# Replicating Digital Synthesizer

Robert Glaser
Eugene Mauro

The replicating digital synthesizer stores in digital form musical notes. These notes may be obtained from an orchestra, individual instruments, or synthesized notes. These notes constitute the raw material from which new sounds are created. By recording and storing one such original note, it is possible to play this data back at different rates than that it was recorded; thereby creating many possible playback notes. The manner in which the data is played back can change the timbre and quality of the sound. Playing back the data the same way it was recorded will result in a complete scale of notes with the same quality as the one original note. Therefore, recording and storing a single orchestral note in the synthesizer will at the very least give the capability of playing back complete scales of orchestral quality.

However, this is merely the very least that the replicating digital synthesizer can do; by implementation with multiple independent playback circuits, chords, each elemental note of which is of orchestral quality, can be played back, both monaurally and with multiple outputs, each of which may be mixtures of several playback circuits. The playback circuits may also play the data back in a different sequence from that which it was recorded at, creating other effects. Once the music is played back digitally, it can be further processed for other effects as well by controlling the amplitude and frequency for tremelo, vibrato, and other techniques.

Once these sounds are created, they may themselves be recorded digitally and used as the raw material for new sounds.

The instrument is played by use of a standard keyboard and a control panel. In this fashion, for elementary operation, simple use of the keyboard will allow the novice to utilize this instrument, while for the advanced musician the sounds created are limited to his imaginaton, since this instrument gives the utmost flexibility in use through the control panel as well as the capability of programming the instrument to perform any possible other function. Clearly, no instrument with the sophistication and versatility of the replicating digital synthesizer has ever before been constructed. Only now with the advent of micro-computer integrated circuits is it possible to build such a device without needing rooms full of equipment and the bank account of Fort Knox.

Figure 1 shows the basic layout of machine functions of the digital synthesizer. The brains of the unit reside in the master microprocessor. The keyboard and control panel in part provide signals to the microprocessor, telling it what note(s) and mode(s) to command. The major task of the master microprocessor is to control the music generator. As will be seen later, the music generator consists of many controllers, and through the command bus, response line, and controller select lines, the master microprocessor accomplishes this control. There is also the capability of the master microprocessor to access the music memory through the data bus.

The master microprocessor also has the comparatively trivial task of controlling the mixers block. If n separate outputs are required from the synthesizer, then there should be n mixers and amplifiers. The master microprocessor commands the mixers as to which of the m audio outputs from the music generator to sum for each of the n outputs desired. The program panel and cassette input/output blocks are optional. To create new, permanent raw data, the cassette interface will be very useful. This is one of the simplest and most quiet digital data storage mediums available. For the sophisticated user of this synthesizer, the program panel permits reprogramming of the master microprocessor. This should not be necessary for most uses, but this maximum flexibility allows the individual to create his own special synthesizer.

The heart of the synthesizer is the music generator. This is shown in Fig 2. The music generator consists of the music memory and from one to sixteen identical controllers. A high frequency clock drives each controller. Command words coming from the master microprocessor are sent to all controllers in parallel, but the individual controller select lines coming from the master microprocessor are used as strobe pulses to load the command word into the proper controller. Each controller has the capability of using the music memory, but only one controller may use it at any one time. It is a function of the master microprocessor to command only one controller at a time to utilize the music memory. The controller in control of the music memory sends out the address and the control lines Memory Read and Memory Write to the master microprocessor. Likewise, only the controlling controller allows the data coming back from the music memory to enter it. The music memory data line also goes to the master microprocessor. The master microprocessor can access the music memory by commanding one of the controllers to send out the proper addresses, and picking up the data itself. This will be done to load new digital information already created from the cassette interface or the front panel. This two level control eliminates the need for a sixteen bit address bus from the master microprocessor to the music memory.

In practice, the master microprocessor will command each controller to load a portion of the music memory into that controller's memory. Thereafter which each controller can independently run. A preferred method would be to have each controller to have simultaneous access to the music memory. However, just for the read mode, this would multiply the access time of the music memory by up to sixteen, and obviously for the write mode confusion could reign. Therefore, this method of giving each controller its own memory allows the controllers to use the music memory, and to maintain speed and independence while so doing.

Figure 3 shows the music memory. It is made up of 4k ROM and 4k RAM boards. These are standardly constructed as in any computer system. A single bidirectional eight bit data bus is used. Each board has three state data bus drivers activated by the Memory Read signal. Standard ROM is supplied with the instrument with specific raw data preprogrammed. The user may select as much RAM as he desires. This is only needed for creating new data from an audio input or from the cassette interface or front panel of the master microprocessor.

The controller is shown in Fig 4. The controller is based on a microprocessor as well as the master microprocessor. At this point it should be stressed that the choice of microprocessors for the controller and the master microprocessor allows ultimate flexibility for the replicating digital synthesizer. To make modifications, updates, or even to correct production errors, all that need be done is to reprogram (swap ROM chips) the proper microprocessor. The entire synthesizer has been designed with this in mind so that any conceivable circumstance could be implemented by reprogramming if desired.

The command and select lines go to the microprocessor and the response line comes from it to the master microprocessor. The data, address, and control lines also emanate from it to seize the music memory. The high frequency clock signal is fed to a programmable divide by N divider. The output of the divider feeds the clock input of the microprocessor. The jam inputs of the divide by N counter are set by the microprocessor itself. By outputting data onto the jam inputs of the counter, the microprocessor can control the speed of its own clock. This is the straightforward way the different notes are created. The microprocessor will be running a small program to read its memory and output data to the digital to analog converter. No delay loops will be programmed into the microprocessor. Therefore, the rate at which the data is output, and hence its frequency at the output of the D/A converter will be directly a function of the microprocessor clock input. The master microprocessor will command the controller microprocessor to output at a specific frequency to create a certain note; the microprocessor will output the proper jam inputs to the divide by N counter to derive the proper frequency clock to itself to do so.

The output of the D/A converter must go through a low pass filter to reconstruct the proper analog signal. However, the dynamic range of the synthesizer frequency wise is so great that it will be necessary to use different cutoff low pass filters. A digitally controlled low pass filter is used, permitting the microprocessor to control the filter characteristics which will depend upon the note being output. The cutoff will be kept as close to the sampling rate as is possible to maintain the highest fidelity. This will result in approximately the same number of harmonics to be passed out of the system thereby maintaing the original quality of the raw data as much as possible.

The analog signal next passes through a digitally controlled attenuator. The microprocessor can control the output level to create any desired attack, decay curves, or tremelo. The audio then passes through an analog attenuator which is of the standard type used in other synthesizers. This allows simple hand control of these characteristics by the novice operator on the control panel.

Optionally, there is an analog to digital converter feeding the microprocessor. This is used to input new raw data to the controller or the music memory. If this is not required, then the A/D converter need not be installed. Even in a maximum system, it is not anticipated that more than one controller will need to have an A/D converter, however the capability is there to add as many as desired.

The controller microprocessor is shown in Figs 5 & 6. In

Fig 5 the necessary circuitry is shown to allow the microprocessor to access the music memory. The address bus is driven by tri-state drivers controlled by an output line from the microprocessor. The data bus is driven in the same manner for a write operation, but for a read operation all that is needed is to activate the CD4066 switches to put the music memory data onto the microprocessor data bus. The input/output ports are CDP1852 latches. the input port which accepts commands from the master microprocessor is at the lower left. The command bus drives all of the input latches in each controller, but the select line strobes the clock input to the 1852. This puts a low on the SR line which interrupts the uP and sends it to a routine which handles the command information. This method eliminates the need for the microprocessor to poll the command lines when outputting data to the D/A. The output routine will be kept as short as possible to permit the highest sampling rate to be used. The Q output of the microprocessor is fed to the response line when the controller is controlling the music memory.

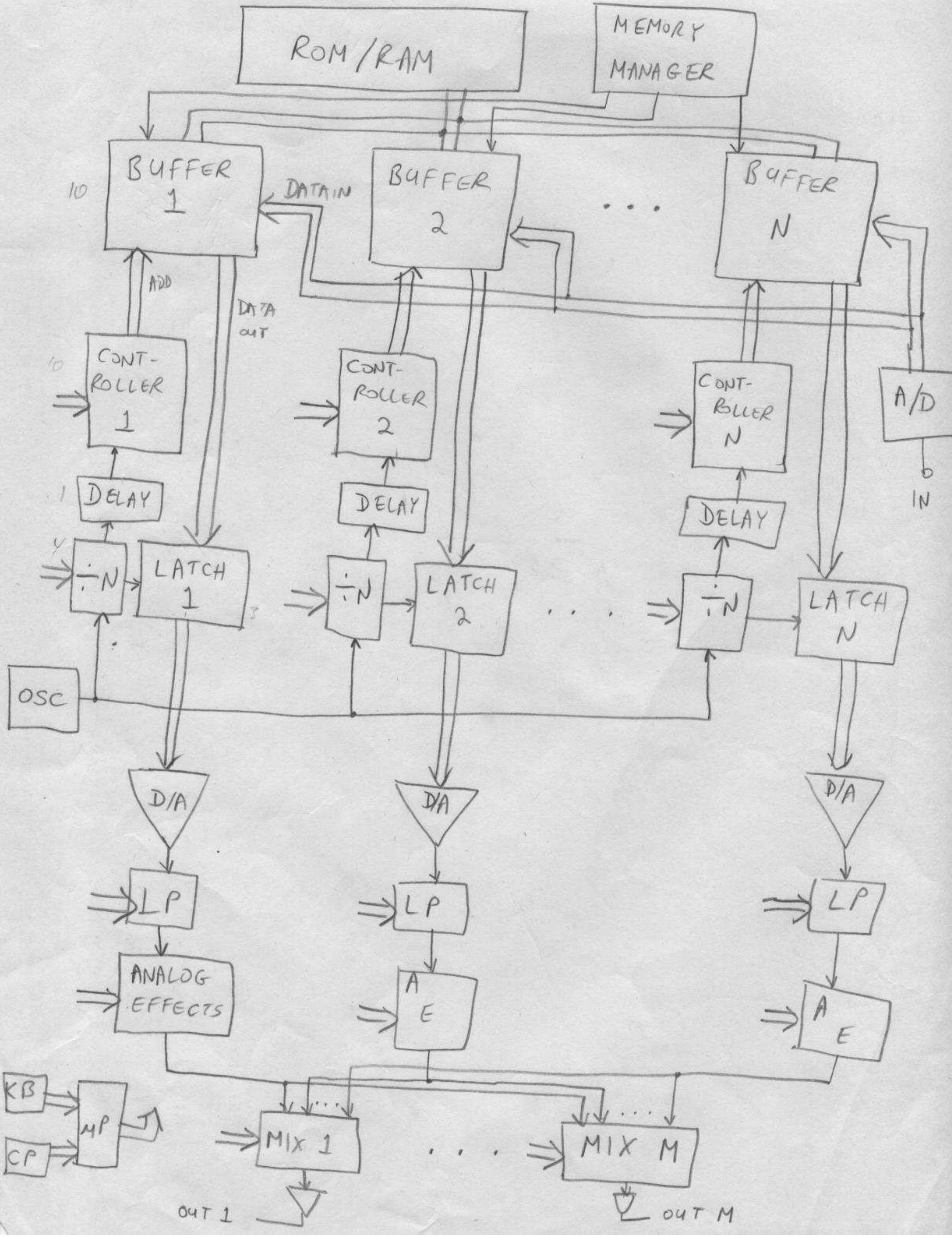
All necessary output ports are implementated as at the lower right with 1852's. The chip select lines of the ports are driven by a CD4028 3 to 8 decoder which is driven by the I/O device select lines of the microprocessor. The device number is assigned by connecting the chip selects to the proper output of the decoder.

As shown in Fig 6, the microprocessor chosen for this application is the RCA CDP 1802. The COS/MOS device draws little current and has high noise immunity. As much of the logic of the synthesizer is COS/MOS as is possible for the same reasons. The

1802 microprocessor is one of the few microprocessors available which can be driven by a clock from DC up to its maximum frequency. This allows the method used to control the notes generated.

Two CD4042 latches are used to latch the high eight address bits. A 1702A ROM is used to store the program. A 4k RAM board identical to those used in the music memory is used for the local microprocessor memory. As many RAM boards as desired may be used, but there should be no need for more than one or two.

It should be clear that much of the implementation of the synthesizer is defined in the programming of the controller microprocessor and the master microprocessor. A considerable amount of software is needed, but this is straightforward. The more of the design that is software, the more flexible the ultimate design will be.


ROM/RAM
MEMORY MANAGER
BUFFER 1
DATA IN
BUFFER 2
...
BUFFER N
ADD
DATA OUT
CONT-ROLLER 1
CONT-ROLLER 2
CONT-ROLLER N
A/D
IN
DELAY
DELAY
DELAY
÷N
LATCH 1
÷N
LATCH 2
...
÷N
LATCH N
OSC
D/A
D/A
D/A
LP
LP
LP
ANALOG EFFECTS
A E
A E
KB
MP
CP
MIX 1
...
MIX M
OUT 1
OUT M

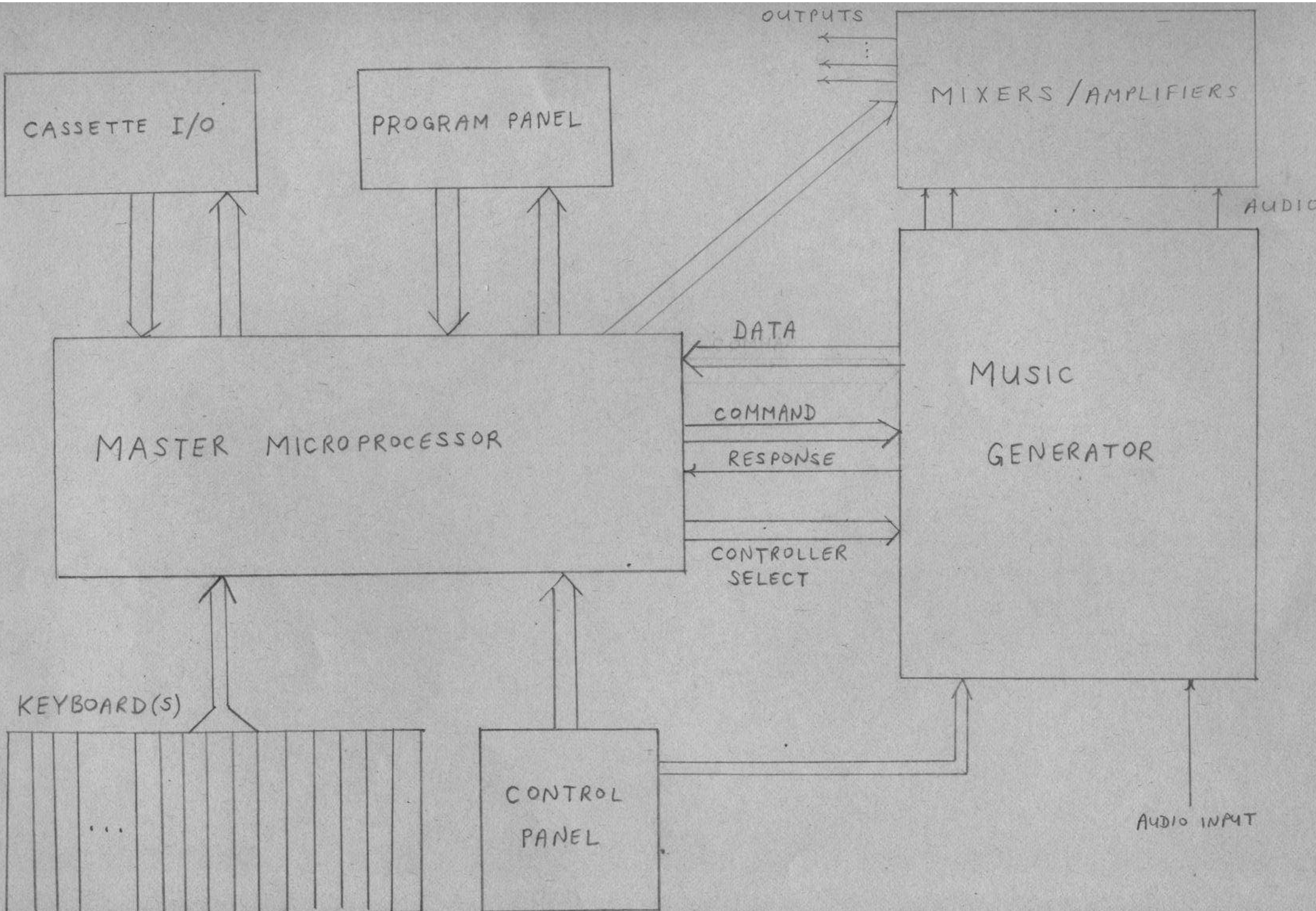
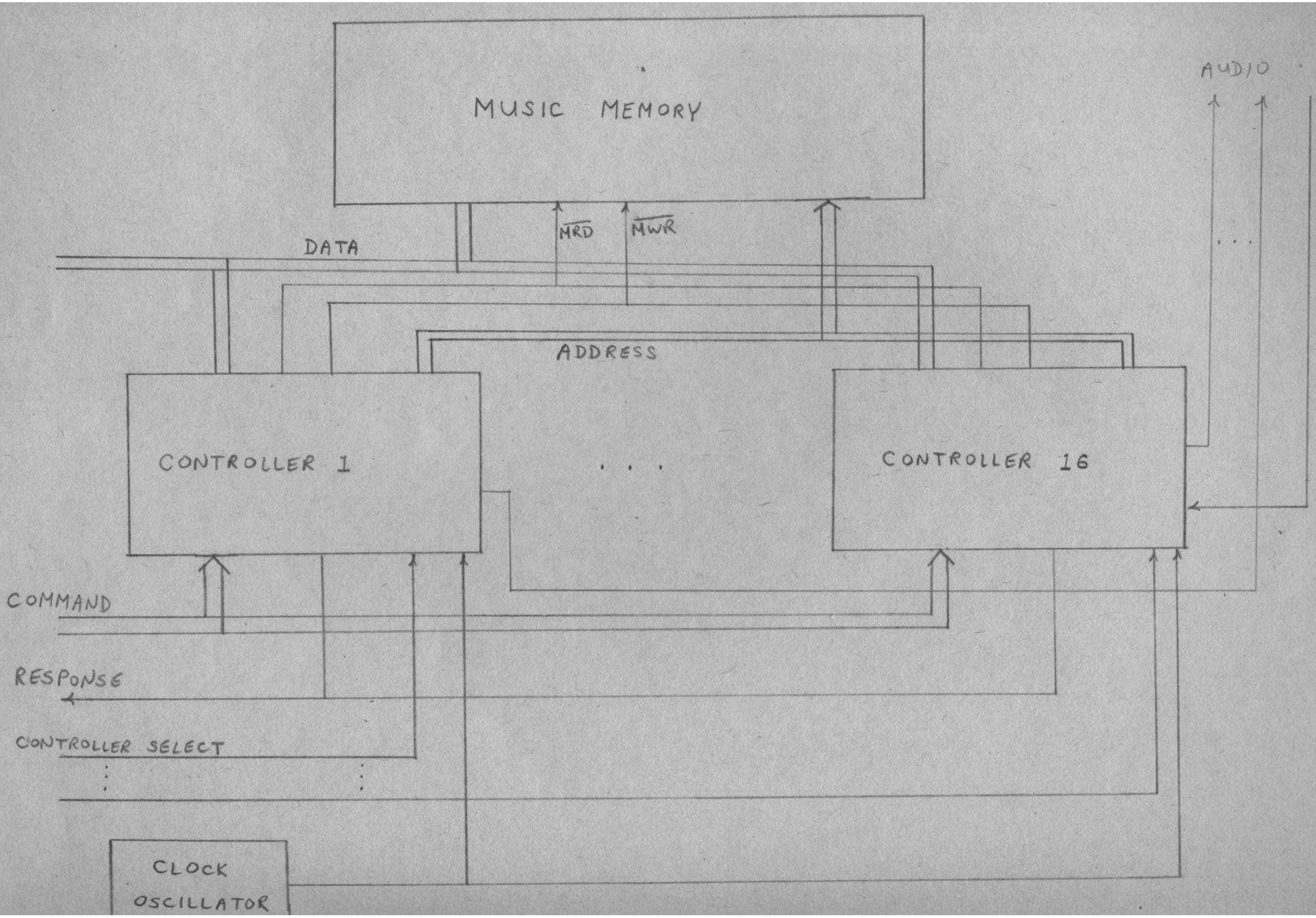



Fig 1 - REPLICATING SYNTHESIZER


MUSIC MEMORY
AUDIO
MRD
MWR
DATA
ADDRESS
CONTROLLER 1
CONTROLLER 16
COMMAND
RESPONSE
CONTROLLER SELECT
CLOCK OSCILLATOR

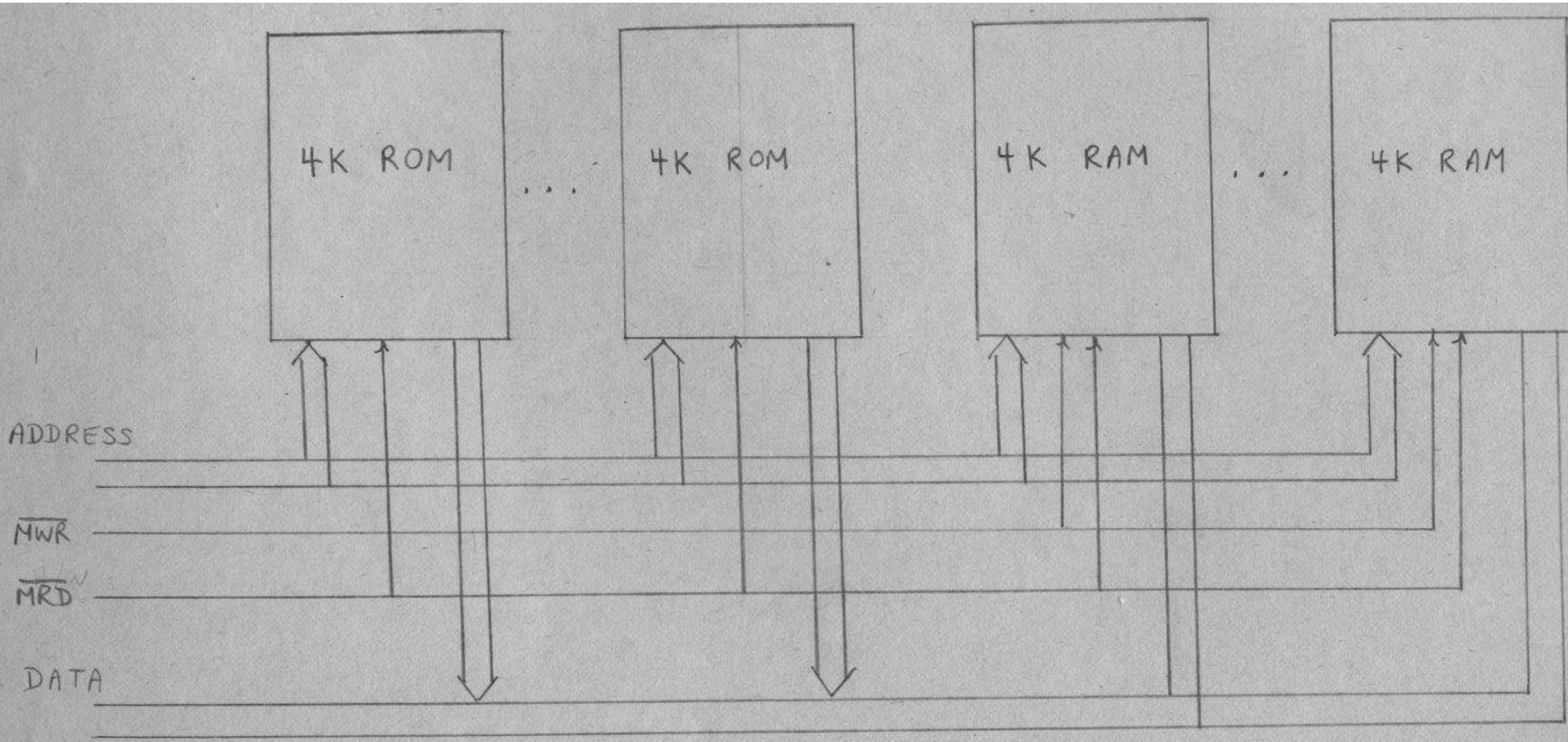
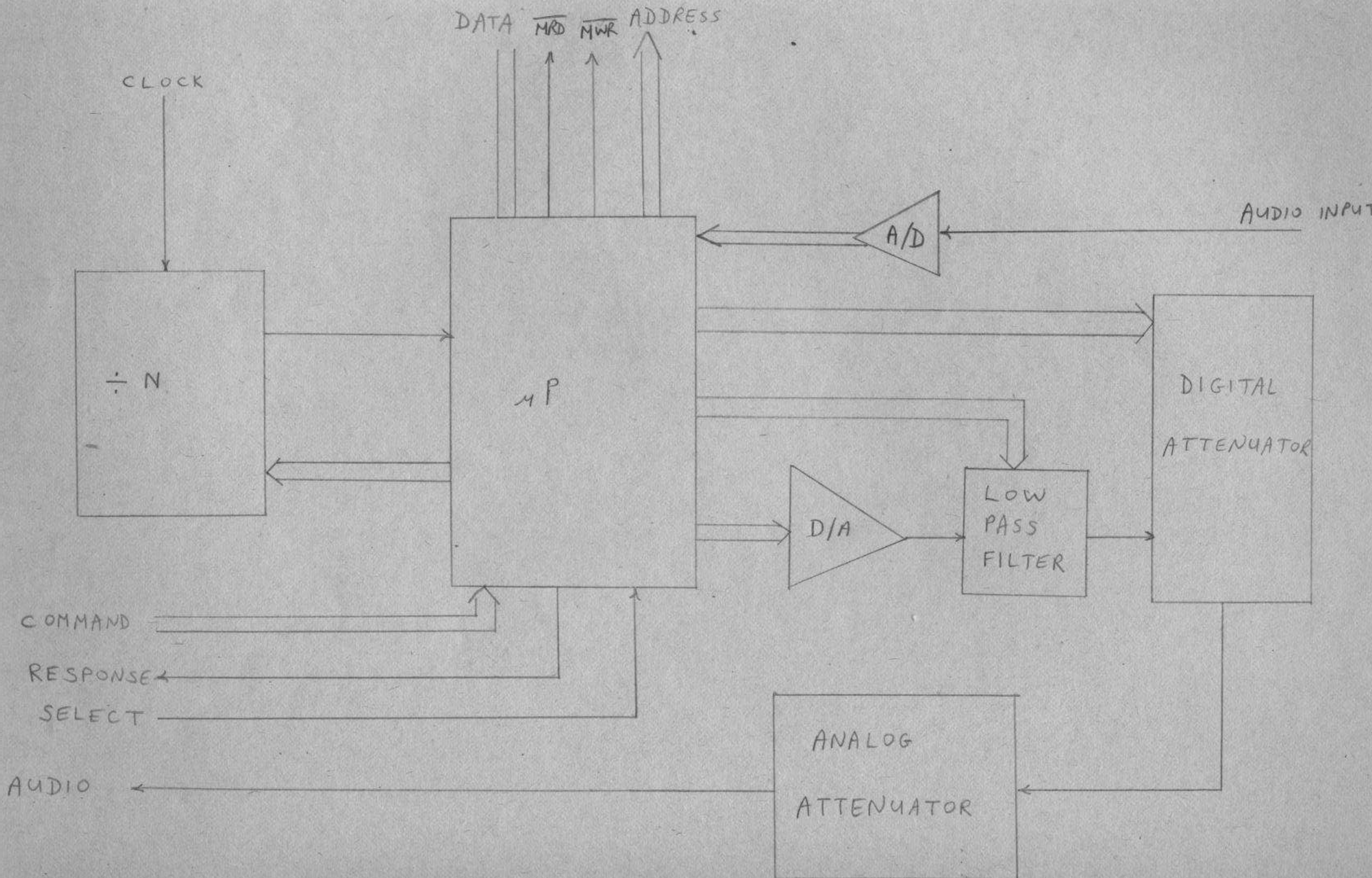



Fig 3 - MUSIC MEMORY

Fig 4 - CONTROLLER



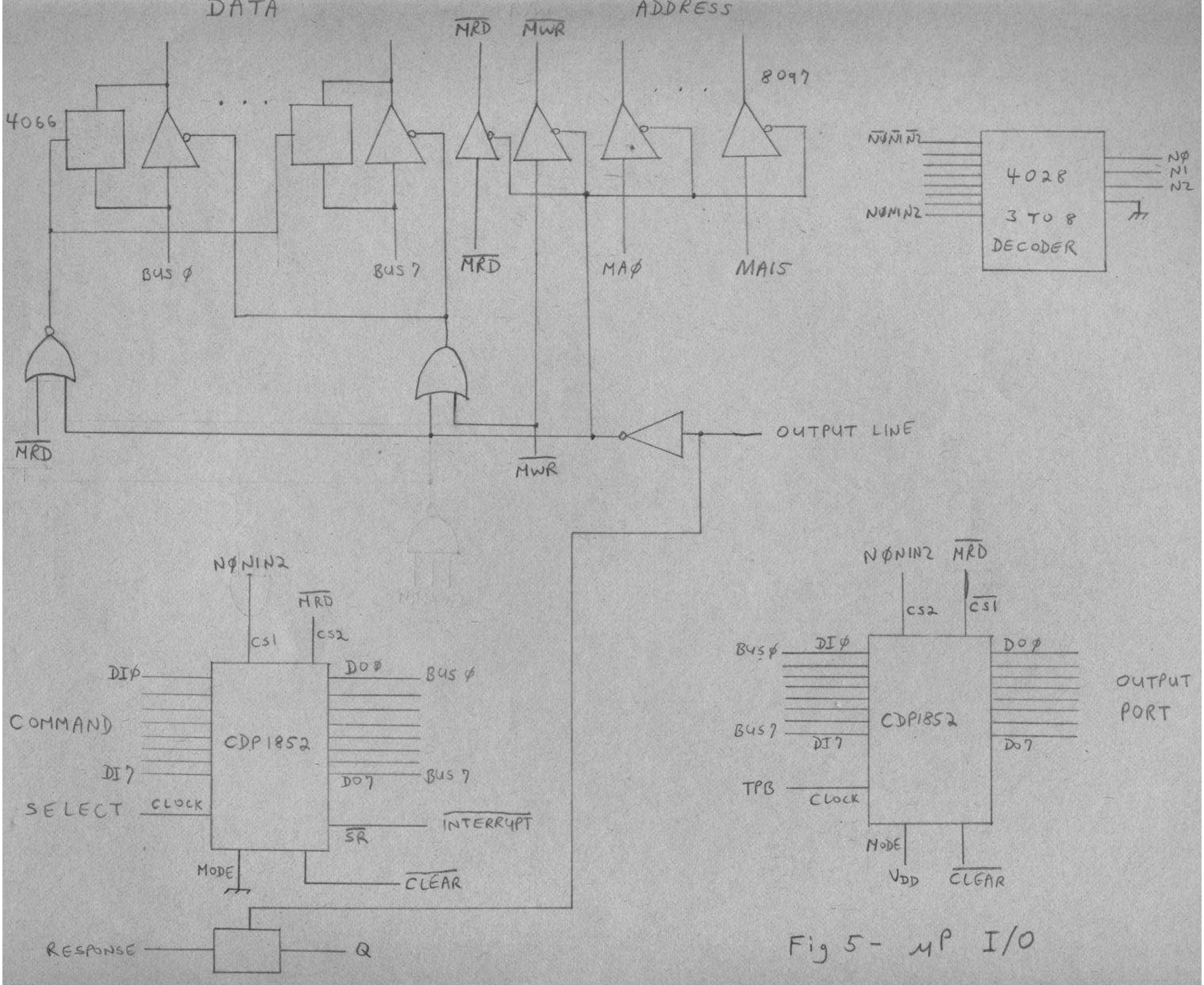
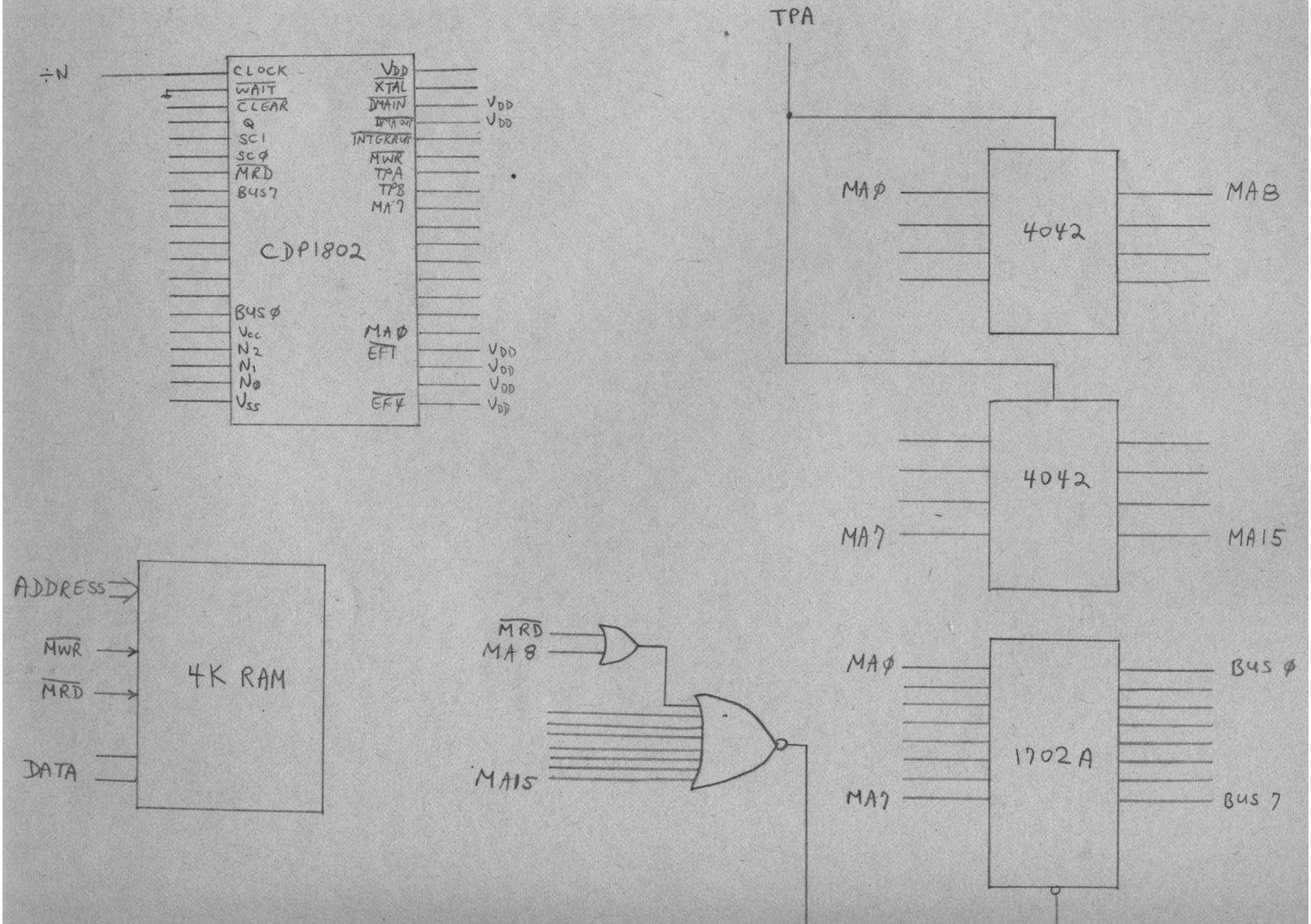

DATA
ADDRESS
MRD
MWR
8097
4066
NØN1N2
4028
3 TO 8
DECODER
NØ
N1
N2
BUS Ø
BUS 7
MRD
MAØ
MA15
MRD
MWR
OUTPUT LINE
NØN1N2
MRD
CS1
CS2
DIØ
COMMAND
DI7
SELECT
CLOCK
CDP1852
DOØ
BUS Ø
DO7
BUS 7
SR
INTERRUPT
MODE
CLEAR
RESPONSE
Q
NØN1N2
MRD
CS2
CS1
BUSØ
DIØ
BUS7
DI7
TPB
CLOCK
CDP1852
DOØ
DO7
OUTPUT
PORT
MODE
VDD
CLEAR

Fig 5 - μP I/O



÷N
CLOCK
WAIT
CLEAR
Q
SC1
SCØ
MRD
BUS7
CDP1802
BUSØ
Vcc
N2
N1
NØ
Vss
VDD
XTAL
DMAIN
MWR
TPA
TPB
MA7
MAØ
EF1
EF4
VDD
VDD
VDD
VDD
VDD
VDD
TPA
MAØ
4042
MA8
4042
MA7
MA15
ADDRESS
MWR
MRD
4K RAM
DATA
MRD
MA 8
MA15
MAØ
1702A
MA7
BUS Ø
BUS 7


Fig 6 - μP