CHANNEL		1. TRUMPET					
MEASURE	STEP	VCO CV1	S	G			
1	1	36	32	28			
	2	40	16	14			
	3	36	16	14			
	4	31	32	8			
	5	31	32	8			
2	1	36	32	28			
	2	40	16	14			
	3	36	16	14			
	4	31	32	8			
	5	43	32	8			
3	1	41	16	14			
	2	40	↓	↓			
	3	38					
	4	36					
	5	35					
	6	36					
	7	35					
	8	36					
4	1	38					
	2	36					
	3	35	↓	↓			
	4	33	16	14			
	5	31	32	20			

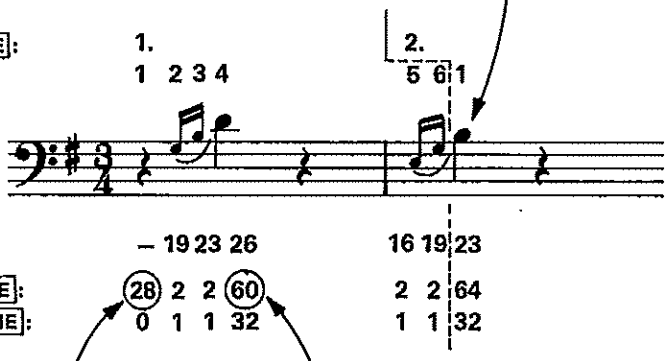
Embelishments

Grace notes and other embelishments must be treated as notes with real time, and the values must be subtracted from the value of the previous note or rest.

PROGRAM 10:

TIME BASE ♩ = 32

Since the B falls on the first beat of the measure, the grace notes should be programmed as being in the previous measure.



𝄽 = 32

If we assign the value of 2 to each grace note, then:

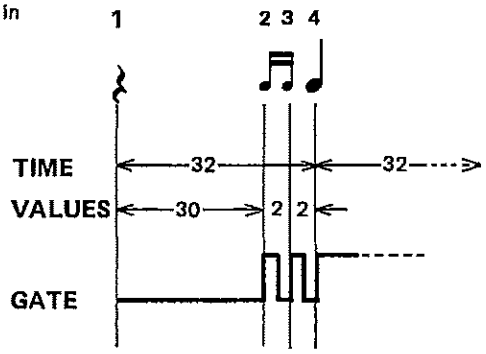
♬ = 2 + 2 = 4

32 – 4 = (28)

♩ + 𝄽 = 32 + 32 = 64

♬ = 2 + 2 = 4; 64 – 4 = (60)

Timing for measure 1:



Measure 1 programmed as:

16 + 8 + 4 = 28 2 2 28

MM ♩ = 82 TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1	19	28	0				
	2	19	2	1				
	3	23	2	1				
	4	26	60	32				
	5	16	2	1				
	6	19	2	1				
2	1	23	64	32				

Program 11 shows a sample program for a trill.

PROGRAM 11:

1.

Programmed as approximately:

MM ♩ = 96 TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	VCO CV1	S	G				
1	1	43	32	32				
	2	41	9	9				
	3	43	7	7				
	4	41	4	4				
	5	43	3	3				
	6	41	|	|				
	7	43	|	|				
	8	41	3	3				
	9	40	32	28				

Since the trill is to last the duration of a quarter note (32), these values must total 32.

Dynamics

With a capacity of 128 increments (numbered 0 to 127), the output voltage range of each CV memory is from 0 to slightly over +10 volts.

CV memory data		CV output
0	=	0.00 volts
12	=	+1.00
24	=	+2.00
36	=	+3.00
48	=	+4.00
60	=	+5.00
72	=	+6.00
84	=	+7.00
96	=	+8.00
108	=	+9.00
120	=	+10.00
127	=	+10.58

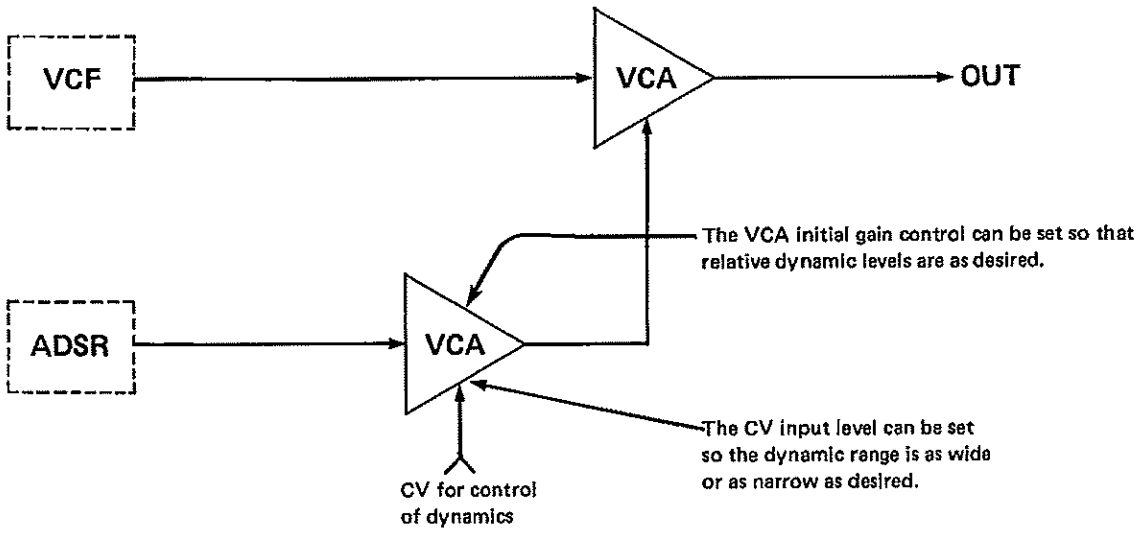
When used to control a VCA for dynamics, the following values may be used:

Loudness	For exponetial VCA	For linear VCA
fff	100	127
ff	90	64
f	80	32
mf	70	16
mp	60	8
p	50	4
pp	40	2
ppp	30	1

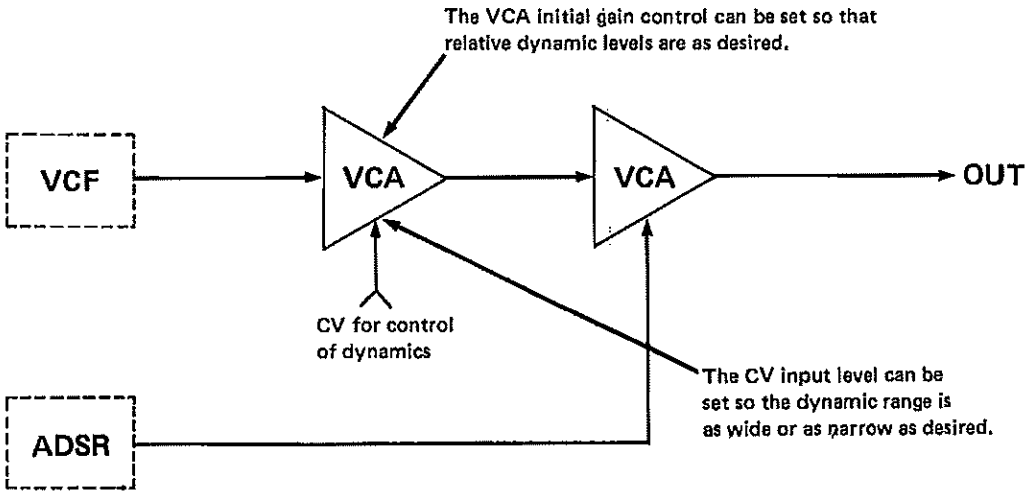
← The correct value here is 128 which is not available; 127 is close enough.

Use numbers between those shown for more delicate shading of dynamics.

The VCA used for the control of dynamics can be patched at the output of the envelope generator so as to control the level of the envelope, as shown below:



The drawing below shows an alternate method of controlling dynamics (which can be used if your VCA's can't pass dc voltages), but the method above will produce a better S/N in the output sound, especially during long rests.



Since VCA controls can be used to control the relative levels and dynamic range, the values loaded into the CV memory are not critical.

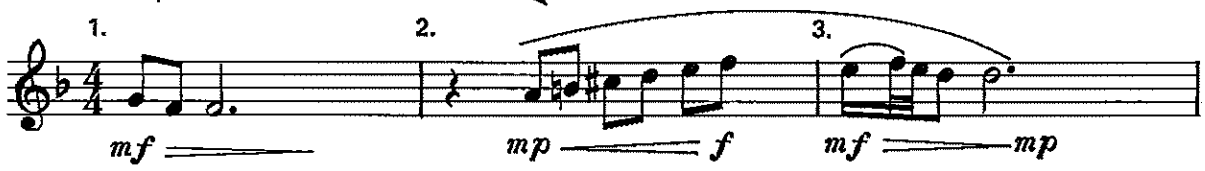
PROGRAM 12.

TIME BASE ♩ = 32

Yesterday

Note the differences in GATE TIME for these groups of sixteenth notes due to the phrasing.

Beatles



MM ♩ = 132 TIME BASE ♩ = 32

CHANNEL		1. FLUTE							
		VCO			VCA				
MEASURE	STEP	CV1	S	G	CV2				
1	1	31	16	8	70				
	2	29	16	8	60				
	3	29	96	88	50				
2	1	29	32	0	50				
	2	33	16	14	40				
	3	35	|	|	30				
	4	37	|	|	60				
	5	38	|	|	70				
	6	40	|	|	75				
	7	41	16	14	80				
3	1	40	8	8	70				
	2	41	4	4	65				
	3	40	4	4	60				
	4	38	16	12	60				
	5	38	96	88	55				

← Use data from previous step in the same way as outlined in Program 5, p. 21.

Dynamics (for exponential VCA)

PROGRAM 13

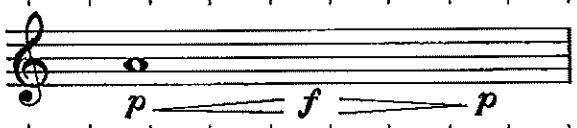
SWELL

MM ♩ = 120 TIME BASE ♩ = 32

CV1

CHANNEL					
MEASURE	STEP	VCO CV2	S	G	VCA CV1
1	1	33	8	8	50
	2				53
	3				57
	4				60
	5				65
	6				70
	7				75
	8				80
	9				72
	10				70
	11				67
	12				63
	13				60
	14				57
	15			8	53
	16	33	8	6	50

Set Interface PORTAMENTO control at about "4" (PORTAMENTO switch at MANUAL).



Tone Color

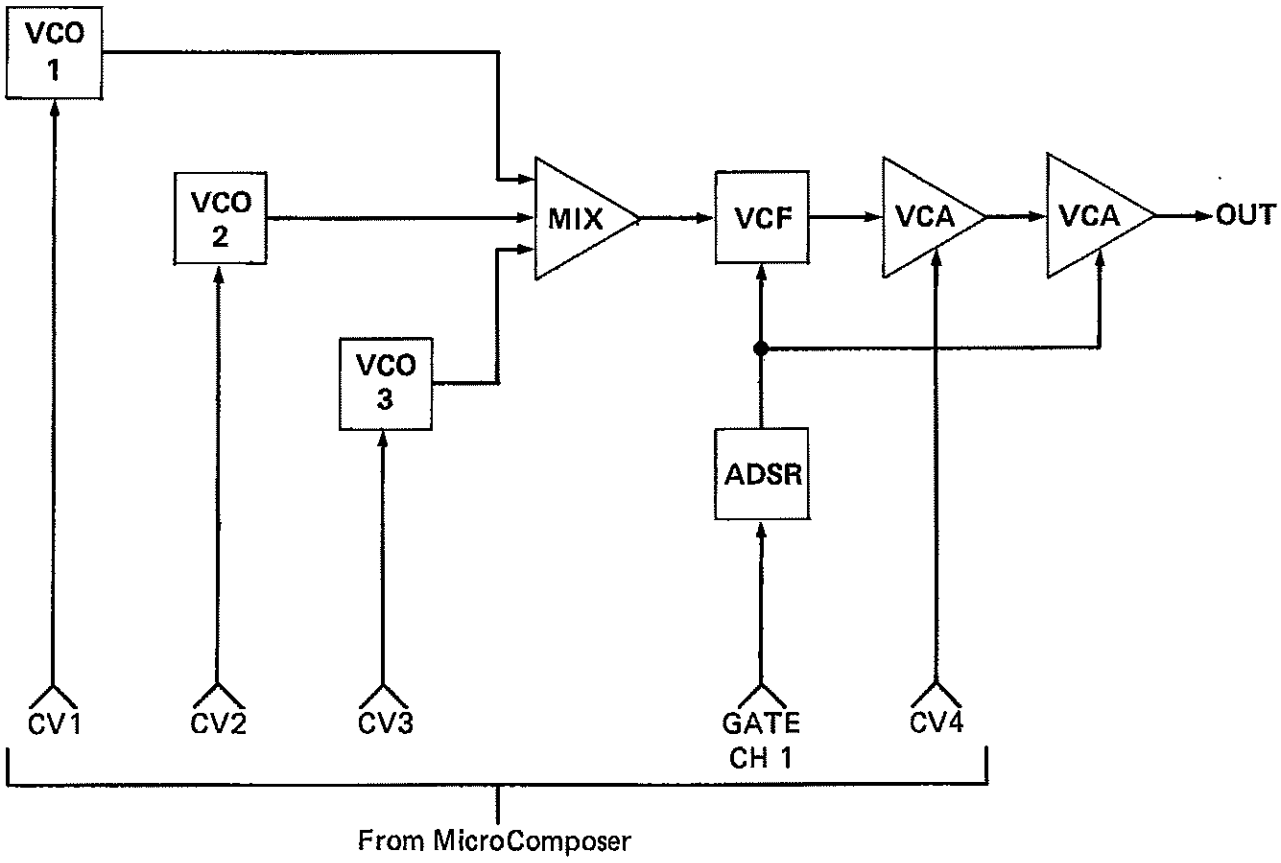
The CV outputs may, of course, be used to control any device or function which will accept external control voltage inputs.

It should be remembered that the tone coloring of different instruments may change radically with the dynamic levels and, therefore, using a CV output to control the cutoff point of a VCF can be very useful, even in the most conventional of music styles. Owners of the ROLAND System 700 Synthesizer which are equipped with the 703E, F, or G VCF also have the option of voltage controlled resonance.

Chords

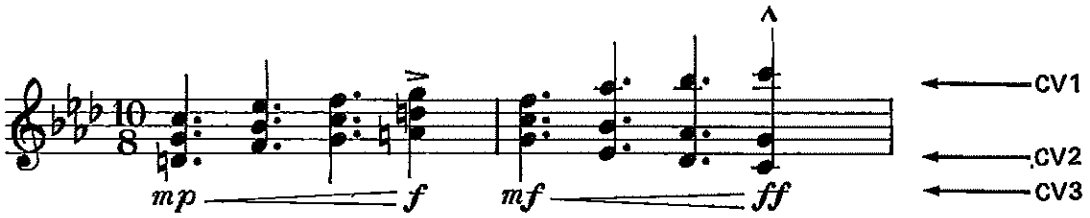
Chords may be produced simply by assigning more CV memories to the related channel. The patch for Program 14 (next page) is shown below.

PATCH for PROGRAM 14:



PROGRAM 14:

TIME BASE ♩ = 16



Note that a different time base was used

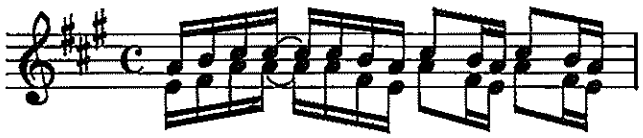
MM ♩ = 124 TIME BASE ♩ = 16

CHANNEL		1. HORNS								
MEASURE	STEP	VCO CV1	VCO CV2	VCO CV3	S	G	VCA CV4			
1	1	36	31	26	24	20	60			
	2	39	34	29	24	20	65			
	3	41	36	31	24	20	72			
	4	43	38	33	16	8	80			
2	1	41	36	31	24	20	70			
	2	44	34	27	24	20	78			
	3	46	32	25	24	20	85			
	4	48	31	24	16	12	100			

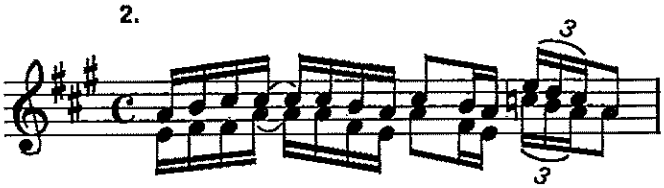
Upper voice
Middle voice
Lower voice
Dynamics

PROGRAM 15:

1.



2.



MM ♩ = 76 TIME BASE ♩ = 24

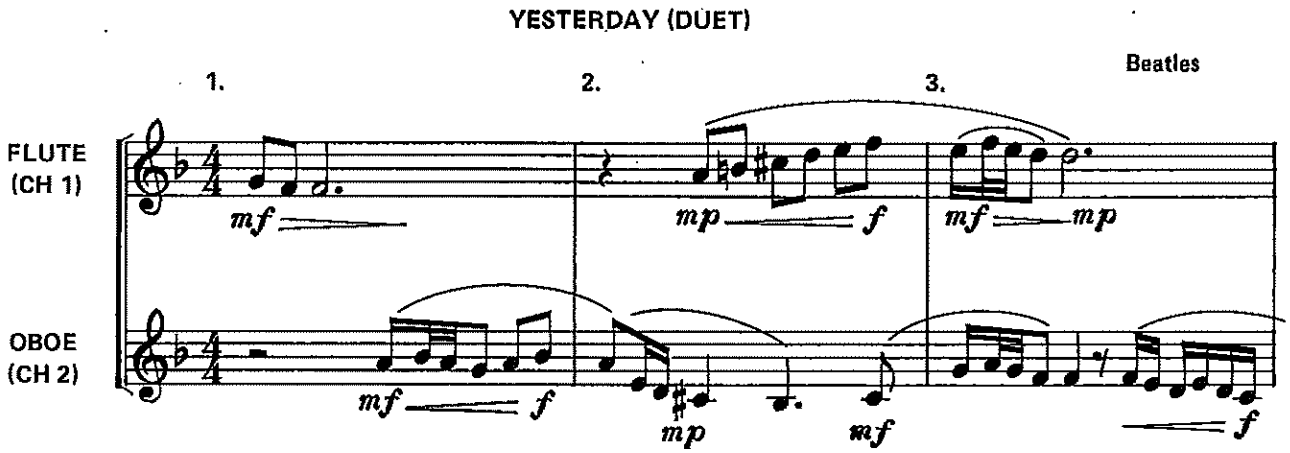
CHANNEL		1. GUITAR						
MEASURE	STEP	1	2	S	G			
1	1	28	33	6	2			
	2	30	35	6				
	3	33	37	6				
	4	33	37	12				
	5	33	37	6				
	6	30	35	6				
	7	28	33	6				
	8	33	37	12				
	9	30	35	6				
	10	28	33	6				
	11	33	37	12				
	12	30	35	6				
	13	28	33	6				
2	1	28	33	6				
	2	30	35	6				
	3	30	37	6				
	4	33	37	12				
	5	33	37	6				
	6	30	35	6				
	7	28	33	6				
	8	33	37	12				
	9	30	35	6				
	10	28	33	6				
	11	36	40	4				
	12	35	38	4				
	13	33	36	4				
	14	33	33	12	2			
	Repeat measures 1 & 2 as desired							

Multichannel Programs

The MicroComposer has an eight channel gate pulse output. In other words, it produces eight completely independent gate pulse trains at the same time.

Program 16 shows the addition of a second voice to Program 12.

PROGRAM 16:



PROGRAM 16

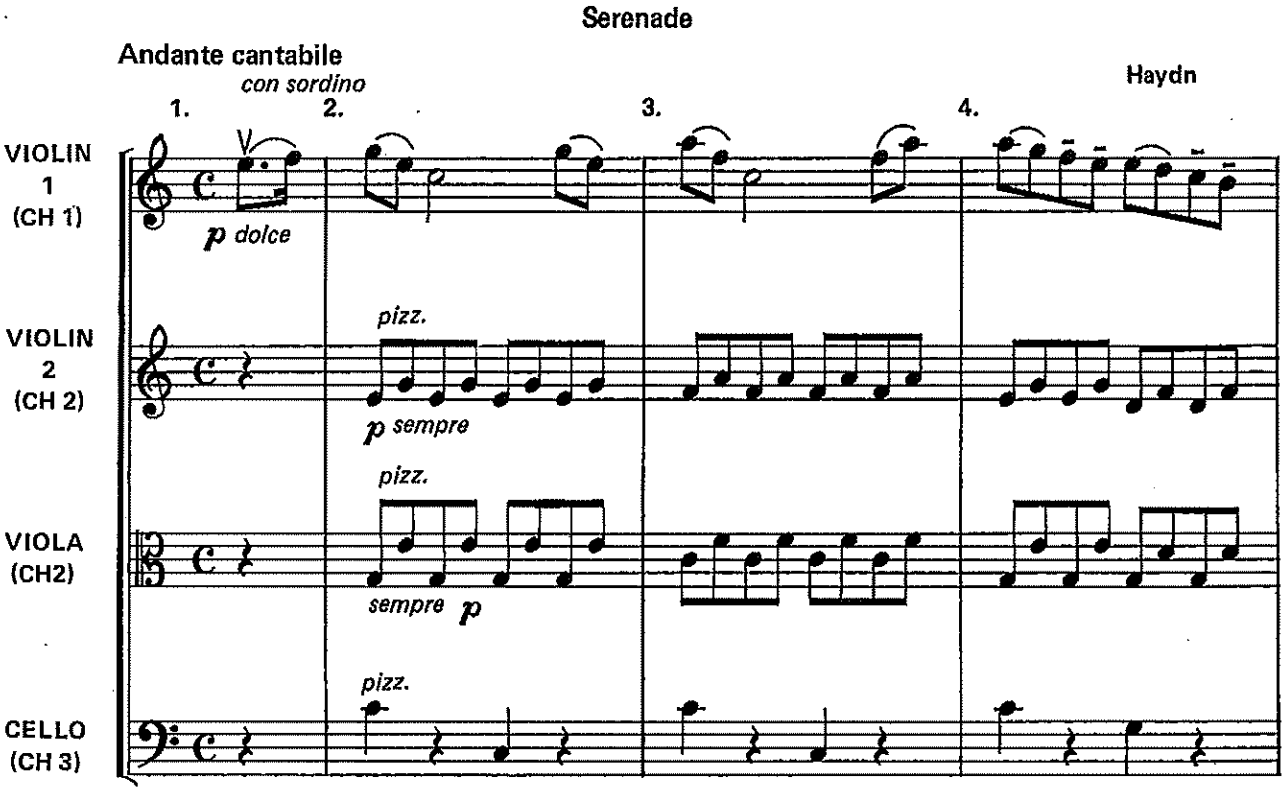
YESTERDAY (DUET)

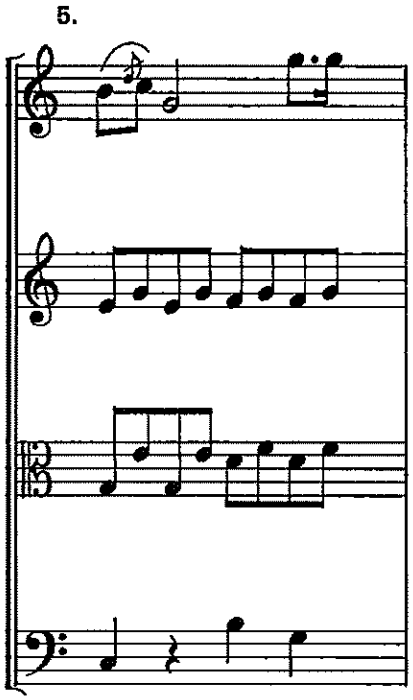
MM ♩ = 132 TIME BASE ♩ = 32

CHANNEL		1. FLUTE						2. OBOE					
MEASURE	STEP	VCO CV1	S	G	VCA CV2			VCO CV3	S	G	VCA CV4		
1	1	31	16	8	70			33	64	0	0		
	2	29	16	8	60			33	8	8	50		
	3	29	96	88	50			34	4	4	50		
	4							33	4	2	55		
	5							31	16	14	60		
	6							33	|	14	65		
	7							34	|	14	70		
2	1	29	32	0	50			33	16	12	75		
	2	33	16	14	40			28	8	4	50		
	3	35	|	|	30			26	8	2	|		
	4	37	|	|	60			25	32	30	|		
	5	38	|	|	70			22	48	40	50		
	6	40	|	|	75			25	16	14	65		
	7	41	16	14	80								
3	1	40	8	8	70			31	8	8	70		
	2	41	4	4	65			33	4	4	65		
	3	40	4	4	60			31	4	2	65		
	4	38	16	12	60			29	16	12	65		
	5	38	96	88	55			29	48	40	70		
	6							29	8	6	60		
	7							28	|	|	60		
	8							26	|	|	65		
	9							28	|	|	65		
	10							26	|	|	70		
	11							24	8	6	75		

This portion of the program is exactly the same as Program 12.

PROGRAM 17:


Serenade
Andante cantabile
con sordino
Haydn
1.
2.
3.
4.
VIOLIN 1 (CH 1)
p dolce
VIOLIN 2 (CH 2)
pizz.
p sempre
VIOLA (CH2)
pizz.
sempre p
CELLO (CH 3)
pizz.


5.

PROGRAM 17

SERENADE

This data represents variable tempo and is explained in Section 14.

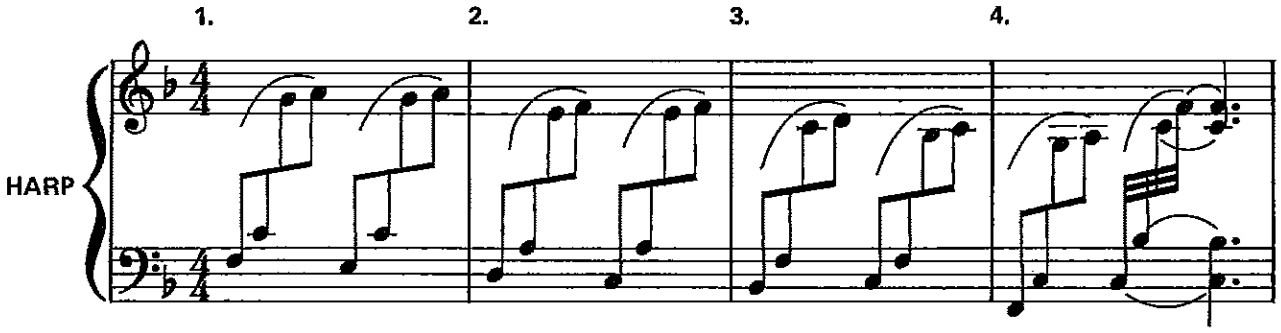
MM ♩ = 100 TIME BASE ♩ = 32

CHANNEL		1. VIOLIN 1							2. VIOLIN 2 / VIOLA PIZZ						3. CELLO PIZZ					4. TEMPO	
MEASURE	STEP	VCO 1	S	G	VCA 2	MPX			3	4	S	G			6	S	G	7		T	S
1	1	40	24	24	60				28	19	32	0			24	32	0	0		80	24
	2	41	8	6	55															90	8
2	1	43	16	16	58				28	19	16	2			24	64	2	60		100	128
	2	40	16	12	52				31	28					12	64	2	55			
	3	36	64	60	45				28	19											
	4	43	16	16	55				31	28											
	5	40	16	12	50				28	19											
	6								31	28											
	7								28	19											
	8								31	28	16	2									
3	1	45	16	16	60				29	24					24	64	2			100	96
	2	41	16	14	53				33	29					12	64	2			94	16
	3	36	64	60	45				29	24										90	16
	4	41	16	16	50				33	29											
	5	45	16	10	55				29	24											
	6								33	29											
	7								29	24											
	8								33	29											
4	1	45	16	16	58				28	19					24	64	2			100	128
	2	43		12	52				31	28					19	64	2				
	3	41		14	58				28	19											
	4	40		14	58				31	28											
	5	40		16	58				26	19											
	6	38		12	52				29	26											
	7	36		14	58				26	19											
	8	35	16	14	58				29	26											
5	1	35	12	12	55				28	19					12	64	2	55		100	96
	2	38	4	4	52				31	28					23	32	2	55		90	32
	3	36	16	14	48				28	19					19	32	2	50			
	4	31	64	60	45				31	28											
	5	43	24	20	60				29	26											
	6	43	8	6	60				31	29											
	7								29	26											
	8								31	29											

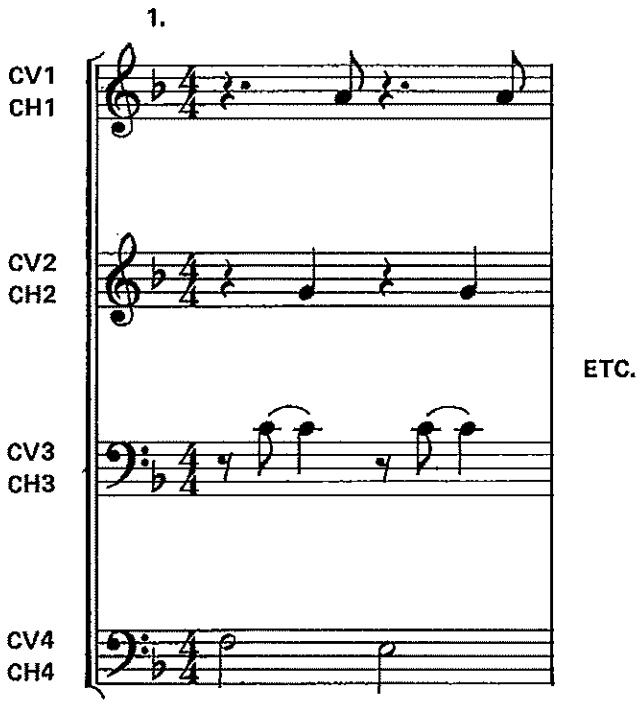
These indicate the use of the MicroComposer COPY function which is explained in detail in Section 9. If you try this program use COPY 3x instead of 31x since this sample only goes to measure 5.

SUSTAINED ARPEGGIOS

PROGRAM 18:



The program was written as if the music were written:



ETC.

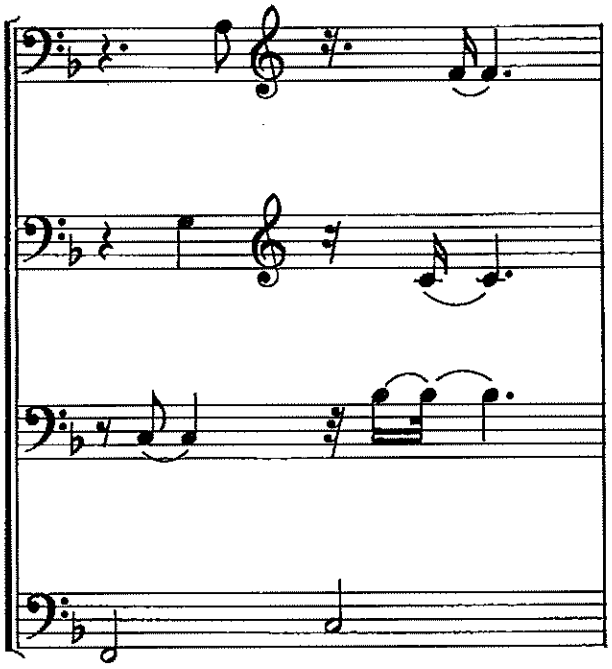
PROGRAM 18

SUSTAINED ARPEGGIOS

MM ♩ = 132 TIME BASE ♩ = 32 C = COPY function (Section 9)

CHANNEL		1.					2.					3.					4.				
MEASURE	STEP	VCO 1	S	G			VCO 2	S	G			VCO 3	S	G			VCO 4	S	G		
1	1	33	48	0			31	32	C G1			24	16	C G1			17	64	8		
	2	33	64	8			31	64	1X			24	64	1X			16	64	8		
	3	33	16	8			31	32	M1 M4			24	48	M1 M4							
2	1	33	C	C			31	C				24	C				14	64	8		
	2	29	2X	3X			28	2X				21	2X				12	64	8		
	3	29	M1 M1	M1 M1			28	M1 M1				21	M1 M1								
3	1	29					28					21					10	64	8		
	2	26					24					17					12	64	8		
	3	24					22					17									
4	1	24	48				22	32				17	16				5	64	8		
	2	21	28				19	40				12	52				12	64	8		
	3	29	52				24	56				22	60								

MEASURE 4:

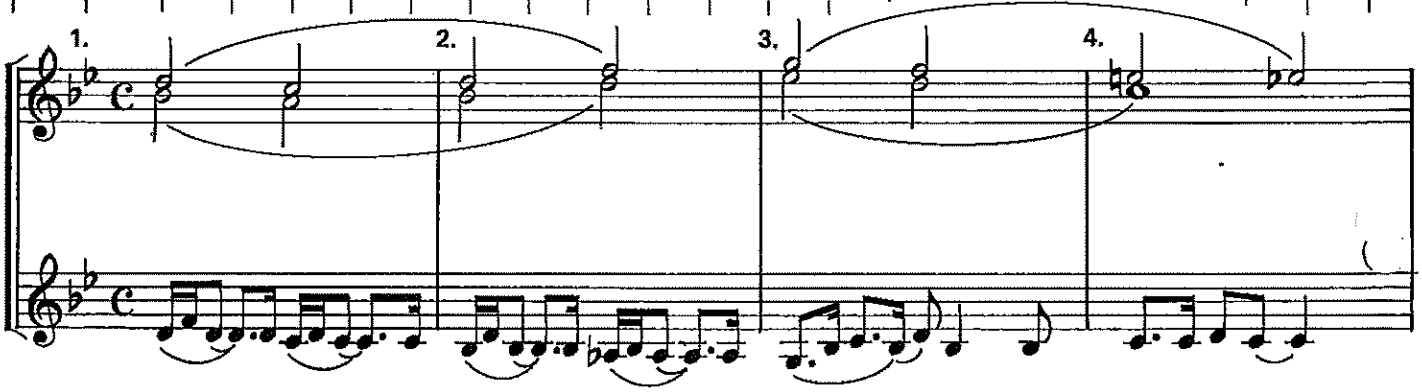


PROGRAM 19

LES CHAPS-ÉLYÉES

MM ♩ = 100 TIME BASE ♩ = 24

CHANNEL		1. FL				2. FL				3. CL			
MEASURE	STEP	1	S	G		2	S	G		3	S	G	
1	1	38	48	48		34	C	C		26	6	6	
	2	36	48	48		33	S1	G1		29	6	6	
	3						1X	1X		26	30	28	
	4						M1	M1		26	6	4	
	5						M3	M3		24	6	6	
	6									26	6	6	
	7									24	30	28	
	8									24	6	4	
2	1	38	C	48		34				22	C	C	
	2	41	3X	40		38				26	1X	1X	
	3		M1							22	M1	M1	
	4		M1							22	M1	M1	
	5									20			
	6									22			
	7									20			
	8									20			
3	1	43		C		39				19	18	18	
	2	41		1X		38				22	6	6	
	3			M1						24	18	18	
	4			M2						22	6	5	
	5									26	12	10	
	6									22	24	4	
	7									22	12	8	
4	1	40				36	96	90		24	18	16	
	2	39								24	6	4	
	3									26	12	10	
	4									24	48	36	



PROGRAM 20

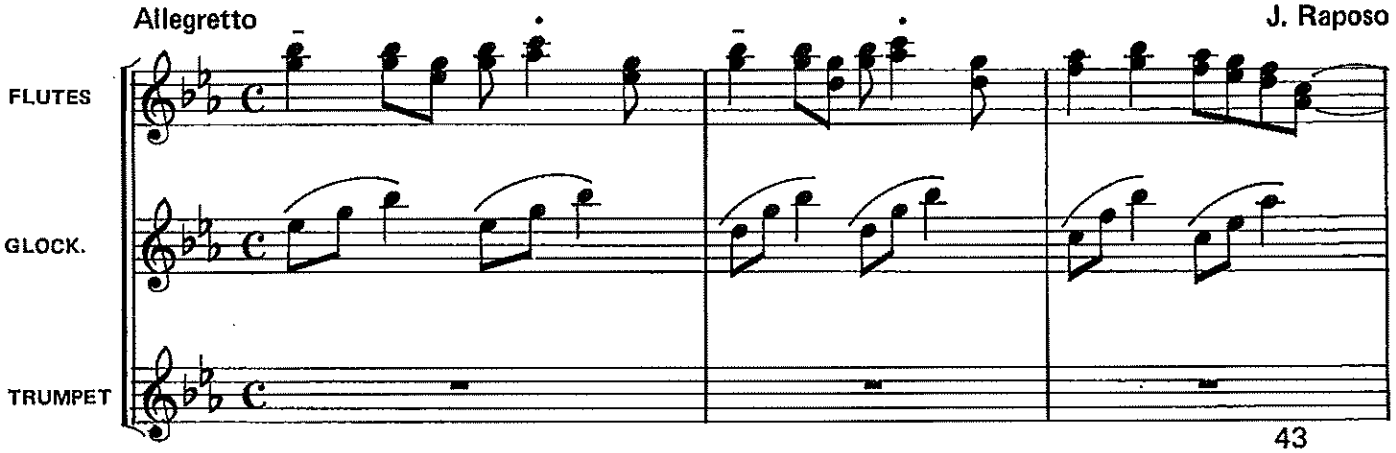
SING

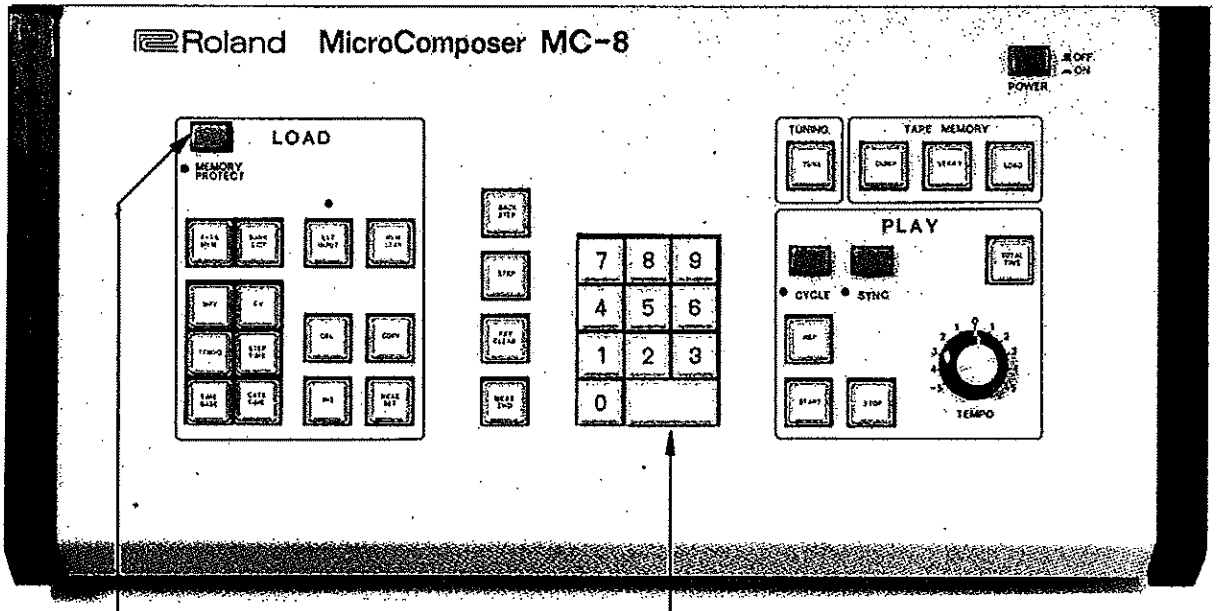
MM ♩ = 150 TIME BASE ♩ = 16

CHANNEL		1. FLUTES						2. GLOCK						3. TRUMPET						4. TEMPO	
MEASURE	STEP	1	2	S	G	VCA 3		4	S	G	5			7	S	G	8			T	S
1	1	46	43	16	14	60		39	8	2	60			34	64	0				150	64
	2	46	43	8	6	60		43	8	2	58										
	3	43	39	8	6	55		46	16	2	55										
	4	46	43	8	6	60		39	8	2	58										
	5	48	44	16	4	63		43	8	2	56										
	6	43	39	8	6	53		46	16	2	54										
2	1	C	43	C	C	C		38	C	3X	59			34	64	0				150	64
	2	1X	43	1X	1X	1X		43	M1	M1	57										
	3	M1	38	M1	M1	M1		46			54										
	4	M1	43	M1	M1	M1		38			57										
	5		44					43			55										
	6		38					46			53										
3	1	44	41	16	10	60		36			58			34	64	0				150	32
	2	46	43	16	10	62		41			56									146	8
	3	44	41	8	8	58		46			53									142	|
	4	43	39	|	6	58		36			56									140	|
	5	41	38	|	6	55		39			54									138	8
	6	36	32	8	8	53		44			52										
4	1	36	32	64	60	50		38			54			34	64	0				136	8
	2							41			52									134	8
	3							46			48									132	16
	4							38			52									126	8
	5							41			48									120	8
	6							46			45									100	16

PROGRAM 20: Sing

Allegretto J. Raposo





ENTER Key

In the following pages this is shown by [ENTER]

NOTE: None of the front panel controls operate when the MEMORY PROTECT switch is depressed (LED on).

5. Loading the MicroComposer from the Program Sheet

Introduction

The following explains the theory of loading the MicroComposer. If you are in a hurry to try the MicroComposer, jump to page 54 and begin there.

The MicroComposer is designed so that it can control all of the variables in a sound. For this reason, at least three of the memories must be loaded with data before the MicroComposer will cause the synthesizer to produce sound:

1. One CV memory (pitch)
2. One STEP TIME memory (note time value)
3. One GATE TIME memory (length of time key would remain depressed if played on the keyboard controller)

2 and 3 above combine to produce the gate pulse output for the synthesizer. In addition to the above, it is also necessary to load TIME BASE and TEMPO data.

Loading data into a memory requires three operations:

OPERATION 1:
Establishing the ADDRESS (location in the memory)
OPERATION 2:
Memory selection and channel assignment
OPERATION 3:
Writing (loading) the data into the memory

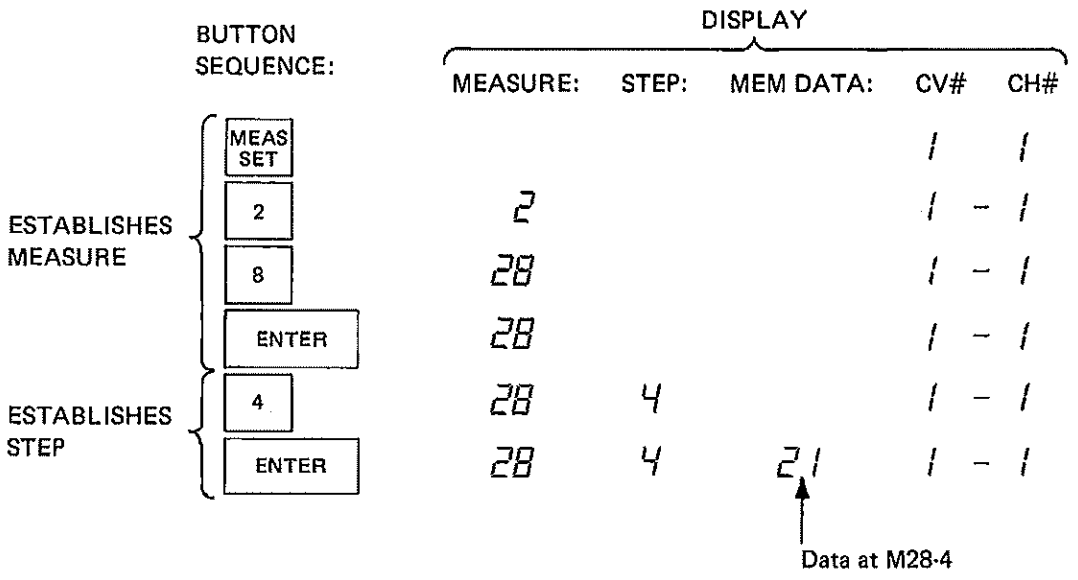
In actual practice, it is not necessary to stick to the above order. For example, Operations 1 and 2 may be reversed with Operation 2 coming first.

OPERATION 1: Establishing the ADDRESS

ADDRESS refers to a specific location within a given memory. A memory may be divided into up to 999 measures, each measure of which may be divided into up to 100 steps (or notes). Note that the STEP display uses only two digits. To indicate Step 100, the display will show "0".*

> **NOTE:** In this manual, memory addresses will be written:
>
> M28-4 (Measure 28, Step 4)

EXAMPLE: Set ADDRESS M28-4.



> **NOTE:** Figures shown in half tone represent a hypothetical display resulting from some previous MicroComposer operation.

*In actual practice, it is possible to load as many as 256 steps into any one measure. (see p. 148)

Exceptions to the above are as follows:

When setting Step 1 of any given measure.

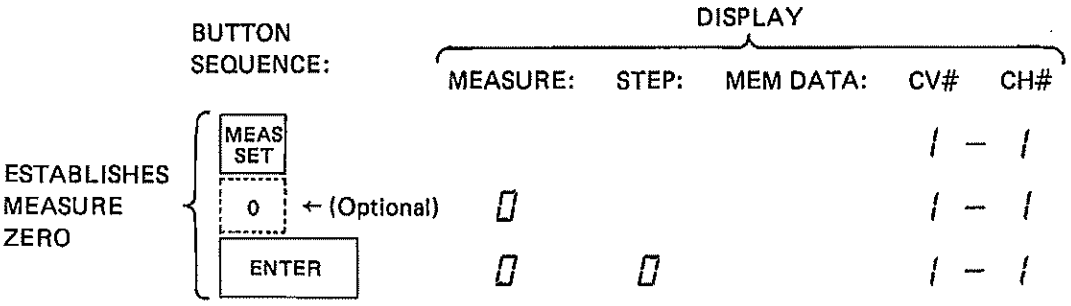
EXAMPLE: Set ADDRESS M14-1.

BUTTON SEQUENCE:		DISPLAY				
		MEASURE:	STEP:	MEM DATA:	CV#	CH#
	MEAS SET				1 – 1	
ESTABLISHES MEASURE	1	1			1 – 1	
	4	14			1 – 1	
	ENTER	14			1 – 1	
ESTABLISHES STEP	ENTER	14	1	34 ↑ Data at M14-1	1 – 1	

The [1] key may be used, but is not needed

When loading data from the beginning of a program.

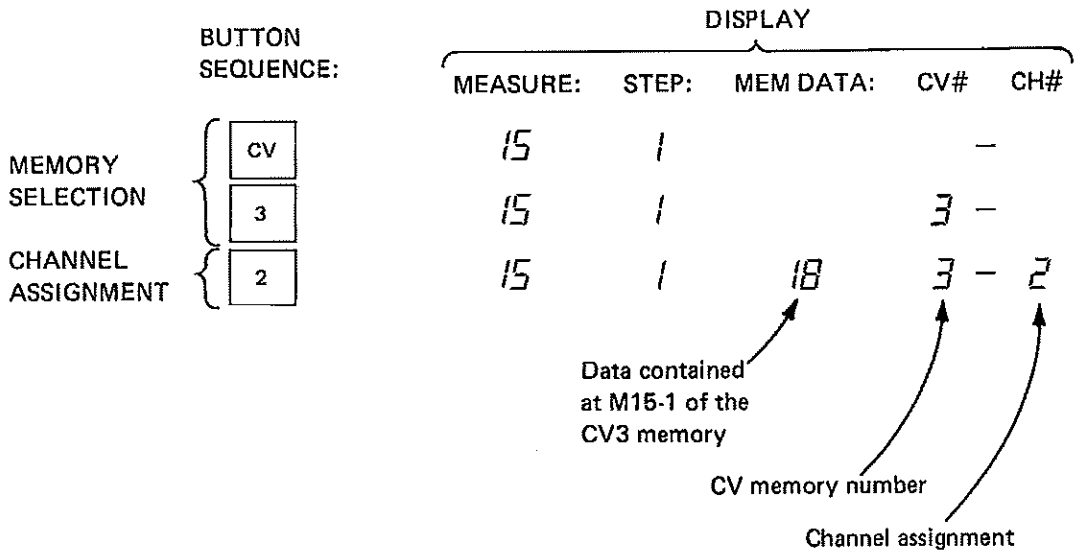
EXAMPLE: Set ADDRESS Measure 0.



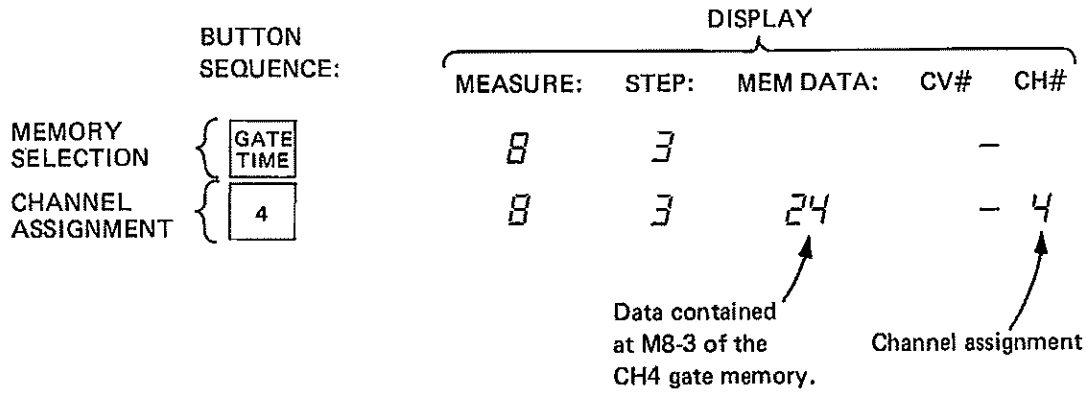
OPERATION 2: Memory Selection

The memories are: Control Voltage (CV) (8), STEP TIME (8), GATE TIME (8), Multiplex (MPX) (1), TEMPO (1), and TIME BASE (1).

EXAMPLE: Select CV3 Memory and assign it to channel 2.



EXAMPLE: Select Channel 4 GATE TIME memory.

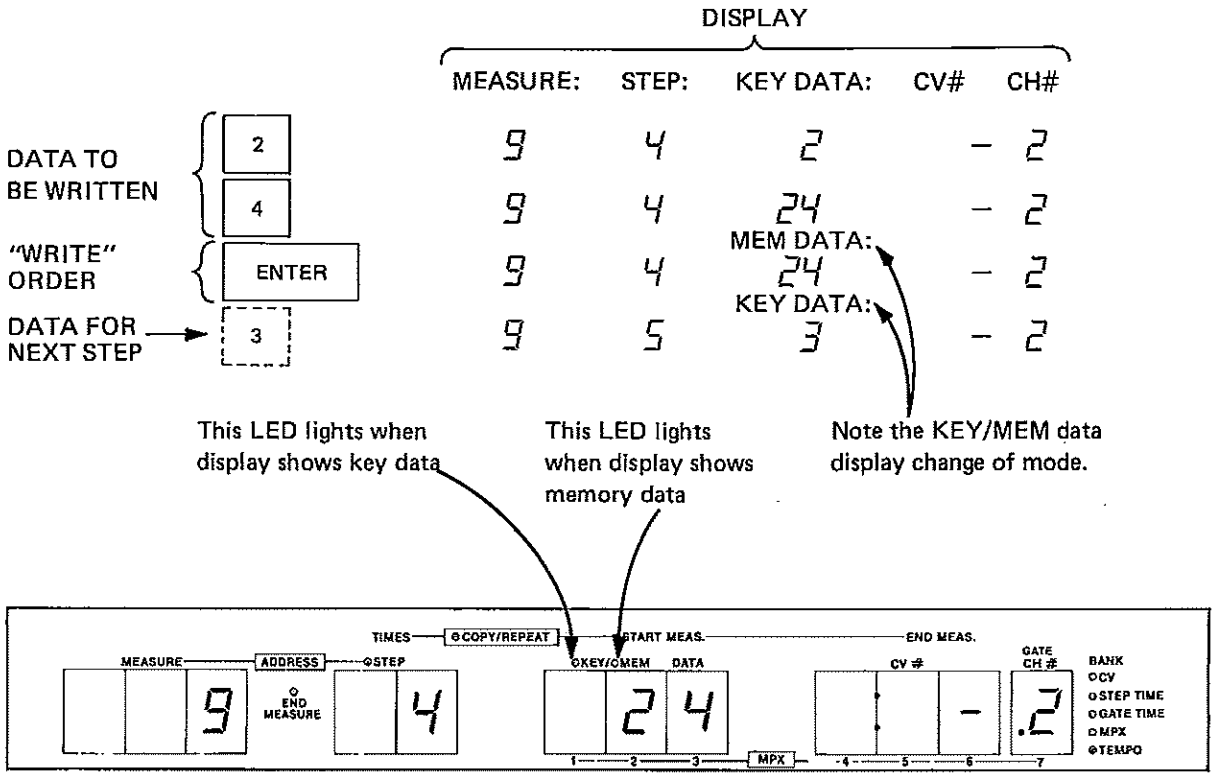


CV memory selection is the only memory selection which requires pushing two of the digit buttons; all others require only one digit.

OPERATION 3: Writing

Operation 3 consists of writing (loading) data into the memories.

EXAMPLE: Write "24" into the memory.



IMPORTANT: The first number key pressed in the write operation advances the ADDRESS one step.

After [ENTER] is pressed, the KEY/MEM DATA display can be checked to make sure that the correct data has been written into the memory at the desired ADDRESS. After that, the pressing of the next key (data for the next step) will advance the ADDRESS thus eliminating the need for using the [STEP] key. This greatly simplifies the loading of long columns of data.

In Operation 3, when loading a CV memory, pushing [ENTER] will produce a short gate pulse at the related GATE OUTPUT jack on the Interface panel which can be used for triggering the synthesizer so that the pitch values being loaded can be monitored by ear as they are loaded.

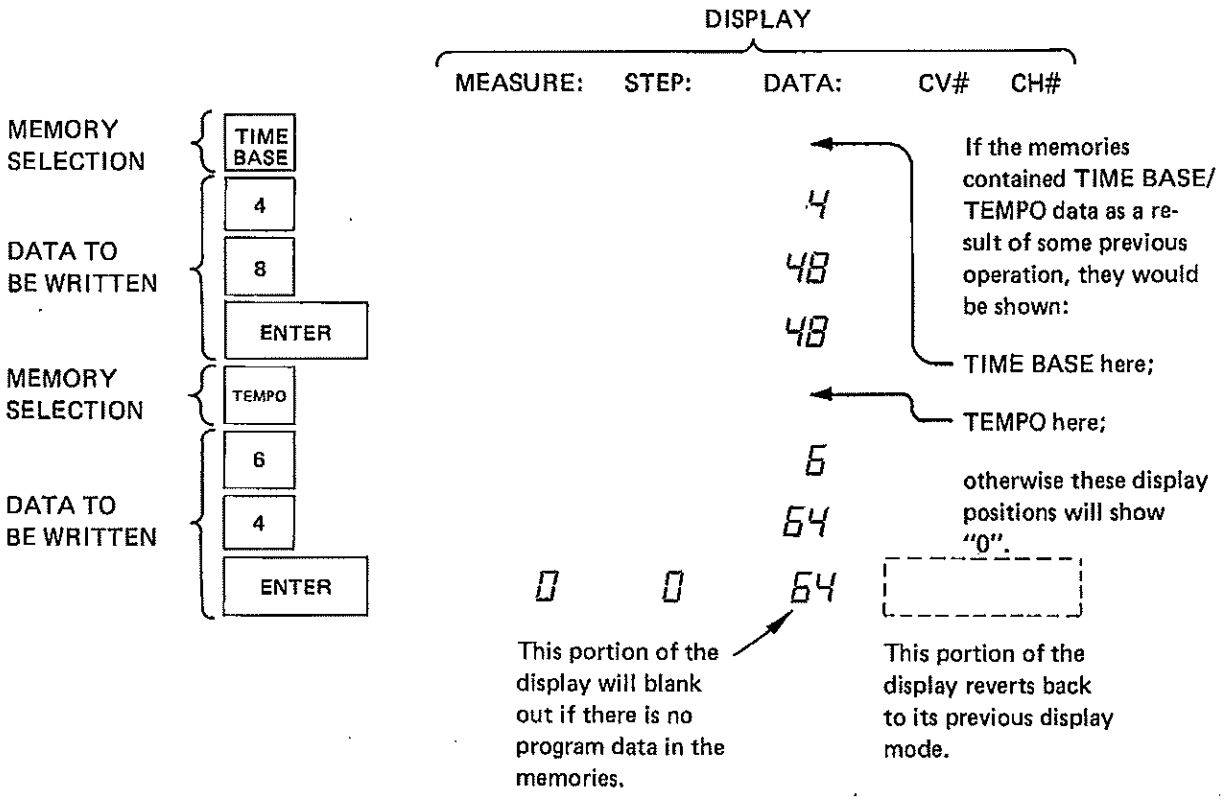
The TIME BASE/TEMPO set Operation

The TIME BASE and TEMPO are stored in their own separate memories. Both must be translated into terms of a quarter note. For example:

If tempo is MM ♩. = 72, then MM ♩ = 108*

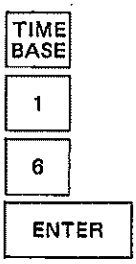
EXAMPLE: Set a TEMPO of MM ♩ = 32 with a TIME BASE of ♩ = 48.

If MM ♩ = 32, then MM ♩ = 64

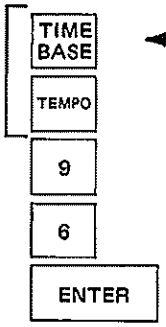


*MM = Maelzel's Metronome. MM ♩. = 72 means 72 dotted quarter notes per minute; therefore, 216 eighth notes per minute (3 x 72 = 216) or 108 quarter notes per minute (216 ÷ 2 = 108).

EXAMPLE: Set TIME BASE only; TIME BASE ♩ = 16



EXAMPLE: Set TEMPO only; MM ♩ = 96



Note that when setting TEMPO only, the TIME BASE button must be used first.
This is to differentiate fixed TEMPO from variable TEMPO which is explained in Section 14.

The data limits are:

TIME BASE:	4 – 255
TEMPO:	2 – 254 (even numbers only)

For a given program, the TIME BASE multiplied by the TEMPO must equal 128 or more. Trying to load a TIME BASE/TEMPO product of less than 128 will activate the ERROR function when [START] is pushed.

The upper extreme for the TIME BASE/TEMPO product will depend on a number of factors and may vary from program to program. The final test would be simply to try to run the program by pushing the [START] button. If the program runs with no trouble, fine.

If the program will not run, pushing [START] will throw the MicroComposer into a deadlock such that the front panel controls seem to have no effect. This is a rare occurrence and should therefore cause little trouble. The deadlock can be broken by turning off the POWER switch. (Section 14 shows how to break this deadlock without destroying the program).

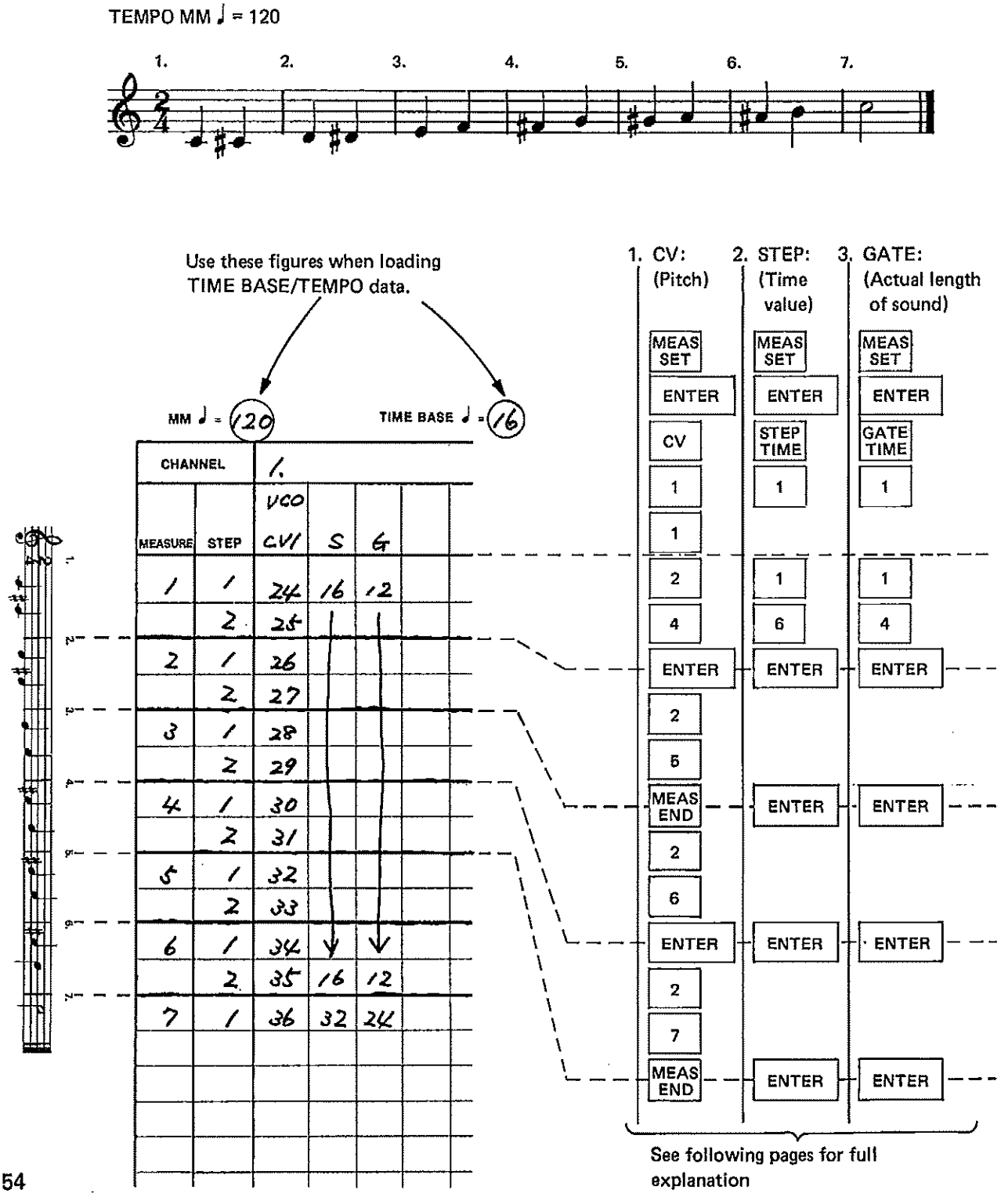
The programming and production of variable tempo is discussed in Section 14. Section 14 also goes into more detail about problems which might be encountered with very high TIME BASE/TEMPO products.

TIME BASE and/or TEMPO may be loaded at any time: before, after, or in the middle of loading the main program data.

Sample Programs

Program 21 is a simple chromatic scale for practice programming. Before trying this program, clear all the memories by turning off the POWER switch for a few seconds. Also, make sure that the MEMORY PROTECT switch is in the raised (LED off) position.

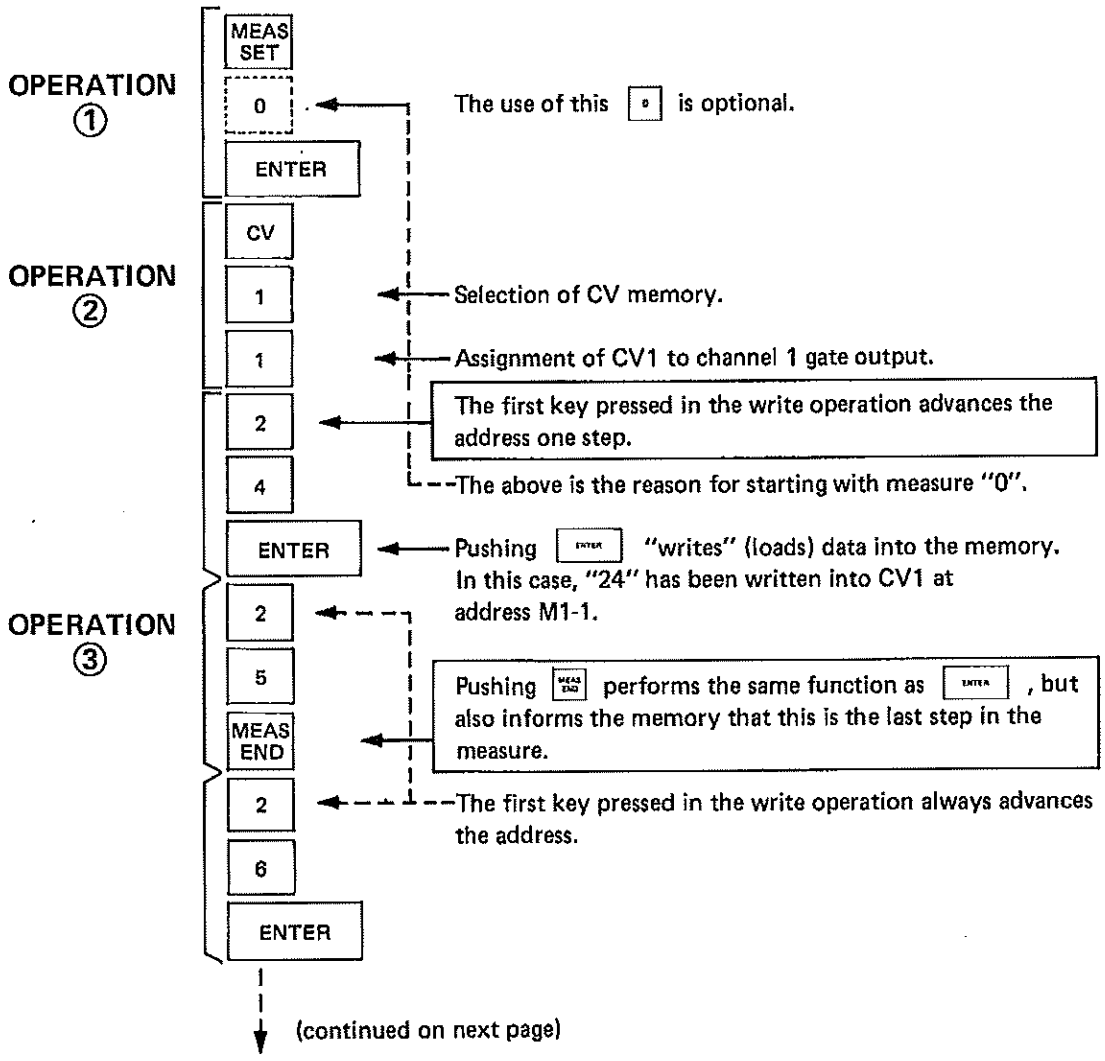
PROGRAM 21:



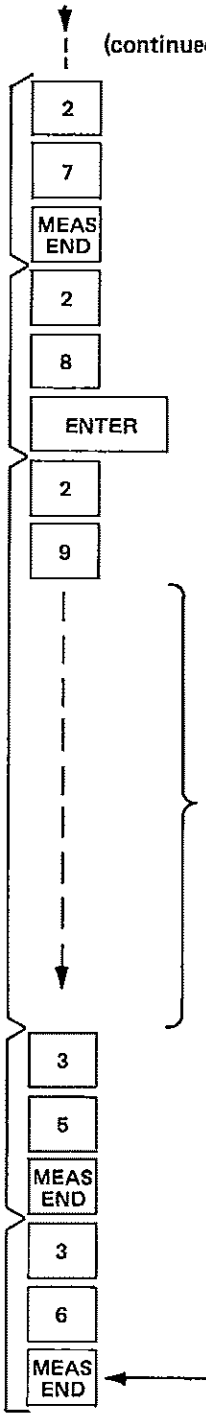
Usually, TIME BASE/TEMPO data is loaded first. Use the procedure shown on page 51.

The best practice is to load the program one column at a time starting at the left and moving column by column towards the right. In any event, at least one CV memory must be assigned and loaded before the MicroComposer will accept data for either of the TIME memories in a given channel.

EXAMPLE: Load CV1 for Program 21.



OPERATION
③
(continued)

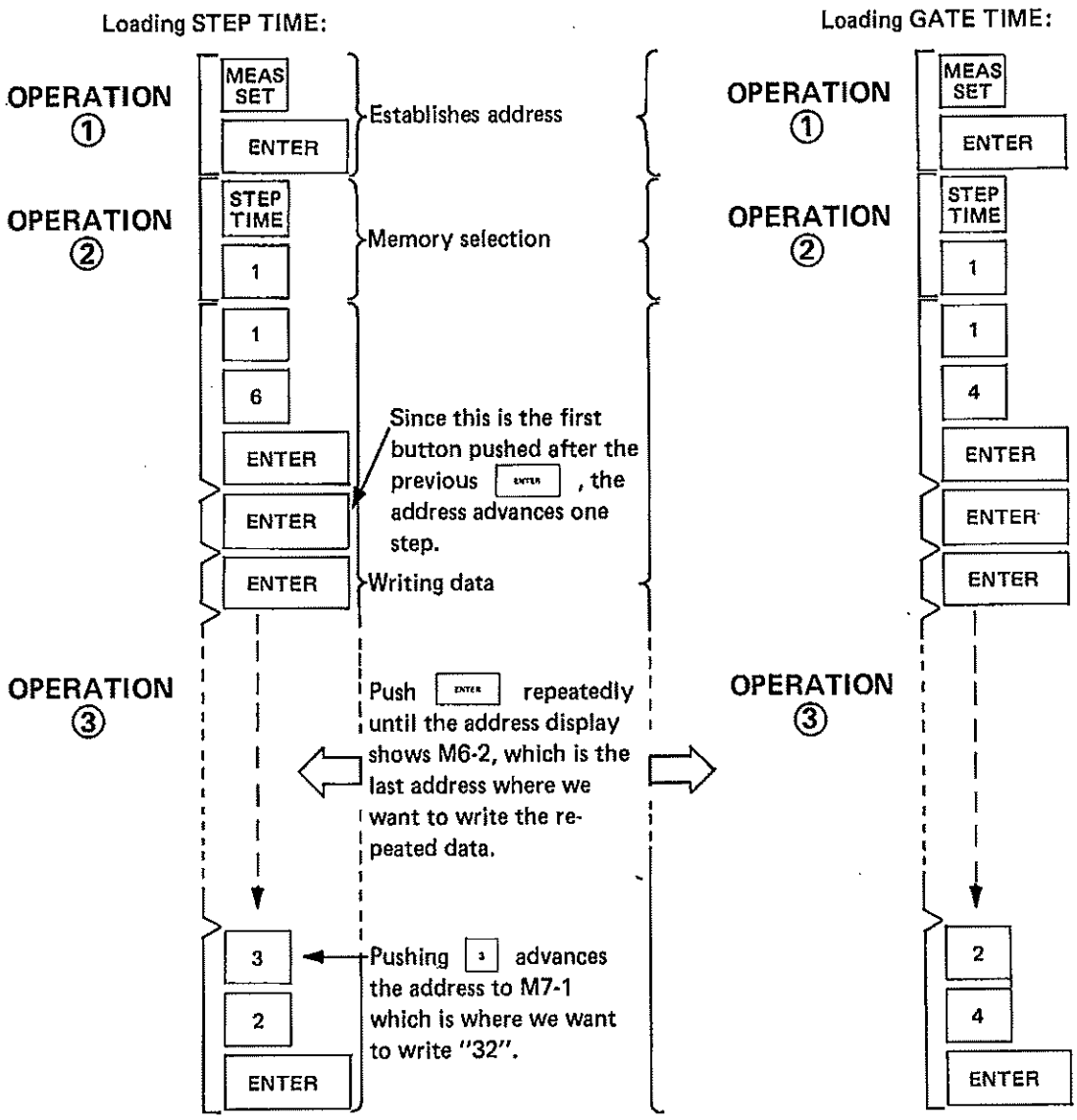


Note that when loading the CV memory, if the synthesizer is set up and connected to the MicroComposer, pushing [ENTER] or [MEAS END] will produce a short gate pulse which will trigger the synthesizer, thus the pitches can be monitored as they are loaded.

Continue in the same manner until the end.
Don't forget to use [MEAS END]

The last step in the program must always be followed by [MEAS END] even if it is not a complete measure, otherwise the MicroComposer may continue to run after the end of the program, giving out random data as it runs.

Next, load STEP TIME, then GATE TIME.



Note that pushing ENTER writes into the memory whatever data is displayed by the KEY/MEM DATA display; therefore, ENTER can be used for data which repeats.

Also note that MEAS SET is not necessary (but may be used, if desired) when loading the TIME memories.

Producing the Sequence

Make sure that the SYNC switch is in the raised position. On the MC-8 Interface, make sure that the PORTAMENTO CONTROLLER is at "0" and the switch at MANUAL. (Or turn the PORTAMENTO knob up a little if you want to add portamento effect to the notes).

Push [TUNE] and tune the VCO to A = 440Hz.

To produce the program, push [START]

The tempo of the program may be varied with the TEMPO control by approximately ±60% of the programmed tempo.

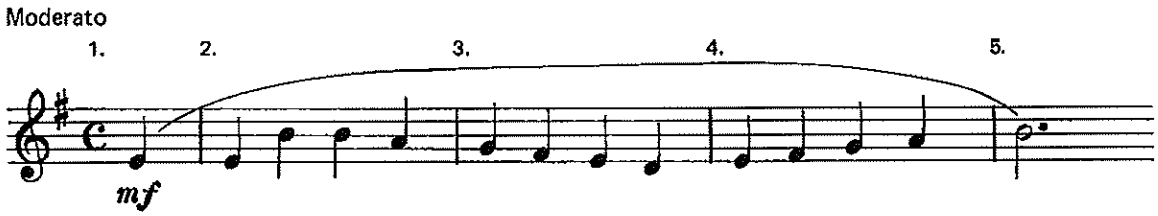
With the CYCLE switch depressed (LED on), the program will repeat itself over and over. With the switch raised (LED off), the program will run one time from the beginning each time the [START] button is pushed.

If you have trouble, double check that the synthesizer is set up and connected correctly. If the trouble is with the MicroComposer, it means a mistake has been made in programming. For the time being, clear the memories (by turning off the POWER switch) and try the program again from the beginning. (Editing and/or the correction of mistakes is discussed in detail in Section 8).

For exercise, also try Programs 22 and 23 which follow. After that, if you want to try some music with a rock flavor, try Program 15 in Section 4 using the same techniques and patching shown for Program 23.

PROGRAM 22:

GOD REST YE MERRY, GENTLEMEN



MM ♩ = 162 TIME BASE ♩ = 16

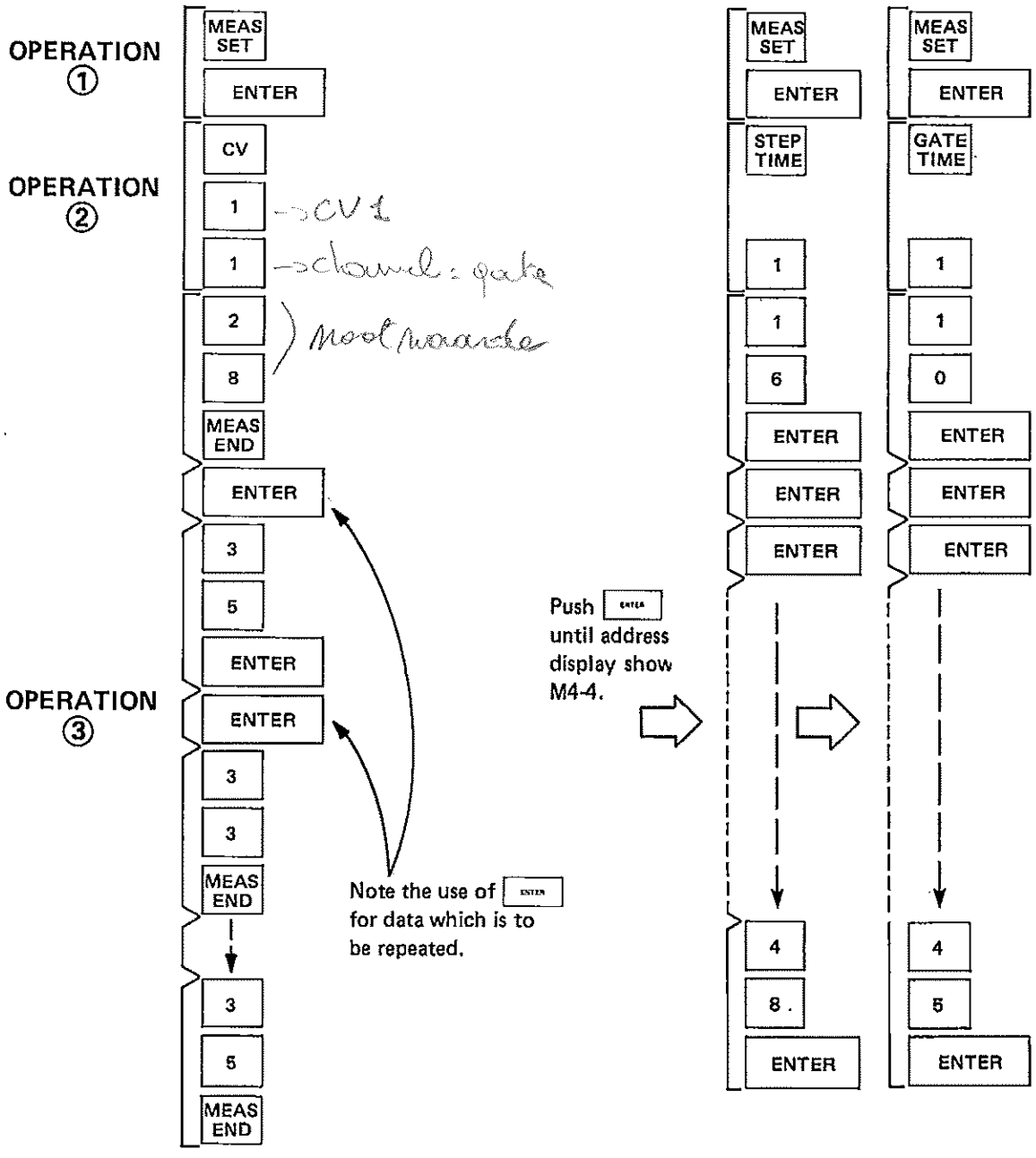
CHANNEL		1.						
MEASURE	STEP	CV1	T	G				
1	1	28	16	10				
2	1	28						
	2	35						
	3	35						
	4	33						
3	1	31						
	2	30						
	3	28						
	4	26						
4	1	28						
	2	30						
	3	31						
	4	33	16	10				
5	1	35	48	45				

PROGRAM 22:

GOD REST YE MERRY, GENTLEMEN

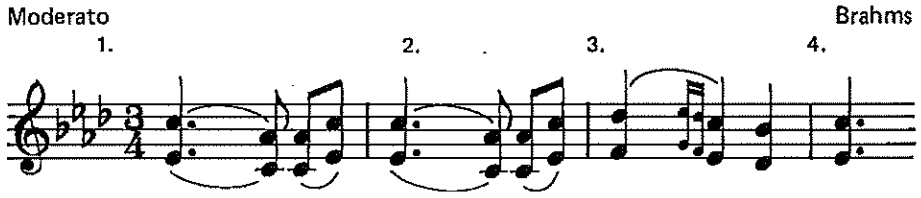
❶ Load TIME BASE/TEMPO data (p. 51)

❷ Load CV1:



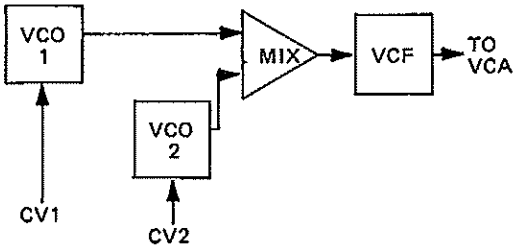
PROGRAM 23:

WALTZ



MM ♩ = 120 TIME BASE ♩ = 32

CHANNEL							
		VCO 1	VCO2				
MEASURE	STEP	CV1	CV2	S	G		
1	1	36	27	48	46		
	2	32	24	16	12		
	3	32	24	16	14		
	4	36	27	16	12		
2	1	36	27	48	46		
	2	32	24	16	12		
	3	32	24	16	14		
	4	36	27	16	12		
3	1	37	29	24	20		
	2	39	31	4	3		
	3	37	29	4	3		
	4	36	27	32	30		
	5	34	25	32	30		
4	1	36	27	48	45		



If you monitor the pitches as you load them, note that:

while loading CV1, since CV2 contains no data, VCO-2 will produce pitches equivalent to data "0".*

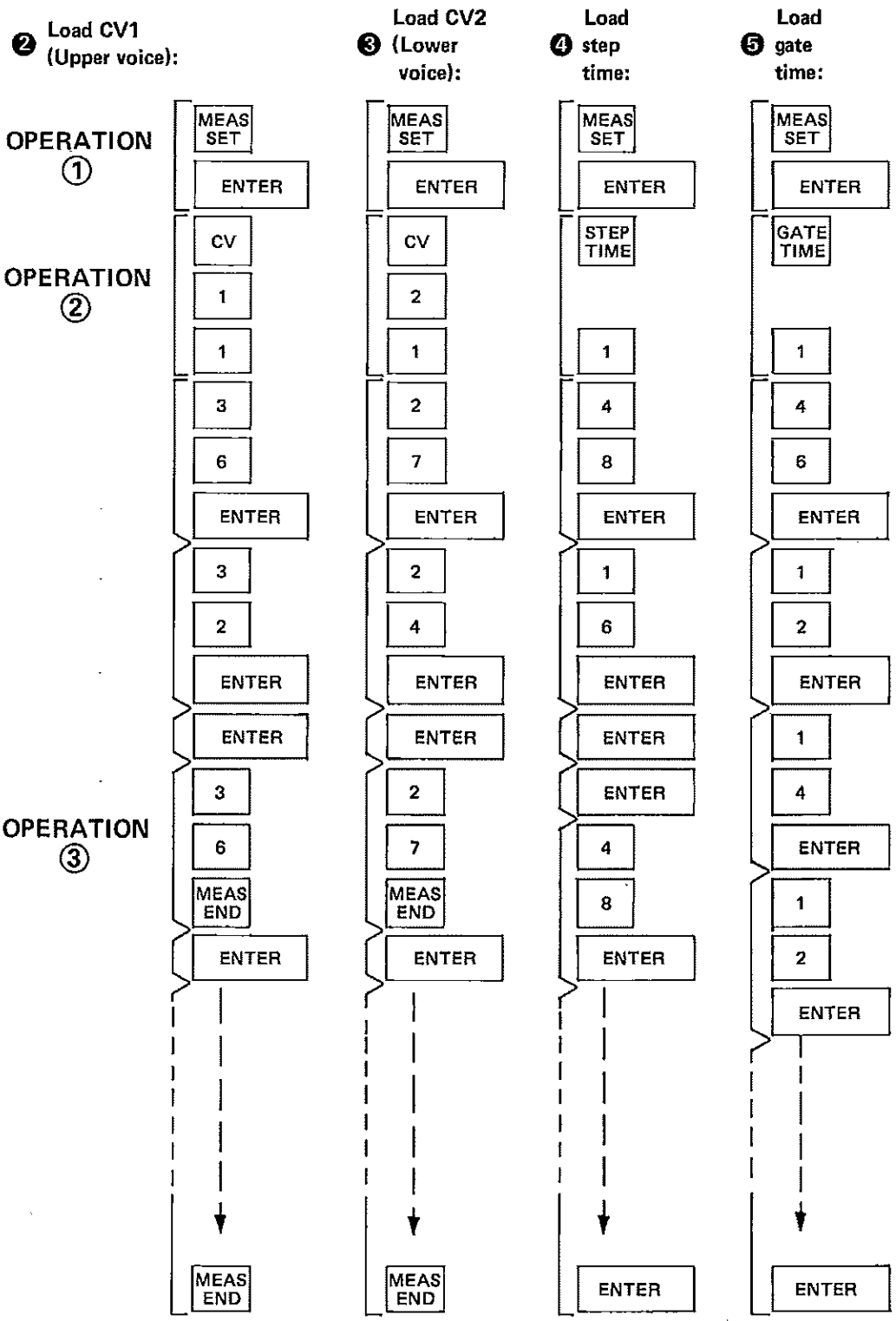
while loading CV2, VCO-1 will produce the pitch equivalent to the last data loaded in CV1; in this case "27".

Push [TUNE] and tune both VCO's to unison (A = 440Hz.) after loading the program.

*This is true only if the POWER switch was used to erase the memories prior to loading.

PROGRAM 23: BRAHMS WALTZ

❶ Load TIME BASE/TEMPO data (p. 51)



Summary

The MicroComposer outputs eight independent gate pulses. The data contained in the STEP TIME and GATE TIME memories for a given channel combine to produce the gate pulse output for that channel. STEP/GATE data for a given channel cannot be loaded into that channel until at least one other memory (CV, MPX, or TEMPO*) has been assigned to that channel and loaded with data. The [MEAS END] button must be used when loading the assignable memories (CV, MPX, and TEMPO), but is not necessary when loading STEP/GATE memories.

Usually, the program is loaded one column at a time starting at the left side of the program sheet. Usually, the entire column is loaded even though it may run for many pages of program sheets. The main exception to this would be where you might want to program a few complete measures at a time for experimentation. For example, for a given set of data in the GATE TIME memory, the phrasing and articulation may change drastically with different settings of the synthesizer envelope generator controls.

*Assigning TEMPO to a gate channel output allows the programming of variable tempo; this is discussed in Section 14.

6. Loading from an External Source

Inputs

The two EXT INPUT jacks on the Interface front panel can be used for loading control voltages only, or for loading control voltages with gate pulses from any external source such as a keyboard controller, analog sequencer, sample and hold, etc.

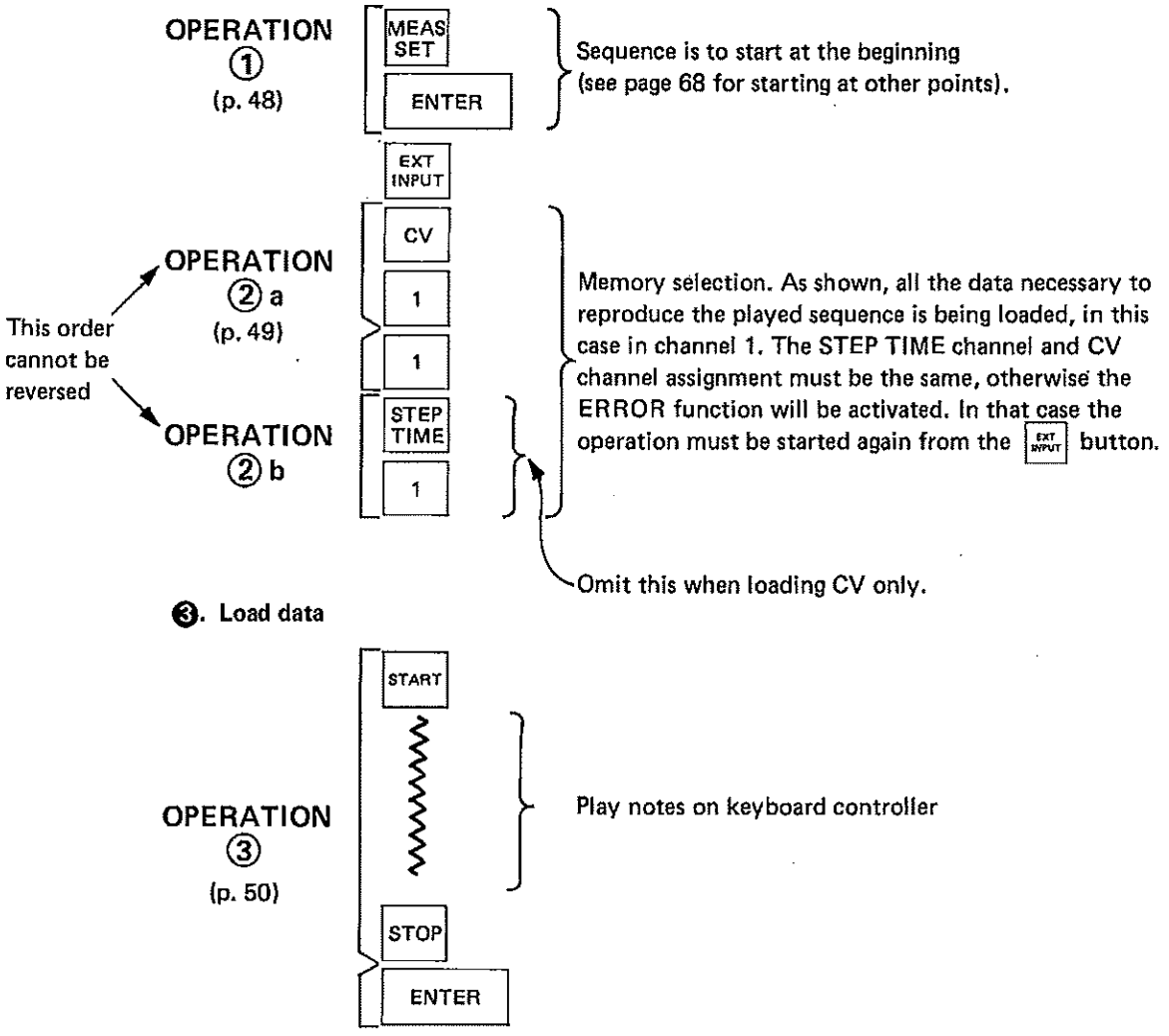
The GATE jack will accept negative gate pulses when the GATE switch on the back of the Interface (see page 5) is in the INV (invert) position.

LOADING FROM AN EXTERNAL SOURCE

NOTE: When loading from an external source, the related memories must be cleared first, otherwise data will not be loaded correctly.

❶. Decide and set TIME BASE and TEMPO (p. 51)

❷. Select start measure and memories to be loaded.



❸. Load data

❹. Push START to produce the sequence. (See p. 58)

Loading CV and Gate Data

When loading from an external source, the MicroComposer uses half of its available memory space as a sort of scratch pad to remember the data being loaded. When [ENTER] is pushed, the data is transfered to its permanent place in the memories. This means that when loading from an external source, the effective memory capacity is half of the available memory space. This is one reason the first step should be to clear the memories of all unwanted data (see "The Memory Clear Function", page 88).

The other reason is that when loading from an external source, any data contained in the place where the external data is to go will be mixed in with the old data.

Once the data from the "scratch pad" has been written into its permanent place, the "scratch pad" area becomes free and may be used for writing more data, either from the external source, or with the keys on the front panel.

Decide TIME BASE and TEMPO in more or less the same way as you would if you were going to load the data with the keys on the front panel of the MicroComposer. This is important because if the TIME BASE times TEMPO equals a smaller number than the fastest gate pulse you intend to input, the resolution will not be fine enough and the MicroComposer will miss notes and many note time values will be noticeably different from the way they were played when loaded.

After preparing for loading from an external source, if you push [START] but there is no external source connected to the EXT INPUT jacks, the MicroComposer will be thrown into a deadlock where none of the front panel controls have any effect. This can be broken into by turning off the POWER switch, or by inputting a single gate pulse into the EXT INPUT GATE jack, then pushing [STOP].

While loading from an external source, the display will show the data being written into the CV memory. Once [ENTER] is pushed at the end of the sequence, the display will revert to the STEP TIME display if that was the last memory selection before loading.

The MEASURE display will remain at "1" and the STEP display will change each time a note is played. If a note time value (or rest) is too long for one step (more than 256 increments long), the STEP display will advance and use the next memory space to continue the note. Remember that the STEP display can run beyond 99, but that only the last digits will be displayed. Also, when editing, Operation 2 (measure set) cannot be used for selecting steps beyond Step 256 (although the program will run beyond this point).

While loading data from the external source, if the memory becomes full, the ERROR function will be activated. If you push [ENTER], the data will be written into the memory. The note before the note which caused the ERROR function will be the last note written into the memory.

Loading CV Only

The connection between the gate source and GATE INPUT jack must be made even when loading CV only.

Internally, the method for writing CV only, or CV + GATE is exactly the same. The only difference is that when loading CV only, the gate data is not written into the memory. Thus, when loading CV only, if a note is held too long, it will take up more than one step in the memory. To get around this, you can temporarily program an extremely slow tempo, such as MM♩ = 4 (TIME BASE times TEMPO must equal 128 or more). With a temporary slow tempo, notes can be held much longer and still have each CV value take up only one step.

Once CV data has been loaded, the STEP TIME and GATE TIME memories can be loaded using the keys on the front panel of the MicroComposer as described in the previous section.

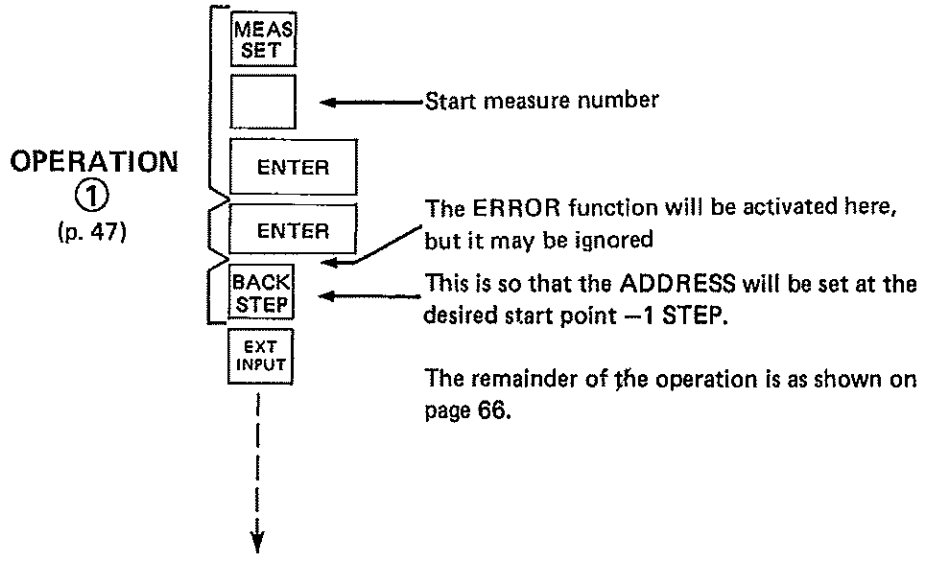
Editing

Editing may be accomplished either by replaying the sequence from any given point to the end, or by using the keys on the front panel as explained in the following sections of this manual.

LOADING EXTERNAL DATA FROM ANY POINT OTHER THAN THE BEGINNING

❶ .The memories to be loaded must be cleared from the start point to the end ("The MEMORY CLEAR Function, page 88). The CV data just before the start point must contain a [MEAS END] flag (page 81).

❷ .Select start measure and memories to be loaded.



Tuning the External Source

If the external source is not tuned correctly, it is possible for mistracking (wrong pitches) to occur. If the external source is a properly calibrated keyboard controller, then usually setting the TUNING control at center is enough.

If you have access to a digital voltmeter, you can set the keyboard controller TUNING control at the center of the range which produces a voltage reading at the CV output with no digits to the right of the decimal point when any C key is pressed. For example, middle C normally produces 2.00 volts.

Another method is to set the MicroComposer to accept data from the external source, then repeatedly play any given note. Adjust the keyboard controller TUNING control while watching the data at the KEY/MEM DATA display. There will be a range on the TUNING control where the data will remain unchanged. Above this range, the data will move up one value; below this range, down one value. The TUNING control should be set in the center of this range. Once set, the memories can be cleared and the desired sequence loaded.

Portamento

The CV memories will accept voltage levels only in chromatic semi-tone increments, thus portamento is not normally used when loading from an external source.

This voltage level effect can also sometimes be used with advantage with the output of a sample and hold circuit. Random notes would become random notes whose pitches fall within the scale system being used. The pitches of notes generated by other wavefomrs (such as a sawtooth wave) would also fall within the scale system being used.

Multichannel Data Loading

The external input function operates in such a way that it is possible to load data by playing along with melodies already loaded into the memories in the same way that recording is done with a multichannel tape recorder. This type of loading, however, will prove impractical because of timing errors which occur due to the fact that the CPU (the "brain" of the MicroComposer) is too "busy" trying to produce correct gate times at the outputs while at the same time trying to accurately read the gate times at the external input. Also, it is humanly impossible to play notes with perfect accuracy. For example, if you happen to play a note whose timing turns out to be 19.53 timing increments, the MicroComposer will either load a 19 or a 20 into the timing memory since fractions cannot be loaded.

7. The RUN Function

Repeat RUN Function

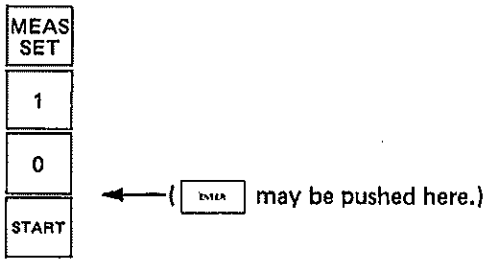
The MicroComposer program may be run from any given measure to the end of the program, or to some other measure in the program. This run may be made one time only, or repeated as many times as desired. This kind of run can be used where it is desired to check only a portion of a program, or for setting up synthesizer sounds to match a given passage.

Note that the [REP] (REPEAT) button is used for selecting a group of one or more measures from the program and that this group will repeat only with the CYCLE button depressed.

Also note that whenever the MicroComposer is running that releasing the CYCLE button will cause the MicroComposer to continue running to the end of the program (or to the end of the group selected with the [REP] button), then stop.

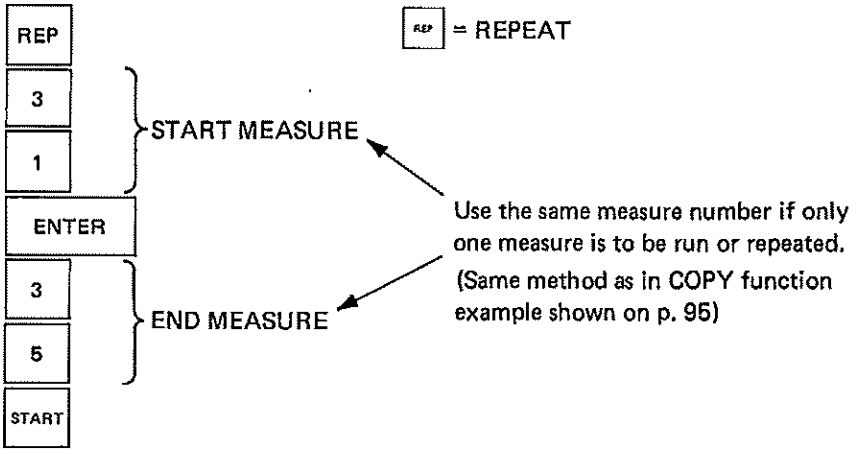
Repeat/Run Functions

EXAMPLE: Run from M10 to end of program



CYCLE switch off: Program will run once from M10-1 to end of program.
CYCLE switch on: The portion of the program from M10-1 to the end will repeat endlessly (until STOP is pushed or CYCLE is turned off).

EXAMPLE: Run from M31 to M35 (inclusive)



CYCLE switch off: Program will run once from M31-1 to the last step in M35.
CYCLE switch on: The portion of the program from M31-1 to the last step in M35 will repeat endlessly (until STOP is pushed or CYCLE is turned off).

Running the Program

The program will run to the end of the channel with the largest number of measures. For example, if Channel 1 has been fully programmed from M1 to M50, but Channel 2 has been programmed from M1 to only M4 (perhaps you are experimenting with Channel 2 phrasing), when [START] is pushed, the program will run all the way to the end of M50. The Channel 1 CV and gate will remain normal throughout; after M4, the Channel 2 data will cease flowing.

If the Channel 2 M4 were an incomplete measure (no [MEAS END]), the Channel 2 voice would produce random and intermittent sounds beyond M4 up to some indeterminate point. If the Channel 1 M50 were an incomplete measure (again, no [MEAS END]), the program would continue running to some indeterminate point beyond M50, with the Channel 1 voice producing random and intermittent sounds.

The MicroComposer RUN function cannot be started from any measure which does not contain data for all memories that have been selected and/or assigned. In other words, using the above example again, the program can be started from M1, M2, M3, or M4, but not from any measure after M4 because the data for Channel 2 only goes to M4. Trying to start the RUN after M4 will activate the ERROR function.

Normally, pushing the [START] button will start the program RUN from M1. If a memory has been selected and/or assigned, but no data loaded (as can happen sometimes if you hit the wrong number key in Operation 2), the MicroComposer will not run at all. This is because there is a memory in the program which contains no data at M1-1. Pushing [START] will activate the ERROR function. The way to correct this is to either load data into a memory or remove the memory assignment with the memory CLEAR function (as shown on page 90; see also page 78).

Tuning

Pushing [TUNE] causes all the CV outputs to produce a voltage equal to the voltage that would be produced if a "33" were loaded into each CV memory. If you use the CV data/pitch relation system used in this manual, this voltage should produce the pitch of 440Hz A (the A above middle C). Using [TUNE] then, all VCO's may be tuned to unison.

Each CV output contains what is known as a latch circuit. This means that each CV output will remember the last CV value received from the internal circuits, much as a synthesizer keyboard controller will "remember" the last key struck. The MicroComposer CV outputs will retain the exact pitch indefinitely and will not "leak" upwards or downwards*. The main advantage of this is that specific chords or intervals within the program may be tuned perfectly beat-free, if desired.

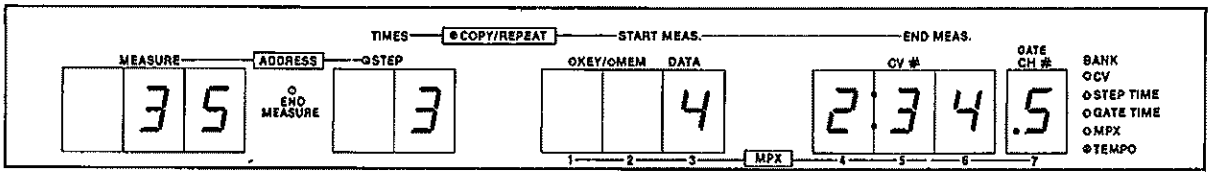
To tune a chord, run the measure which contains the chord. When the chord sounds, immediately push [STOP]. Turn up the VCA gain controls and tune the VCO's as desired.

*When using any of the tape memory functions (Section 12), the CV levels will "leak" upwards, but will return to normal when the function ends.

The Display

When the MicroComposer is running, the right hand portion of the display acts as a TIMER (Section 13), and the rest of the display will show memory data as the program runs. For example, to monitor CV data, use Operation 2 to select the desired CV memory, then while the program is being run you can watch the pitch data follow the pitches that they represent. When the program stops, the TIMER display remains. This can be canceled with the BACK STEP or STEP key so that the right hand portion of the display goes back to showing which memory is being monitored by the display.

This portion of the display acts as a TIMER when the program is run.



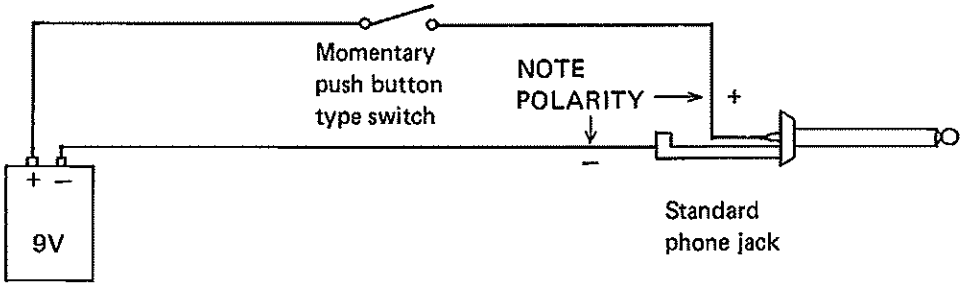
When setting up to run the program from any point other than M1 (p. 72) this portion of the display shows the START MEASURE number.

When setting up to use the END function, this portion of the display shows the END MEASURE number.

(See also page 77 for explanation of NON-DISPLAY mode of display).

Remote START / STOP

The REMOTE START and REMOTE STOP jacks on the rear panel of the MicroComposer may be used to control these functions from a remote position. All that is needed is a momentary push button type switch and a voltage source of from about +5 to about +10 volts. A 9 volt transistor radio battery can be used, if desired, and should last a long time since these jacks require very little current and only when the push button is closed.



A multiplex pulse (explained in Section 11) could be put into the program and recorded on a spare track of the tape recorder, then used later to trigger the MicroComposer RUN function for joining two long programs together in series.

8. Program Revisions

Key Punching Mistakes

If the wrong number key is accidentally pressed, press [KEY CLEAR], then the correct number.

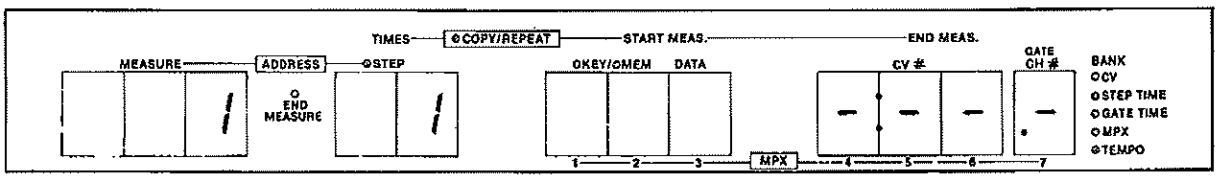
The only exception to the above is when selecting a memory (Operation 2). This is because pressing a number key in Operation 2 (channel number, or CV number and channel assignment) completes the operation without the need for the [ENTER] key. To correct this kind of mistake, simply begin Operation 2 over again. (See also the last paragraph in "Running the Program, p. 73)

The [BANK DISP]* Button

When editing the program, the [MEAS SET], [STEP], and [BACK STEP] buttons can be used for manually running through portions of the program vertically to check data contained in a given memory. The memory select buttons and the [BANK DISP] button can be used for running through the program horizontally for checking the data in each memory at a specific ADDRESS.

The order in which the memories are displayed is determined by the order in which they were loaded; they are not necessarily displayed in numerical order. If the memories were loaded in exactly the order shown by your program sheets, then pushing the [BANK DISP] button will have the effect of reading the data in steps across the program sheet from left to right at one given ADDRESS. Each time [BANK DISP] is pressed, the CV#/GATE CH# display will change to show the next memory to be read, and the KEY/MEM DATA display will change showing the data contained in that memory. The ADDRESS display will not change since you are, of course, reading across the program sheet horizontally.

Once you have read through all the memories, pressing [BANK DISP] once more will put the display in the NON-DISPLAY mode which is indicated by the CV#/GATE CH# portion of the display showing all dashes. This is a neutral state of display. The KEY/MEM DATA display will usually remain blank or show a "0", but when the program is run, the ADDRESS display will show the measure number from which the program run was started. (Also, the TIMER will operate normally).



The dashes indicate the NON-DISPLAY Mode.

After the NON-DISPLAY mode, pressing [BANK DISP] will start the process over again from the beginning. Also, pushing some other button will interrupt the [BANK DISP] mode so that pressing [BANK DISP] again will start the whole process over again, rather than continuing with the next memory in line.

* [BANK DISP] = (Memory) BANK DISPLAY

If you are using the [BANK DISP] at an ADDRESS where a given channel contains no data, the ERROR function will be activated when you reach this memory. Pressing the [BANK DISP] button again will start the process from the beginning again.

In the loading process, if the wrong number key is pressed while selecting a memory, and the memory CLEAR function is not used, this memory selection will remain and the program will not run because this memory contains no data. An accidental memory selection can be discovered simply by setting M1-1, then using the [BANK DISP] button. The CV#/GATE CH# display will show in turn each memory which was selected. Any memory which was accidentally selected and does not belong in the program will contain no data at M1-1, and, therefore, will activate the ERROR function when you come to it with the [BANK DISP] key. The MEMORY CLEAR function (page 88) can be used to get rid of it.

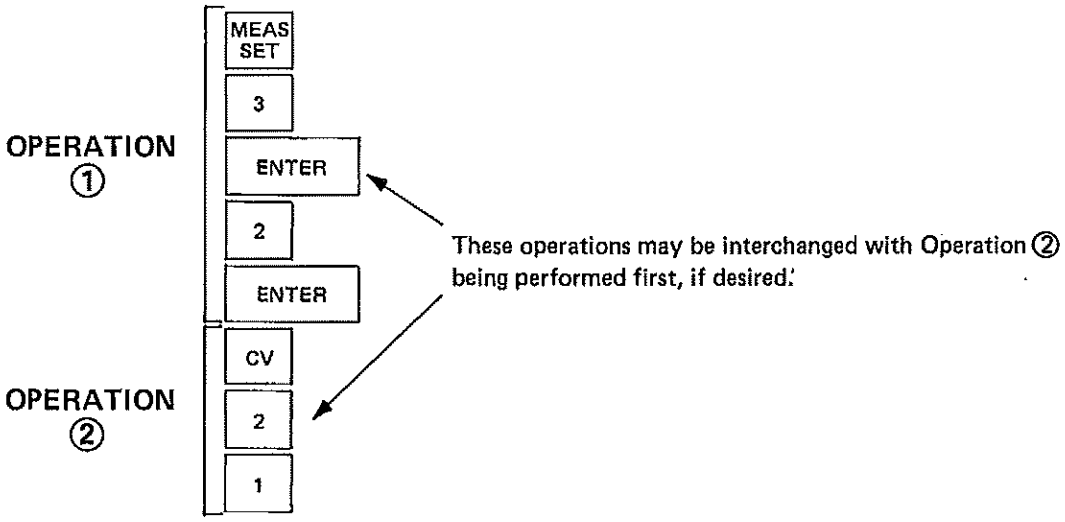
Data Revision

When data is written into a memory, it automatically erases data previously written there. This means that revisions are made simply by rewriting the data where it is desired.

When working with revisions, it is important to keep in mind the automatic address advancing feature (as shown on p. 50).

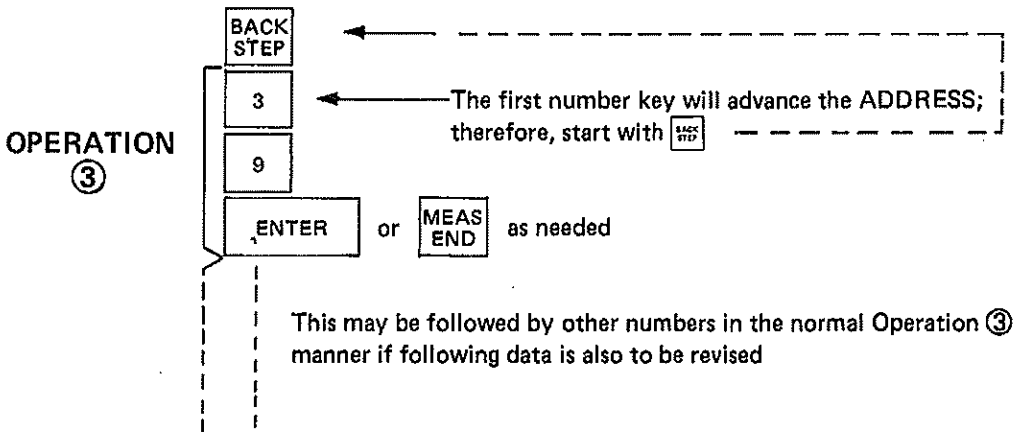
To check data contained at a given address in the memory, perform Operations 1 and 2. If the data shown by the KEY/MEM DATA display is to be revised, push [BACK STEP] and write the new data in the normal Operation 3 manner.

EXAMPLE: Check CV2 data (channel 1) at M3-2.



The above will show the CV2 data at M3-2; if revision is desired, continue as shown on the next page.

EXAMPLE: Revise data to read "39"



Revision of [MEAS END]

The purpose of the [MEAS END] button is to divide the program into musical measures for convenience. In a long composition, this makes the finding and checking of individual notes much easier. Also, the COPY and REPEAT RUN functions depend on these [MEAS END] flags for proper operation.

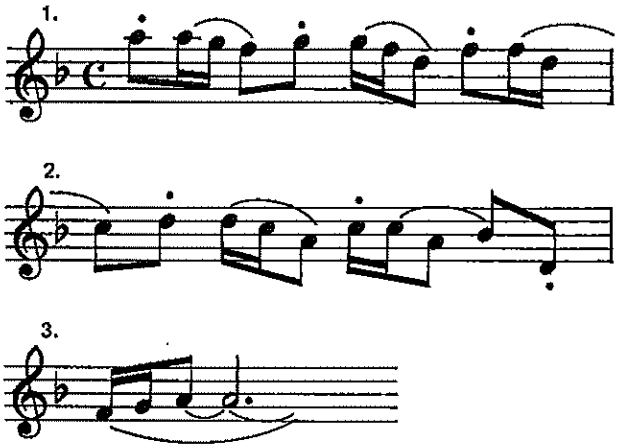
The insertion of [MEAS END] in the wrong place (or its omission) will affect the program only if the COPY function is involved, or if the program is run from the measure with the [MEAS END] mistake or from any measure which follows.

[MEAS END] is also required after the last step in the program. If [MEAS END] were not used after the last step, the MicroComposer would continue running to some undetermined point after the end of the program.

Mistakes in [MEAS END] placement may be pinpointed by simply stepping through the suspected portions of the program manually (with the [STEP] button) while comparing the ADDRESS display with the program sheet.

Mistakes in the placement of [MEAS END] may be corrected by simply rewriting the data just in front of the point to be corrected.

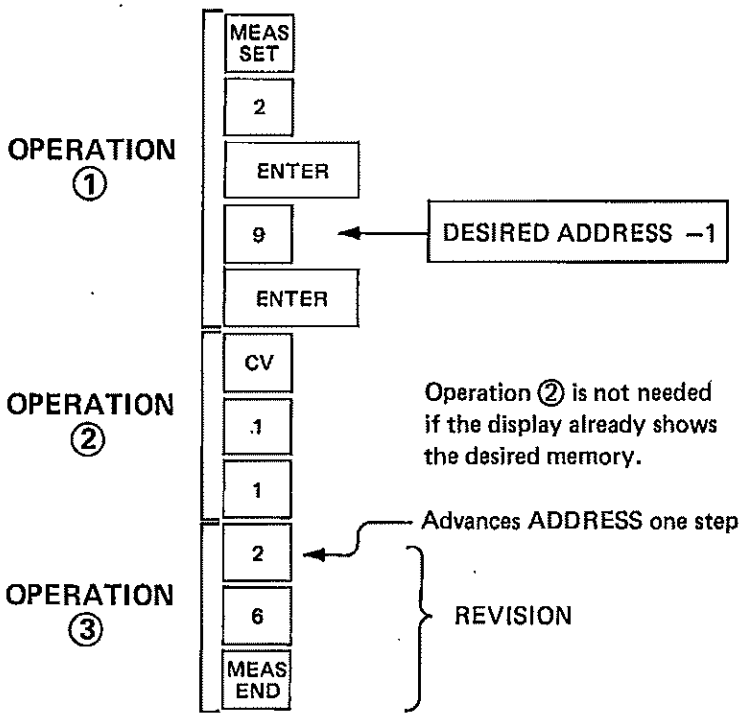
PROGRAM 24:



MM ♩ = 150 TIME BASE ♩ = 16

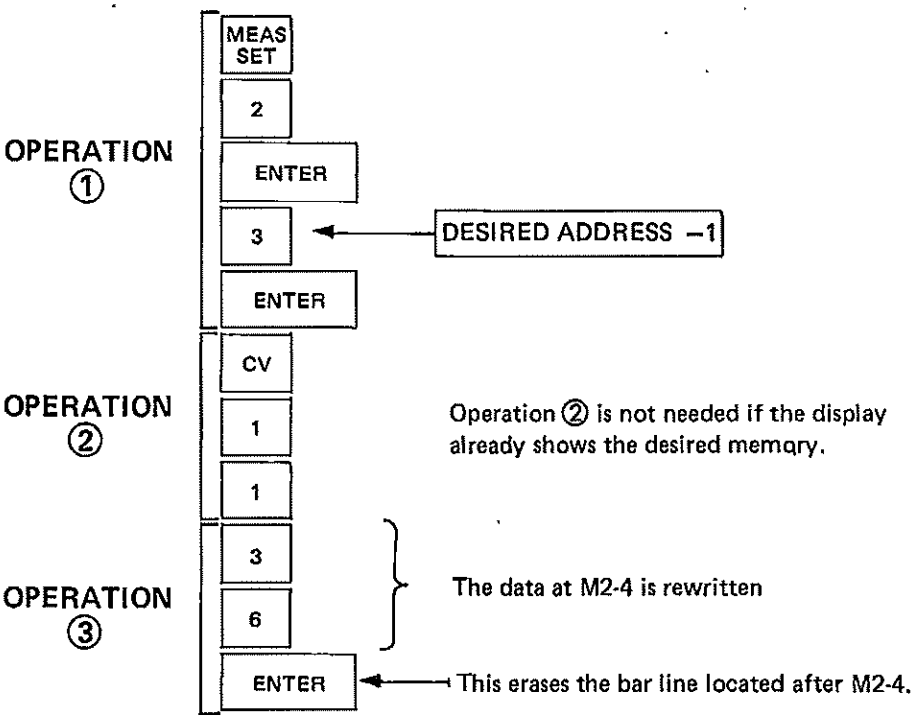
CHANNEL		1. TRUMPET						
MEASURE	STEP	1	S	G				
1	1	45	8	2				
	2	45	4	4				
	3	43	4	4				
	4	41	8	6				
	5	43	8	2				
	6	43	4	4				
	7	41	4	4				
	8	38	8	6				
	9	41	8	2				
	10	41	4	4				
	11	38	4	4				
2	1	36	8	6				
	2	38	8	2				
	3	38	4	4				
	4	36	4	4				
	5	33	8	6				
	6	36	4	2				
	7	36	4	4				
	8	33	8	8				
	9	34	8	6				
	10	26	8	2				
3	1	29	4	4				
	2	31	4	4				
	3	33	60	60				

EXAMPLE: Bar line omitted between M2-10 and M3-1 (PROGRAM 24)



In this case, the data at M2-10 has been rewritten followed by the correct MEAS END

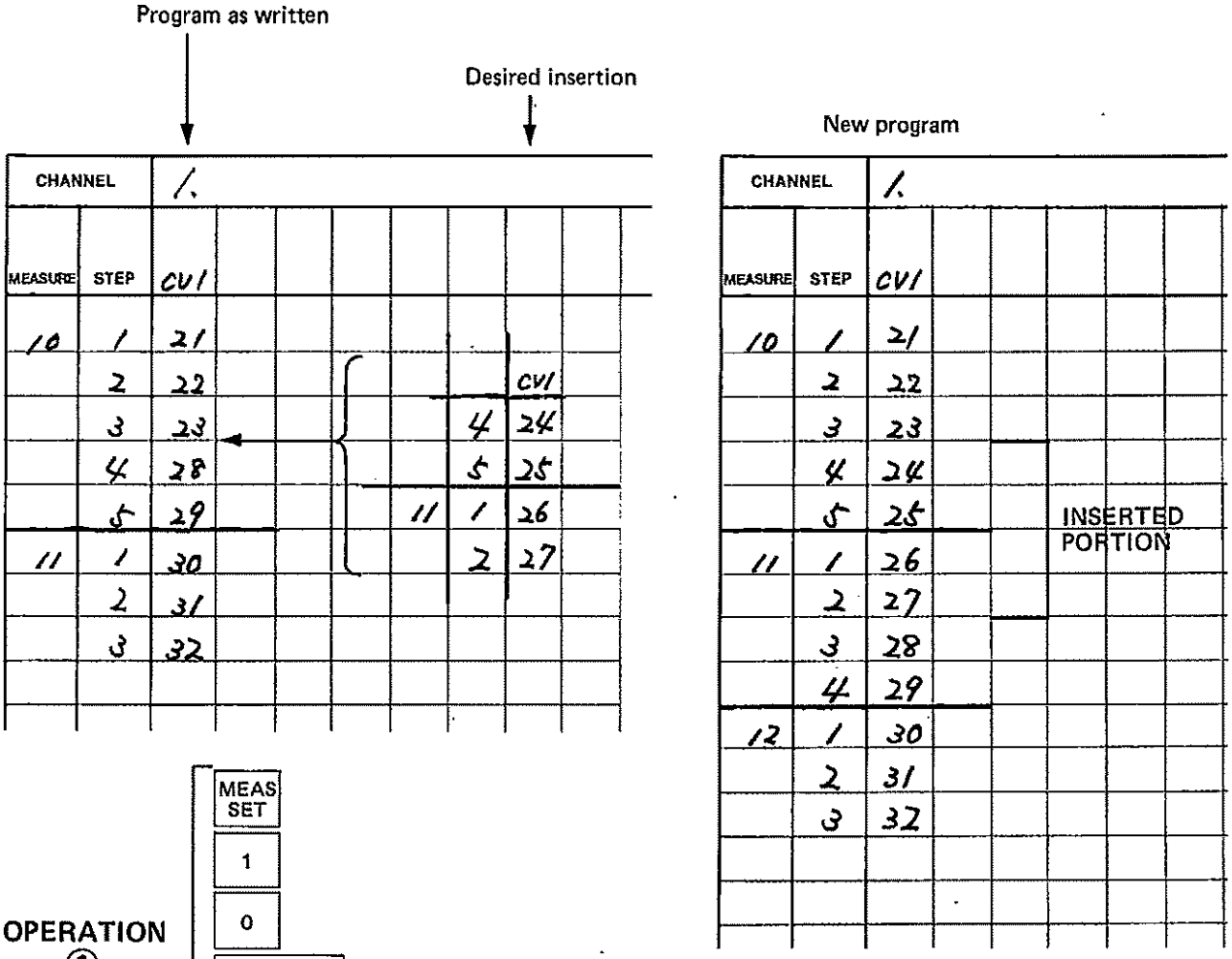
EXAMPLE: Bar line accidentally inserted between M2-4 and M2-5 (PROGRAM 24).



The INSERT and DELETE functions

Any number of notes may be inserted anywhere in the program without having to reload the program.

EXAMPLE: Make the following insertion.



OPERATION ①

MEAS SET → 1 → 0 → ENTER → 3 → ENTER

← Last address before insertion

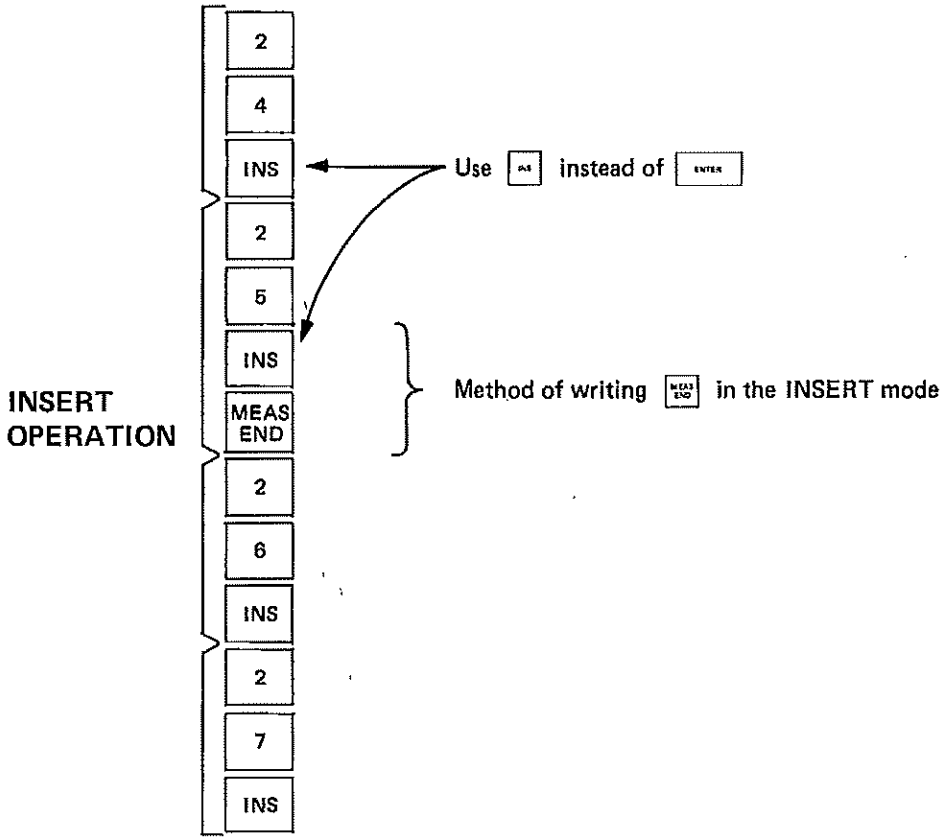
OPERATION ②

CV → 1 → 1

Operation ② is not needed if display already shows desired memory.

(continued on next page)

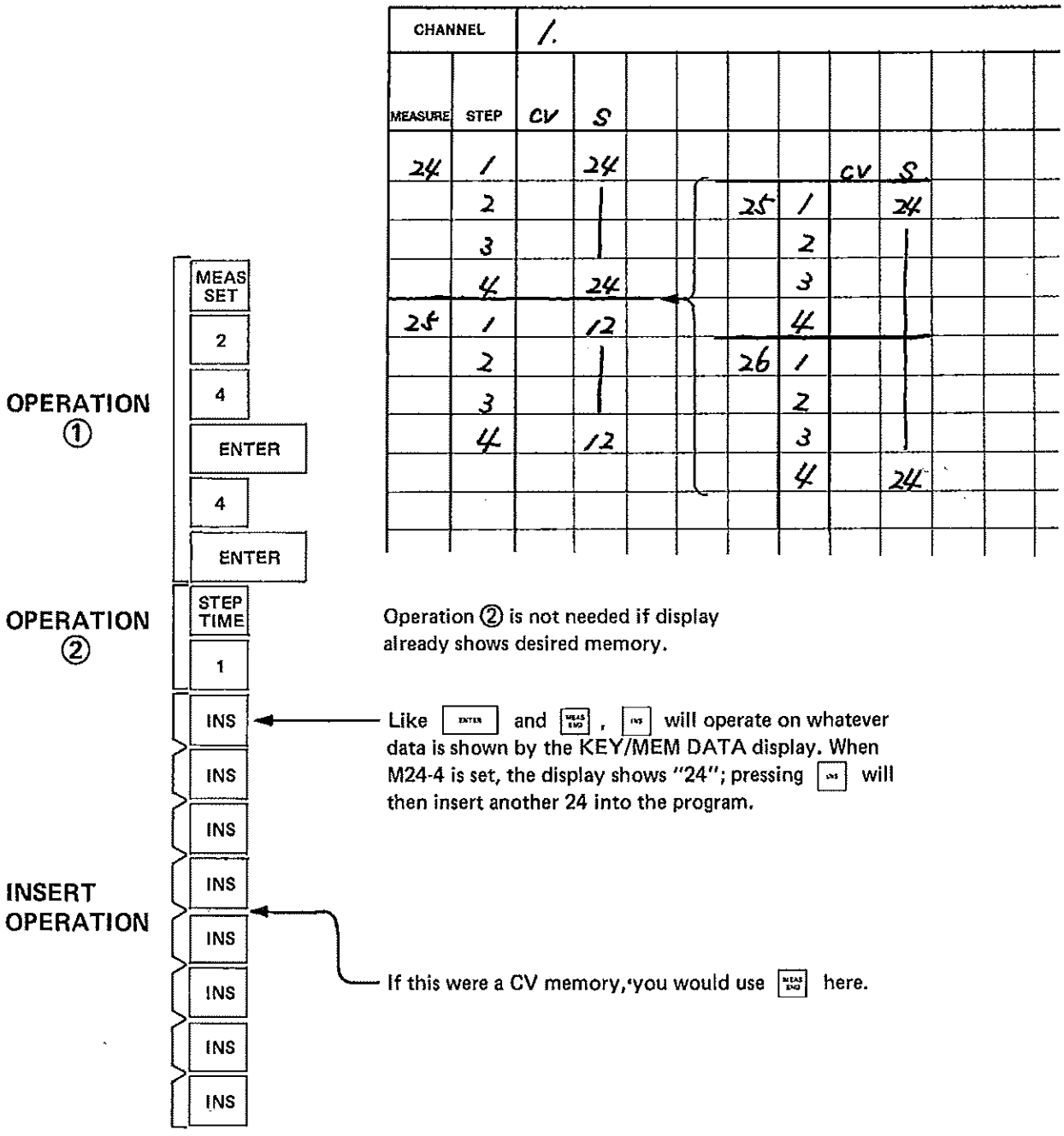
(continued from previous page)



[INS] = INSERT

The [INS] button may be used like the [ENTER] and [MEAS END] buttons when repeating data.

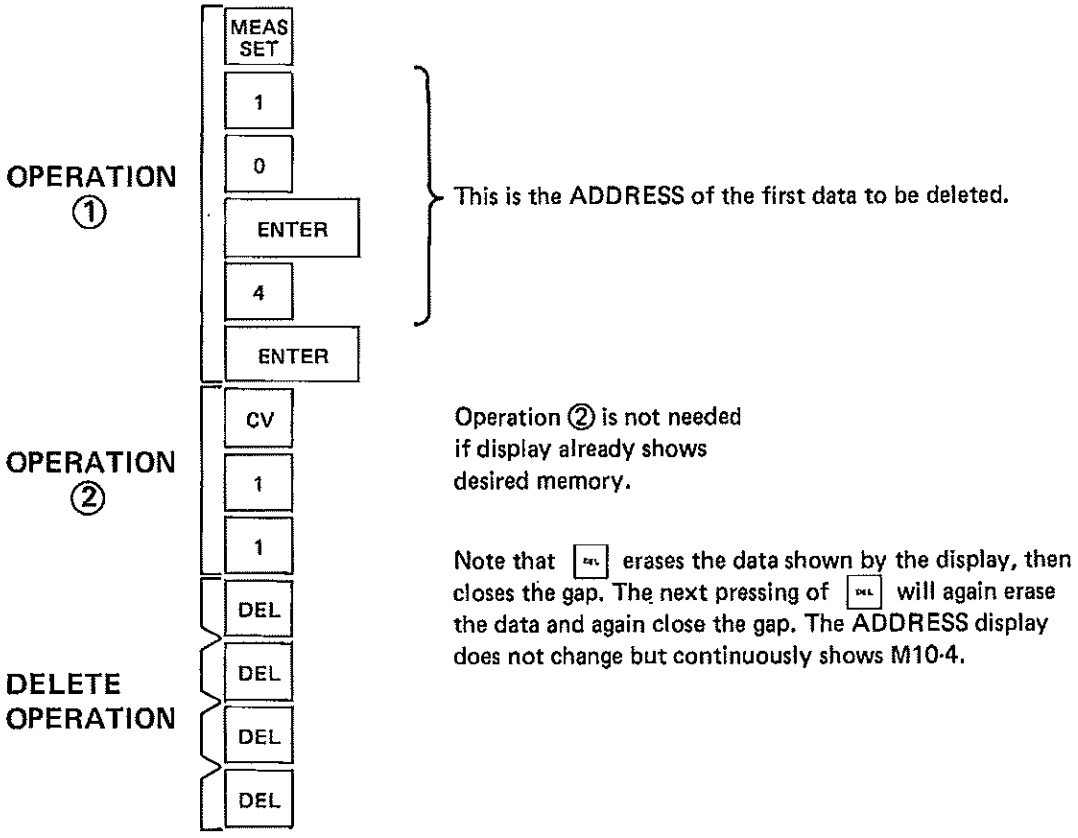
EXAMPLE: Make the following insertion



(Note that the COPY function discussed in Section 9 also operates in the INSERT mode).

Any number of notes may be deleted from the program with the space left by this operation being automatically closed up.

EXAMPLE: Delete the data which was inserted in the example on p. 84.



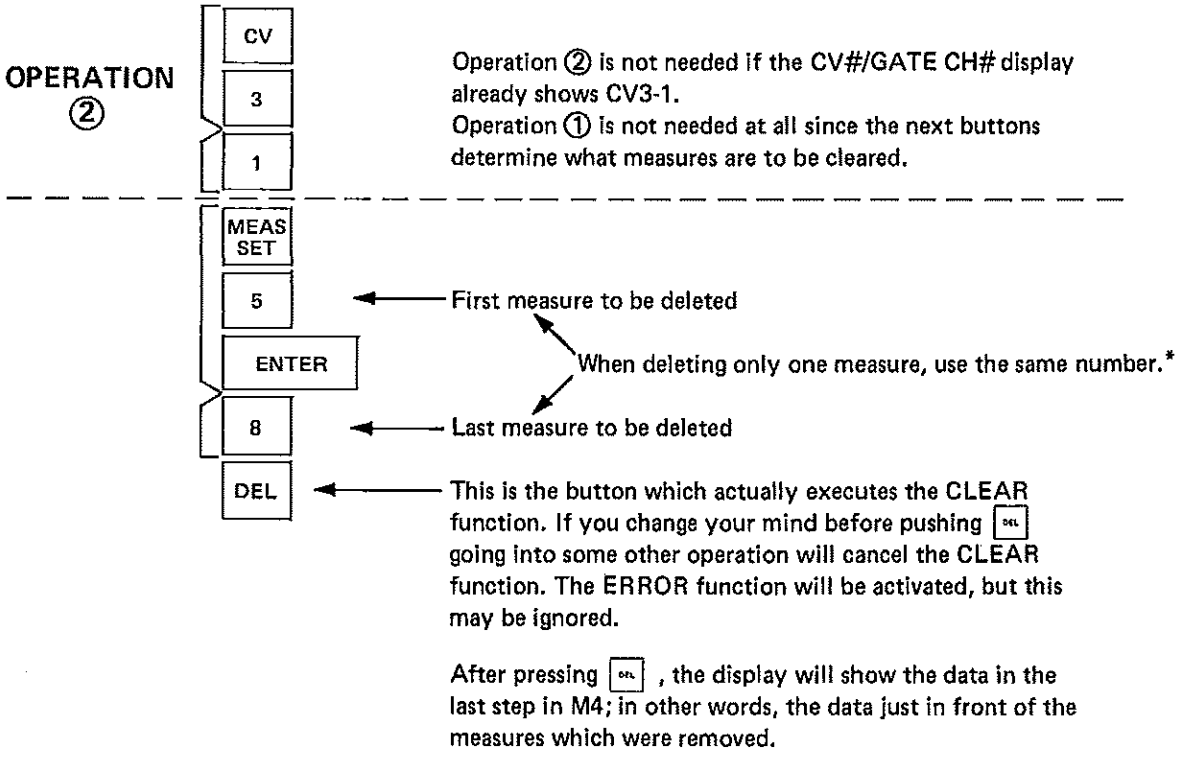
DEL = DELETE

The MEMORY CLEAR function

As mentioned before, to clear all of the MicroComposer memories of all data, turn off the POWER switch for a few seconds.

The MEMORY CLEAR function allows any given MicroComposer memory to be cleared of data from any point to any other point in the program.

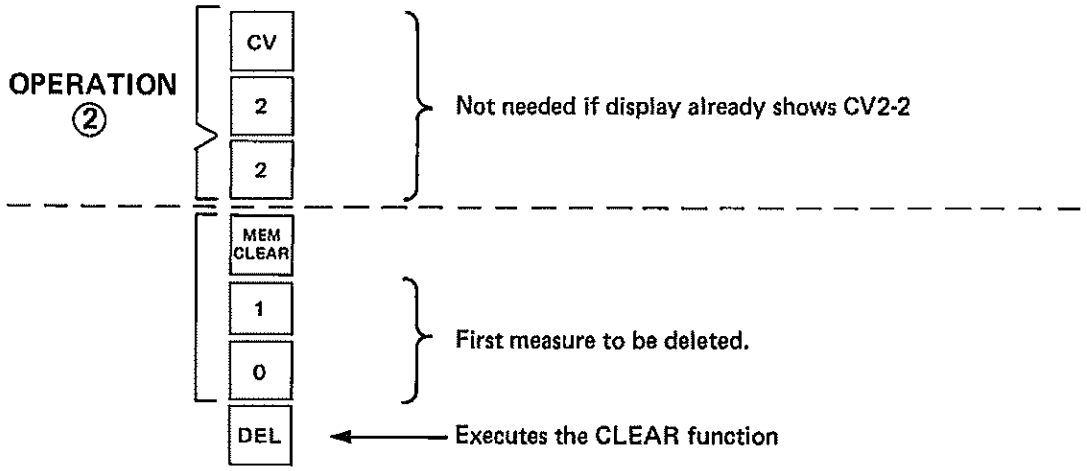
EXAMPLE: Clear CV3-1 from M5 to M8.



As the DEL button implies, the data after M8 is brought forward to close up the hole left in the program by the CLEAR function.

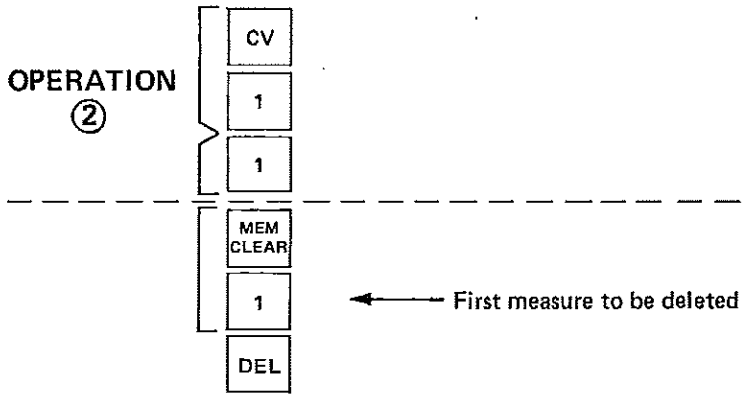
*(Same method as in COPY function example shown on p. 95)

EXAMPLE: Clear CV2-2 from M10 to end of program.



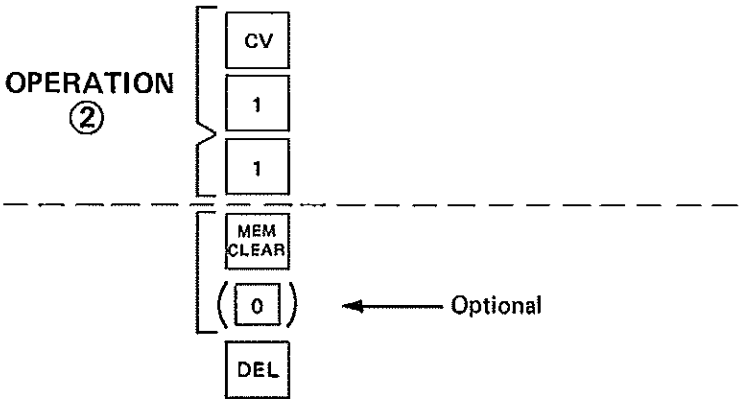
After pressing [DEL], the ADDRESS display will show the last step in M9; in other words, the display will show the last data now contained in CV2-2.

EXAMPLE: Clear all data from CV1-1



In the above, the assignment of CV1 to Channel 1 remains intact, but CV1 now contains no data. The ADDRESS display will show M0 (measure zero) and the KEY/MEM DATA display will show "0". Remember that the MicroComposer will not run in this condition; data must be reloaded into CV1, or the channel assignment must be cleared.

EXAMPLE: Clear all data from CV1-1 along with the channel assignment



The above will activate the ERROR function indicating that the memory has been cleared of all data and the channel assignment cancelled.

The other MicroComposer memories are cleared in the same way as shown above.

CAUTION:

When loading data into the MicroComposer, one of the assignable memories (CV, MPX, or TEMPO) must be assigned to a channel and loaded before the MicroComposer will accept data for the related STEP or GATE TIME memory. When clearing memories, the opposite is true.

If all the assignable memories for a given channel are cleared first, you no longer have access to the STEP/GATE memories. Trying to select these STEP/GATE memories will only activate the ERROR function. If the assignable memories are cleared but their assignment left intact, you can select the related STEP/GATE memories, but you still will not have access to the data. Also, the MicroComposer will not run.

If you find that you have inadvertently cleared all the assignable memories, the only way to get access to the related STEP/GATE memories is to temporarily reassign a memory to that channel and load data into it up to the point where access is desired. For example, if you want to clear STEP TIME completely (in other words from M1 to the end of the program), you need access to M1-1. Once the assignable memory has data at M1-1, you have access to the STEP/GATE M1-1 and can clear them.

9. Repetition

Repeated Notes

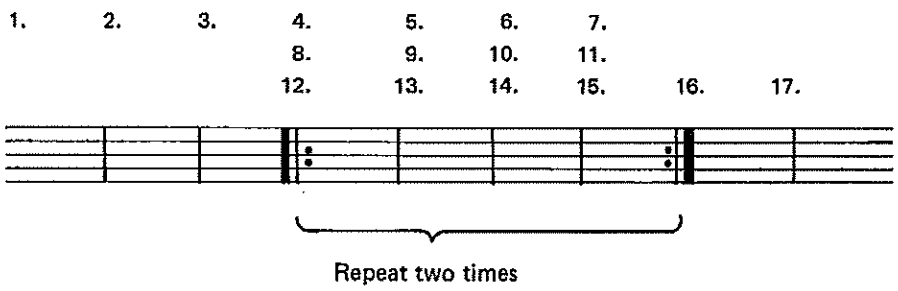
Repeated data within a given measure can be handled as shown in Section 5 using the [ENTER] or [MEAS END] buttons.

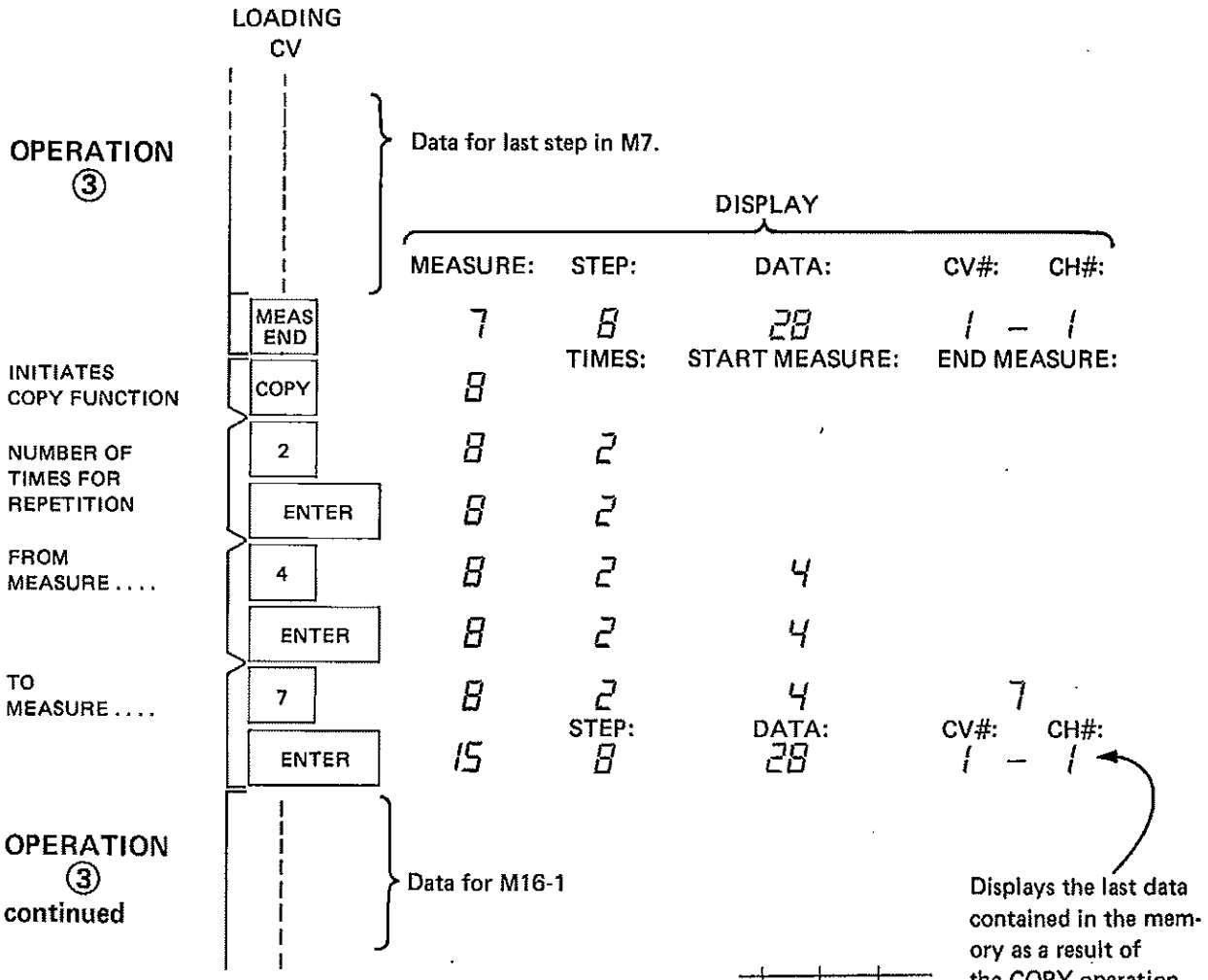
COPY Function

Repetitions of musical patterns may be easily accomplished with the COPY function.

There are two types of COPY function. The first involves the transfer of data from one place to another in the same memory. This operation is shown on the next page and in Program 25 (p. 96).

COPY function (same memory)



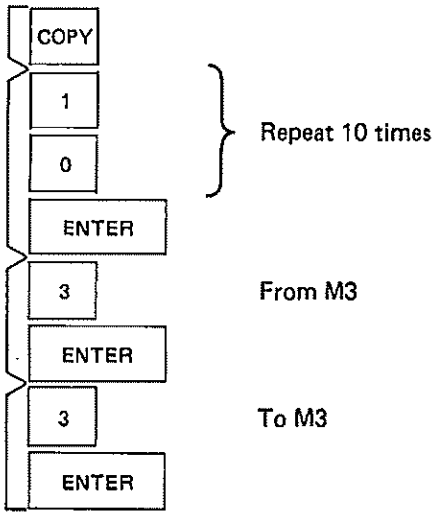


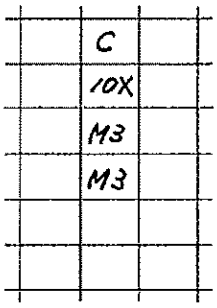
On the program sheet the COPY function can be written →

	C	
	2X	
	M4	
	M7	

To repeat only one given measure a number of times:

EXAMPLE: Repeat M3 ten times





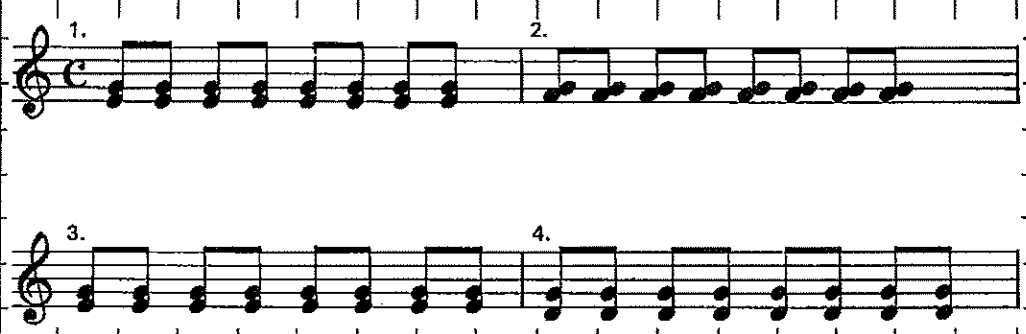
PROGRAM 25

COPY FUNCTION

MM ♩ = 150 TIME BASE ♩ = 32

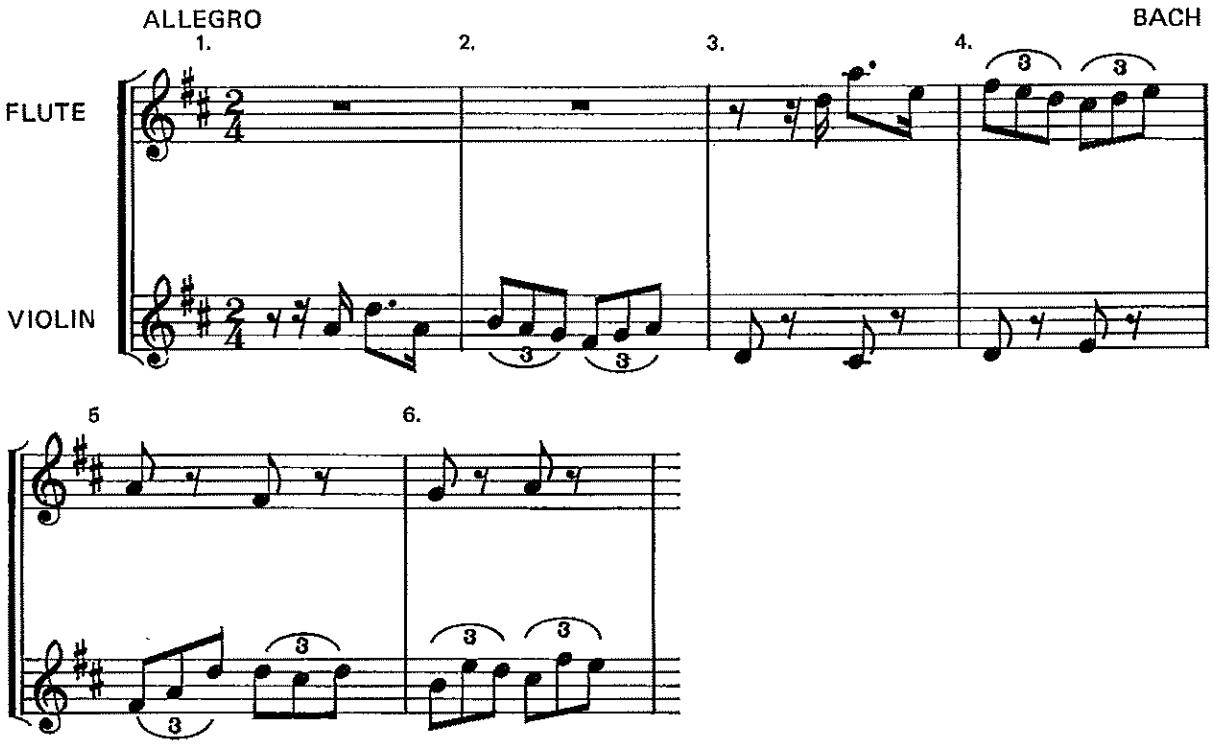
CHANNEL		1.			
MEASURE	STEP	VCO 1	VCO 2	S	G
1	1	31	28	16	6
	2				
	3				
	4				
	5				
	6				
	7				
	8	31	28	16	6
2	1	C	29		
	2	3X		C	
	3	MI		3X	
	4	MI		MI	
	5			MI	
	6				
	7				
	8		29		
3	1		C		
	2		1X		
	3		MI		
	4		MI		
	5				
	6				
	7				
	8				
4	1		26		
	2				
	3				
	4				
	5				
	6				
	7				
	8		26		

NOTE:
In actual practice the use of the COPY function here is questionable since it would be just as easy to load the "28" in the normal manner for this measure.



The second type of COPY function is the transfer of data from one memory to another. This operation is shown in Program 26.

PROGRAM 26:



PROGRAM 26

BRANDENBURGH CONCERTO

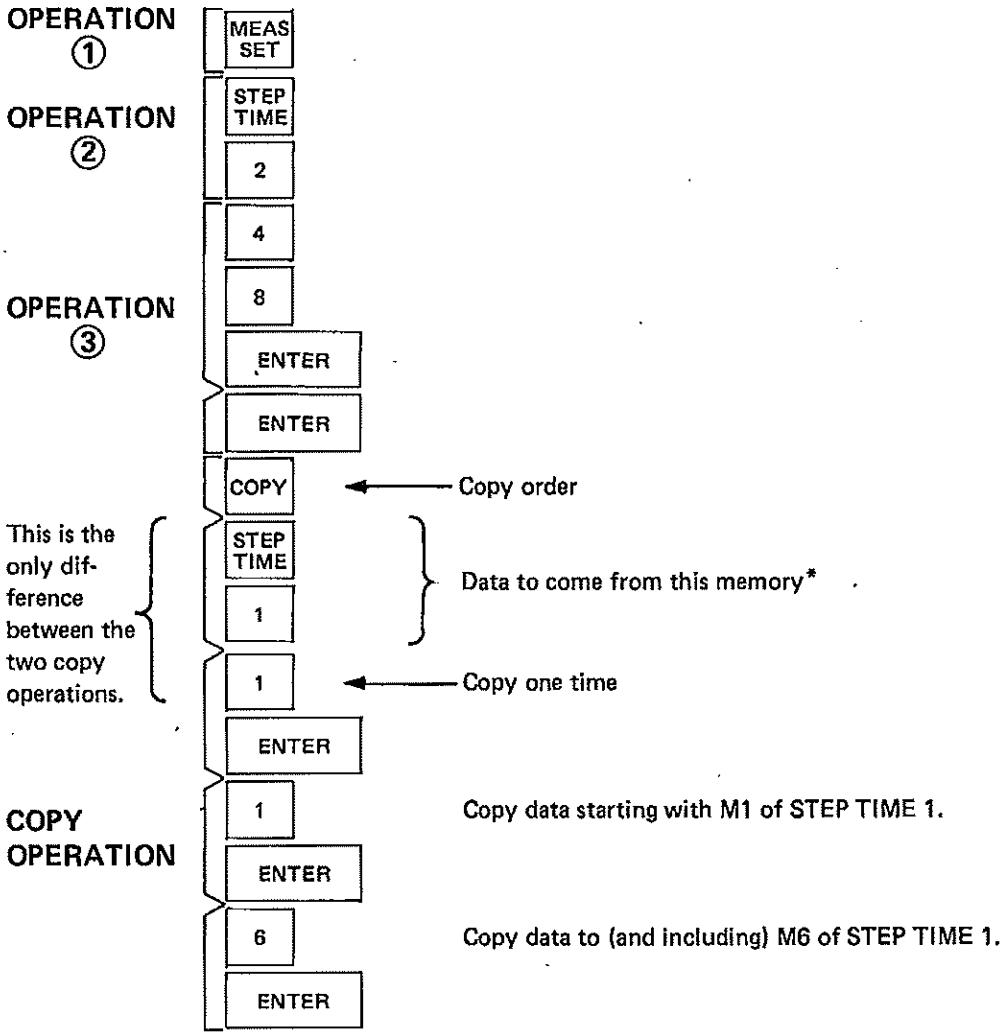
MM ♩ = 120 TIME BASE ♩ = 24

CHANNEL		1. VIOLIN							4. FLUTE						
MEASURE	STEP	VCO 1	S	G	VCA 2				3	S	G	4			
1	1	33	18	0	0				38	48	0	0			
	2	33	6	4	72										
	3	38	18	14	80										
	4	33	6	4	72										
2	1	35	8	7	80				38	48	0	0			
	2	33	|	7	78										
	3	31	|	6	76										
	4	30	|	7	78										
	5	31	|	7	76										
	6	33	8	6	74										
3	1	26	24	8	76				38	C	C	C CV			
	2	25	24	8	70				38	G1	G1	2-1 1-X M1 M2			
	3								45	1X					
	4								40	M1	M2				
4	1	26	24	8	71				42						
	2	28	24	8	72				40						
	3								38						
	4								37						
	5								38						
	6								40						
5	1	30	|	|	74				33			80			
	2	33	C	2X	73				30			72			
	3	38	M2	M2	72										
	4	38			76										
	5	37			75										
	6	38			72										
6	1	35			78				31			74			
	2	40			74				33			78			
	3	38			72										
	4	37			80										
	5	42			78										
	6	40			76										

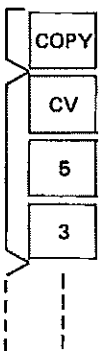
This operation is shown on the opposite page.

COPY function (from memory to memory)

EXAMPLE: Load STEP TIME 2 for Program 26.



*When copying from a CV memory, include the CV channel assignment. In the example at the right, the copied data is to come from CV5 which is assigned to Channel 3.



Data in a CV memory cannot be transfered to a STEP or GATE TIME memory because the CV memory contains the [MEAS END] flags.

Data may be transfered between STEP and GATE TIME memories if the following is kept in mind:

The capacity of the STEP and GATE TIME memories is 256 increments per step. They are numbered:

STEP TIME: 1 to 256
GATE TIME: 0 to 255

The internal circuits of the MicroComposer actually uses only one numbering system for both of these memories: 0 to 255. Thus, when a "1" is programmed into STEP TIME, the internal circuits are actually remembering the number "0". For this reason, if data is transfered from STEP TIME to GATE TIME, the data will become one less in value.

EXAMPLE: STEP TIME = 24;
copied to GATE TIME it becomes: 23

Also, for the same reason, data transfered from GATE TIME to STEP TIME will have a "1" added to it.

EXAMPLE: GATE TIME = 15;
copied to STEP TIME it becomes: 16

In a legato passage where you might want the GATE TIME to be the same as STEP TIME, instead of trying to use the COPY function, write some arbitrary large number into GATE TIME as shown in Program 8 (page 24).

Once an assignable memory has been assigned and loaded with data, this channel assignment cannot be changed without first erasing all the data. To get around this, use the COPY function. For example, CV2 has been assigned to Channel 3 but now it is desired to have CV2 assigned to Channel 1. This can be done as follows: Copy the CV2 data into an unused CV memory; clear CV2 of its data and channel assignment; reassign CV2 to Channel 1; copy the data back into CV2; clear the CV used for temporary storage.

Tempo and multiplex channel assignments can be changed in exactly the same way. Copy the tempo or multiplex data into an unused CV memory, change the channel assignment, then return the data. This is possible because the CV, tempo and multiplex memories use exactly the same data storage system even though it appears different to us on the front panel displays and program sheets.

If you have no spare channel for temporary storage, the data could be temporarily copied onto the tail end of an already occupied channel.

The maximum number of times for the COPY function repetition is 99 times. To copy data more than 99 times, use the COPY function more than once. For example, to copy 150 times, first copy 99 times, then copy again 51 times.

Sometimes, even with measures which are not exactly the same, it may still prove useful to use the COPY function rather than writing the data out. For example, in a long measure where only a few notes are different from some previous measure it may prove easier to copy the measure, then go back and change the data which is different.

The [MEAS END] flags in the CV memories determine the boundaries of the measures. The STEP TIME and GATE TIME memories do not contain [MEAS END] flags (even if [MEAS END] was used in loading them). This means that it is possible to copy STEP and GATE data into following measures even though they are of uneven length.

In Program 27, the COPY order will cause the four 2's in M1 to be written four times, filling the program to the point shown. The MicroComposer depends on the [MEAS END] flags for orientation in the program, and because of the "C4X" order, will jump to the last step in M5 (four measures away from M1). Since M5-6 contains no data, the ERROR function will be activated. To defeat this, once the COPY operation has been carried out, merely [BACK STEP] (to M5-2 in this example) until the ERROR function stops, then continue loading data from there.

PROGRAM 27:

MM ♩ = TIME BASE ♩ =

CHANNEL		1.		
		VCO		
MEASURE	STEP	1	S	G
1	1			2
	2			2
	3			2
	4			2
2	1			C
	2			4X
3	1			M1
	2			M1
	3			
4	1			
	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			
5	1			
	2			
	3			2
	4			2
	5			2
	6			2

Using the COPY function as shown will load a "2" into all of these steps.

In Program 28, the [MEAS END] flags will cause the MicroComposer to jump ahead to M5-4 because of the "C 4X" order, but in this case there will be an overflow of data so the ERROR function will not be activated. In this case, push [STEP] until the ERROR function is activated indicating that you have reached the first step with no data, then [BACK STEP] once and continue loading from there. (Or you could use the measure set operation to bring you up to the correct place).

PROGRAM 28:

MM ♩ = TIME BASE ♩ =

CHANNEL		1.		
MEASURE	STEP	VCO 1	S	G
1	1			2
	2			2
	3			2
	4			2
2	1			C
	2			4X
3	1			M1
	2			M1
4	1			
	2			
5	1			
	2			
	3			
	4			
6	1			
	2			
7	1			
	2			
8	1			
	2			
	3			2
	4			2

As before, the COPY function will load a "2" into these steps.

An important point to keep in mind is that:

> The COPY function operates in the INSERT mode (see Section 8, pp. 84-86).

In this way, inserting new repeat measures into a composition to fill a given time space (such as in commercial work) becomes quite simple.

In the normal loading operation, writing data into a memory slot will erase any data which existed there previously; this is not true of the COPY function. If a mistake is made with the COPY function and you try to do the COPY operation over again at the same place, the new COPY data will be inserted in front of the previous COPY data. The way to avoid this is to use the MEMORY CLEAR function (see Section 8, pp. 88-91) before doing the corrected COPY operation.

Canceling the COPY Function

The following shows how you can change your mind or correct mistakes made when writing the COPY order.

If you change your mind immediately after pressing [COPY], simply push [KEY CLEAR]. This will activate the ERROR function which may be ignored. (To stop the flashing display, push [STOP]).

If you make a mistake with either the number of repetitions or one of the measure numbers, but have not yet pushed the related [ENTER], use [KEY CLEAR].

If you have pushed [ENTER] after the END MEASURE (the last [ENTER] when writing the COPY order), the COPY order has been completed. If data has been transfered, then it must be removed with the MEMORY CLEAR function (Section 8), then the COPY order rewritten.

If you have pushed [ENTER] at some other point, the COPY order has been partially written but the order has not been carried out; it must be canceled and rewritten from the beginning. This can be done by pressing [BACK STEP]. The ERROR function will be activated, but this can be ignored; press [COPY] and start again.

The same procedures work for the COPY function used for copying data from another memory.

CAUTION: If you cancel the COPY order after selecting the memory from which the data is to be copied, this memory selection remains and the MicroComposer continues to monitor this memory after the cancelation of COPY rather than the memory into which the data was to be loaded. You must re-select the original memory before doing anything else.

10. Synchronous Recording

The SYNC function allows for perfect synchronization of separate programs in multichannel recording.

Normally, the sync signal is recorded on a separate track first, then the sync track is used to drive the MicroComposer during the recording of the programs on the other tracks.

Note that with the SYNC switch depressed (LED on), the MicroComposer will not run when there is no signal present at the SYNC IN jack.

Recording the Sync Signal

The sync carrier signal always appears at the SYNC OUT jack on the rear panel of the Micro-Composer when the SYNC switch is in the raised position (LED off). When the START button is pressed, the sync carrier is modulated by the sync data.

Patch the SYNC OUT jack to the LINE INPUT of the desired tape channel and set the RECORD level. A safe value to start with is –3VU. With a good recorder and good tape, you will probably be able to lower this to –10VU (or more) to help reduce crosstalk; a little experimentation will show if this is possible. (The output at the SYNC OUT jack is approximately 0dBm).

When the RECORD level is set, put the recorder in the RECORD mode and let it run for about five to ten seconds so as to record some of the unmodulated carrier, then push the MicroComposer START button to record the sync. When the program ends, wait until a few seconds after the TIMER display stops running before stopping the recorder.

The TEMPO knob on the front panel of the MicroComposer can be used to manually change the tempo of the music while the program runs. These tempo changes will also change the modulation rate of the sync carrier and, therefore, all following programs will exactly follow these tempo changes.

Recording the Programs

Patch the LINE OUT of the channel used for the sync signal to the MicroComposer SYNC IN jack on the rear panel. The output level for that channel may usually be left in the calibrated position (in other words, the VU meter will read the same level in PLAY as was used for recording the sync). (The SYNC IN jack will accept almost any level starting at about −10dBm).

Use the sel sync mode for the sync track the same as you normally use for overdubbing. In this way, music and sync remain synchronized and can be bounced to other tracks or tapes, if desired. Also, during the final mixdown, one final program can be added to the mix.

Set the tape at a point a few seconds before the start of the recorded sync carrier. Start the tape transport (punch in the RECORD mode whenever desired) and watch the sync channel VU meter. When the meter jumps up showing the presence of the sync carrier, push the MicroComposer's SYNC switch in, then push [START]. The program will begin running when the modulation of the carrier begins.

Record all subsequent programs in the same manner.

If the music is to use programmed variable tempo, the program used during the recording of the sync track must contain the tempo data. Once the sync has been recorded, the tempo data is not needed. In fact, both TIME BASE and TEMPO (fixed or variable) may be omitted from the MicroComposer programs when the MicroComposer is being driven from an external sync source.

The above is the normal procedure for multichannel recording.

When the [START] button is pushed, the program begins to run and after a very slight delay, the modulation of the sync carrier follows. When the recorded sync is used to drive the MicroComposer, there will again be a very slight delay before the first pulse modulations produce the first note in the program. If the first program were recorded at the same time as the sync signal, this combined delay in some cases could cause a perceptible hesitation between the entry of the first program and the following programs. When the sync is recorded first, all programs receive the same delay and are therefore in perfect sync.

"False" Starts

Occasionally, due to circuit noise, pushing the SYNC switch in will cause the synthesizer to produce sound. As long as this sound has not been recorded on tape, there is no problem (a good reason for punching in the record mode after pushing SYNC), The next step, pushing [START], will always cause the program to run from the beginning.

If the GATE OUTPUTS are latched "on" (the synthesizer continues to produce sound), push [STOP], raise the SYNC switch and start the tape from the beginning again.

11. Multiplex

The multiplex (MPX) memory is an assignable memory. This means that it requires assignment to a STEP TIME channel. If it is the only memory assigned to a given channel, then, like the CV memories, it must be loaded first before the related STEP TIME memory will accept data. Also, like the CV memories, MEAS END must be used to designate the bar lines in the program.

The STEP TIME determines the timing between the multiplex pulses. This means that the minimum required to produce multiplex pulses would be MPX data and STEP TIME data. If, in addition to the multiplex pulses, a gate pulse is also desired, GATE TIME data for that channel can be added. Of course, CV memories can be assigned to the same channel for the control of pitches or any other voltage controlled fuction along with the multiplex pulses.

The STEP TIME determines the timing of the multiplex pulses. Any number of the available six bits may be programmed to appear at the respective output jacks for each step in the program. When a bit is programmed to occur at a given step, the output will be a +15 volt pulse whose time duration will be exactly the same as STEP TIME for that step.

A few practical examples will help to clarify what multiplex is and how it can be used.

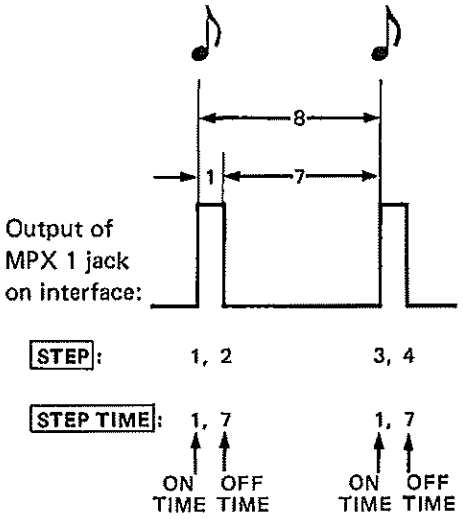
Rhythmic Patterns

One of the uses of the multiplex outputs is the generation of multiple pulse trains for the triggering of percussion voices.

Since the multiplex pulse is exactly the same length as STEP TIME, we have to give each note two steps in the program: one for "gate on" time and one for "gate off" time. The total of these two steps would equal the desired time value for the note. With percussion voices usually a short pulse is enough to trigger the envelope, thus we have decided to use a STEP TIME of "1" for the "gate on" time in Program 29.

PROGRAM 29:

TIME BASE ♩ = 16



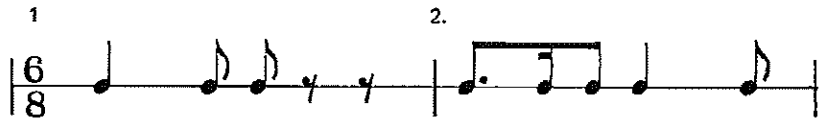
MM ♩ = 102 TIME BASE ♩ = 16

CHANNEL		1					
MEASURE	STEP	MPX	S				
1	1	1	1				
	2	0	7				
	3	1	1				
	4	0	7				

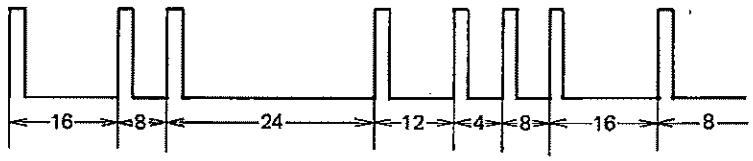
ON OFF

PROGRAM 30:

TIME BASE ♩ = 16



STEP TIME: 16 8 24 12 4 8 16 8

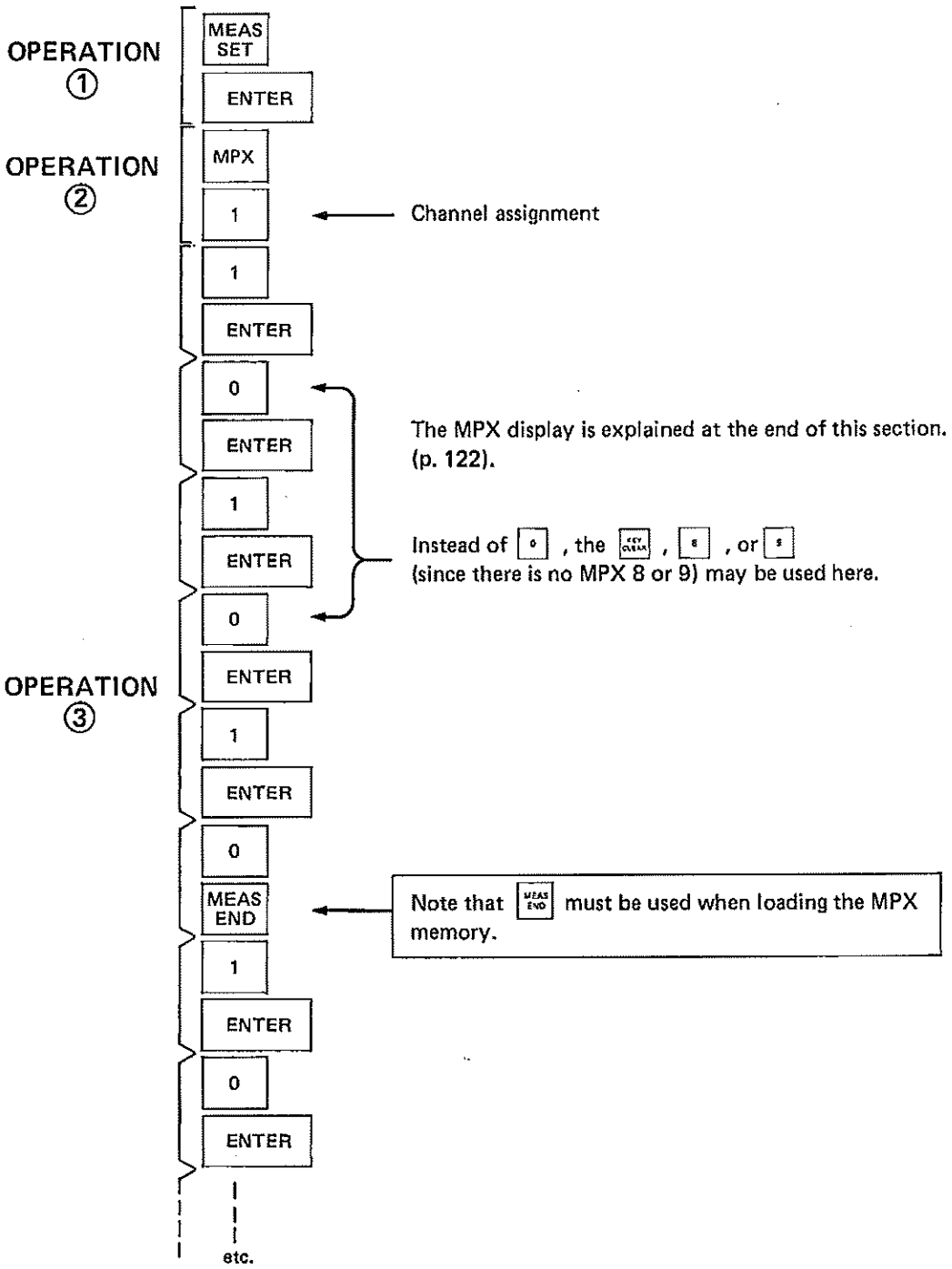


MM ♩ = 120 TIME BASE ♩ = 16

CHANNEL		1.						
MEASURE	STEP	MPX	S					
1	1	1	1					
	2	0	15					
	3	1	1					
	4	0	7					
	5	1	1					
	6	0	23					
2	1	1	1					
	2	0	11					
	3	1	1					
	4	0	3					
	5	1	1					
	6	0	7					
	7	1	1					
	8	0	15					
	9	1	1					
	10	0	7					

Patch the MPX 1 output on the Interface to the GATE input of the envelope generator.

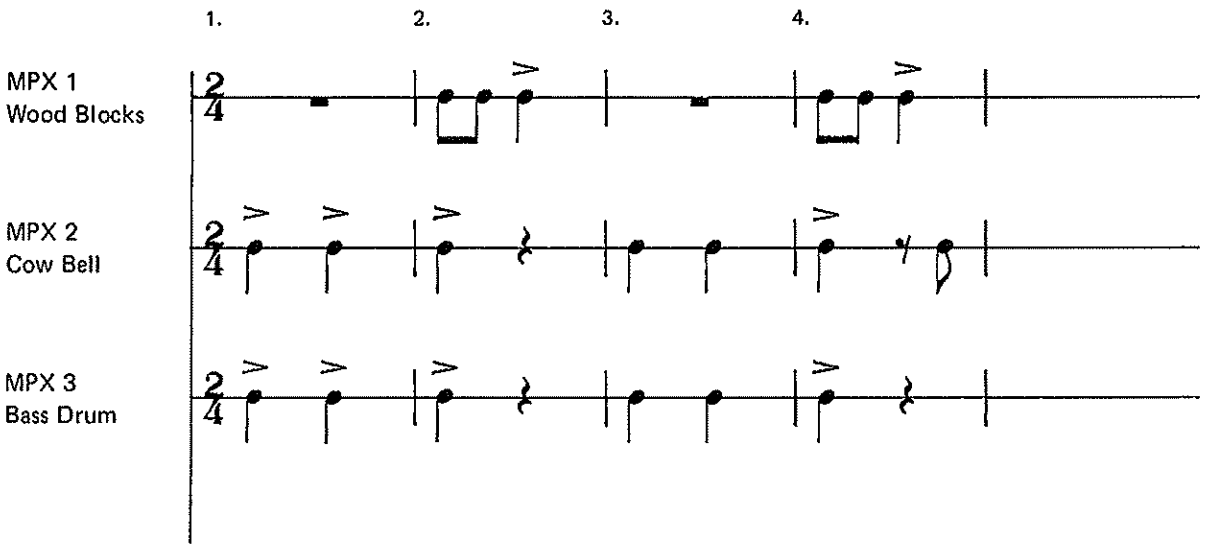
EXAMPLE: Load MPX data for Program 30.



Up to this point the use of the multiplex over the standard gate output would seem to be a disadvantage since the multiplex requires twice as many steps for the same number of notes. By using multiplex, however, six completely independent rhythm voices can be programmed at the same time using only two memories (one channel): MPX and STEP TIME.

The first step would be to determine the overall rhythm of the passage so that you know how many steps are needed in each measure.

PROGRAM 31:



TIME BASE ♩ = 24

MEASURE:	1.				2.				3.				4.			
STEP ON:	1		3		1	3	5		1		3		1	3	5	7
OFF:		2		4	2	4	6		2		4		2	4	6	8
OVERALL RHYTHM:																
Time value:	24		24		12	12	24		24		24		12	12	12	12
STEP TIME ON:	1		1		1	1	1		1		1		1	1	1	1
OFF:	23		23		11	11	23		23		23		11	11	11	11

Each rhythm voice is assigned to one of the multiplex outputs; each step in the program is then programmed to contain the multiplex pulses for the rhythm voices which are to sound at that point in the program.

PROGRAM 31:

MM ♩ = 100 TIME BASE ♩ = 24

CHANNEL		1. PERCUSSION							
MEASURE	STEP	MPX			CV3	CV4	S		
1	1		2	3	0	90	1		
	2						23		
	3		2	3			1		
	4				0		23		
2	1	1	2	3	75		1		
	2				75		11		
	3	1			72		1		
	4				72		11		
	5	1			80		1		
	6					90	23		
3	1		2	3		75	1		
	2						23		
	3		2	3			1		
	4				80	75	23		
4	1	1	2	3	75	90	1		
	2				75		11		
	3	1			72		1		
	4				72		11		
	5	1			80		1		
	6					90	11		
	7		2			70	1		
	8				80	70	11		

MPX 1 = Wood blocks
MPX 2 = Cow bell
MPX 3 = Bass drum

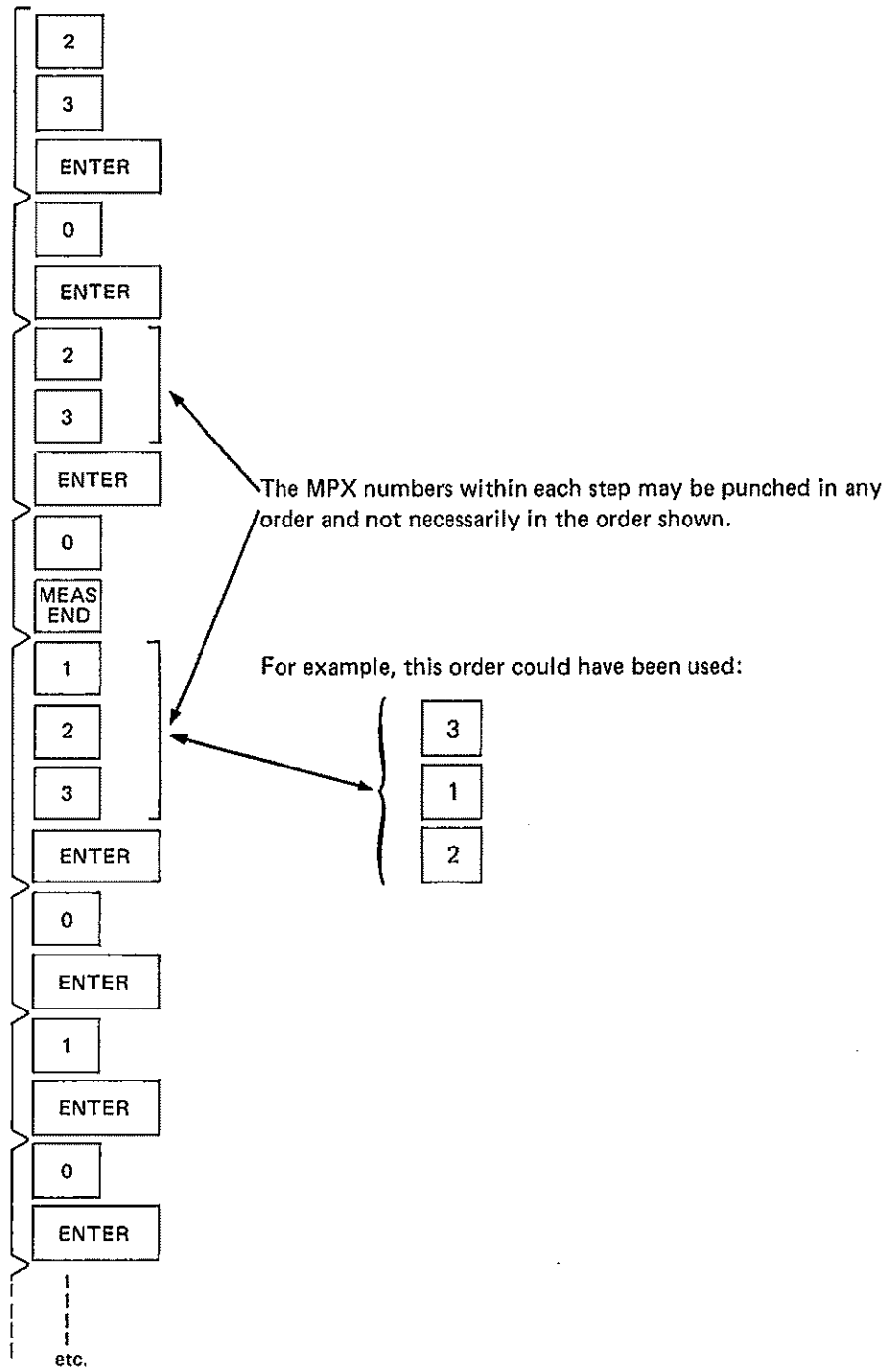
Dynamics for wood blocks

Dynamics for cow bell and bass drum

EXAMPLE: Load MPX data for Program 31.

Perform Operations (1) and (2) exactly as shown in Program 30. (p. 113)

OPERATION (3)



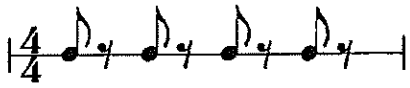
PROGRAM 32:

Mambo :

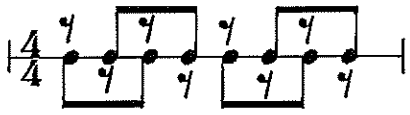
MPX 1: GUIRO

MPX 2: MARACAS

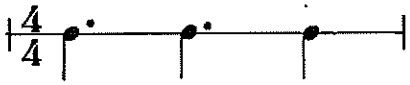
MPX 3: COW BELL



MPX 4: HIGH BONGO
MPX 5: LOW BONGO



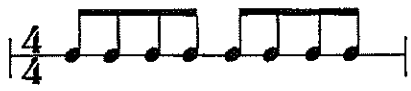
MPX 6: BASS DRUM



MEASURE: 1.

STEP: 1 2 3 4 5 6 7 8

OVERALL RHYTHMIC PATTERN

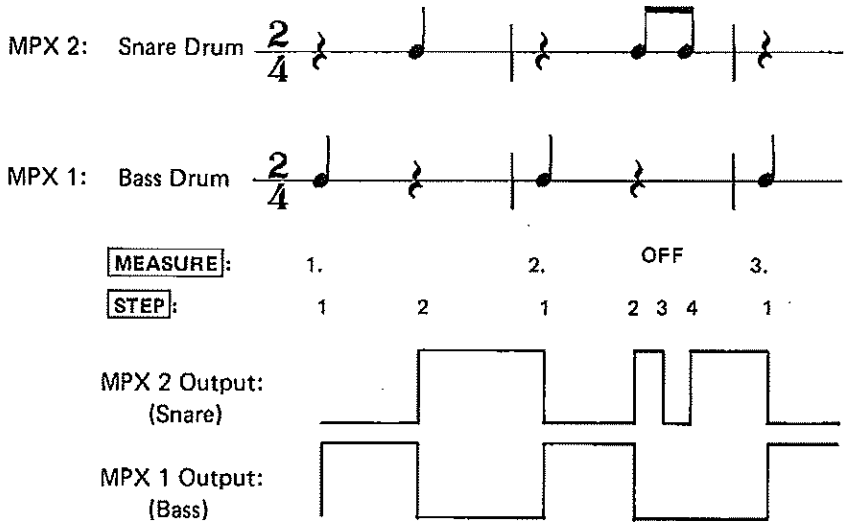


MM ♩ = 100 TIME BASE ♩ = 32

CHANNEL		1. MPX							
MEASURE	STEP	1	2	3	4	5	6	S	
1	1	1	2	3		5	6	1	
	2							15	
	3		2		4			1	
	4							15	
	5	1	2	3		5		1	
	6							15	
	7	1	2		4		6	1	
	8							15	
	9	1	2	3		5		1	
	10							15	
	11		2		4			1	
	12							15	
	13	1	2	3		5	6	1	
	14							15	
	15	1	2		4			1	
	16							15	

Many times, for percussion voices, the length of the pulse used to trigger the envelope generator is not important. This fact can be used to simplify some of the percussion portion of a program.

PROGRAM 33:



MM ♩ = 96 TIME BASE ♩ = 16

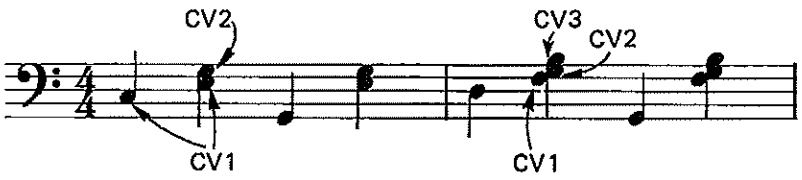
CHANNEL		1. PERCUSSION						
MEASURE	STEP	MPX		S				
1	1	1		16				
	2		2	16				
2	1	1		16				
	2		2	1				
	3			7				
	4		2	8				
3	1	1		16				

Switching Functions

The multiplex may also be effectively used for switching functions in conjunction with an analog switch (such as the ROLAND 723A) or an analog sequencer.

Program 34 shows a simple on/off switching function to get a piano-like bass pattern. CV1 is programmed for the lowest notes in the sequence; MPX 1 and 2 serve to add the other voices at the proper places.

PROGRAM 34:

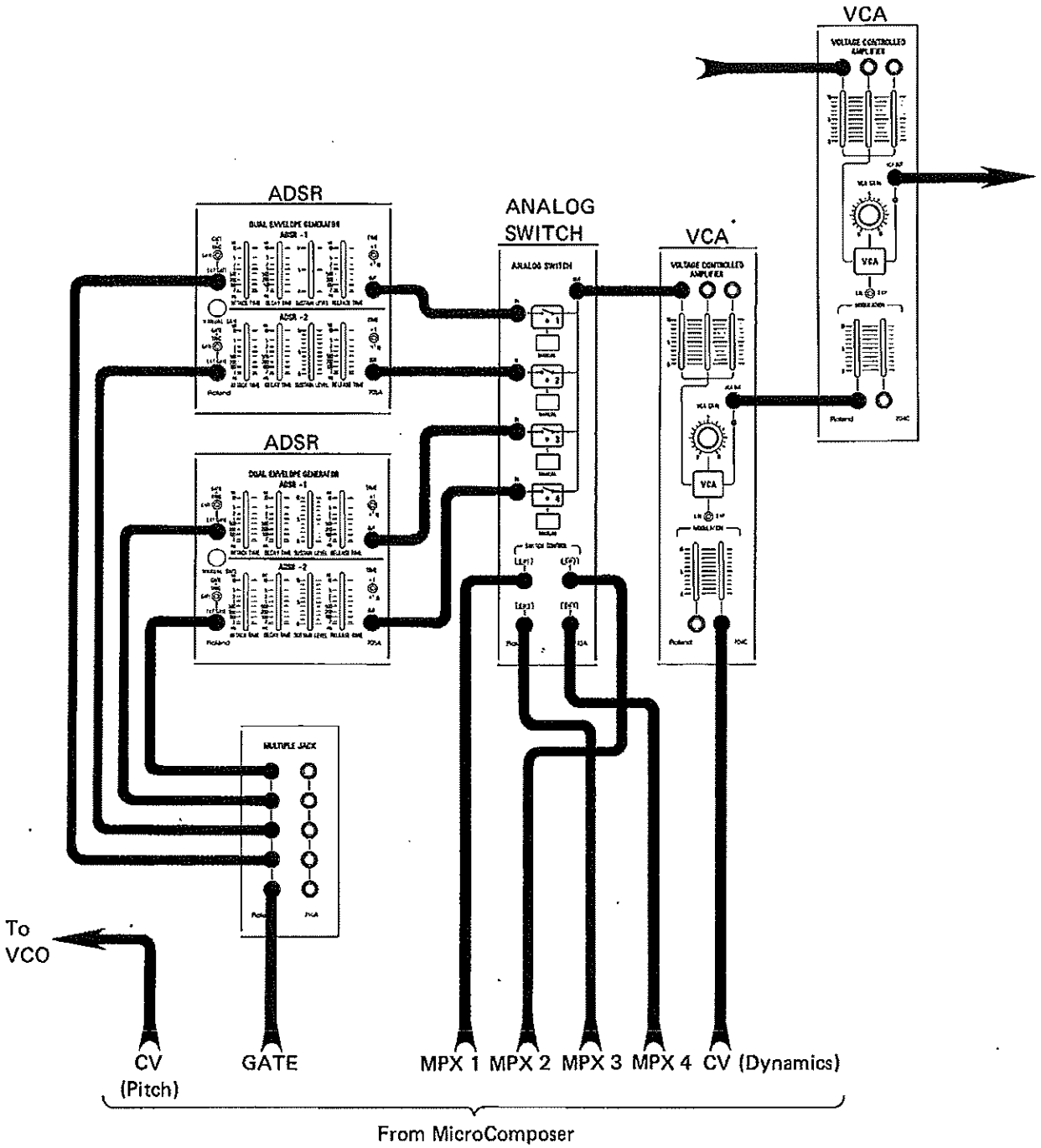


MM ♩ = 80 TIME BASE ♩ = 32

CHANNEL		1. PIANO							
MEASURE	STEP	VCO CV1	S	G	VCO CV2	VCO CV3	MPX		
1	1	12	32	2	19	23	0		
	2	16					1		
	3	7					0		
	4	16					1		
2	1	14					0		
	2	17					12		
	3	7					0		
	4	17	32	2	19	23	12		

The drawing below shows a patch for envelope switching. Each step in the program must contain an MPX 1, 2, 3, or 4 (or a combination of 1, 2, 3, and 4 for special effects). As an example, one envelope might have a longer attack time to give a swelling effect on certain notes. Other envelopes could produce other expression effects.

Output for Switching



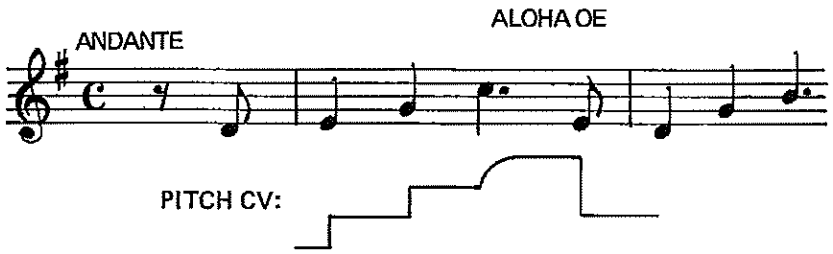
Portamento

The multiplex memory contains a special seventh bit which is used for portamento on/off control.

The PORTAMENTO controls on the front panel of the Interface affect the output of CV1 only. The knob controls the amount of portamento (lag time). When the switch is in MANUAL, the portamento is always in effect (and the LED next to the switch is lit). When portamento is not desired, the PORTAMENTO knob should be left at "0".

With the PORTAMENTO switch in the "MPX-7" position, portamento will be in effect only in those steps in the program where a multiplex bit "7" has been programmed. The LED next to the PORTAMENTO switch will light for those steps where portamento occurs as the program runs.

PROGRAM 35:



Set the PORTAMENTO switch at "MPX-7"; try the PORTAMENTO knob at about "4".

MM ♩ = 80 TIME BASE ♩ = 32

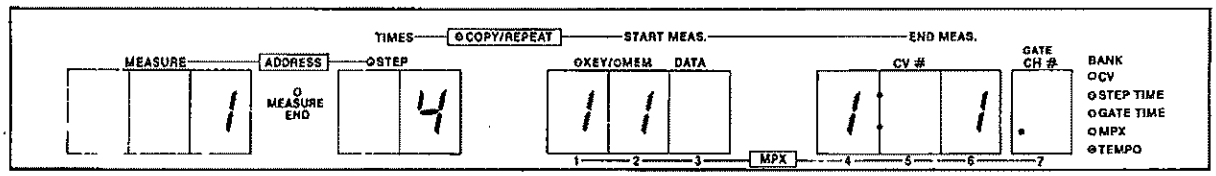
CHANNEL		1.							
MEASURE	STEP	CV1	S	G	MPX				
1	1	26	16	0					
	2	26	16	14					
2	1	28	32	28					
	2	31	32	28					
	3	36	48	40	7				
	4	28	16	14					
3	1	26	32	28					
	2	31	32	28					
	3	35	48	40	7				

When loading the MPX memory, use [0] for blank places. Also, don't forget to use [MEAS END].

The MPX Display Mode

Normally, the circuits which drive the displays convert the digital information in the Micro-Composer memories into decimal numbers which are easier for us to use and understand. When the display is in the MPX mode (which happens when the multiplex memory is selected using Operation 2), this conversion to decimal numbers does not take place; the MPX mode displays directly the on/off state of each MPX pulse output for each step in the program.

The Multiplex Display



ADDRESS display remains normal.

This portion of the display is used for the multiplex. The digits displayed show the channel assignment; in this case, multiplex is assigned to Channel 1. The position of the digits shows the outputs. In this example, pulses will appear at the 1, 2, 4, and 6 MPX OUTPUT jacks at M1-4 in the program.

Since the TIMER uses the right hand portion of the display whenever the MicroComposer is running (or when [TOTAL TIME] is pushed), the MPX mode of display would be incomplete and meaningless when the TIMER operates. For this reason, when the TIMER operates during the MPX mode of display, the multiplex pulse data is converted into decimal numbers and displayed at the KEY/MEM DATA position. The resulting numbers are of not much use in the studio, but displaying them is better than leaving the KEY/MEM DATA position blank and these numbers do serve to show that monitored memory (MPX) does contain data.

One more idiosyncrasy of the MPX mode of display should be mentioned. If the TIME BASE/TEMPO set operation is performed while the display is in the MPX mode, the display will not show decimal numbers, but will display the tempo and time base figures as if they were a series of pulses (which they actually are inside the memory circuits).

12. External Tape Memory

The digital information contained in the MicroComposer memories can be recorded onto tape for permanent storage of program data. No matter what the condition of the program, even if it is not finished, or it will not run, or it continually produces the ERROR function, as long as the MicroComposer memories contain data, this data can be transferred to tape.

Almost any tape recording format can be used, even cassettes.

Recording the Tape Memory

Patch the TAPE MEMORY DUMP jack on the rear panel of the MicroComposer to the LINE INPUT of the tape recorder. Like the SYNC OUT jack, a carrier signal always appears at this jack (at about 0dBm). Set the RECORD level for OVU.

To record the memory data, set the recorder in the RECORD mode and push [DUMP].

When [DUMP] has been pushed, all the displays will go dark and a "0" will appear at the KEY/MEM DATA display. If you monitor the output of the tape recorder you will hear the memory carrier; after a five second delay, the carrier will become garbled with the memory data. At this point, the KEY/MEM DATA display will start counting slowly upwards at an even rate of speed (very roughly about one second between counts). Each of these counts represents a thirty-two byte* block of data.

When the DUMP function terminates, the display will revert back to its original display mode; the ADDRESS display will show M0-0, and the data display will retain the last data block count. If you are monitoring the recorder output, the garbling of the memory carrier also ends.

It may prove to be a good habit to write the final data block number at the end of the written program sheet, and also on the tape identification labels so that you can judge the time required for the DUMP/LOAD operations.

About eight minutes of tape will be required if the MicroComposer memories are loaded to capacity (16 kilobytes; see Section 16). Less tape is required for shorter programs.

The DUMP function merely reads the data contained in the memories and does not remove or alter it in any way.

*byte: a measure of memory space; one byte is required for each number written into the memory. (See Section 16).

Verification of the Tape Memory

The VERIFY function allows the data recorded on tape to be compared with the data remaining in the memories to make sure that it has been recorded correctly.

Patch the LINE OUTPUT of the recorder to the TAPE MEMORY LOAD jack on the rear panel of the MicroComposer.

To verify, set the tape at a point several seconds before the beginning of the recorded memory carrier signal. Put the recorder in PLAY and watch the VU meter. When the meter jumps up indicating the presence of the recorded memory carrier, push the [VERIFY] button. As in the DUMP mode, all displays will go dark, leaving a "0" at the KEY/MEM DATA display. When the tape reaches the point where the data starts, the display will start counting blocks of data as before. When the data ends, the counting will stop and the display will show the number of the last block of data counted. At this point the ERROR function will be activated if there was a mistake in the data as recorded. This type of error is most often caused by tape drop-out; it is often possible to correct it by simply re-recording the data.

Retrieving the Tape Memory

If the MicroComposer has just been turned on from cold in a very cold room, it may be necessary to let it warm up for at least one minute before trying to load the memories from a data tape.

To retrieve the tape memory, use the procedure outlined above for VERIFY, but use the [LOAD] button instead of [VERIFY]. Pushing the [LOAD] button clears all the memories so that memory clearing before initiating the LOAD function is not necessary.

The LOAD function is automatically terminated when the data for a given program ends so that once initiated, it no longer needs attention. Once terminated, the MicroComposer will not be affected by following data for other programs recorded on the same tape (unless [LOAD] is pushed again). (The same is true of the VERIFY function).

If you change your mind during the DUMP, VERIFY, or LOAD operations, the [STOP] button can be used to terminate these functions. In the DUMP mode, if the [STOP] button is pressed while the display still shows "0", the function will not be canceled until the data count actually begins.

Data Errors

In the VERIFY mode, an ERROR indicates that the data on tape does not match the data in the memories. This is corrected by re-recording the data.

In the LOAD mode, an ERROR indicates that the MicroComposer has missed some data somewhere. If the ERROR was caused by noise, the tape can be run again.

If an ERROR occurs due to drop-out, the ERROR function is held off until all the tape data has been read and loaded into the memory. In this way, valuable data tapes may be used even though they contain drop-out errors. Once loaded, it remains only to find where in the program the error is and to correct it by editing the program in the normal manner.

If the ERROR function is activated, the MEASURE display shows a flashing number which represents the data block number where the error occurred (or the number of the data block containing the last error if there is more than one). If you remember that the data blocks are recorded on tape in exactly the same order in which they were loaded into the MicroComposer memories, you can see that the block numbers can sometimes give you a rough idea where in the program the mistake occurs.

For example: if you loaded CV1-1, STEP 1, then GATE 1; if the last data block is numbered 96 and the block containing the error is number 31, since CV1-1 represents the first third of the total memory space used, the error is located near the end of CV1-1.

Under normal studio conditions, the activation of the ERROR function when using any of the tape memory functions should be quite rare.

Data Tapes

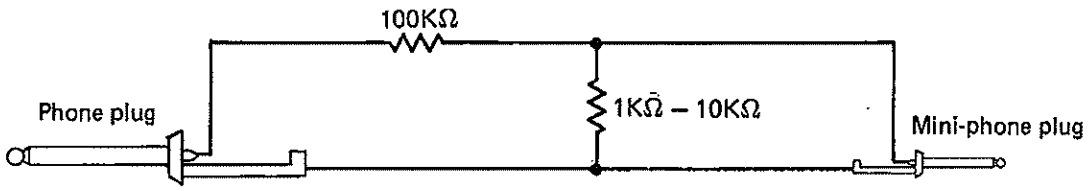
As mentioned before, most troubles encountered with memory recorded on tapes will be caused by tape drop-out. For this reason, to protect valuable data tapes use the same standards of machine care, tape quality, and tape storage conditions as used for valuable audio tapes.

There are two exceptions to the above. The difference in levels between the recorded data and the signal level of print-through is great enough that print-through is no problem; therefore, tapes need not be stored tail out and can be recorded in both directions to economize on tape. Like audio tapes, however, they should be stored tightly wound.

The second exception is that you can usually get away with using lower recording speeds to help economize tape.

Cassette Recorders

Cassette recorders may prove to be the best medium for data recording from the cost and convenience stand point. Even a cheap pocket type cassette machine should cause little trouble. If the recorder has only a MIC input, you will need to make a special patch cord:



From MicroComposer
TAPE MEMORY DUMP
jack.

To recorder
MIC IN

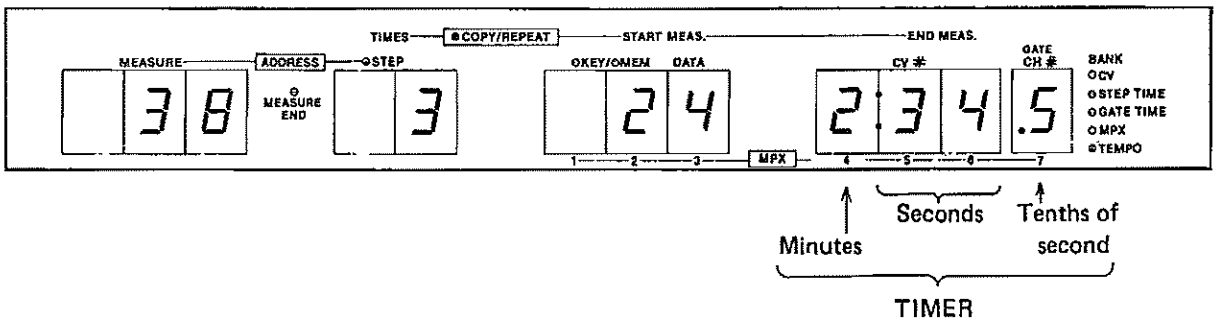
(1/4 watt resistors okay)

The earphone jack can be used as is for data output from the recorder. You may want to measure the output level and mark the volume control at the point that produces 0dB but this is usually not critical. Also, you may want to install a switch to cut off the speaker if it produces sound during the processing of data tapes.

With a stereo cassette recorder, record data on both channels at the same time to minimize drop-out problems. If you have trouble retrieving the memory, mix the stereo output together.

13. The Timer

When the MicroComposer is running, the right hand portion of the display acts as a stop watch timer.



The timer is activated whenever the [START] button is pushed and deactivated whenever the program ends or the [STOP] button is pushed. This means that not only can the program be timed from the beginning to the end, but it can also be timed from any point to any other point using the RUN functions shown on page 72.

The TIMER portion of the display contains four digits; the maximum amount of time which can be displayed is 9'59.9", after which the timer returns to 0'00.0".

The [TOTAL TIME] Button

Pushing the [TOTAL TIME] button will cause the timer to display the time required for the program to run from the beginning to the point shown by the ADDRESS display. (This does not work when the display is in the NON-DISPLAY mode; p. 77).

When a specific time duration is required for a program (such as in commercial work), set the last address in the program (both measure and step number), push [TOTAL TIME] and note the timing. Adjust the TEMPO control up or down slightly towards the desired timing and push [TOTAL TIME] again. Repeat until pushing [TOTAL TIME] produces the desired timing.

To cancel the TOTAL TIME display, use [STEP] or [BACK STEP].

For long programs, the difference between the timing indicated when using [TOTAL TIME] and when the program is actually run may vary ±0.1 second. Also, the time indicated could vary from channel to channel if the total number of timing increments loaded into each STEP TIME memory is not exactly the same.

In animation work, or any other work where musical punctuation must exactly match action on a screen, timing of different portions of the music may become more important than total time. This is discussed in Section 15.

(Note that the TOTAL TIME function does not work to show correct timing when programmed variable tempo is used; see Section 14, p. 135.)

If the STEP TIME values in a given channel totals more than 65,535 timing increments, pressing TOTAL TIME will activate the ERROR function since this figure is above the capacity of memory used to calculate the timing. Normally, this should cause little inconvenience since this could occur only with programs which would run from 20 to 30 minutes, depending on the tempo.

14. Variable Tempo

TIME BASE/TEMPO

The TIME BASE figure determines how many timing increments in the timing circuits will equal a quarter note.

TEMPO determines how many of these quarter note groups of time increments will occur per minute; in other words, the number of quarter notes per minute (with the TEMPO control on the front panel at center).

TIME BASE times TEMPO, then, gives the number of time increments produced per minute.

As an example, if you load a program using a TIME BASE of 16 and run it at a TEMPO of MM ♩ = 100, quarter notes will be produced at a rate of 100 per minute, and the time increment rate will be 1,600 (16 x 100) per minute. Next, if you change the TIME BASE to 32 and the TEMPO to MM ♩ = 50 without changing anything else in the program, it will sound the same as before. Doubling the TIME BASE without changing the note time values in the program has the effect of cutting all these time values in half. Since the tempo is also cut in half, the program will sound the same. The time increment rate will also be the same (32 x 50 = 1,600).

From the above it can be seen that changing either the TIME BASE or TEMPO will change the increment rate and thus the rate at which notes in the program are produced.

As mentioned on page 53, the minimum increment rate which the MicroComposer will accept is 128 increments per minute. (In other words, TIME BASE times TEMPO must equal 128 or more). The upper limit will depend a great deal on the program itself.

In the following three test cases, only one channel was used and all STEP TIME values were the same for every step in the program.

STEP TIME	TIME BASE		TEMPO		INCREMENT RATE
1	12	x	218	=	2616
2	24	x	214	=	5136
16	32	x	194	=	6208

The increment rates shown were the maximum rates which allowed exactly correct timing of the steps. You would probably be able to go to twice these increment rates before your ear could detect timing errors, however.

In most programs, of course, the STEP TIME values will vary from step to step. In programs with very high TIME BASE/TEMPO rates, steps with larger STEP TIME values will be normal while STEP TIMES with smaller values (such as "1" or "2") would be slightly late. Most often, this would go

undetected, except for extremely high rates in large multichannel programs. In many cases if the large program is run by itself, no timing errors will be noticed; however, if a second long program is run in sync with the first, it is possible that the new material will gradually go out of sync with the first due to inaccuracies in timing. This can be avoided by using a TIME BASE times TEMPO of 3,000 or less, or by avoiding the use of "1" or "2" in the STEP TIME memories of such programs.

With extremely high rates, pushing [START] will occasionally cause the MicroComposer to jam itself into a deadlock where nothing seems to work. Short of destroying the program by turning off the POWER switch, this deadlock can be broken into as follows:

Push the SYNC switch in, the TIMER portion of the display should begin running; push [STOP], push the SYNC switch again to raise it (LED off). Change the TIME BASE and/or TEMPO and try again.

Manually Controlled Tempo

There are three ways in which tempo may be varied as the program runs.

The first would be, of course, to vary the tempo manually with the TEMPO control on the front panel as the program runs. These tempo variations will be reflected in the sync signal output so that in multichannel recording all following programs will exactly follow any tempo changes when the MicroComposer is being driven by the recorded sync signal.

Varying STEP and GATE Times

The other two methods of varying tempo involve programming. Programmed tempo has the advantage of remaining the same every time the program is run. In this way, you can listen to the effects of the tempo changes, then edit them so that eventually you get exactly the nuances you want, and these nuances will remain the same each time the program is run.

The first and most obvious way of changing tempo within the program is to alter STEP TIME. Program 36 shows an example of a ritardando done in this way.

PROGRAM 36

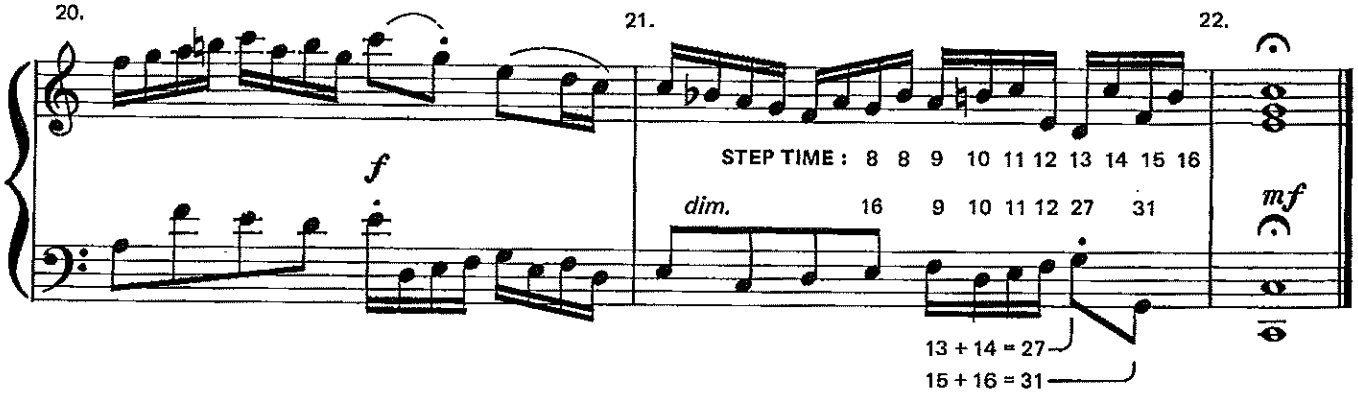
BACH INVENTION

MM ♩ = 96 TIME BASE ♩ = 32

CHANNEL		1. HORN			2. HORN			3. HORNS				
MEASURE	STEP	1	S	G	2	S	G	3	4	5	S	G
21	1	36		6	16	16	14	12	28	35	164	0
	2	34			12							
	3	33			14							
	4	31			16	16	14					
	5	110			17	9	7					
	6	33			14	10	8					
	7	31			16	11	9					
	8	34	8	6	17	12	10					
	9	33	9	7	19	27	15					
	10	35	10	8	7	31	25					
	11	36	11	9								
	12	28	12	10								
	13	26	13	11								
	14	36	14	12								
	15	29	15	13								
	16	35	16	14								
22	1	36	128	120	0	128	120	12	28	31	128	120

Altered times

Note that the fermata has been omitted since these are the normal full measure values



Below are shown two other examples of how Channel 1 in M21 of Program 36 could be programmed to produce different ritardando rates.

(PROGRAM 36)

Slower rate:

		1	S	G			
21	1	36	8	6			
	2	34	|	|			
	3	33	|	|			
	4	31	|	|			
	5	29	|	|			
	6	33	|	|			
	7	31	|	|			
	8	34	8	6			
	9	33	9	7			
	10	35	9	7			
	11	36	10	8			
	12	28	10	8			
	13	26	11	9			
	14	36	11	9			
	15	29	12	10			
	16	35	12	10			

(PROGRAM 36)

Faster rate:

		1	S	G			
21	1	36	8	6			
	2	34	|	|			
	3	33	|	|			
	4	31	|	|			
	5	29	|	|			
	6	33	|	|			
	7	31	|	|			
	8	34	8	6			
	9	33	9	7			
	10	35	10	8			
	11	36	12	10			
	12	28	14	12			
	13	26	16	14			
	14	36	19	17			
	15	29	22	20			
	16	35	26	24			

The more complicated the music becomes, however, the more cumbersome this system becomes. But even so, there are still advantages. Different rates of tempo or change are possible with different voices in the same measures. Another use would be to purposely miss-time certain voices. For example, in Program 17 (p. 39), if the STEP TIME in M1 of Channel 3 (cello part) is made "31" instead of "32", the cello part will consistently be slightly ahead of the beat as long as the program is run from M1. (If run from any other measure, the cello part will be "normal").

Programmed Variable Tempo

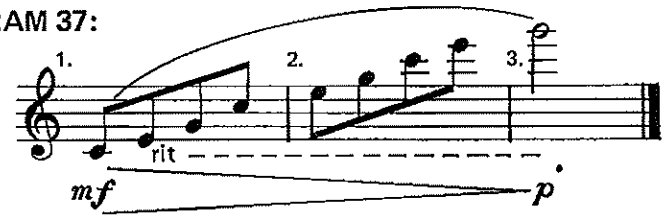
TEMPO may be assigned to any channel and loaded in exactly the same way as a CV memory so that the tempo may be different for every note in the program, if desired.

The data limits are the same as for fixed TEMPO:

MM ♩ = 2 – 254 (even numbers only)
(Data must be written in terms of a quarter note)

Program 37 shows another ritardando, but using programmed variable tempo.

PROGRAM 37:



MM ♩ = (120) TIME BASE ♩ = 32

CHANNEL		1.										
MEASURE	STEP	VCO 1	S	G	VCA 2	(T)						
1	1	24	16	10	70	120						
	2	28			68	120						
	3	31			64	114						
	4	36			60	110						
2	1	40			51	100						
	2	43			53	80						
	3	48			50	64						
	4	52	16	10	45	54						
3	1	55	64	58	40	40						

TEMPO
(Page 138 shows how to load this data)

If the full eight channel output is not being used, it may prove better to assign TEMPO to an unused channel rather than combining it with a CV channel. Using a separate channel makes it easier to program sections where tempo changes do not occur and is more economical of memory space; combining TEMPO with a CV channel means that all steps in that channel must also be loaded with a TEMPO value as well as pitch value.

PROGRAM 38:

MM ♩ = (120) TIME BASE ♩ = 32

CHANNEL		1.						2. TEMPO			
MEASURE	STEP	VCO CV1	S	G	VCA CV2			(T)	(S)		
1	1	24	16	10	70			120	32		
	2	28	|	|	68			114	16		
	3	31	|	|	64			110	16		
	4	36	|	|	60						
2	1	40	|	|	57			100	16		
	2	43	|	|	53			80	|		
	3	48	|	|	50			64	|		
	4	52	16	10	45			54	16		
3	1	55	64	58	40			40	64		

TEMPO

STEP TIME

STEP TIME detemines the length of time the TEMPO in each step will be held.

This program sounds exactly the same as Program 37.

Fixed/Variable TEMPO Selection Logic

The functions of the MicroComposer are controlled by button pushing and the logic of the internal master programming. TEMPO is the only case where a problem arises. If you push [TEMPO], then a digit, the MicroComposer would not know what to do with this digit. Is it the first digit in fixed TEMPO? If so, then the digit must be loaded in one portion of the memory and a place left for the possible loading of one or two more digits, and once [ENTER] is pushed, this data will control the tempo of the program. Or is this digit a TEMPO channel assignment? If so, then the digit must activate the memory assignment, then wait for TEMPO data which will control the tempo of the program.

This conflict was solved by deciding that if [TEMPO] was pushed after performing the TIME BASE set operation (even with a long delay), this would mean that the digits are to be fixed tempo. If [TEMPO] is pushed at any other time, it means that the next digit will be the TEMPO channel assignment.

Once variable tempo has been programmed, it is still possible to run the program again with a fixed tempo by performing the fixed TEMPO set operation again. The variable tempo data portion of the program remains intact, although it is not used.

When it is desirable to return to the variable tempo program, simply push [TEMPO] plus the correct channel assignment.

When dumping a program with variable tempo, remember that the data recorded on tape will be exactly the same as that contained in the memories. In other words, if the program was run with fixed tempo before dumping, when it is retrieved from tape it will run at a fixed tempo as before. To reinstate the variable tempo, push [TEMPO] and the channel assignment.

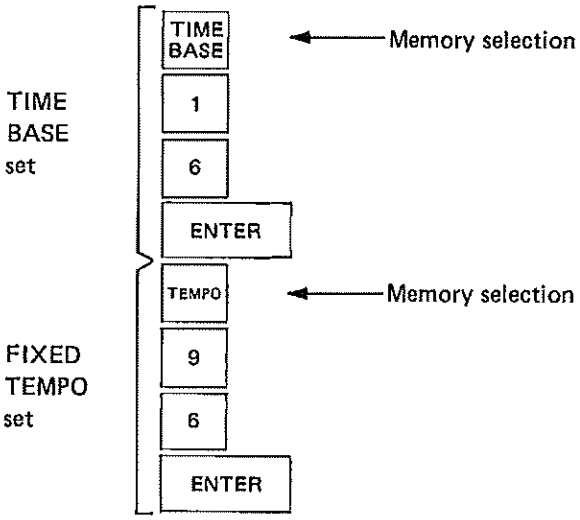
With fixed tempo, when [TOTAL TIME] is pushed, the MicroComposer adds up all of the STEP times from the beginning of the program up to and including the STEP time at the ADDRESS shown by the display. This total is than used, along with the fixed tempo rate, to calculate the time required for the program to run to the ADDRESS shown by the display.

With variable tempo (using the TEMPO memory), the MicroComposer will calculate the time required to run to the displayed ADDRESS using the tempo value loaded at that ADDRESS. Thus, with Program 37 or 38, if the ADDRESS display shows M3-1 and [TOTAL TIME] is pushed, the MicroComposer will calculate the time required to run the program to the end of M3-1 at a fixed tempo rate of MM ♩ = 40.

Fixed Tempo

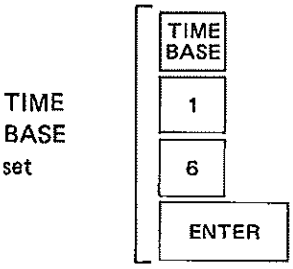
TIME BASE/TEMPO Set Operation

EXAMPLE: Set TEMPO at MM ♩ = 96 with a TIME BASE of 16.



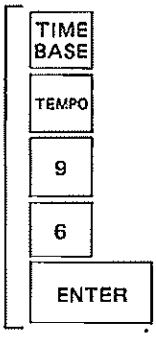
TIME BASE Set Operation

EXAMPLE: Set TIME BASE only (TIME BASE ♩ = 16)



Fixed Tempo (cont'd)

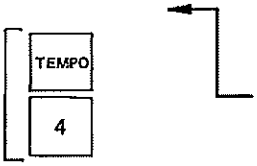
EXAMPLE: Set TEMPO only. (TEMPO MM ♩ = 96)



VARIABLE TEMPO

Variable Tempo set Operation

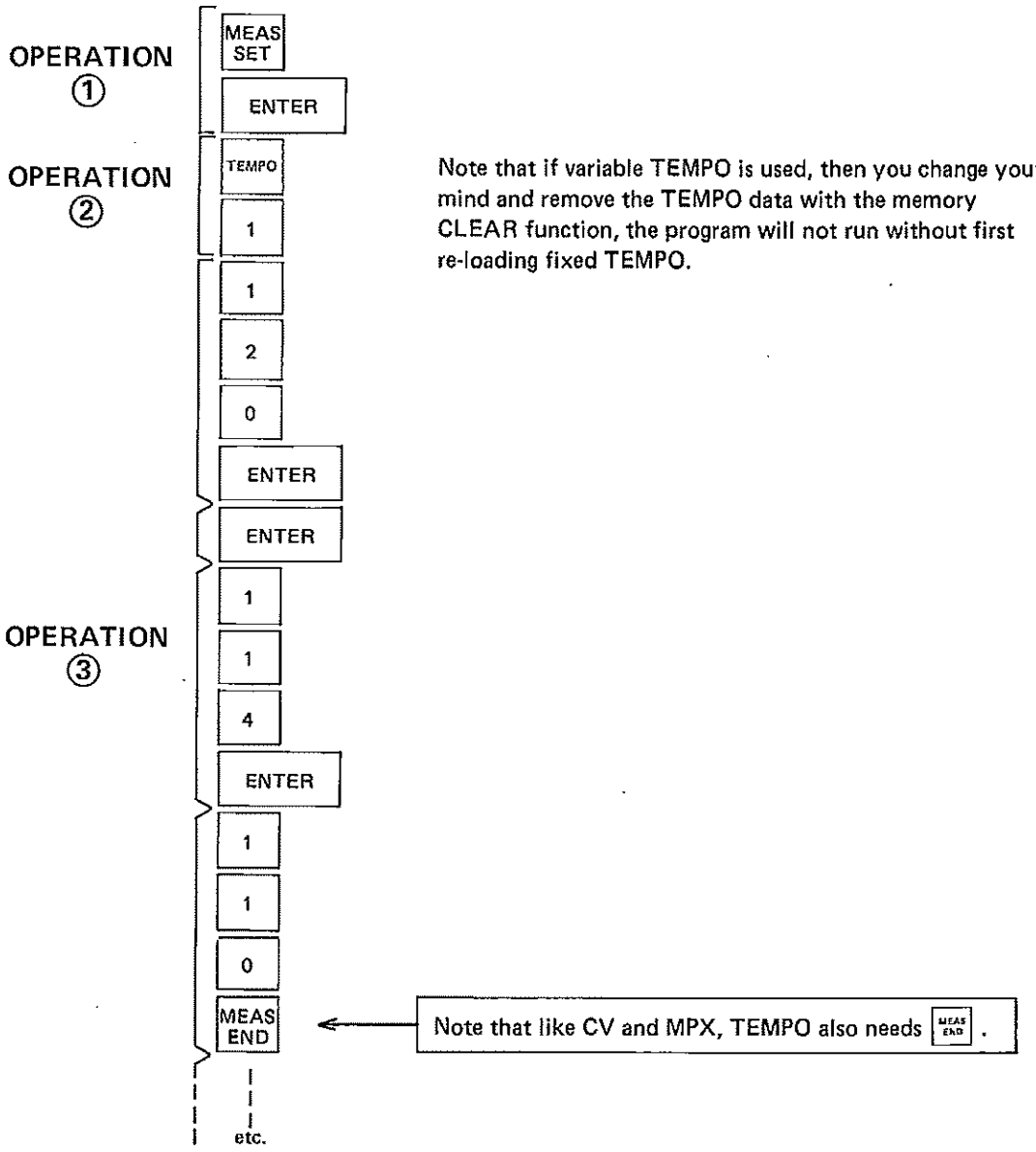
EXAMPLE: Assign TEMPO to Channel 4 to be used for variable TEMPO.



This must not be preceded by the TIME BASE Set Operation.

Variable Tempo set Operation (complete, with data)

EXAMPLE: Load TEMPO for Program 37 (p. 133)



Altering Overall Tempo

Because TIME BASE also affects the tempo of the program, if it is desired to alter the overall tempo of a variable tempo program, (outside the limits of the front panel TEMPO control), it is necessary only to change the TIME BASE. Of course, all editing and additions to the program must be done as if the old TIME BASE were still in effect.

15. Miscellaneous MicroComposer Applications

This section is designed to briefly present a few ideas on expanding the use of the MicroComposer.

Two or More MicroComposers in Parallel

The SYNC OUT jack of one MicroComposer may be used to drive other MicroComposers in parallel with the first through their SYNC IN jacks. The "slave" MicroComposers should be set with their SYNC switches down (LED lit) and the [START] buttons pushed. When the control MicroComposer is started, the others will follow.

If you push [STOP] on the control MicroComposer in the middle of the program run, then push [START] again, the control MicroComposer will start at the beginning and the others will start where they left off. To start the slave MicroComposers from the beginning, push their [STOP], then [START] buttons. This is where the remote [START]/[STOP] function would come in handy; all MicroComposers would be controlled by the same remote [START]/[STOP] buttons.

If the program has been run to the end, the slave MicroComposers must be reset by pushing their [START] buttons to run the program again.

It would be possible to work out an elaborate live show using a number of MicroComposers. One or more could control several synthesizers. Other MicroComposers could be used to control the mix, including a separate mix for cueing any "live" musicians. And other MicroComposers could control the lighting. The possibilities are unlimited.

Two or More MicroComposers in Series

There are two methods in which MicroComposers could be used in series for joining programs end to end.

The first would be to program a multiplex pulse at the end of the first program and use it to trigger the START function of the following MicroComposer by connecting the related MPX OUT jack directly to the START IN jack of the next MicroComposer in line. Between the entry of the start pulse and the actual beginning of the program there is a very slight delay. It can be eliminated by making the STEP TIME just before the multiplex pulse slightly shorter than normal.

The second method would be to run them in parallel with sync, but insert rests at the beginning of the "following" MicroComposer program, and at the end of the "lead" MicroComposer program.

Automated Mixing

The sync track on the multichannel master tape can be used to drive the MicroComposer during the final mix so that this mix could be partially or completely controlled by the MicroComposer program. Many mixing consoles have ACCESSORY jacks where VCA's could be inserted to control the levels in each of the input channels.

Overdubbing "Live" Music

When combining electronic music and "live" musicians, the normal procedure is usually to record the electronic portion first. This would be an almost absolute necessity with MicroComposer controlled electronic music because it would be extremely difficult to synchronize the Micro-Composer to already recorded sound.

One of the main advantages of using the MicroComposer is that if the sync track is left intact after the electonic music is recorded, it can be used to drive the MicroComposer to produce a metronome beat for the musicians who have to perform for the overdub. In this way, the musicians can be carried over sections where the rhythm is ambiguous, or where long rests occur in the music.

Other Scale Systems

The MicroComposer is designed to handle the standard musical scale system in which the octave is divided into twelve semi-tones. Other scale systems can be generated by simply attenuating the control voltage input to the VCO. Since the CV memories have a capacity of 128 (0-127) voltage levels, it would be possible to have a scale system where the octave has 127 divisions.

Consider the quarter-tone scale as an example. The quarter-tone scale has 24 divisions per octave. This means that if a "0" programmed into a CV memory produces a given pitch, "24" should produce the pitch one octave above. This is the key to tuning and adjusting the VCO to produce the pitches of different scale systems.

Program 39 is designed to demonstrate a quarter-tone scale. For comparison, M1 produces a standard diatonic scale running upwards for one octave, M3 produces a one octave chromatic scale, and M5 produces a one octave quarter-tone scale.

Load the program and temporarily insert a "0" and a "24" in front of M1-1 in the CV memory. With no CV input to the VCO, temporarily tune it to unison with any convenient reference pitch. On the MicroComposer, select the CV1 memory so that you can monitor the CV data at the display. Set the MicroComposer at Measure Zero and push [STEP] twice so that display shows "24". The voltage at the CV1 OUTPUT jack now corresponds to the data "24". Move the VCO CV input attenuator up until the VCO is tuned to one octave above the reference pitch. Use [BACK STEP] to return the CV output to "0". If the VCO is now in unison with the reference pitch, no further adjustment is necessary; otherwise, repeat the above steps until both "0" and "24" produce beat free sounds with the reference.

After the above is finished, delete the "0" and "24" and tune the VCO with its tuning control to the desired frequency. (Do not alter the VCO modulation input attenuator).

PROGRAM 39

Page 1

MM ♩ = 100 TIME BASE ♩ = 32

CHANNEL		1					
MEASURE	STEP	1	S	G			
1	1	48	16	14			
	2	52					
	3	56					
	4	58					
	5	62					
	6	66					
	7	70					
	8	72	16	14			
2	1	72	C	0			
3	1	48	SX	14			
	2	50	M1				
	3	52	M1				
	4	54					
	5	56					
	6	58					
	7	60					
	8	62	2				
	9	64					
	10	66					
	11	68					
	12	70					
	13	72		14			
4	1	72		0			

PROGRAM 39

Page 2

MM ♩ = 100 TIME BASE ♩ = 32

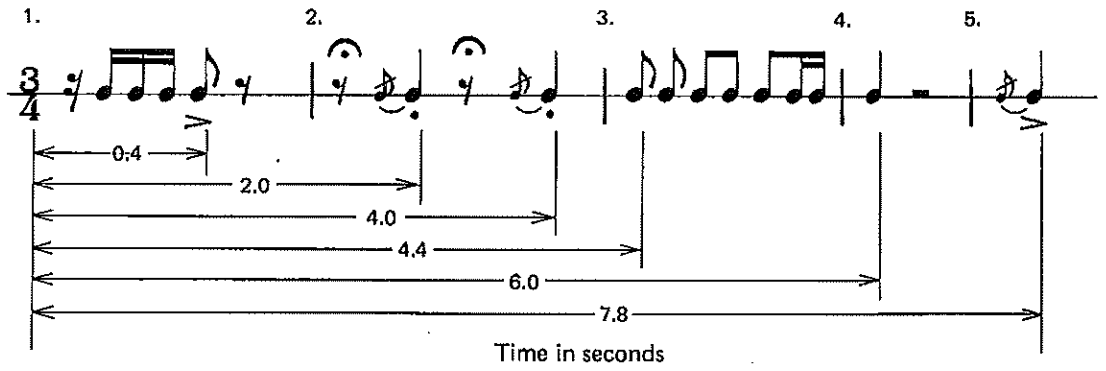
CHANNEL		1					
MEASURE	STEP	1	S	G			
5	1	48		14			
	2	49	3				
	3	50					
	4	51					
	5	52					
	6	53					
	7	54					
	8	55					
	9	56					
	10	57	4				
	11	58					
	12	59					
	13	60					
	14	61					
	15	62					
	16	63					
	17	64					
	18	65	5				
	19	66					
	20	67					
	21	68					
	22	69					
	23	70					
	24	71					
	25	72		14			

Timing in Music

In commercial work musical punctuation must sometimes exactly follow actions on a screen.

Let us say that the followings timings are needed:

PROGRAM 40a:

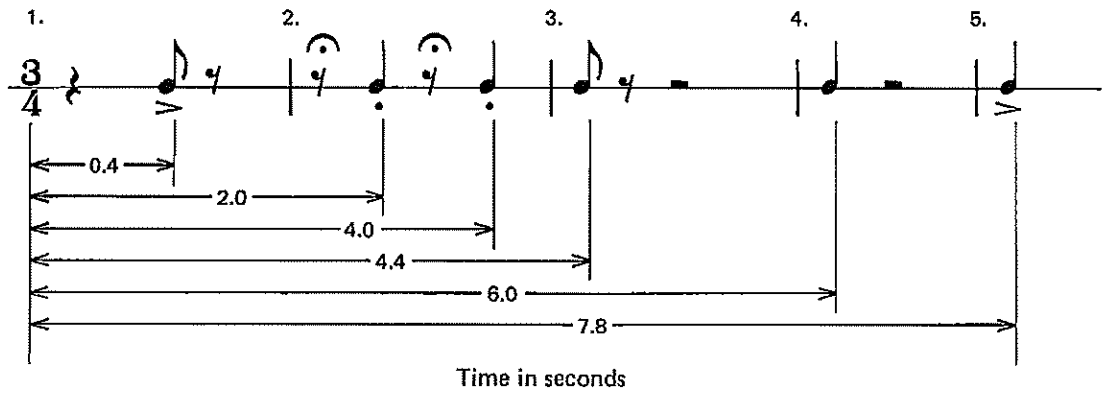


For all of the following, the position of the TEMPO control must remain unchanged. Also, it is assumed that the score already approximates the desired timings.

The first step is to write out a test program. Write the program using only one channel and as if the music contained only those notes where timing is important. Load the test program.

PROGRAM 40b:

TEST PROGRAM



PROGRAM 40b:

MM ♩ = 106 TIME BASE ♩ = 32

CHANNEL		1.								
MEASURE	STEP	VCO 1	S	G		DESIRED TIME	NEW STEP TIME			
1	1		32	0		0.4	28			
	2		32	2		—	32			
2	1		16	0		2.0	60			
	2		48	2		4.0	114			
	3		32	2		4.4	23			
3	1		96	2		6.0	90			
4	1		96	2		7.8	106			
5	1		96	2		—	96			
						See text				

Normal STEP TIMES
for TIME BASE ♩ = 32

Now select the STEP TIME memory, set M1-1, and push [TOTAL TIME]. If the time is not correct, push [BACK STEP] and re-write the STEP TIME data. A larger number will increase the playing time and a smaller number will decrease it. Push [TOTAL TIME] again. Again, re-write STEP TIME, if necessary. You will probably find that several values of STEP TIME will produce the correct time reading. Since M1-1 in the test program represents four notes in the actual music program, we used a number divisable by four: 28.

Push [STEP] and check the timing for M1-2. Continue as above until all the timings in the program are as desired.

After doing two or three of these steps, you will be able to judge more easily how much change is needed to get the correct time reading, and the process will go much more quickly.

The music program can now be written out using the STEP TIME values from the test program to determine the STEP and GATE TIME values.

PROGRAM 40a:

MM ♩ = 106 TIME BASE ♩ = 32

CHANNEL		1						
MEASURE	STEP	1	S	G				
1	1		7	0				
	2		7	5				
	3		7	5				
	4		7	5				
	5		32	10				
2	1		56	0				
	2		4	2				
	3		110	8				
	4		4	2				
	5		32	8				
3	1		15	10				
	2							
	3							
	4							
	5		15	10				
	6		8	5				
	7		7	5				
4	1		106	6				
	2		4	2				
5	1		96	6				

The total here is 28, the same as M1-1 in the test program.

This total equals test program M2-1.

16. The Memories

Memory Capacity

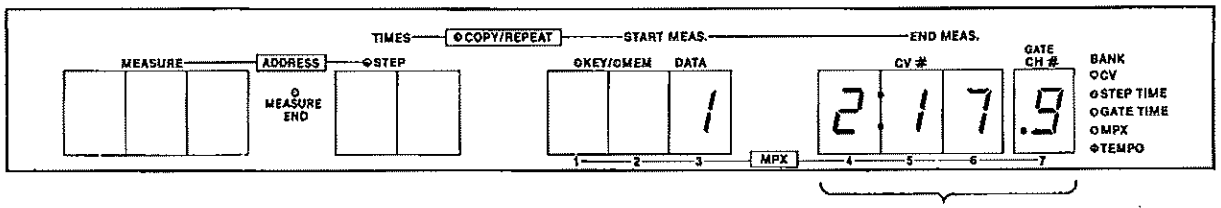
Memory capacity is measured in bytes. One byte is required for each data number written into the memory. The memory space required for the storage of a given program may be estimated by remembering that one byte will be needed for each square on the program sheet which contains a number (excluding, of course, MEASURE, STEP, and column headings; also excluding fixed TEMPO and TIME BASE).

The MicroComposer is delivered with a standard total memory capacity of 16,384 bytes. This is known as a 16K (16 kilobyte*) memory. 448 of these bytes are used as a "scratch pad" for the internal programming of the MicroComposer. This leaves 15,936 bytes for music programming.

A single voice line program would require three bytes per note, one byte each for: pitch, STEP TIME, and GATE TIME. This means that a single line sequence of over 5,300 notes is possible.

Available Memory

When [AVAIL MEM] is pressed, the right hand portion of the display will show how many bytes remain free for programming music. If you subtract the available memory figure from the actual capacity, you know how much memory space your program requires. The memory required by your program can be divided by 32 to determine approximately how far the data count will go and to estimate how much tape will be needed for dumping the data onto tape (see Section 12).



AVAILABLE MEMORY display

In this example 12,179 bytes of memory space remain.

*Kilo = 1,000

Memory Protect

Depressing the MEMORY PROTECT button (LED on) deactivates all of the MicroComposer front panel controls (except, of course, the POWER switch). With the MEMORY PROTECT button down, then, it becomes impossible to alter the contents of the memories by inadvertant key punching.

Memory Search Delay

Each time data is written into a memory, the microprocessor inside the MicroComposer requires a small amount of time to search the memory being loaded for an empty space for the new data. The farther you get from Measure 0, the longer this search for memory space takes. With very long programs this memory search time becomes noticeable.

You can observe this effect by trying Program 40. This program loads only the CV1 memory almost to the full capacity of the memories. After you have loaded the CV as shown, go back to the beginning and try loading a few steps of Channel 1 STEP TIME and you will note that there is no noticeable delay at the beginning of the program when loading a new memory.

The above is an exaggerated case since Program 41 is actually useless. Since most of the memory space is being used for CV data, there is no room for STEP and GATE data.

In the event you do need a long program and this delay becomes objectionable, you could load CV1 (for example) up to the point where the search time becomes annoying, then stop. At this point continue loading program data, but load it into Channel 2 from M1. When the length of Channel 2 nears that of Channel 1, stop and use the COPY function to transfer the Channel 2 data so that it follows the Channel 1 data, then clear Channel 2. Repeat this procedure, if needed.

PROGRAM 41

TEST

MM ♩ = TIME BASE ♩ =

CHANNEL		1.																						
MEASURE	STEP	I	S	G																				
1	1	24																						
	2	26																						
	3	28																						
	4	29																						
	5	31																						
	6	33																						
	7	35																						
	8	36																						
2	1	C																						
		99X																						
		M1																						
		M1																						
101	1	C																						
		18X																						
		M1																						
		M100																						
1901	1	C																						
		88X																						
		M1																						
		M1																						
1988	1	24																						
	2	26																						
	3	28																						
	4	29																						
	5	31																						
	6	33																						
	7	35																						
	8	36																						

← (Note the time required for the MicroComposer to carry out this COPY order after pressing the final [DATA]).

Loading this data will show the amount of delay encountered near the end of a program which takes up most of the memory space. (See text).

Measure and Step Capacity

The MEASURE display contains three digits and therefore gives a practical limit of 999 measures. In actual practice, as shown by Program 41, it is possible to load almost any number of measures. If each measure contains only one note, it is possible to have more than 5,300 measures, but it is impossible to use the [MEAS SET] button to select any measure above M999.

The STEP display contains two digits which gives a practical limit of 100 steps. In actual practice it is possible to load as many as 256 steps and to use Operation 1 to gain access to this data. For Step 100, the STEP display will show "0"; for Step 101, it will show "1", etc. For Steps 200 and 201, the display will be the same, "0" and "1" respectively. When loading from an external source (Section 6), any number of steps may be loaded, but Operation 1 will give you access only up to Step 256. For access to steps beyond, insert [MEAS END].

17. The ERROR Function

When all the display LED's begin flashing on and off, this indicates that the ERROR function has been activated. The ERROR function is activated whenever you do something which is beyond the capabilities or comprehension of the MicroComposer. In some cases, this ERROR indication is mundane and can be completely ignored, but in other cases it is helpful for showing that you may have made a mistake which will affect the program.

If the flashing display distracts you while you are trying to decide what to do or checking your program sheets, the flashing may be stopped by pushing [KEY CLEAR] or [STOP]. Remember that this does not rectify the reason for the ERROR display and that **the display in this condition does not show correct information.**

Following is a list of the causes which activate the ERROR function.

WRONG DATA ERROR

The ERROR function will be activated anytime you try to load data which is beyond the legal limits of the memories.

Example: Loading 128 into a CV memory will cause an ERROR. (CV memories accept data only from 0 to 127)

Example: Loading a TIME BASE of 16 with a TEMPO of 4 will cause an ERROR. (TIME BASE times TEMPO equals less than 128)

In the second example given above, the ERROR function will not be activated until [START] is pressed.

Trying to write illegal data into an occupied memory space (as when correcting a mistake) will not affect the data contained there. Trying to write illegal data into a new space in the memory will cause some random data to be written into the memory if the loading process is continued. In this case, when the ERROR function is activated, the correct data may be written without using the [BACK STEP] button.

Any combination of number keys or any number of number keys may be pressed without causing an ERROR; the ERROR function will be activated only when [ENTER] is pushed if the KEY/MEM DATA display shows data outside the memory limits.

The exception to the above is TEMPO. TEMPO (both variable and fixed) is designed to accept even numbers only from 2 to 254. If you try to load an odd number, it will automatically be reduced to the next lower number without causing an ERROR. When loading variable TEMPO, the TEMPO memory will accept "0" (or any other figure which brings the TIME BASE times TEMPO below the 128 limit), but when the program is run, it will stop at the point where the low tempo is and the ERROR function will be activated. Also, in some cases where TIME BASE times TEMPO is too large, the ERROR will not occur until you try to run the program. With extremely high TIME BASE/TEMPO rates, pushing [START] may cause the MicroComposer to jam into a deadlock where nothing seems to work. (See top of page 130).

WRONG OPERATION ERROR

The MicroComposer requires certain sequences of button pushing to perform different operations such as measure selection, copying data, memory clearing, etc. In most cases, if this order is deviated from, the ERROR function is activated.

Example: Pushing [STEP], then any key other than a number key (including pushing [STEP] again) will cause an ERROR.

This kind of ERROR may be corrected by pushing the desired button a second time.

Example: Push [CV]; if you change your mind and push [START] to run the program, the ERROR function will be activated. Push [START] again and the program will run.

The exception is the Measure Set Operation (Operation 1) which may be interupted at any point. The ERROR function will be activated only if the display is in the NON-DISPLAY mode.

NON-EXISTENT ADDRESS ERROR

This is a very common ERROR because there are many places in a multichannel program where an ADDRESS exists in one channel but not in another.

Example: If Channel 5 contains a simple bass line which consists of only one note per measure, trying to select any step other than Step 1 in Channel 5 will produce this type of ERROR.

Example: If the display shows the data at M6-4 in Channel 3 and you select one of the memories for the Channel 5 described above, the ERROR function will be activated since Channel 5 contains no M6-4.

This type of ERROR will often occur while editing multichannel programs and may usually be ignored.

NO DATA ERROR

To have access to a given STEP TIME or GATE TIME memory, at least one of the assignable memories (CV, MPX, or TEMPO) must have been assigned and loaded with data. Trying to select a STEP or GATE memory to which another memory has not yet been assigned and loaded will produce an ERROR.
CAUTION: This memory selection will remain and the MicroComposer will not run (See p. 73).

An ERROR will also be produced if you try to load a STEP or GATE memory beyond the last ADDRESS contained in the related assignable memory.

Trying to run the program from any point where there is a memory with no data will also produce an ERROR.

CV MEMORY IN USE ERROR

If you select and assign a CV memory, then later try to select the same CV but assign it to a different channel, the ERROR function will be activated.

The same is true of the (variable) TEMPO and MPX memories; the ERROR function is activated if you try to later assign them to different channels without first clearing both data and channel assignment.

COPY ERROR

The ERROR function will be activated if you go to something else in the middle of writing the COPY order (WRONG OPERATION ERROR).

The ERROR function will also be activated if you use strange measure numbers.

Example: C 1X M10 – M3

Example: Memory loaded only to M25, but: C 1X M20 – M30

In the above examples, no data will be copied.

Example: CV (or TEMPO, or MPX) is loaded to the end of M5, you forget [MEAS END], then write: C 3X M5 – M5.

CAUTION: In the last example above, data will be copied, but not necessarily as desired. To correct this, insert the [MEAS END] in the correct place, clear the memory of data from M6 (in this example) to the end of the program, then COPY again.

The above applies to assignable memories. STEP and GATE data will be copied correctly from previous measures into following measures even if the count does not come out right. The ERROR function will be activated if the amount of data is not enough to fill the open spaces (see page 101).

NON-DISPLAY MODE ERROR

An ERROR will sometimes occur when the display is in the NON-DISPLAY mode. Usually this may be ignored.

Example: Selecting an ADDRESS in the NON-DISPLAY mode (NO DATA ERROR).

When clearing all of a given memory (including channel assignment), the display reverts to the NON-DISPLAY mode and the ERROR function is activated. This is normal; it is actually an indication that the memory has been correctly cleared rather than an indication that a mistake (ERROR) has been made.

MEMORY OVERLOAD ERROR

Any operation which attempts to use more memory space than is available will activate the ERROR function.

TAPE MEMORY ERROR

The ERROR function may occasionally be activated in relation to tape DUMP/LOAD operations; see Section 12.

18. Calibration

This section is designed for qualified electronic technicians or repairmen. **DO NOT ATTEMPT ADJUSTMENT WITHOUT THE PROPER TEST EQUIPMENT.**

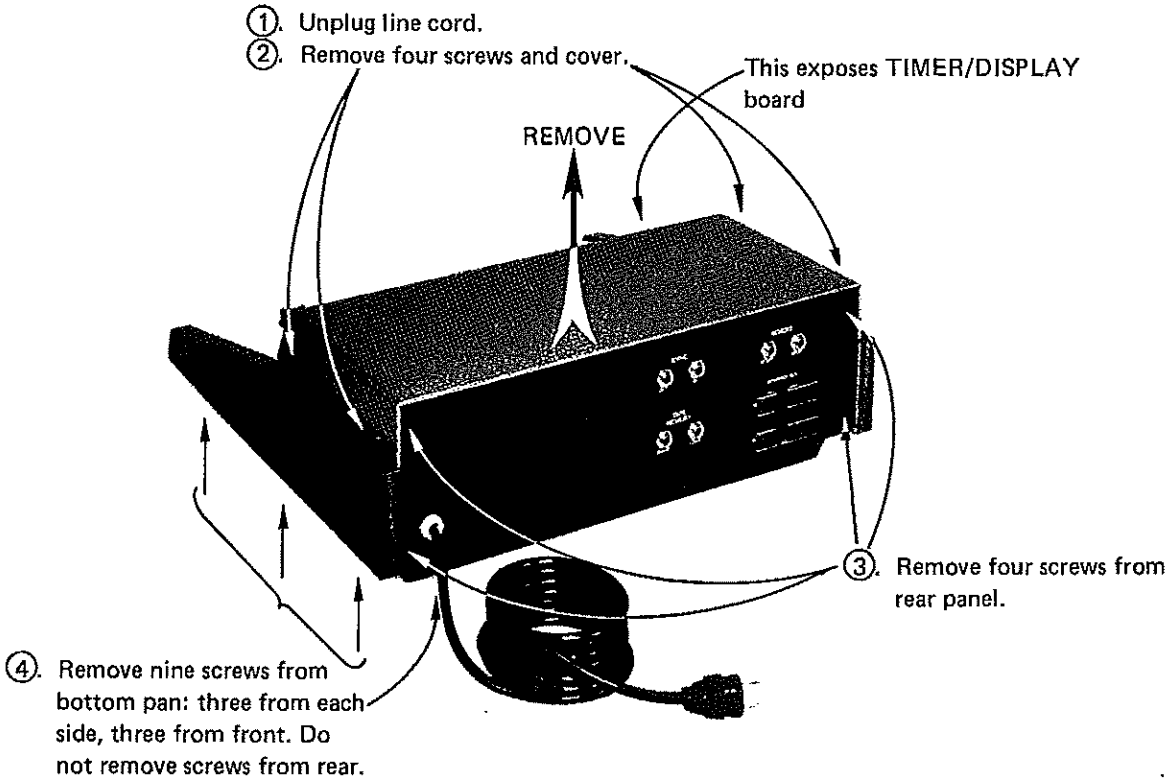
Since they are very stable, these adjustments should not be made unless it is determined necessary. Even then, only the adjustments at fault should be made.

The adjustments are divided into three groups. If adjustment is necessary, the groups may be done in any order desired. Within the groups, the order should be as shown.

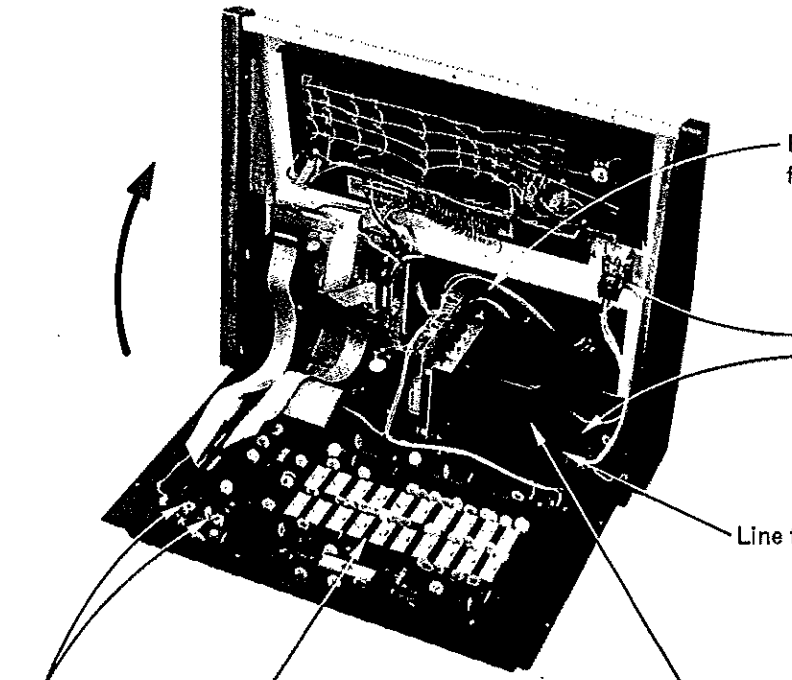
The following test equipment is needed:

- Digital Voltmeter
- Oscilloscope (dual trace)
- Frequency Counter
- Synthesizer Keyboard Controller

MicroComposer



⑤. Lift upper half of MicroComposer up as if it were hinged at the back.

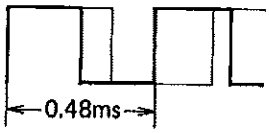


WARNING!
Unplug the power cord. Line voltage appears at the POWER switch and the line fuse.

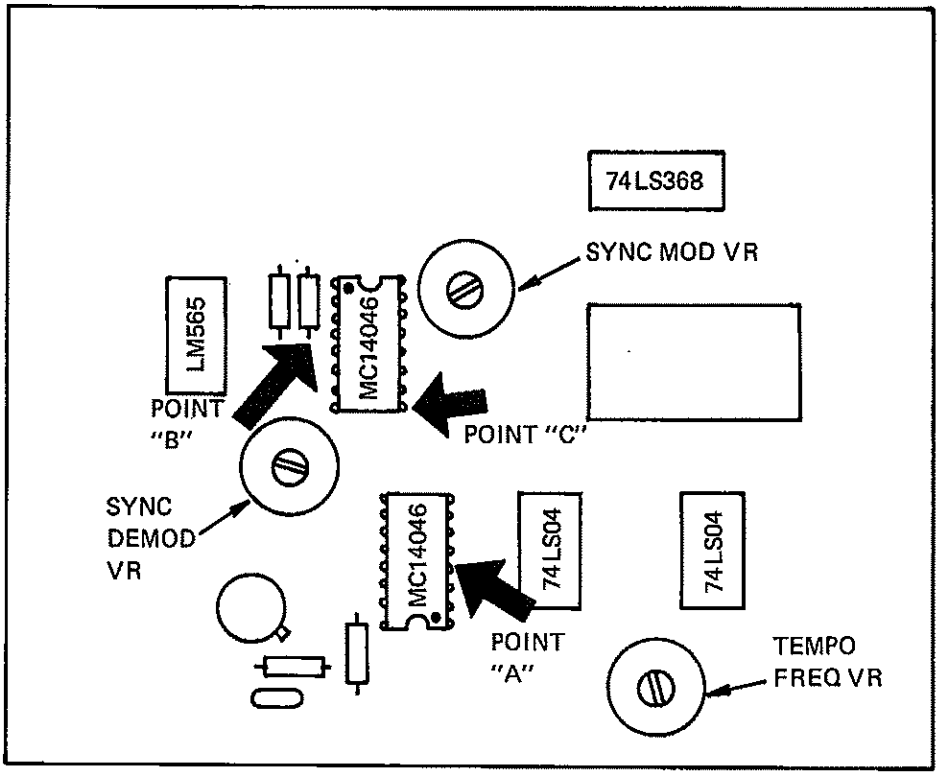
CAUTION:
Do not try removing any of the socket mounted IC's; static electricity generated by your body can easily and completely destroy them simply by touching them.

TIMER/DISPLAY BOARD Adjustments (see text)

② Load test program. Push [START].
Adjust SYNC MOD VR for Point "B" waveform:



③ Connect patch cord between SYNC IN and SYNC OUT. Push [START]. Adjust SYNC DEMOD VR for 50% duty cycle square wave at Point "C", (SYNC switch up).



① Adjust TEMPO FREQ VR so Point "A" = 139.8kHz.
(TEMPO control at "0")

The TIMER/DISPLAY Board Adjustments

TEMPO FREQ VR

With the front panel TEMPO control at "0", adjust the TEMPO FREQ VR so that Point "A" reads 139.8kHz on the counter.

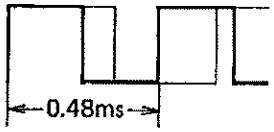
SYNC MOD VR

Both the SYNC VR adjustments require that a program be loaded into the MicroComposer. Any program will do so long as the MicroComposer will run. If a data tape is not handy, the following might be used:

MM ♩ = 250 TIME BASE ♩ = 32

CHANNEL		1.						
MEASURE	STEP	1	5	9				
1	1	33	16	8				
	2	|	|	|				
	3	|	|	|				
	4	|	|	|				
	5	|	|	|				
	6	|	|	|				
	7	|	|	|				
	8	33	16	8				
2	1							
		c99X						
		M1	M1					

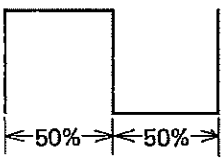
A 2.1 – 1.3kHz FSK is used for the sync signal. Load a test program into the MicroComposer. Push [START]. The waveform at Point "B" will be a composite of the two FSK frequencies. The frequency of one of these waveforms will change when the SYNC MOD VR is changed. Adjust this portion of the waveform so that one full cycle equals 0.48 milliseconds.



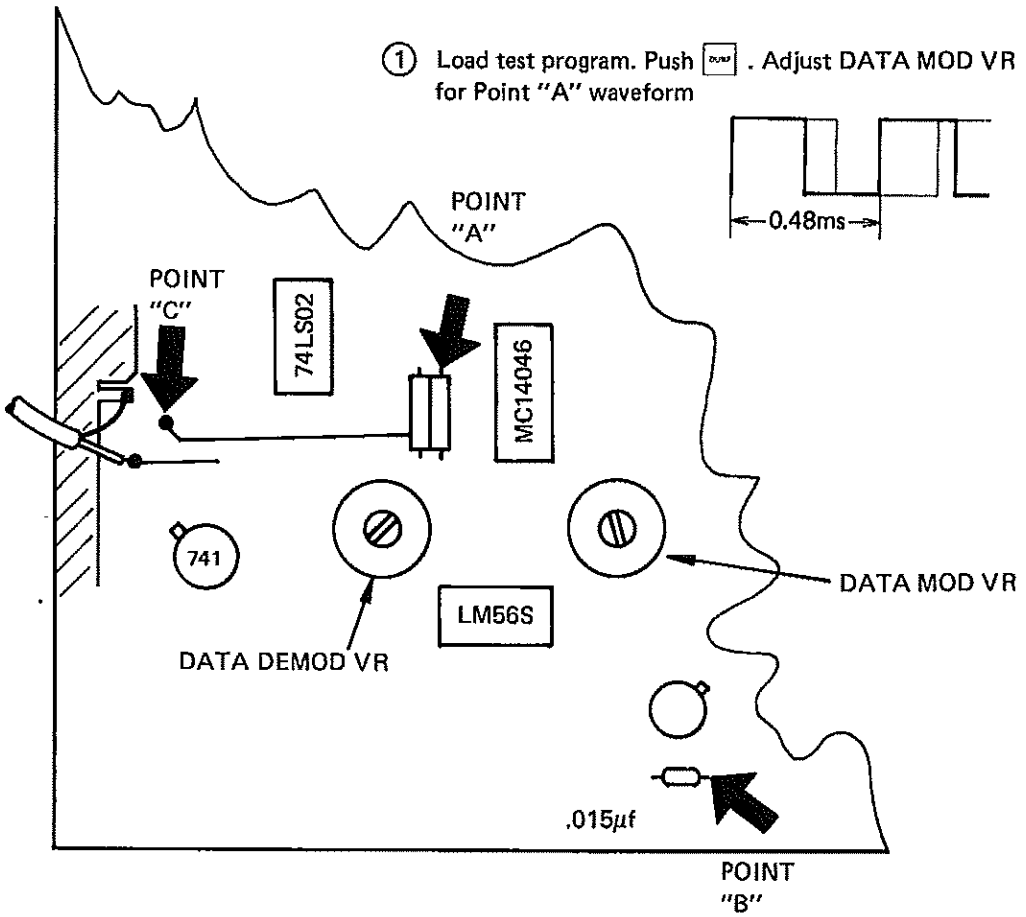
Portion to be adjusted is shown darker although it will not be this way on the oscilloscope.

SYNC DEMOD VR

Patch the SYNC OUT jack to the SYNC IN jack on the rear panel of the MicroComposer. With the test program loaded, push [START]. Adjust the SYNC DEMOD VR so that the waveform at Point "C" has a duty cycle of 50%.



CPU Board Adjustments (see text)



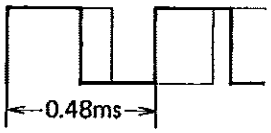
② Connect patch cord between TAPE MEMORY DUMP and LOAD jacks. Push [~] . Adjust DATA MOD VR so Points "B" and "C" show the same waveform.

The CPU Board Adjustments

DATA MOD VR

For data storage on tape the MicroComposer uses a 1.3 – 2.1kHz FSK. The frequency of the square wave appearing at Point "A" will depend on the latch condition of the controlling IC.

If the frequency at Point "A" is changeable with the DATA MOD VR, use a counter to adjust for 2.1kHz. If not, load a test program, push [DUMP], and after the data begins to flow, adjust the DATA MOD VR for 0.48 millisecond cycle in exactly the same way as the SYNC MOD VR is adjusted.



Portion to be adjusted is shown darker although it will not be this way on the oscilloscope.

DATA DEMOD VR

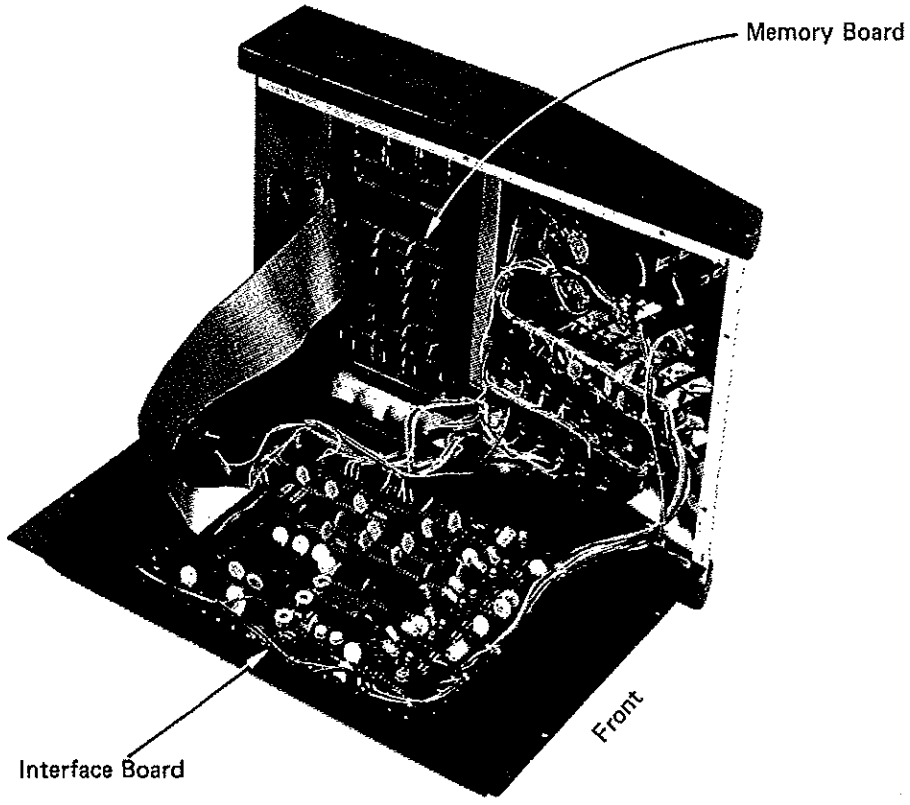
Load a test program. (It does not have to be a program which will run). Patch the TAPE MEMORY DUMP jack to the TAPE MEMORY LOAD jack on the rear panel. Push [DUMP] and adjust the DATA DEMOD VR after the modulation of the carrier begins; there are two methods:

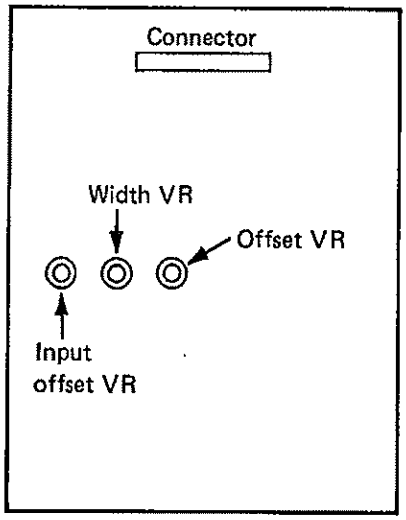
One is with a dual trace oscilloscope. Adjust so that the waveforms at Points "B" and "C" are the same while data is being dumped.

The other is to set the DATA DEMOD VR in the center of the range which produces the cleanest leading and trailing edges to the waveform at Point "B".

INTERFACE

① Remove all screws around edges of bottom pan.





The Interface Board Adjustments

CV OUTPUT ADJUSTMENTS

The OFFSET VR and WIDTH VR adjustments affect all of the CV outputs and are usually adjusted while monitoring the voltage at the CV1 OUTPUT jack.

Connect a digital voltmeter to the CV1 jack on the front panel of the Interface.

Program a "0" in the CV1 memory and adjust the OFFSET VR for 0.00 volts. (If "0" is programmed into the other CV memories, their outputs should read 0 volts, ±10mv).

Program "120" in the CV1 memory and adjust the WIDTH VR for +10.00 volts. (If "120" is programmed into the other CV memories, their output will read +10 volts, plus/minus the error noted in the "0" data reading).

Check that the following numbers programmed into the CV1 memory will produce the following readings:

0	0.00 volts
12	+1.00
24	+2.00
36	+3.00
48	+4.00
60	+5.00
72	+6.00
84	+7.00
96	+8.00
108	+9.00
120	+10.00

INPUT OFFSET VR

The accuracy of this adjustment depends entirely on the accuracy of the keyboard controller used.

Adjust the keyboard controller TUNING so that pressing the lowest key produces 0.00 volts at the keyboard CV output.

Connect the keyboard controller to the EXTERNAL INPUT jacks and set up the MicroComposer programming so that one of the CV memories will accept data from the external source.

The INPUT OFFSET VR is set so that playing a chromatic scale starting with the bottom key on the keyboard will produce numbers at the KEY/MEM DATA display which start at "0" and move upwards one step at a time all the way to the top of the keyboard without skipping or repeating numbers.

There will be a short range of settings of the INPUT OFFSET VR which will produce the correct data readings; the VR should be set in the center of this range. Below this range will produce data which starts at some number higher than "0"; above this range will produce several zeroes before the count moves upwards.

19. Specifications

MC-8 MicroComposer

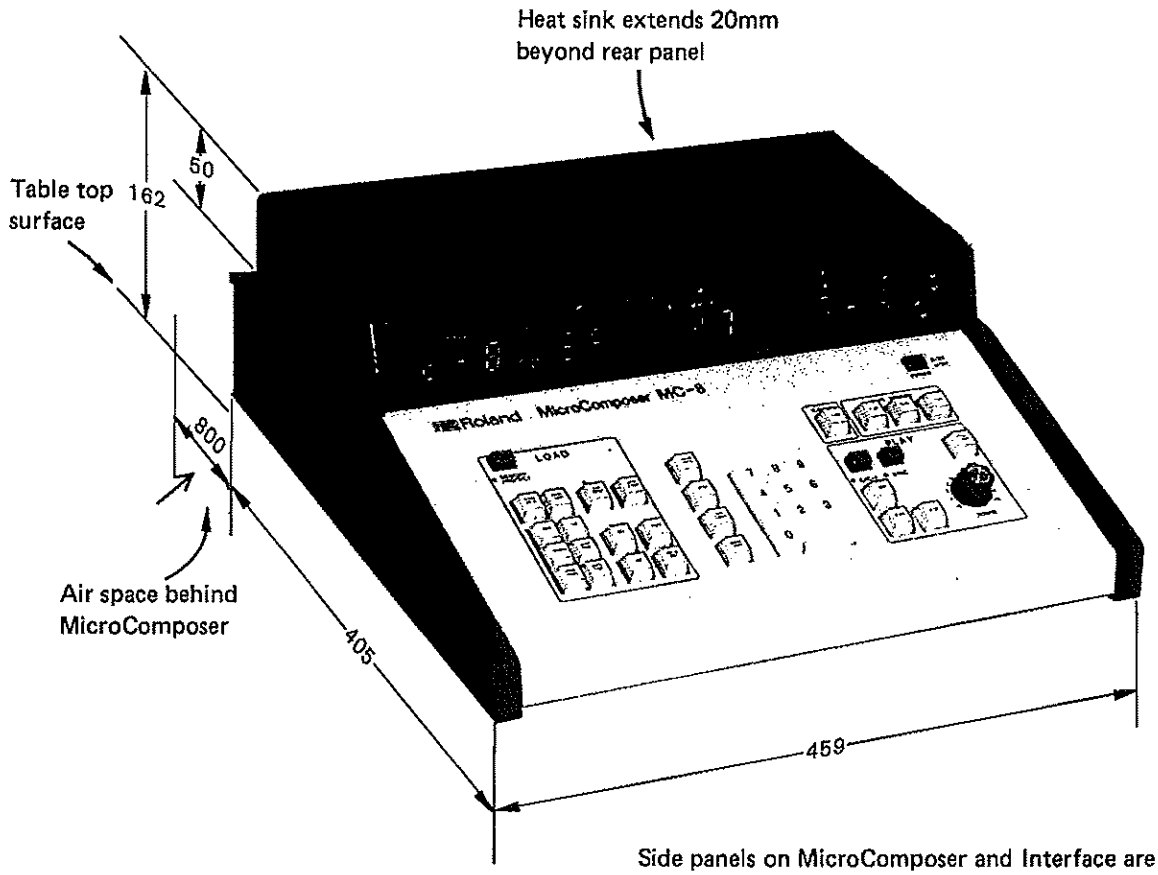
Power requirements:

100V, ±10%	50/60Hz	60W max
117V, ±10%	50/60Hz	60W max
230V, ±10%*	50/60Hz	60W max

*The 230V version can be used with either 220V or 240V.

Weight: 10.7kg

Dimensions (shown in millimeters):



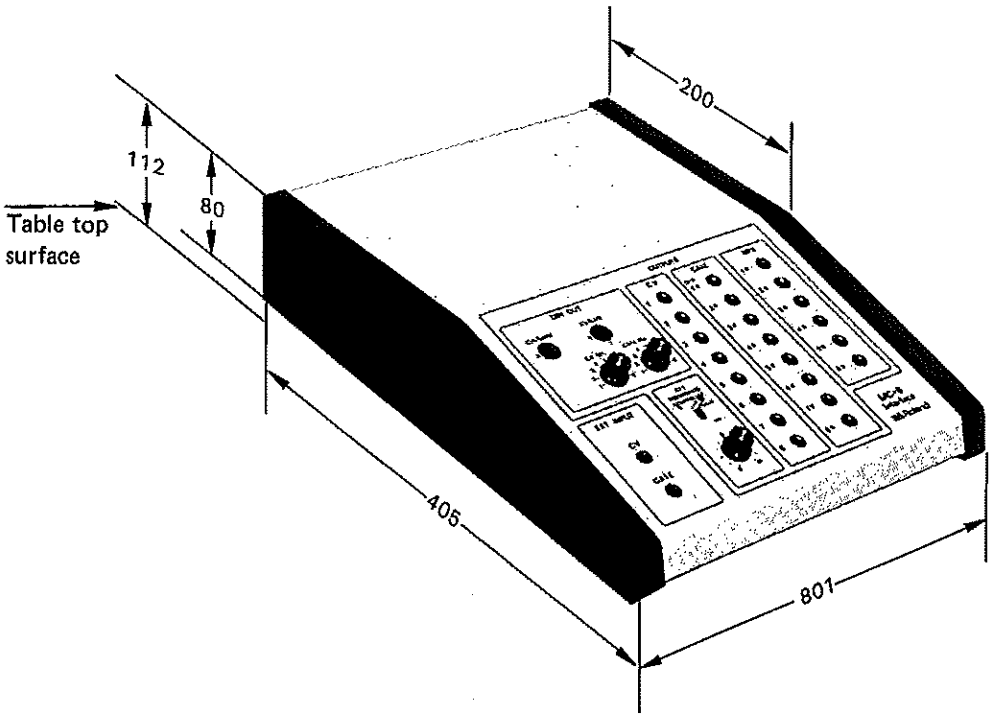
Side panels on MicroComposer and Interface are exactly the same.

MC-8 INTERFACE

Power requirements: None (power supplied from MicroComposer)

Weight: 4.5kg

Dimensions (shown in millimeters):



CV OUTPUTS

8 independent outputs; program assignable to any GATE output channel

Voltage range:	0 – +10.58V; 1/12V steps (128 steps)
Min. permissible load:	10kΩ
Voltage steps:	0 – 127
Pitch control sensitivity:	1V/8va; more than 10 octaves range
CV1 portamento range:	0 – approx. 3 seconds
CV1 portamento control:	Manual, or programmable

GATE OUTPUTS (switchable to minus gate)

8 independent outputs

Gate voltage; ON:	+15V (no load)
OFF:	0V
Output impeadance:	4.7kΩ
Current sink (minus gate only):	2mA

MULTIPLEX OUTPUTS

6 bits; plus one bit for CV1 portamento on/off control

Program assignable to any GATE output channel

Pulse voltage; ON:	+15V (no load)
OFF:	0V
Output impeadance:	4.7kΩ

EXTERNAL INPUTS

CV sensitivity:	1V/8va; 1/12 volt steps
CV range:	0 – +10.58V (128 steps)
CV input impeadance:	100kΩ
Gate pulse:	+5 – +15V (switchable to minus gate)
Gate input impeadance:	100kΩ

TAPE MEMORY OUTPUT

Signal:	FSK; 1.3 – 2.1kHz
Level:	0dBm (approx.)
Impeadance:	10kΩ

TAPE MEMORY INPUT

Minimum level:	–10dBm
Impeadance:	10kΩ

SYNC OUTPUT

(Same as TAPE MEMORY OUTPUT)

STNC INPUT

(Same as TAPE MEMORY INPUT)

REMOTE START

Minimum voltage:	+2.0V
Minimum pulse length:	100ms
Impeadance:	10kΩ

REMOTE STOP

(Same as REMOTE START)

MEMORY CAPACITY

Standard:	16 kilobytes (Single voice program: over 5,300 notes)

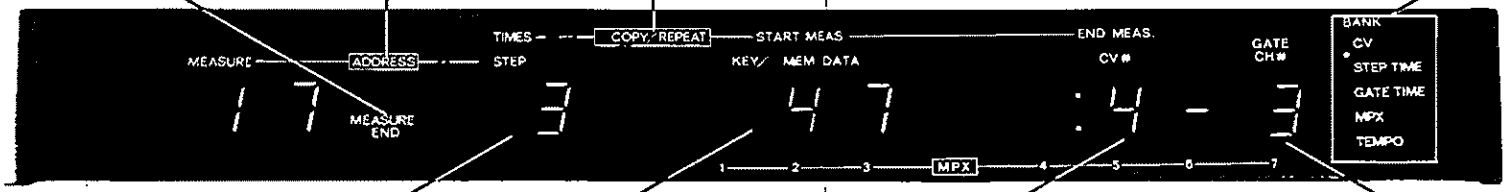
ADDRESS (p)
Location in a memory.

COPY/REPEAT Display (pp. 74, 94)
Shows START and END MEASURES when music is being repeated or data copied.

RE Indicator
TEP display represents the last data in a

splay (pp. 46-48)
neasure is being monitored.

BANK Displa
Shows what t

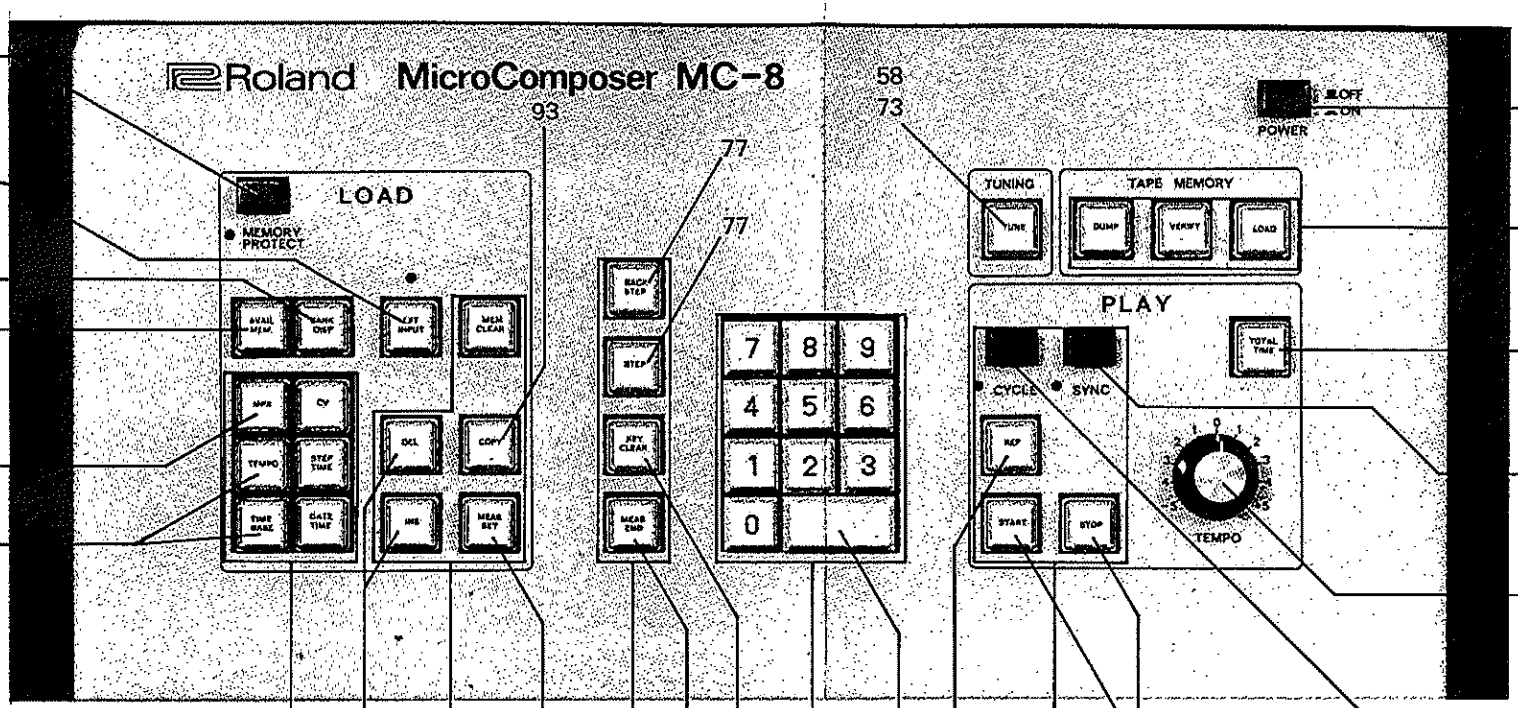


(pp. 46-48)
ıote within the measure is
ed.

KEYBOARD/MEMORY DATA Display (pp. 50, 51)
Shows data written with the KEYBOARD; or shows data contained in memories.

CV# Display (pp. 49, 74)
Shows which control voltage memory is being monitored; this portion of the display also used for TOTAL TIME (p. 127) and AVAILABLE MEMORY (p. 145)

GATE CH# D
Shows the ch
indicated by t



ECT (44; 146)
emory data from being
roved.

JT (SECTION 6)
g from an external source
d.

77

MORY (145)
memory space remains

SECTION 11

51
52
SECTION 14

93

58
73

77

77

POWER Swit

TAPE MEMC
For loading p
storage.

TOTAL TIM
To determine

SYNC Switch
Used for syn

TEMPO Cont
Used when m

71
72

84

46
47
48

55 77
57
81
93

ENTER Key
44
55
57
93

58
72
73

58
71
72

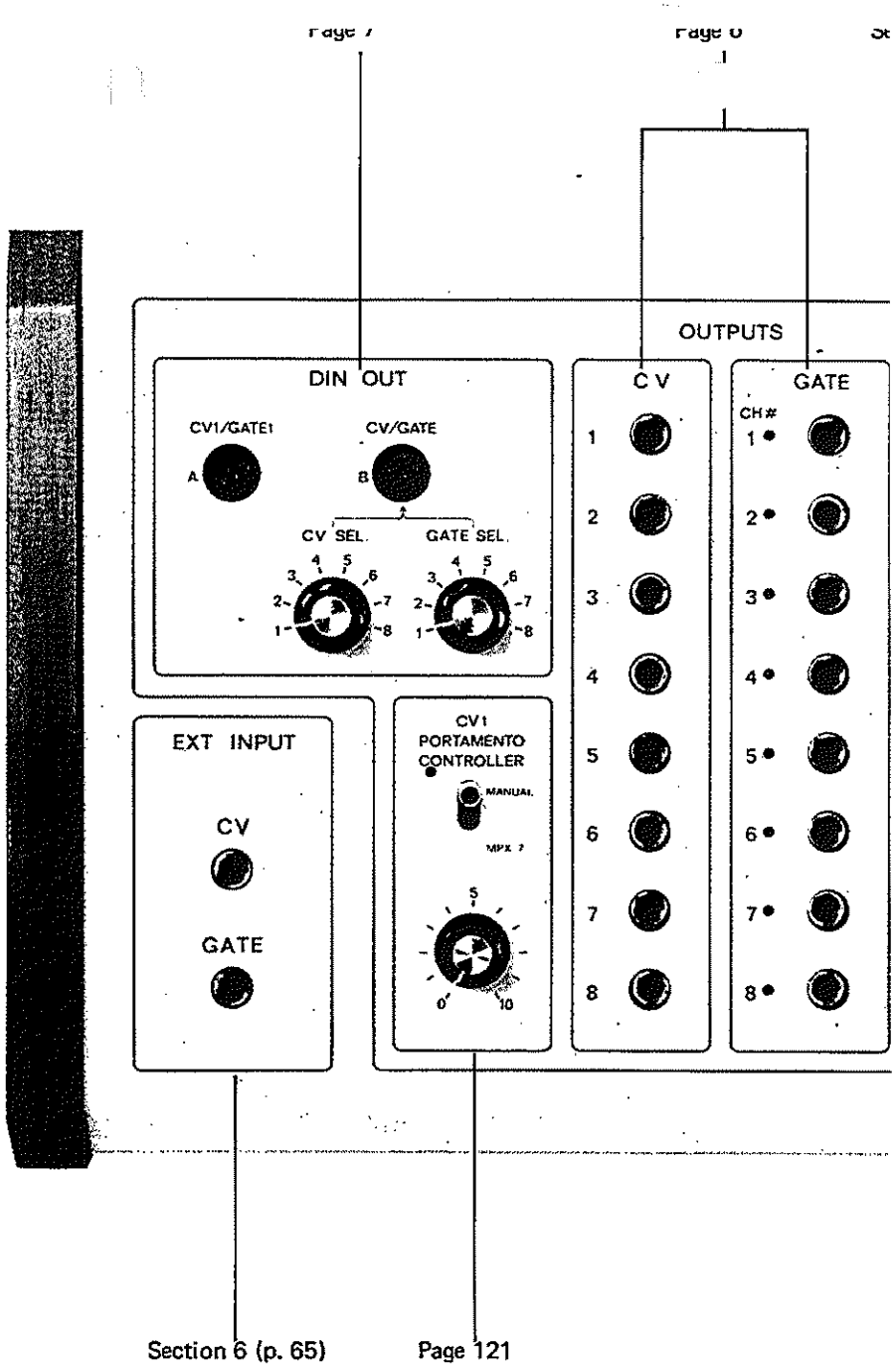
Memory Selection Keys
(49)

Editing Keys

KEYBOARD (SECTION 5)
For selecting memory locations and loading data

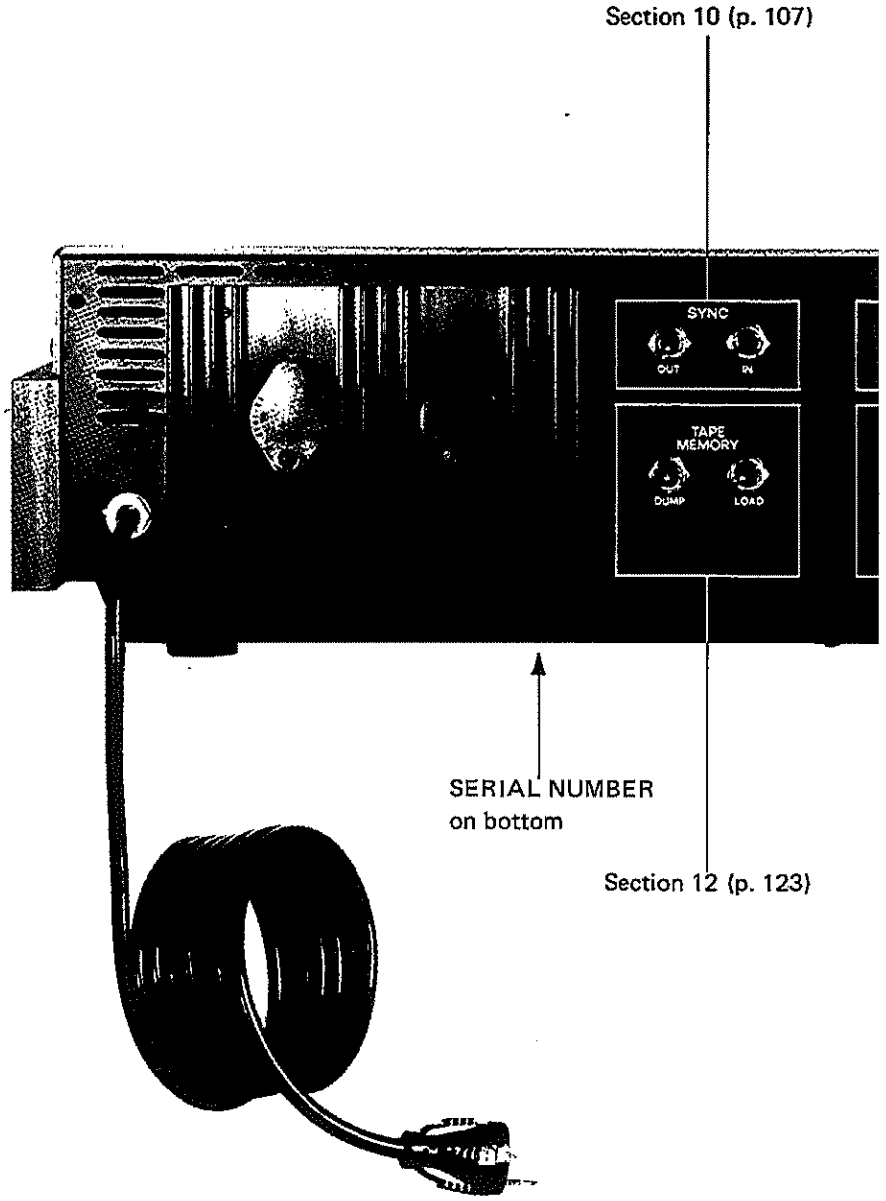
RUN Function Controls
(SECTION 7)

This group of keys is related to the
KEYBOARD functions.



Section 6 (p. 65)

Page 121


Section 10 (p. 107)
SYNC
OUT
IN
TAPE MEMORY
DUMP
LOAD
SERIAL NUMBER
on bottom
Section 12 (p. 123)