Università di Napoli

Dipartimento di Scienze Fisiche

AIMI - Associazione di Informatica Musicale Italiana

International Workshop on models of singing voice and musical sounds

Sorrento (Naples)
28-29-30 October 1988

This workshop is promoted by Department of Physics - University of Naples and by AIMI (Italian Computer Music Association) in cooperation with other scientific organizations.

The purpose of the workshop is to estabilish the state-of-the-art in the representation of musical signals and to provide a forum for discussion among participants and invited lecturers in the fields of music, digital signal processing,phonetics,psychoacoustics and computer science.

In addition to the plenary lectures, time will be made available for shorter contribution related to the program. Partecipants are encouraged to present contributed papers.

The workshop will be held at Caravel Hotel,c.so Crawford 69 -telephone 081/ 8782081 - Sorrento -(Naples). Sorrento is 1/2 hour by train from Naples.
The workshop will begin on Thursday, 27 October 1988 with afternoon registration.Technical events will start on the morning of Friday 28 October.

Introduction

New models of the singing voice and musical instruments are the result of applications developed in a wide spectrum of disciplines.

The representation of musical sounds is a fundametal problem, both theoretical and applied point of view. In recent years, several new representation techniques have been devised.

It is worthwhile to compare and verify these new techniques with classical methods of spectral estimation. Furthermore,perceptual criteria in modeling of the singing voice and musical instruments, may act as a link between micro and macro structural representations.

A Multi-DSP System for Audio Synthesis and a Graphic Editor for Synthesis Algorithms

Leonello Tarabella *

Graziano Bertini **

Special thanks to

Ing. Stefano Toni

Dr. Andrea Lombarini

* Computer Music Dept. of CNUCE/CNR - PISA, via S.Maria 36

** Digital Signal Processing Group of IEI/CNR - PISA, via S.Maria 46

A Multi-DSP system for Audio Synthesis and a Graphic Editor for Synthesis Algorithms

Leonello Tarabella
Computer Music Dept. of CNUCE/CNR, Pisa

Graziano Bertini
Signal and Image Processing Dept. of IEI/CNR, Pisa

Special Thanks to
Ing. Stefano Toni
Dr. Andrea Lombardini

A system for digital signal processing, designed around microprocessor TMS320C25 to be used in acoustic and musical activities, is described. The system consists in a Multi-DSP architecture made of a single Master processor and a number (up to 16) of Slave processors: Master processor performs communication with Host Computer, D-to-A and A-to-D conversion, and control of Slave processors. Each Slave Processor, being programmable, behaves as an highly sophisticate co-processor; communication with Master Processor and single Slave processor is performed using Dual Port Ram devices. Further, Slaves Processors can communicate among themselves using programmable up to 5Mbit/sec. serial lines.

The system was designed to implement any kind of synthesis algorithm and digital filter for audio signals (e.g. FM, Karplus-Strong, etc.). In order to get easier the definition of algorithms and filters, a prototype of a graphic editor was developed on MacIntosh computer: using the mouse, is possible to place onto the graphic plane basic symbols (generators,envelope shapers, delays, adders, etc.) taken from a palette, and to link them using appropriate tools (pencil, rubber, etc.). The editor has an on-line *syntactic analyzer* that guarantees the correctness and consistence of the algorithm drawn. When ordered, the drawing is translated into a meta-language resembling very closely to machine code: a specific mapping operation issues the binary code for a particular µprocessor (like the TMS320C25 itself).

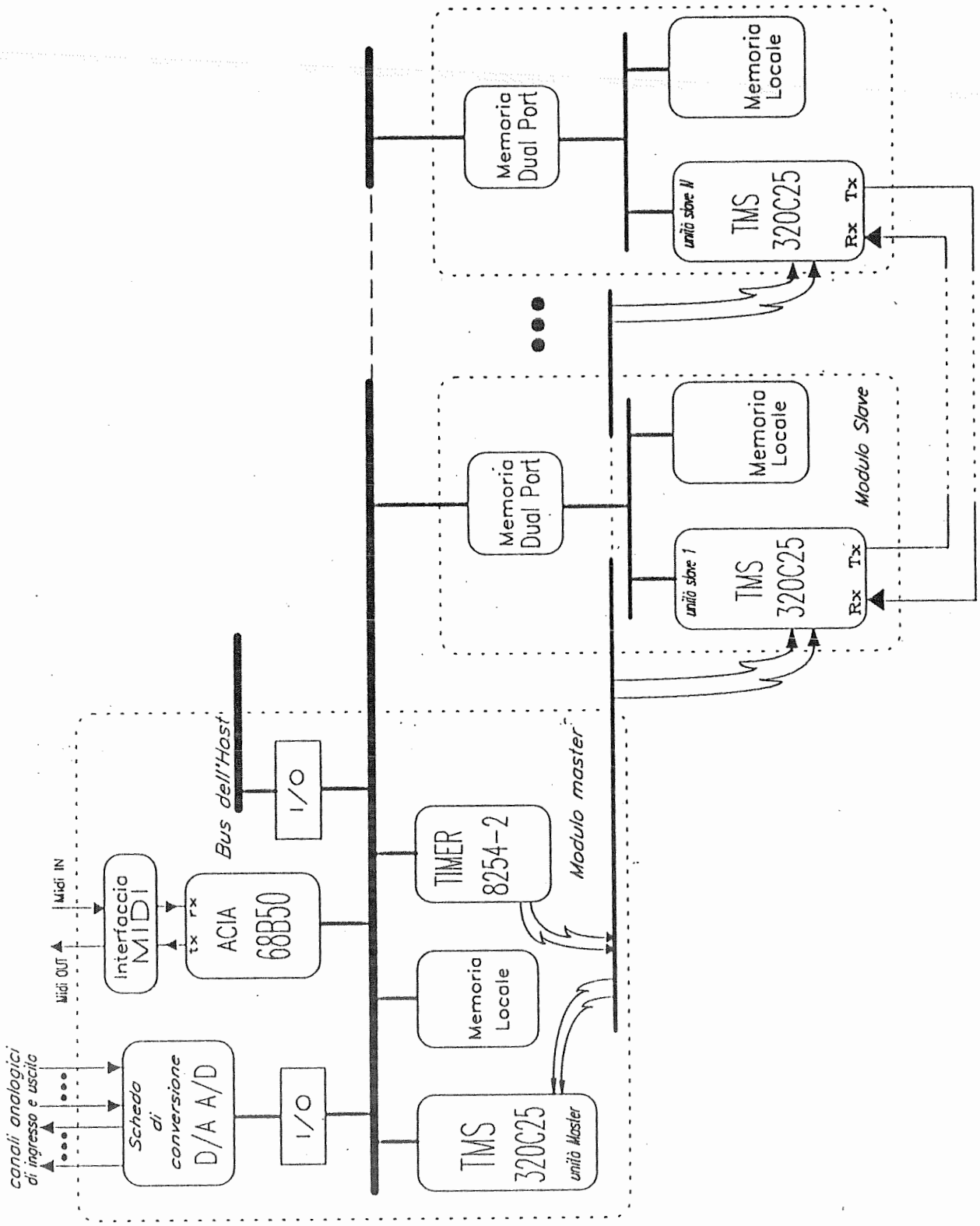
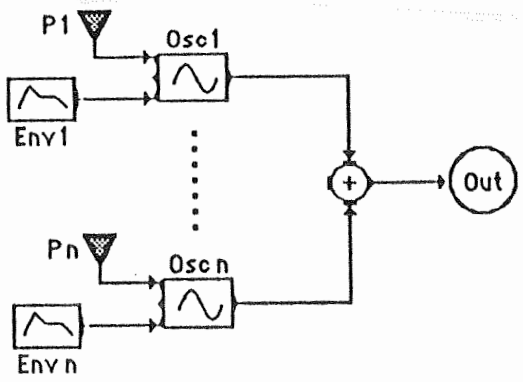
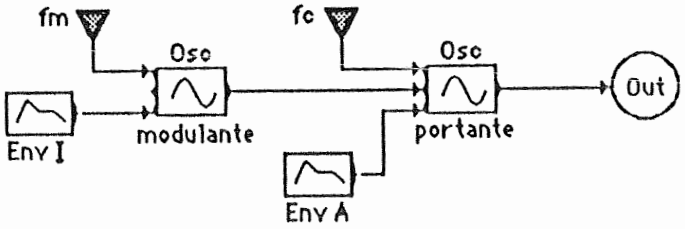


Fig.0.1 schema a blocchi dell'architettura del sistema

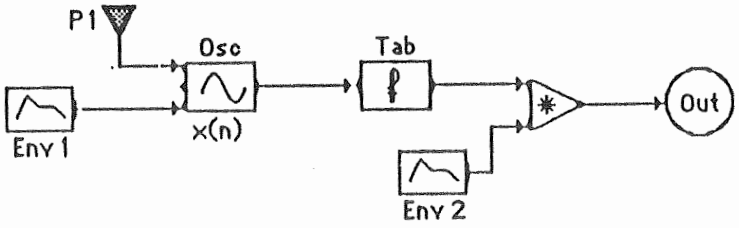
Algorithms



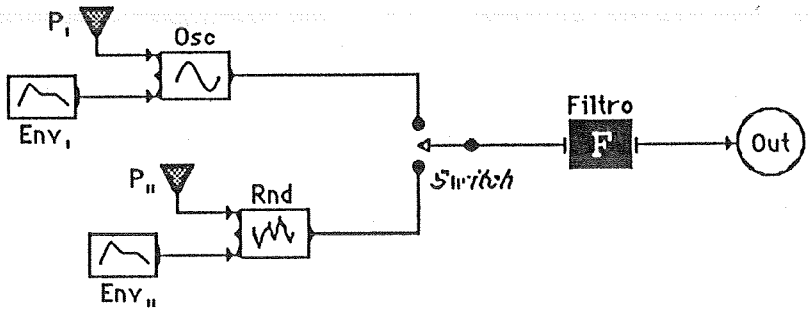
Additive



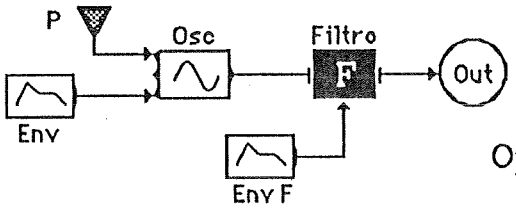
FM



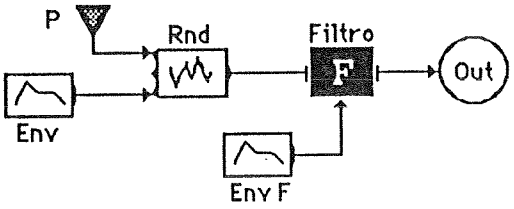
WaveShape



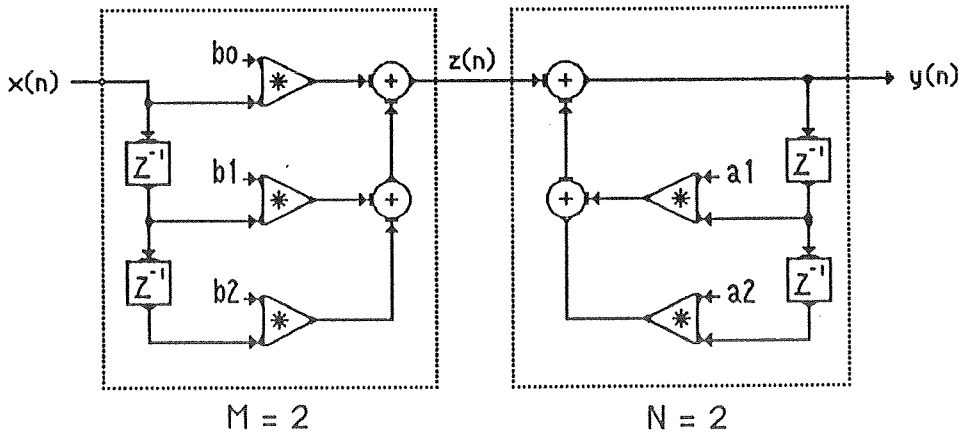
LPC

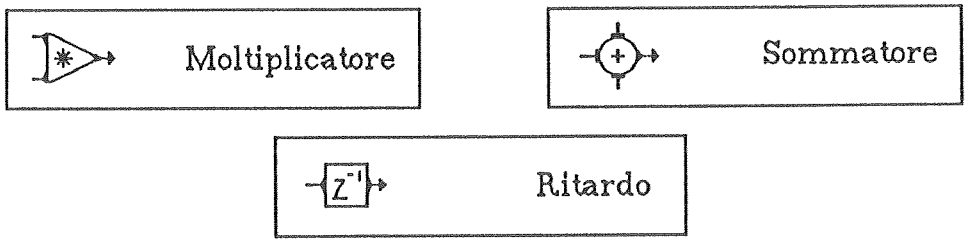


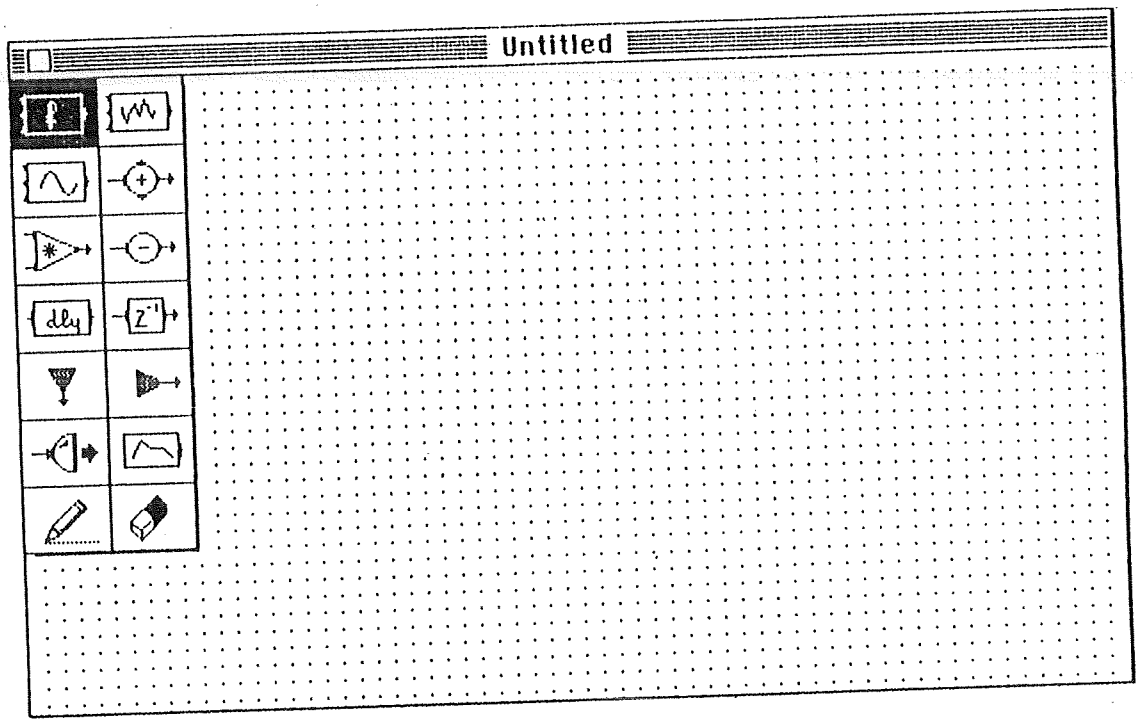
Oppure:



FIR





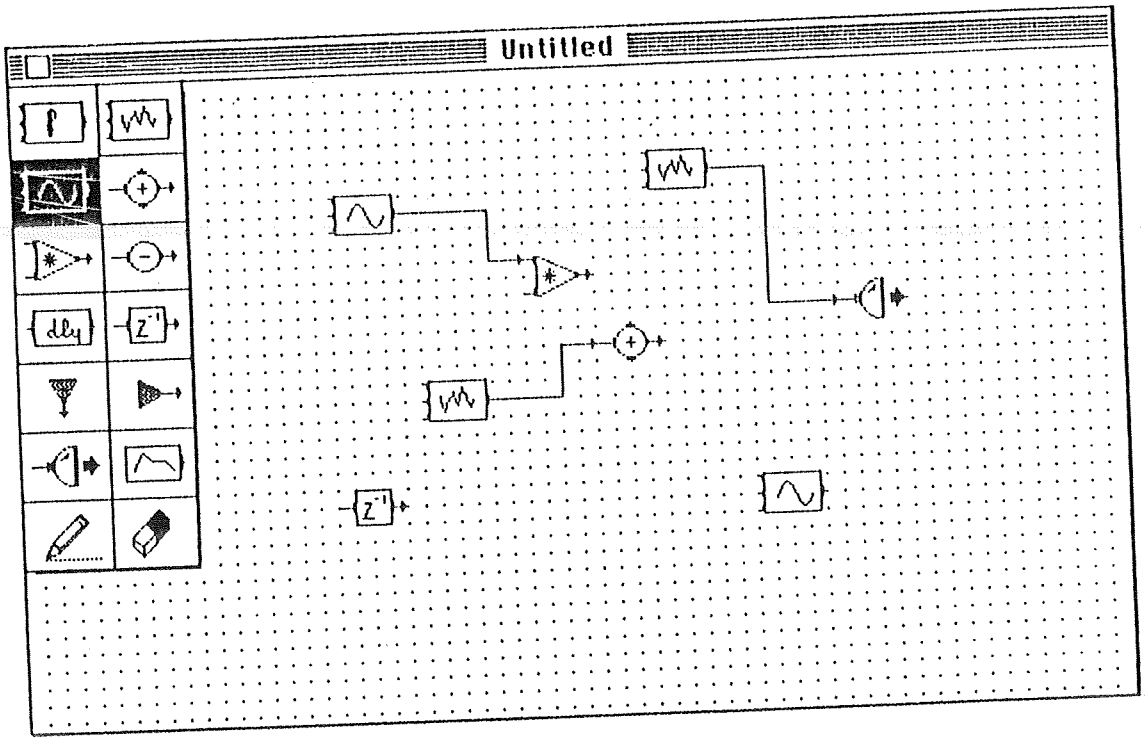


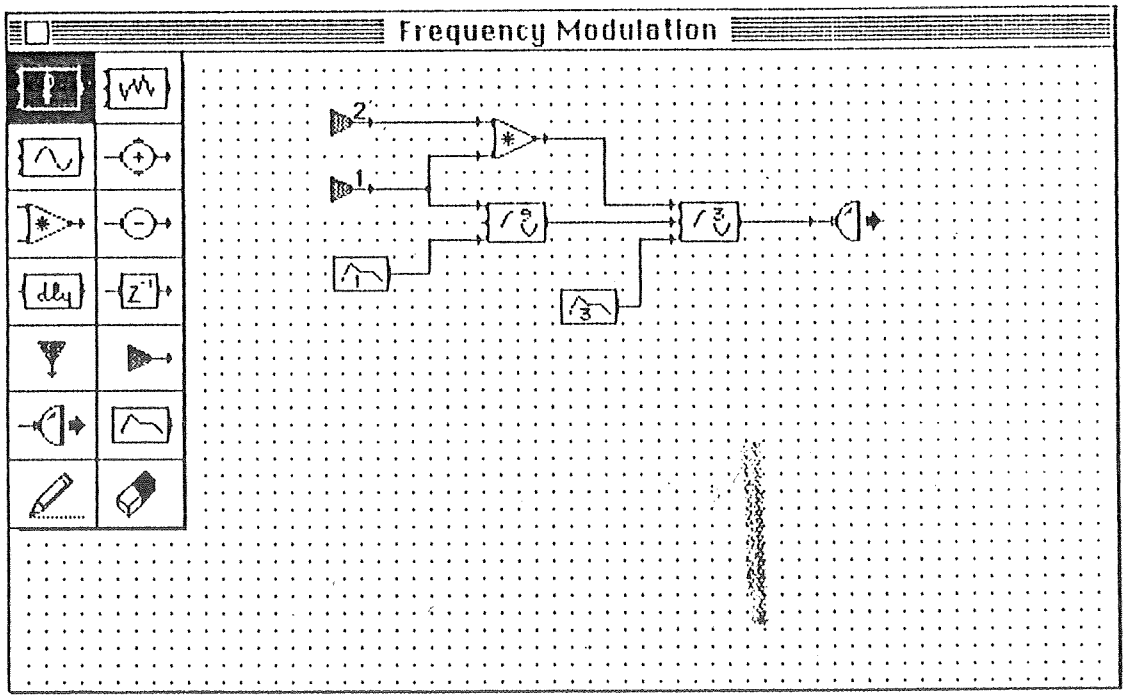
Tipo	Simbolo	Ingressi	Uscite
Oso			
Tab			
Mul			
Dly			
In1		∅	
Out			∅

Tipo	Simbolo	Ingressi	Uscite
Rnd			
Sum			
Min			
Zet			
In2		∅	
Env		∅	

Atoms

Graphic Tools


Untitled


Frequency Modulation

sfruttar

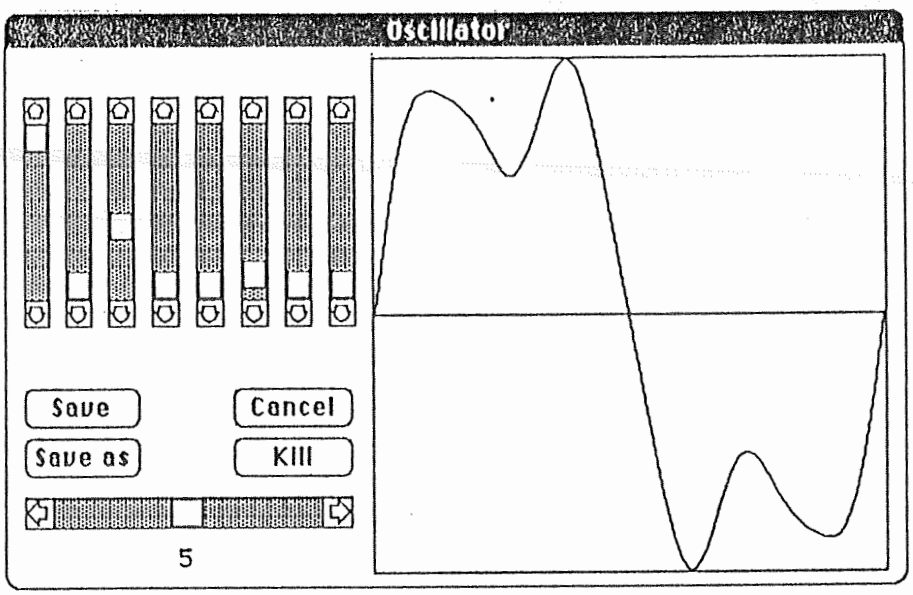
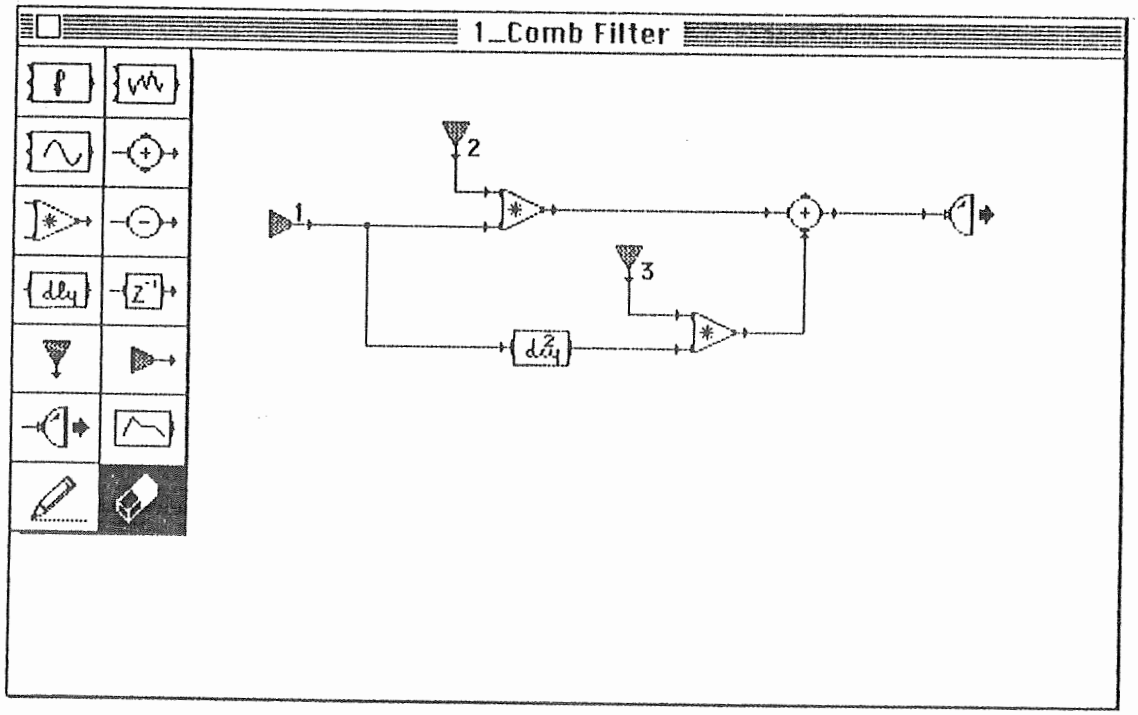
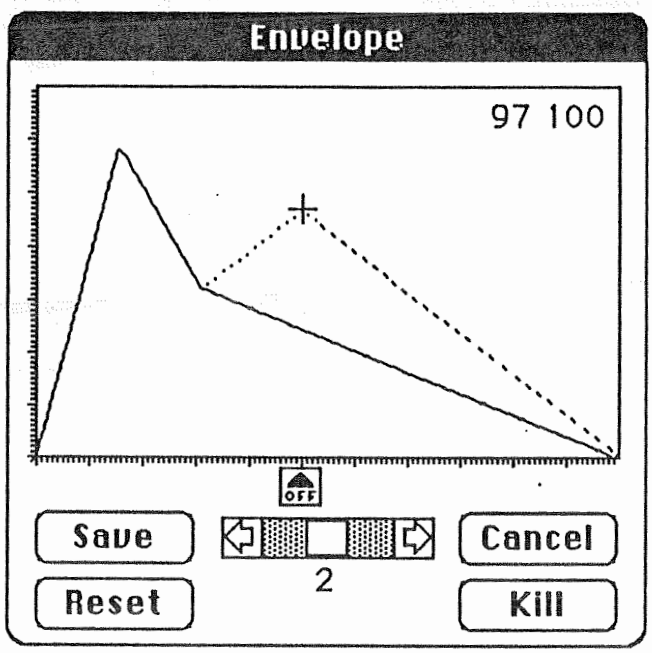
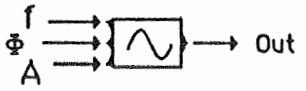


Fig 3.11 Finestra per la definizione dell'oscillatore.

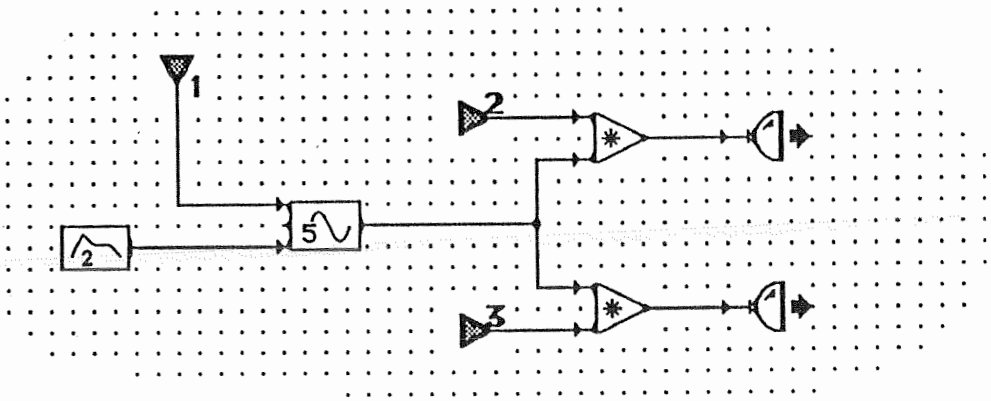


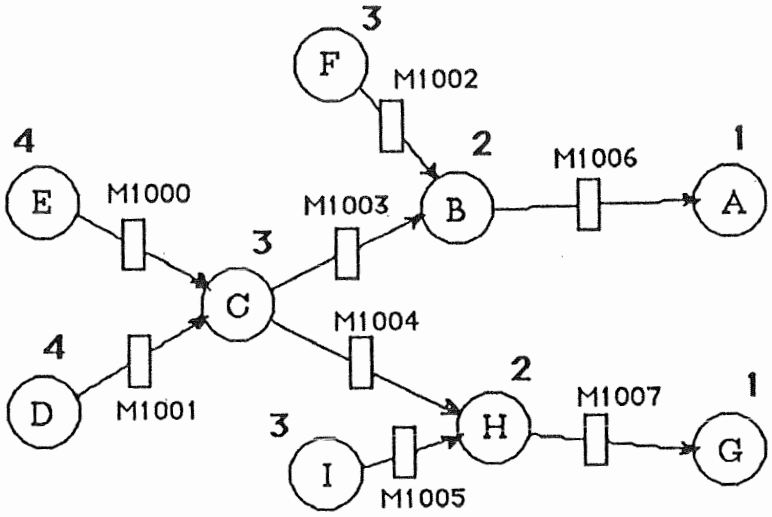




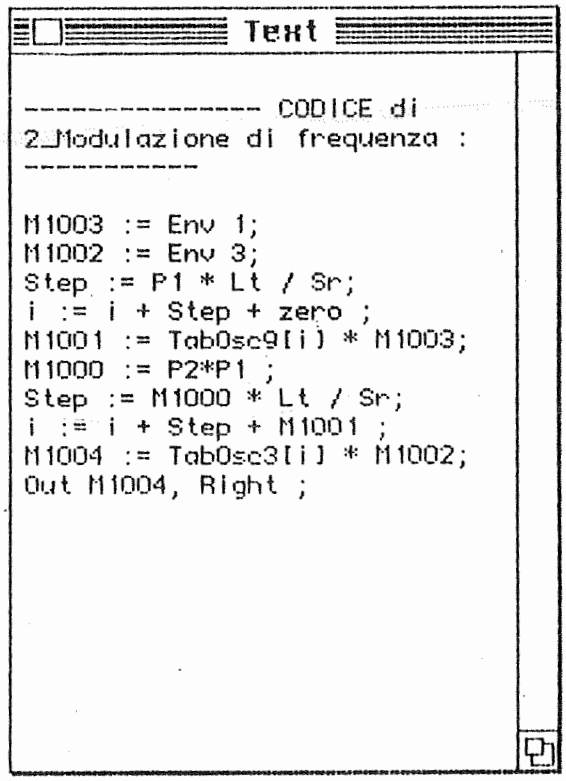

```
repeat
  Step := f * lt / Sr;
  Fase := (Fase + Step + Φ) mod 2^n;
  Out := Tab[Fase] * A;
until False;
```

TIPO	Blocco di Istruzioni Relative
Tab :	? Ininfluente; Acc := TabFun [?]*?;
Osc :	Fase := ?*Lt*Sr; i := i+Fase+?; Acc := TabOsc D [i]*?;
Mul :	Acc := ?*?;
Dly :	Acc := ? Delayed of D*1024 Samples;
In1 :	Acc := P D;
Out :	Out ? ;
Rnd :	Acc := TabRnd[?*t + ?]*?;
Sum :	Acc := ?+?+?;
Min :	Acc := -?;
Zet :	Acc := Buf; Buf := ?;
In2 :	Acc := P D;
Env :	Acc := Env D;






```
M1002 := P2;
M1000 := P1;
M1001 := Env 2;
Step := M1000 * Lt / Sr;
i := i + Step + zero ;
M1003 := TabOsc5 [i] * M1001;
M1004 := M1003;
M1006 := M1002 * M1003;
M1005 := P3;
M1007 := M1004 * M1005;
Out M1007;
Out M1006;
```


Text
---------- CODICE di
2_Modulazione di frequenza :

M1003 := Env 1;
M1002 := Env 3;
Step := P1 * Lt / Sr;
i := i + Step + zero ;
M1001 := Tab0sc9[i] * M1003;
M1000 := P2*P1 ;
Step := M1000 * Lt / Sr;
i := i + Step + M1001 ;
M1004 := Tab0sc3[i] * M1002;
Out M1004, Right ;