

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets

(11)

Publication number:

0 127 586
A2

(12)

EUROPEAN PATENT APPLICATION

(21)

Application number: 84830087.7

(51)

Int. Cl.³: **G 10 H 7/00**

(22)

Date of filing: 26.03.84

(30)

Priority: 01.04.83 IT 1790483

(43)

Date of publication of application:
05.12.84 Bulletin 84/49

(84)

Designated Contracting States:
AT BE CH DE FR GB LI LU NL SE

(71)

Applicant: "ISELQUI"-S.p.A.
Via delle Torri, 32
I-63100 Ascoli Piceno(IT)

(72)

Inventor: Pezzani, Enzo
Via Petroniano, 5
Massa(IT)

(72)

Inventor: Bosetto, Antonio
Via Bellini, 1
Osimo (AN)(IT)

(74)

Representative: Baldi, Claudio
Viale della Vittoria 97
I-60035 Jesi (Ancona)(IT)

(54)

Polyphonic digital synthesizer.

(57)

This invention relates to a polyphonic and polytimbric digital synthesizer characterized by the definition of the synthesizer sound through a finite set of (n) periodic wave forms and an algorithm that defines the evolution of the sound.

Each sound is generated by the association in real time of the timbre and the frequency relative to it.

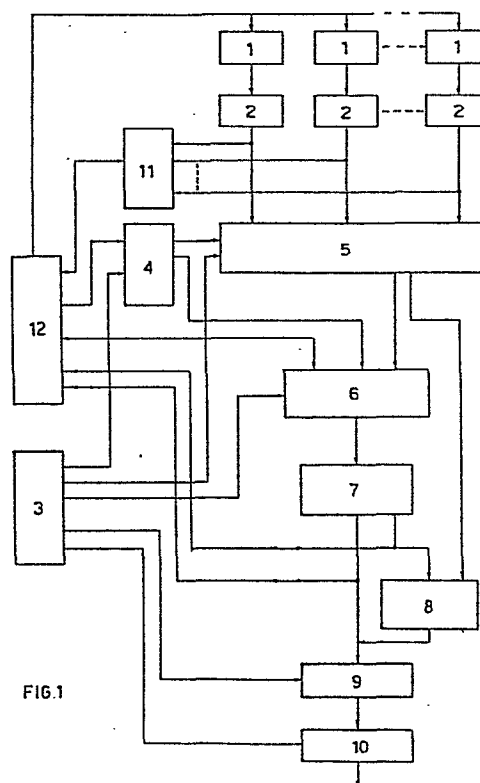


FIG.1

EP 0 127 586 A2

see front page

see front page

- 1 -

Polyphonic and polytimbric digital synthesizer.

This petition of patent for industrial invention is related to a polyphonic and polytimbric digital synthesizer: it simultaneously generates various sounds eventually distinguished by different timbres.

5 The generation of a sound means the building of an electric signal that, amplified and sent to the loud-speaker, will originate the sound itself, in this particular case, having a digital synthesizer, it means to build the patterns of that signal at a certain patterning frequency (from now on referred to as FS) that the last step of the system, the
10 analog digital converter, will convert to the analog signal that can be sent to an amplifier and then to the loud-speaker.

The synthesizer determines, for each sound to be generated, the frequency patterns (FS) of the relative signal, in every patterning moment it adds up the patterns relative to the various signals, originating the patterns of the complex signal, sum of the signals that make up the single sounds, this is so because the patterning operation is linear and therefore the sum of the patterns of the single sounds is equal to the patterns of the sum of the signals.

A signal is characterized by its fundamental frequency (FF) and by its timbre, that is by the time evolution of its spectral composition.

The system stores the signal timbres to be synthesized and at the moment of synthesis it generates the sound associating timbre and fundamental frequency.

The timbres are stored in this way: starting from the time evolution

analysis of the spectral composition, the period is subdivided into a number (n) of intervals delimited by the times: (t1, t2, ... tn) and in each interval (ti-1, ti) this time evolution is approximated attaching to the signal on the whole interval the spectral composition it has at the center of the interval.

For each of the timbres that the synthesizer makes available the following is stored:

a) the patterns on a period of (n) periodical wave forms of period normalized to (1), with the spectral composition given to the signal in the (n) intervals as defined above, later on it will be illustrated how, starting from these patterns, it is possible to synthesize periodical signals characterized from those spectral compositions and from different fundamental frequencies.

b) an algorithm that defines the times and the modalities with which the wave forms referred to in point a) come together to create the signal.

This algorithm is implemented in the system through a simple language: for each timbre there exists in the EPROM memory a coded program in this language that is interpreted by the processing unit supported by a microprocessor (CPU).

The microprocessor loads into a RAM area, that we will name RAMTIMBRI, the wave forms relative to the timbres that the user selects as immediately executable and it coordinates the sound generation programming the fundamental frequency and executing the operations defined by the timbre algorithm.

Sounds distinguished by the timbres whose wave forms are loaded in the RAMTIMBRI can be synthesized, at any rate the (CPU) can still load a new timbre in the RAMTIMBRI, substituting it, if the memory is completely occupied, with one of the timbres already present, even when the synthesizer is generating sounds.

All the sounds are built by the withdrawal of patterns stored in the

RAMTIMBRI: the microprocessor based on the signal timbre makes it possible to withdraw patterns of the wave forms relative to it and, interpreting the program that implements the timbre algorithm, it makes it possible to access at any moment the appropriate wave form
5 between those relative to the timbre.

It was above mentioned that a signal is characterized by a timbre and a fundamental frequency (FF).

10 The system provides a set of (p) programmable oscillators; they are programmed by the microprocessor such as to generate a frequency oscillation equal to (m) times the frequency (FF), where (m) is the number of patterns that define the wave forms.

15 This oscillation is utilized by a dedicated digital hardware (HW1), substantially a modulus counter (m), to sequentially address all the patterns of the wave form selected by the microprocessor.

Scanning the (m) patterns of the wave form with frequency (m x FF), the
20 repetition period of the scanning cycle is equal to (FF), that is the generated signal has fundamental frequency (FF).

Summarizing, during the synthesis of a signal, at every moment there is an addressing of the pattern stored in the RAMTIMBRI: the (CPU)
25 provides the addresses that locate the wave form to access, the hardware (HW1) provides the addresses to determine which one of those of the wave form is the concerned pattern.

All this happens for all the signals that have to be utilized simultaneously.
30

Following is the description of the operation of pattern determination of the various signals at the pattern frequency (FS), and their sum.

35 For this operation we have another block of dedicated hardware (HW2), driven by an oscillator of frequency (q) times (FS), where (q) is the maximum number of signals that can be added up (degree of polyphony of the synthesizer).

1. F. U. H. - M. V. 1
Don (ing) BALDI
1970

During each of these time slots the hardware (HW2) executes the following operations for the determination of the pattern of the relative signal:

- After all these operations are executed in all the time slots, on all active signals, at the end of the patterning period the accumulator contains the sum of the patterns of all the signals: at this point the accumulator contents are given to the converter, the accumulator is reset to zero, and a new patterning cycle is started.

The described time slot structure allows for an interpolation between two wave forms of a timbre, permitting a gradual transition from one to the other.

This is possible when the number of the present sounds is small; in this case it is possible to assign more than one time slot to each signal, so that in the summing phase its patterns are added up various times (a convenient scale factor acting on the final signal makes up for this fact).

When the (CPU), implementing the timbre algorithm, has to carry out the

transition from one wave form to another, this commutation is executed on a "copy" of the signal at the time, that is on one slot at the time of the slots assigned to that signal.

- 5 The result is that in the transition phases the same signal appears coming out with the patterns of two different wave forms, thus carrying out an interpolation between the two waves.

The maximum degree of interpolation comes up in the presence of a
10 single signal, that occupies all available time slots: in this case the transition occurs in (q) (time slots number) subsequent steps, with the wave of the final wave form always greater.

Another peculiar feature of the synthesizer is the way in which the wave
15 form patterns which make up the timbre are stored to minimize the required memory quantity.

It was mentioned above that for each wave form (m) patterns are stored; the compacting form of the adopted memory permits to reduce these
20 patterns to (m/r) and to calculate in real time the (m) global patterns through an interpolation technique approximated between each of the (m/r) stored and the next.

The technique used consists of:

- 25 a) a set of (s) values ($I_1, I_2, \dots, I_s$) has been appointed and, in correspondence of each of the values (I_i), (r) values ($0, I_i/r, 2 I_i/r, \dots (r-1) I_i/r$) have been stored in a PROM memory.

b) in the off-line generation phase of the (m/r) patterns of the wave
30 forms to be stored, for each of them the difference is calculated:
 $A_a = \text{value of next pattern} - \text{value of pattern}.$

c) for each pattern the value (I_i) closer to the value (A_a) relative to it is determined: (l_i) is assumed as an approximated increment
35 between one pattern and the next,

d) still in the pattern generation phase the index (i) of the value (I_i) determined as mentioned in part c) is stored together with the

values of the (m/r) patterns.

e) in real time the hardware block (HW2) calculates the value of the patterns of the various signals in the following way:

5 1) gets $(i, j = \log_2 m)$ address bits provided by the hardware (HW1) relative to the signal in consideration; uses (i, k) most significant bits, where $(k = \log_2 m/r)$ to get one of the (m/r) wave form patterns stored in the RAMTIMBRI; moreover it gets the index (i) of the increment (I_i) to use: this increment is expressed in $(h = \log_2 s)$ bits;

10 2) uses the (h) bits above defined and the remaining $(j - k = \log_2 r)$ bits provided by the hardware (HW1) respectively as high and low part of the address with which one can access the PROM increments memory: the value obtained is added up to the one obtained from the RAMTIMBRI thus building the pattern of the signal.

15 Another feature of the system is the fact that the commutation between the two wave forms of a timbre always occurs in a situation in which the patterns of the wave forms are both (0), avoiding any discontinuity in the resulting signal.

20 This function is given to a hardware modulus (HW4) which verifies the addresses provided by the (HW1) moduli and signals to the microprocessor where to carry out the commutation.

25 We now continue with the description of the block diagram (fig. 1):

30 1) Programmable oscillators, programmed by the processing units for the generation of oscillators of frequency equal to (m) (number of the patterns that define the wave forms) times the fundamental frequency (FF) of the signal to synthesize.

35 2) Moduli (HW1), driven by the previous, they are modulus counters (m) that scan, for the associated signal, the patterns of the wave form selected by the microprocessor, generating some of the addresses (the rest are generated by the microprocessor) which are stored in the RAMTIMBRI memory.

3) Modulo (HW2): it's the block that coordinates the building of the patterns at the patterning frequency (FS) of the active signals, their sum and their conversion in analogic signal.

5 4) Modulo (HW4) programmed by the microprocessor, makes available for block (HW2) the following information concerning each time slot:

- a) which (HW1) is associated to the time slot;
- b) if there is an active signal associated to the time slot;
- c) which timbre is associated to the time slot;

10 d) which wave form of the timbre is associated to the time slot;

5) Modulus of address patterning: in each time slot it patterns and makes available the exits of block (HW1) associated to the time slot.

15 6) Modulus (HW3), handles the access to the patterns of the wave forms stored in RAMTIMBRI; it ensures the microprocessor of the transparent access to RAMTIMBRI and, during the adding phase, it carries out the correct sequencing of access to RAMTIMBRI and to PROM.

20 7) RAMTIMBRI memory: it stores in compact form the patterns of the wave forms of the rimbres immediately executable.

It provides part of the address with which to access the PROM increments memory.

25 8) PROM increments memory, in which are stored (r) values for each appointed increment (Ii): in the accessing phases, part of the address is obtained from RAMTIMBRI, and another part is provided by modulus (5).

30 9) Adder-accumulator block: driven by (HW2) it sums up the patterns of the active signals during the patterning period (TS), it makes available the frequency sequence (FS) of the patterns of the sum signal.

35 10) Converter block: it converts the frequency sequence (FS) into the analogic sum signal. It moreover provides a width check to account for the fact that the same signal might have been added up various times to allow for interpolation between two wave forms of the timbres in the transition between one and the other.

STUDIO CONSULTING
BY STEPHEN
DATE 09/10/80 PA: DI
V. 1.0

11) Block (HW5): modulus programmed by the microprocessor which controls the exits of the moduli (HW1) relative to the various signals, and it can tell the central unit, if properly programmed, that the addressed pattern of a certain signal is (0).

12) Central processing unit, which consists of a microprocessing system that manages the whole sound generation system.

CONSULENZA
VERBOSI
BALDI

- 1 -

Claims

1) Polyphonic and polytimbric digital synthesizer characterized by the definition of the timbres of the sounds to be synthesized through a finite set of (n) periodic wave forms and an algorithm that defines the evolution of the sound. Each sound is generated associating in
5 real time the timbre and the frequency relative to it. The wave forms that define the timbre are determined in the following way: starting from the analysis of the evolution in time of the spectral composition of the signal to synthesize, the length of the sound is subdivided in a number (n) of intervals and in each of these such evolution is approximated giving to the signal a constant spectral composition: these
10 (n) spectral compositions are memorized by the system as (n) periodical wave forms. Of these the system memorizes (m) patterns on a period considered normalized to (1). Scanning the (m) patterns with scanning frequency ($m \times FF$) one obtains signals with that spectral composition and with fundamental frequency (FF). The algorithm of the evolution of
15 sound defines in a complete way the times and the modalities with which the (n) wave forms relative to the timbre contribute to generate the signal; it's the processing unit that, carrying out the operations described by the algorithm, makes it possible for the scanning above
20 mentioned to be executed on the different wave forms of the timbre: going from one wave form to the other a signal is built whose spectral composition has the course of the approximated spectral composition from which we had started to determine the (n) wave forms of the timbre in the above mentioned way. The algorithm moreover defines the times
25 and variation modalities of the fundamental frequency (FF) of the sound. The algorithm of sound evolution is described by means of a simple

language: for each timbre there exists in the EPROM memory a program coded in this language that is interpreted in real time by the processing unit.

- 5 2) Polyphonic and polytimbric digital synthesizer distinguished by the method by which it builds the pattern at the patterning frequency (FS) of the polyphonic signal sum of the single signals generally characterized by different frequencies and timbres. The processing unit loads into a RAM area, which we will call RAMTIMBRI, the patterns of
10 the wave forms of the timbres that the user selects as immediately executable: the information is obtained from it to build the polyphonic signal, with the following method:
 - a) the patterning period ($TS = 1/FS$) is subdivided into (q) equal intervals, which we will call time slots ((q) is the polyphony degree of the
15 synthesizer);
 - b) each time slot is assigned to a signal to be added up (it might also not be assigned);
 - 20 c) during each time slot the pattern relative to the signal assigned to the time slot is obtained from RAMTIMBRI and it is added-up in an adder-accumulator modulus;
 - d) at the end of all time slots the accumulator contains the sum of the
25 patterns of all the signals; at this point the sum value is moved to the analogic digital converter, which builds the analogic signal to be sent to the amplifier, the accumulator is initialized to zero and a new cycle starts. The structure executes what was above mentioned with a high degree of flexibility; in fact, it is possible to associate to
30 each time slot some signals distinguished by each of the present timbres and by each of the generated frequencies (FF); moreover, each timbre can be present with each of the wave forms that define it.
- 3) Polyphonic and polytimbric digital synthesizer distinguished by the
35 information compacting method used to represent in memory the (m) patterns of the wave forms mentioned in point 1). It allows the memorization of only (m/r) patterns and from these the reconstruction in real time

of the (m) global patterns, using a technique of approximated interpolation between each of the (m/r) stored and the next. The adopted technique consists of:

a) a set of (s) values

5 I1, I2, ..., Is)

has been appointed and, in correspondence of each of the values (Ii), (r) values (0, Ii/r, 2Ii/r, ..., (r-1) Ii/r) have been stored in a PROM memory;

10 b) in the off-line generation phase of the (m/r) patterns of the wave forms to be stored, for each of them the difference is calculated:

(Aa) = value of next pattern - value of pattern;

c) for each pattern the value (Ii) closer to the value (Aa) relative
15 to it is determined: (Ii) is assumed as an approximated increment between one pattern and the next;

d) still in the pattern generation phase the index (i) of the value (Ii) determined as mentioned in part c) is stored together with the
20 values of the (m/r) patterns.

e) the system calculates, in real time, the generic pattern (u) of the (m) of the wave forms in the following way: it adds the pattern (U1) of the (m/r) stored, where

25 (U1) = whole portion of (u/r) $0 \leq U1 < m/r$

with the value

(U2 x Ii/r) $0 \leq U2 < r$

stored into PROM, where

(U2) = remainder of (u/r)

30 (i) = order number of the increment stored with the pattern (U1) of the (m/r) stored.

4) Polyphonic and polytimbric digital synthesizer distinguished by the fact that the (m/r) patterns (pattern value and relative increment
35 index) mentioned in part 3), relative to the wave forms of the immediately executable timbres, are stored into a double-access RAM (RAMTIMBRI),

so that the processing unit can change the timbres stored there even during sound execution.

5) Polyphonic and polytimbric digital synthesizer distinguished by the fact that the values of the increments (I_i) stored in PROM, mentioned in part 3), are bound by the relation.

6) Polyphonic and polytimbric digital synthesizer distinguished by the fact that the timbre movement between one wave form to the other occurs in the moment when both wave forms have a null value, avoiding discontinuity in the movement. This operation is obtained by means of a programmable hardware that controls the scanning of the wave forms and that, if properly programmed, communicates to the processing unit the exact moment to carry out the commutation.

7) Polyphonic and polytimbric digital synthesizer distinguished by the fact that, if few sounds are present, the patterns of the same signal can be added up various times in the determination of the pattern of the polyphonic signal; in the transition from one wave form to the other (see part 1)), this allows for a linear interpolation between the two wave forms in a certain number of steps. For this purpose, referring to the time slot structure mentioned in point 2), the same signal is associated to (q_1) time slots, where (q_1) = whole portion of (time slot number/present signals number). If ($q_1 > 1$) the patterns of a same signal are added up various times. During the transition phase between one wave form and the other, it is possible to commute on one time slot at the time, thus carrying out the commutation between one wave form to the other in (q_1) steps. This allows the analogic digital convertor to work with a high dynamic degree even though only one sound is present. A convenient scale factor under the digital analogic convertor makes up for the fact that the same signal might have been added various times.

STUDIO DI CONSULENZA
ING. CLAUDIO BALDI

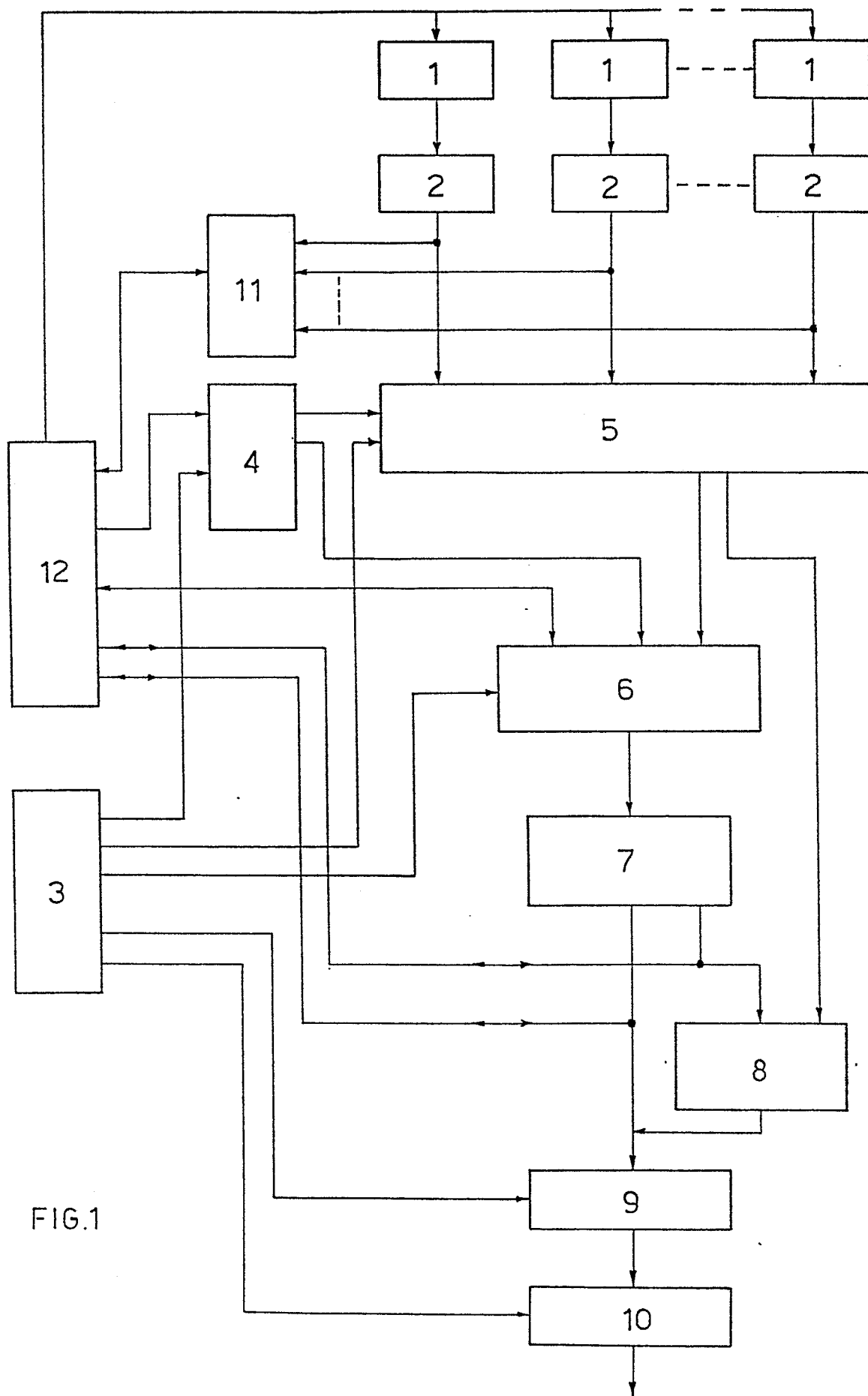


FIG.1