



159159

IMPROVED INTEGRATED CIRCUIT AND PROCESS FOR USING
AN INTEGRATED CIRCUIT OF THIS TYPE

The present invention relates to an improved integrated circuit and the process for using it. The invention is particularly applicable to microprocessors and microcomputers and also to hardwired logic circuits which require protection.

It is known that microprocessors and microcomputers sequentially execute successive instructions of a program stored in a memory, in sync with one or more timing signals referenced relative to one of the clock signals supplied to the microprocessor or microcomputer either internally or externally.

Thus, it is possible to correlate the various phases of this program execution with the clock signals, since the execution of each particular instruction breaks down into several steps timed by one or more successive clock pulses. In effect, in the microprocessors of the prior art, the operation is regularly timed by clock signals generally originating from a sequencing circuit which generates the necessary electrical pulses, particularly by phase shifting the signals relative to the reference clock. Moreover, the sequencing of the actions must factor in the times required to access the various registers, memories and internal devices, but also and especially the signal propagation times in the busses and through the various logic circuits. Consequently, the start and end times of each instruction being perfectly known, it is theoretically possible to know which instruction is being executed in the processing unit of the processor at a given moment, since the program that is running is constituted by a predetermined sequence of instructions.

It is possible, for example, to determine the number of clock pulses delivered since the startup of the program or the resetting of the processing unit, or even the time that has elapsed since an event or an external or internal reference signal.



ACTION

1 This capability of being able to observe the running of a
2 program in a microprocessor or a microcomputer is a major
3 drawback when the microprocessor or microcomputer is used in
4 high-security applications. In effect, an ill-intentioned
5 individual would thus be able to know the successive states of
6 the processor and use this information to gain knowledge of
7 certain internal output data.

8 It is possible to imagine, for example, that a given action
9 on an external signal could take place at different instants as a
10 function of the result of a determined security operation, such
11 as the testing of confidential internal information or the
12 decryption of a message, or even the integrity checking of
13 certain information. Depending on the instant in question, this
14 external signal could supply information on the output data or on
15 the confidential content of the information, and in the case of
16 cryptographic calculations, on the secret encryption key used.

17 Moreover, there are known microprocessors or microcomputers,
18 such as those marketed by the company SGS Thomson under the
19 reference number ST16XY, which comprise a microprocessor
20 incorporating a random number generator, the reading of which
21 makes it possible to obtain a random number used, for example,
22 for the calculation of encryptions and decryptions.

23 One of the objects of the invention is to equip the circuit
24 with means for preventing the type of investigation described
25 above, and more generally for preventing observations, whether
26 illicit or not, of the internal behavior of the circuit.

27 This object is achieved through the fact that the improved
28 integrated circuit has means for decorrelating the running of at
29 least one instruction sequence of a program from the internal or
30 external signals of the circuit.

31 According to another characteristic, the electrical signals
32 of the circuit are timing, synchronization or status signals.

33 According to another characteristic, the decorrelation means
34 comprise one or more circuits generating a sequence of clock or
35 timing pulses which are dispatched at random times.

1 According to another characteristic, the decorrelation means
2 comprise a random number generator which makes it possible to de-
3 synchronize the execution of the program sequence in the
4 processor.

5 According to another characteristic, the decorrelation means
6 comprise a clock calibration circuit which makes it possible to
7 eliminate the timing pulses that are too short.

8 According to another characteristic, the decorrelation means
9 comprise a random interrupt generating system.

10 According to another characteristic, the decorrelation means
11 comprise the execution of secondary sequences in which the
12 instructions and execution times are different and which are
13 selected at random.

14 According to another characteristic, the variable time of
15 the secondary process depends on a value supplied by a random
16 number generator.

17 According to another characteristic, the secondary process
18 does not modify the general operational context of the main
19 program, thus making it possible to return to the latter without
20 having to re-establish this context.

21 According to another characteristic, the secondary process
22 re-establishes the context of the main program before returning
23 the control of the processor to it.

24 According to another characteristic, the main program can
25 enable or disable one or more decorrelation means.

26 According to another characteristic, it has means for phase
27 shifting the timing, synchronization or status signals of the
28 processor.

29 According to another characteristic, the phase shifting
30 means generate a random phase shift of the timing,
31 synchronization or status signals of the processor.

32 According to another characteristic, the random phase
33 shifting means partially or completely de-synchronize the
34 operation of the processor from the external clock during the
35 execution of a program.

1 According to another characteristic, the random number
2 generator uses counters, which may or may not be looped, and
3 which are initialized to a random value.

4 According to another characteristic, the initialization
5 value originates from a non-volatile memory.

6 According to another characteristic, the initialization
7 value is modified during the execution of a program.

8 According to another characteristic, the random number
9 generator uses an algorithm of the cryptographic type or a hash
10 function initialized to the initialization value.

11 According to another characteristic, the sequencing of the
12 actions factors in the times required to access the various
13 registers, memories and internal devices, but also and especially
14 the signal propagation times in the busses and through the
15 various logic circuits.

16 Another object of the invention is to offer a process for
17 using the integrated circuit.

18 This object is achieved by the fact that the process for
19 using an integrated circuit is comprised:

20 of triggering the sequencing of one or more instructions or
21 operations using a random-pulse clock;

22 or of randomly triggering interrupt sequences;

23 or of triggering the processing of a random sequence of
24 instructions or operations during the execution of a main
25 sequence of instructions or operations;

26 or of combining at least two of the above-mentioned
27 possibilities.

28 Other characteristics and advantages of the present
29 invention will become more clearly apparent with the reading of
30 the following description, given in reference to the appended
31 drawings in which:

32 Fig. 1 represents the schematic diagram of the electronic
33 circuits of a first embodiment of the invention;

34 Fig. 2 represents a second, simplified variant of embodiment
35 of the invention;

1 Fig. 3A represents the diagram of an embodiment of the
2 calibration circuit;

3 Fig. 3B represents the logical sequencing diagrams of the
4 calibration circuit;

5 Fig. 4A represents the diagram of the logic circuits of an
6 embodiment of a phase shifting circuit;

7 Fig. 4B represents the diagram of the sequences of the
8 signals of this circuit;

9 Fig. 5 represents a third variant of embodiment of the
10 invention;

11 Fig. 6 represents the diagram of the logic circuits of an
12 embodiment of an internal clock;

13 Fig. 7A represents the logic diagram of an embodiment of the
14 random number generator;

15 Fig. 7B represents the logic diagram of an embodiment of
16 each cell of the random number generator;

17 Fig. 8 schematically represents an example of randomly
18 selected sequences of the secondary program. In the description,
19 the term microcomputer is intended to mean a monolithic
20 integrated circuit incorporating a microprocessor with a read-
21 write memory of the RAM type associated with at least one non-
22 volatile memory which may or may not be programmable such as, for
23 example, a RAM with battery backup, or a ROM, or PROM, or EPROM,
24 or EEPROM, or RAM of the Flash type, etc... or a combination of
25 these memories. The invention will now be explained with the aid
26 of Fig. 1 in which a CPU (1) comprises a random number generator
27 (2) which can run on an internal clock (11). Processors of this
28 type, as mentioned above, are particularly known through the
29 ST16XY family of microcomputers. However, these microcomputers or
30 microprocessors, which use a shift register with parallel input-
31 outputs looped back to at least one of its inputs, wherein the
32 shift is timed by an internal clock, to constitute the random
33 number generator, use the external clock for sequencing the
34 machine cycles of the microprocessor to execute the instruction
35 to read the contents of the register. The invention makes it

1 possible to generate a random number that is not pseudo-random
2 and is based on the fact that the internal clock of the random
3 number generator, which has a multiple frequency from the
4 external clock, is randomly phase shifted relative to the latter.

5 The invention is comprised of using the principle of a
6 microprocessor of this type with a random number generator,
7 adding to it a certain number of elements which will allow the
8 microprocessor running the main program to go from an operation
9 that is perfectly in phase and correlated to the external
10 sequencing clock, to a decorrelated operation in which, as
11 desired and depending on the embodiment chosen, the execution
12 time of a given instruction will no longer be identical, even
13 when the same instruction is executed several times, or in which
14 the execution duration of an instruction sequence will be
15 variable even if the same sequence is re-executed several times
16 by the main program, or even in which the execution duration of
17 an instruction will be variable, the execution time of the same
18 instruction itself being variable. This is obtained by the
19 circuit in Fig. 1 in which, in addition to the random number
20 generator (2), the internal clock (11) is embodied by a free
21 fixed frequency oscillator, de-synchronized and phase shifted
22 relative to the external clock CLKE of the microprocessor or
23 microcomputer. In the prior art, one skilled in the art would not
24 have considered timing the operation of a microcomputer or a
25 microprocessor with an irregular clock. On the contrary,
26 everything was done to ensure that the operation was regularly
27 timed by clock signals generally originating from a sequencing
28 circuit which generated the necessary electrical pulses,
29 particularly by phase shifting the signals relative to the
30 reference clock. This was particularly due to the fact that the
31 sequencing of the actions must factor in the times required to
32 access the various registers, memories and internal devices, but
33 also and especially the signal propagation times in the busses
34 and through the various logic circuits. In the invention, the
35 random number generator (2) is used either to supply a random

value to the various devices through the data bus (3) and load it into the various devices which will be described below, or to generate a pulse signal of variable periodicity at its output (22). In a microprocessor or microcomputer of the invention, the signals required for the loading and execution of the instructions can therefore be generated from randomly dispatched clock pulses, but these pulses must adhere to a minimum cycle time so that the processor (1) has enough time to execute the various operations. This signal, in order to serve as a clock for the microprocessor (1), must be sent to a calibration circuit (9). The output (95) of this calibration circuit is sent to a multiplexing circuit (18) whose input (19) for controlling the multiplexing receives the signal of one or more bits of a register (8) which can be loaded either by the random number generator (2) or with a value determined by the main program (5). When this register (8) is loaded with a random value, the decision which selects the clock signal sent to the processor is made randomly, whereas when this register (8) is loaded with a value determined by the main program, it is the main program which will choose whether the clock for sequencing the microprocessor will be the external clock CLKE or a decorrelation clock CLK2. Likewise, one or more bits of the register (8) are sent through the link (82) to a logic circuit (28) which makes it possible, as a function of the bit or bits of the register (8), to validate or not to validate the transmission of the internal clock signal (11) to the random number generator (2). This random number generator can then also run on the external clock CLKE by receiving its signal through the link (26) and the logic circuit (28). In this latter case, the values generated will be pseudo-random values. The random number generator (2) can operate using the internal clock (11) validated through the circuit (28) by the bit or bits of the register (8), and in this case, the values generated will be random values. The signal I generated at the output (22) of the random number generator (2) and received by the calibration circuit (9) corresponds to a pulse signal whose

1 periodicity varies, either randomly or in pseudo-random fashion.
2 The fact that this periodicity varies in pseudo-random fashion is
3 of little concern since, as will be seen below, the calibration
4 circuit (9) introduces an internal clock signal (FRC) which will
5 itself reintroduce a decorrelation, through a different frequency
6 and a phase shift relative to the external clock signal CLKE, and
7 consequently relative to the pseudo-random clock signal
8 synchronized to this external clock signal.

9 The device can also comprise a register R2 which is loaded,
10 either by the random number generator (2) with a random number,
11 or by the main program (5) with a value determined by the
12 program. This register R2 is totally or partially used by a logic
13 circuit (4) for triggering an interrupt, which receives at one of
14 its inputs the decorrelated clock signal CLK2 issuing from the
15 output (95) of the calibration circuit (9). The output of the
16 circuit (4) is sent through a gate (48) controlled by one or more
17 bits of the register (8) to the interrupt input (12) of the CPU.
18 The bit or bits of this register (8) play the role of an
19 interrupt mask control, which is standard in certain
20 microprocessors. When an interrupt is received at the interrupt
21 input (12) of the processor, the interrupt handling program
22 contained, for example, in the operating system or in the
23 secondary program will introduce a different processing time for
24 the interrupted sequence of the main program. It must be
25 understood that there are two phases in the interrupt mode of
26 operation.

27 A first phase, in which the microprocessor controlled by the
28 so-called main program authorizes the decorrelated operation by
29 unmasking, for example, the interrupts.

30 A second phase, in which the interrupt automatically
31 reroutes the operation to the secondary program. This operation
32 can actually occur without the intervention of the main program.

33 Lastly, the device of the invention can also comprise a
34 secondary program (6) which, as will be seen below, can generate
35 a variable duration time which varies each time this secondary

program (6) is called by the main program (5). Thus, the variant of embodiment represented in Fig. 1 allows the main program (5) to change the desired degrees of protection, either by triggering the sequencing of the execution of one or more instructions with the aid of the decorrelated clock CLK2, or by deciding during the execution of an instruction sequence to introduce, or not to introduce, a randomly triggered interrupt handler, or by deciding during the execution of the sequence to introduce, or not to introduce, a jump to the secondary program (6), which also generates a process with a variable time, or even by combining these various possibilities. Thus, in a variant of the invention, this secondary program (6) can be constituted as represented in Fig. 8 by a plurality of sequences (61, 62, 63....6n) which are called at random; and each sequence (0, 1, 2 or 2^{n-1}) will implement a different set of instructions which will result in a variable processing time in each branch and different behaviors of the microprocessor. The sequences can be called at random; for example, after the main program has executed the jump to the secondary program, the latter loads, in the steps (64 and 65), a random value V originating from the memory (7) into two registers, for example R10 and R11, of the microprocessor (1). The secondary program increments this value V, then the program orders the storage of this incremented value ($V + 1$) in the non-volatile memory NVM (7) in the step (66). This value stored in the non-volatile memory (7) is intended to be used later. The secondary program, in the step 67, then extracts n high-order or low-order bits in R10 in order to obtain a value r which will make it possible to indicate which program sequence among the secondary program sequences (61, 62, 63, ..., 6n) is to be executed. Each secondary program sequence will produce a different process: for example, the sequence (0) is comprised, first of all, of the step 611 for transferring the content of the register R11 of the microprocessor into a register R12. In the step 612, the content of R12 is added to the carry value (CARRY), then in the step 613, an exclusive OR is executed between the

1 content of the register R11 and the content of the register R12,
 2 and the result is placed in the register R12. In the step 614,
 3 the processor decrements R12; in the step 615, a test is carried
 4 out on the value of R12 to determine whether or not R12 is equal
 5 to zero. In the case where $R12 = 0$, the processor returns to the
 6 execution of the main program. In the opposite case, in the step
 7 616, the secondary program (61) executes a rotation of the
 8 content of the register R10. The next step consists of extracting
 9 n bits of a determined order from the register R10, in order to
 10 then access one of the sequences determined by this value r in
 11 the secondary program. Thus, it is possible to access, for
 12 example, the sequence (2^{n-1}) which, in the step (6n1), is
 13 comprised of transferring the result of the multiplication of the
 14 values of R10 and R11 into R13 and R14. In the step (6n2), this
 15 sequence executes a rotation of R13 and R14, then, in the step
 16 (6n3), the content of R13 is transferred into R11. In the step
 17 (6n4), R11 is decremented in order to then, in the step (6n5),
 18 perform a test on the value R11. This test is comprised of
 19 determining whether or not the content of $R11 = 3$. If so, the
 20 process returns to the main program and if not, the program
 21 proceeds to the step (6n6) by rotating R10 to the left, then
 22 executing the instruction (67) in order to access a new secondary
 23 program sequence.

24 In the case where the secondary program is intended to be
 25 combined with a decorrelated clock or interrupt handlers, it is
 26 possible in a combination of this type to restrict oneself to one
 27 secondary program, thus producing a simpler process. A simplified
 28 secondary program of this type can be constituted by the
 29 following instructions:

30 MOV B, R2 which is comprised of loading the register R2 into
 31 the microprocessor register B
 32 LOOP DCX B which is comprised of decrementing the register B
 33 with the value A
 34 JNZ B LOOP which is comprised of performing a test on the
 35 value of the register B and of looping back to the

1 label LOOP if this value is different from zero.

2 This sequence ends with an instruction to return to the
3 instruction of the main program which immediately follows the
4 last instruction executed before the jump to the secondary
5 program (6). The register R2 is pre-loaded by an instruction of
6 the main program (5), before the jump to the secondary program
7 (6), with a random value supplied by the random number generator
8 (2). Thus, the execution of the secondary program defined above
9 will always generate a variable duration.

10 Another embodiment of a secondary program of variable
11 duration can be comprised of defining an area of the program
12 storage corresponding to the secondary program (6) in which a set
13 of instructions is stored. Preferably, the instructions chosen
14 require different numbers of machine cycles in order to be
15 executed, as is known to be the case, for example, with the
16 instructions J, CALL, RET, RST, PCHL, INX in relation to
17 instructions requiring a number of shorter machine cycles such as
18 ADC, SUB, ANA, MOV, etc. Thus, in this storage area, there are a
19 certain number of available instructions having execution
20 durations that are different from one another in terms of the
21 number of machine cycles. The main program (5) comprises an
22 instruction to jump to an indexed address whose index corresponds
23 to the content of the register R2 and whose address corresponds
24 to the first address of the area (6). The execution of this
25 instruction of the main program (5) therefore causes the
26 addressing by the processor (1), at random, of instructions whose
27 execution durations will be different depending on the position
28 addressed. In a known way, the random number generator (2) will
29 be initialized at the start to a variable. This initial variable
30 is contained in a non-volatile memory (7) and constituted, for
31 example, by the last random value generated by the generator (2)
32 before the pausing of the microprocessor (1). Thus, the
33 microprocessor, controlled by a program it will execute, will be
34 able to use this program to activate the means for decorrelating
35 the instruction execution sequencing of this program by loading,

1 for example, registers R2 or 8, or by calling secondary programs.

2 Fig. 2 represents another variant of a simplified embodiment
3 of the invention, in which the content of the register (8) will
4 command the multiplexer (18) to decide whether the external clock
5 CLKE is sent to the processor (1) or whether simply the
6 decorrelated clock CLK is used by the CPU (1). This register (8)
7 is loaded by the bus (30) upon execution of an instruction of the
8 main program (5) which is designed to decide at a given moment to
9 trigger the protected mode by generating instruction execution
10 sequences of variable duration. The random number generator (2)
11 is in communication through a bus (31) with the non-volatile
12 memory (7), which makes it possible, for example, to store the
13 last value generated so that when there is a new connection of
14 the monolithic circuit, the random number generator is re-
15 initialized with a value different from the preceding initial
16 value. This bus (31) may be controlled by the processor (1). In
17 another variant, the writing into the memory (7) can be
18 controlled by a hardwired logic unit.

19 In another embodiment, it is possible to introduce a
20 variable phase shifting circuit (45) at the output of the clock
21 circuit, as shown in Fig. 4A, which phase shifting circuit is
22 constituted for example by a shift register D1 through D5 timed
23 by the signal FRC issuing from the circuit (11) or the
24 recalibrated FRC supplied by the output (95) of the circuit (9)
25 and phase shifting the signal I supplied by the output (22),
26 which can be divided by a slowdown factor in a divisor (452). The
27 output of the phase shifting circuit (45) can be produced with
28 the aid of a multiplexer (451) MUX which makes it possible to
29 extract any of the output signals Q1, Q2, ..., Q5 from the shift
30 register as a function of the content of the register RM which is
31 loaded, either directly by the random number generator (2), or
32 indirectly by the main program (5) or even by the secondary
33 program (6), through the bus (3). In this case, the clock leading
34 edges S delivered as output can be delayed or advanced relative
35 to a median pulse supplied by the central level of the shift

1 register, by a value which depends on a random number, thus
2 proportionally delaying or advancing the instruction execution
3 sequencing of the program in progress.

4 In another embodiment, the random number generator and the
5 phase shifting circuit can be used continuously during certain
6 particularly sensitive periods; during these phases, the
7 processor is timed in a completely random fashion, since the time
8 intervals which separate the clock pulses are variable, and not
9 constant as is the case with the standard processors.

10 The organization of the programs executed by the processor
11 can be carried out in such a way that the operation of the
12 processor (1) is controlled by a genuinely protected operating
13 system which chooses the type of scrambling to be used as a
14 function of the type of program run by the machine. In this case,
15 it is the operating system which manages, as it sees fit, the
16 various signals resulting from the random number generator, the
17 calibrator, the interrupts or commands from the phase shifting
18 circuit, and the start up of the main and secondary programs. It
19 is clear that the secondary program can be used to perform
20 functions other than a simple time-out, particularly by executing
21 processes which may be necessary to the main program in order to
22 make use of the time dedicated to the secondary program, which
23 processes can be constituted, for example, by preparations of
24 calculations subsequently used by the main program. It is
25 understood that the devices of the invention can easily be
26 generalized when the processor is used in multiprogramming, the
27 application programs in this case being considered to be a number
28 of main programs. The random number generator and the clock phase
29 shifting circuit seen above do not pose any particular production
30 problem and are known to one skilled in the art when they are
31 used separately for other usages not linked to the invention.

32 It is also possible to produce a fifth simplified embodiment
33 of the invention which does not use an interrupt. When the main
34 program needs to be protected, it independently activates a
35 secondary program, which generates a process of random length at

1 instants it selects, either at the start or during execution, so
2 as to scramble the various sequences.

3 The various circuits which allow the embodiment of the
4 invention will now be described in connection with the other
5 figures. Thus, a random number generator represented in Figs. 7A
6 and 7B is constituted, for example, by a set of cells (B0 through
7 B7) each of which is formed by an exclusive OR gate (23) with two
8 inputs connected to a type D switch (24) whose output (Q) is
9 connected to one of the two inputs of the exclusive OR gate of
10 the next cell. The second input of the exclusive OR gate receives
11 the input signal of the data issuing from the bus (3) in order to
12 allow the initial loading or, for the cells (B0) and (B3), for
13 example, a loop-back signal (25) issuing from the last cell (B7).
14 The output (22) of the last cell (B7) also constitutes the output
15 which delivers the pulse signal (I) of randomly variable
16 periodicity. This signal (I) is then used in the calibration
17 circuit (9) represented in Fig. 3A. Fig. 3B represents the
18 sequencing of the input and output signals of this calibration
19 circuit (9) in Fig. 3A. This calibration circuit is constituted
20 by two NAND gates (90, 91) with three inputs, each of which
21 receives, at one input, the signal I issuing from the output (22)
22 of the random number generator (2). A first NAND gate (91)
23 receives the output (Q2) of a toggle of the JK type, while the
24 other gate (90) receives the inverted output (NQ2) of this toggle
25 (93). This toggle (93) receives at its clock input a clock signal
26 FRC which constitutes an internal clock of the circuit. This
27 internal clock is generated, for example, by a circuit
28 represented in Fig. 6. The inputs J and K of this toggle (93) are
29 connected to the supply voltage representing the logical level
30 "1". The internal clock signal FRC is sent by an inverter circuit
31 (92) to each of the three inputs of the NAND gates (90, 91). The
32 output of the first NAND gate (90) is sent to the input for
33 setting the second logical toggle (94) to 1, while the output of
34 the second NAND gate (91) is sent to the input for resetting the
35 second toggle (94) to zero. The clock input and the input (J) of

1 this second toggle (94) are connected to the supply voltage
 2 representing the level (1) and the input (K) is connected to the
 3 supply voltage representing the level zero. The output (Q1) of
 4 this second toggle (94) delivers the signal CLK2 supplied by the
 5 link (95) to the multiplexer (18). The internal clock FRC
 6 delivers, through the link (111), periodic pulse signals having a
 7 minimum pulse width T_m which is defined by the circuit in Fig. 6.
 8 This circuit (11) is constituted, for example, by a series of
 9 inverters (111 through 115), in this case five, which each have a
 10 determined propagation time, for example 10 nanoseconds, making
 11 it possible to obtain at the output FRC a pulse of 50
 12 nanoseconds. This output FRC is looped back through the link
 13 (116) to the input of the first inverter (111), and the input of
 14 the first inverter (111) is also fed through a resistor (117) by
 15 the 5-volt supply voltage. The pulse width selected is 50
 16 nanoseconds, but it is quite obvious that varying the number of
 17 inverter gates varies the value T_m . This value T_m will be used,
 18 as represented in Fig. 3B, by the logic circuit (9) of Fig. 3A to
 19 generate, from the pulse signal of randomly variable periodicity
 20 (I), a pulse signal CLK2 whose pulses of variable width have a
 21 minimum value T_m and whose periodicity is also variable and de-
 22 synchronized relative to the external clock CLKE. In effect,
 23 since the internal clock begins to function as soon as the
 24 integrated circuit is powered up, if the initial periodicity of
 25 this clock is different from the periodicity of the external
 26 clock, there is no chance for the clocks to be synchronized at
 27 the startup. The signals from this calibrator (9) have a period
 28 at least equal to twice the minimum time T_m required for the
 29 processor to execute an internal cycle. All the leading edges of
 30 the signal CLK2 will be separated by at least the value T_m , but
 31 their position and their exact duration will be random.

32 It may also be seen that no matter what the variant of
 33 embodiment, the running of the main program occurs with an
 34 unpredictable sequencing which, depending on the variant, depends
 35 on the random number generator, on the random clock, on the

secondary program, on the random interrupts, or on a combination of at least two of these devices. When the main program executes functions that are not sensitive from the point of view of security, it can also return to the external clock CLKE, for example in order to deliver output data to the external world or to mask the decorrelation interrupt in order to optimize the processing time. As soon as a security function is implemented, the main program (5) authorizes the random mode of operation by validating either the random clock or the decorrelation interrupt (or both) in order to "scramble" the various operational signals, particularly by de-synchronizing the clock relative to the main program, or by calling the secondary program.

For the random number generator (2), it is possible, for example, to use looped counters having different periods, which counters are initialized with a "seed" (information) stored in a non-volatile memory (7). When the processor starts up, the counters factor in the stored value as the initial value. During the calculation, or at the end of the calculation, the non-volatile memory (7) is updated with a new value which will serve as a seed for initializing the counters at the next initialization. The interrupt generating circuit (4) can be designed so that the generation of interrupt pulses seen above can occur, for example, when the number generated has certain characteristics, such as equality with certain data of the program. This circuit (4) can also take on the value of one or more bits of one or more counters. It is also possible to produce a very good random number generator using a cryptographic algorithm (69), as shown in Fig. 5 or a hash function initialized by the "seed" (information) seen above. In this case, the generator can be in the form of a program which implements the algorithm executed by the processor (1) and which, for example, implements the cryptographic algorithm by receiving a variable stored in the non-volatile memory (7) and a key for generating an output stored in a buffer register (41). This output stored in the buffer register is then processed by a hardware or software

1 decoding device (42) for generating either the decorrelated clock
2 signal CLK2 or a signal for interrupting the processor (1). It is
3 easy to see that this random number generator can also be used to
4 generate the various random numbers seen above. Another way to
5 produce a generator of this type is to amplify the voltage
6 generated at the terminals of a so-called "noise" diode and to
7 shape the signals after a low pass filtering for preventing the
8 noise pulses that are too rapid from disturbing the operation.

9 For the clock phase shifting circuit, there are
10 possibilities other than the one seen above. For example, a shift
11 register driven by a clock 10 times faster than that of the
12 processor. If the register is assumed to comprise ten toggles,
13 there will be ten available pulses having different phases, which
14 can be selected by the processor with the aid of a multiplexer
15 with ten inputs and one output. The output of the multiplexer, as
16 before, is used to provide the internal clock signal of the
17 processor.

18 Another embodiment is comprised of using a circuit of the
19 same type as the random number generator seen above, and of
20 extracting pulses at the different levels of the counters. In
21 this case, the processor is actually timed by pulses that are
22 dispatched at random times.

23 Another embodiment is comprised of using the signals from
24 the random number generator to extract the pulses from the shift
25 register. Many combinations are possible for refining the
26 devices, but the principles of the invention remain valid.

27 The variant of embodiment in Fig. 1 is the most complete;
28 quite obviously, the monolithic circuit of the microprocessor
29 type or the microcomputer type could incorporate only one, or
30 several, or any combination, of the elements represented.

31 Thus, according to a variant, the monolithic circuit can
32 incorporate a microprocessor, the random number generator, the
33 internal clock (FRC) and the calibration circuit which forms the
34 decorrelated clock.

35 In another variant, the monolithic circuit can incorporate

1 the microprocessor, the random number generator, and the
2 interrupt generating circuit.

3 In another variant, the monolithic circuit can incorporate
4 the microprocessor, the secondary program and the decorrelated
5 and calibrated clock circuits.

6 In another variant, the monolithic circuit can incorporate a
7 microprocessor, the decorrelated and calibrated clock circuit,
8 and the interrupt circuit.

9 In other variants of the monolithic circuit, the
10 microprocessor is replaced by a microcomputer.

11 In other variants of the monolithic integrated circuit, the
12 microprocessor can be replaced by a combinational logic element
13 which makes it possible to execute a limited number of
14 instructions for specific applications. It is obvious that in
15 such a case, the same security devices can be applied to the
16 integrated circuit.

17 Other modifications within the capability of one skilled in
18 the art are part of the scope of the invention.

CLAIMS

1. An improved integrated circuit characterized in that it has means for decorrelating the running of at least one instruction sequence of a program from the internal or external electrical signals of the integrated circuit .

2. The integrated circuit according to claim 1, characterized in that the electrical signals of the microprocessor or microcomputer are timing, synchronization or status signals.

3. The integrated circuit according to claim 1, characterized in that the decorrelation means comprise one or more circuits which generate a sequence of clock or timing pulses which are dispatched at random times.

4. The integrated circuit according to claim 1, characterized in that the decorrelation means comprise a random number generator which makes it possible to de-synchronize the execution of the program sequence in the processor.

5. The integrated circuit according to claim 4, characterized in that the decorrelation means comprise a clock calibration circuit which makes it possible to eliminate the timing pulses that are too short.

6. The integrated circuit according to claim 1, characterized in that the decorrelation means comprise a random interrupt generating system.

7. The integrated circuit according to claim 1, characterized in that the decorrelation means comprise the execution of secondary sequences in which the instructions and execution times are different and which are chosen at random.

THIS SHEET HAS BEEN
REPLACED

1 8. The integrated circuit according to claim 7,
2 characterized in that the variable time of the secondary process
3 depends on a value supplied by a random number generator.

1 9. The integrated circuit according to claim 7,
2 characterized in that the secondary process does not modify the
3 general operational context of the main program, thus making it
4 possible to return to the latter without having to re-establish
5 this context.

1 10. The integrated circuit according to claim 7,
2 characterized in that the secondary process re-establishes the
3 context of the main program before returning the control of the
4 processor to it.

1 11. The integrated circuit according to claim 1,
2 characterized in that the main program can enable or disable one
3 or more decorrelation means.

1 12. The integrated circuit according to claim 1,
2 characterized in that it has means for phase shifting the timing,
3 synchronization or status signals of the processor.

1 13. The integrated circuit according to claim 12,
2 characterized in that the phase shifting means generate a random
3 phase shift of the timing, synchronization or status signals of
4 the processor.

1 14. The integrated circuit according to claim 13,
2 characterized in that the random phase shifting means partially
3 or totally de-synchronize the operation of the processor from the
4 external clock during the execution of a program.

1 15. The integrated circuit according to claim 4,
2 characterized in that the random number generator uses counters

3 which may or may not be looped, and are initialized to a random
4 value.

1 16. The integrated circuit according to claim 15,
2 characterized in that the initialization value originates from a
3 non-volatile memory.

1 17. The integrated circuit according to claim 16,
2 characterized in that the initialization value is modified during
3 the execution of a program.

1 18. The integrated circuit according to claim 15,
2 characterized in that the random number generator uses an
3 algorithm of the cryptographic type or a hash function
4 initialized to the initialization value.

1 19. The integrated circuit according to claim 1,
2 characterized in that the sequencing of the actions factors in
3 the times required to access the various registers, memories and
4 internal devices, but also and especially the signal propagation
5 times in the busses and through the various logic circuits.

1 20. A process for using an integrated circuit comprising
2 means for decorrelating the running of at least one instruction
3 sequence of a program from the internal or external electrical
4 signals of the integrated circuit, characterized in that it
5 comprises the steps consisting:

6 of triggering the sequencing of one or more instructions or
7 operations with the aid of a random-pulse clock;

8 or of randomly triggering interrupt sequences;

9 or of triggering the processing of a random sequence of
10 instructions or operations during the execution of a main
11 sequence of instructions or operations;

12 or of combining at least two of the above-mentioned
13 possibilities.

21. An integrated circuit comprising a microprocessor controlled by at least one program and means for decorrelating the sequencing of the execution of the instructions of this program, characterized in that a part of this program makes it possible to authorize, modify or disable the operation of the decorrelation means.

22. The integrated circuit according to claim 21, characterized in that the decorrelation means comprise means for generating a timing signal, or a sequence of clock pulses which are dispatched at random times, associated either with means for randomly generating interrupts or means for triggering the execution of a secondary sequence.

23. An integrated circuit comprising a microprocessor or means for executing instructions, characterized in that it comprises means for selecting the clock for timing the microprocessor or means for executing instructions, which selection means make it possible to select either a clock CLKE external to the integrated circuit or a random clock CLK2 or F.

24. The integrated circuit according to claim 23, characterized in that the random clock is generated from a random number generator to which either an internal clock (FRC) or an external clock (CLKE) is applied.