(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 August 2007 (02.08.2007)

PCT

(10) International Publication Number
WO 2007/087559 A2

(51) International Patent Classification:
H03K 3/84 (2006.01)

(21) International Application Number:
PCT/US2007/060964

(22) International Filing Date: 24 January 2007 (24.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/761,821 24 January 2006 (24.01.2006) US

(71) Applicant *(for all designated States except US)*: **PUFCO, INC.** [US/US]; 2225 E. Bayshore Rd., Suite 220, Palo Alto, California 94303 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **SUH, Edward** [KR/US]; 408 Grant Ave #311, Palo Alto, CA 94306 (US). **DEVADAS, Srinivas** [US/US]; 7 Whittier Road, Lexington, MA 02420 (US).

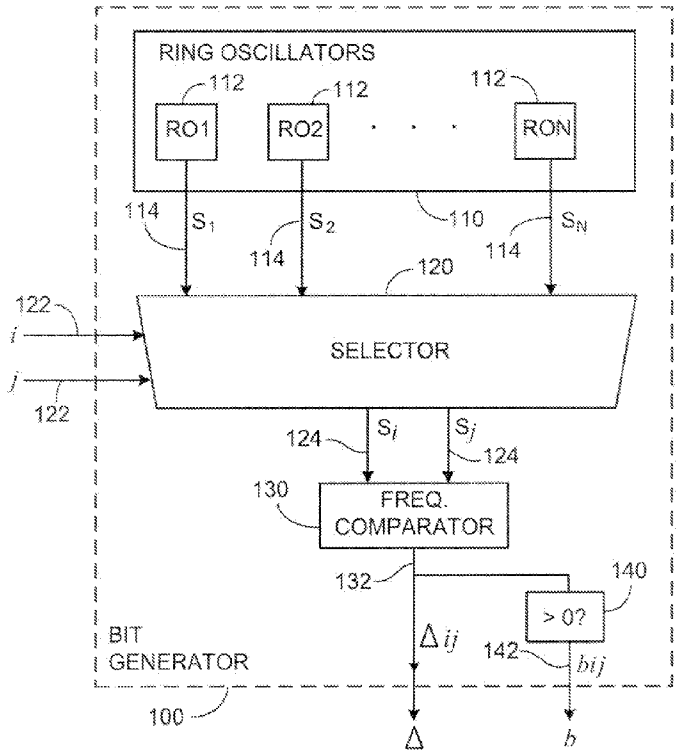
(74) Agent: **ROHLICEK, J., Robin**; Fish & Richardson P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: SIGNAL GENERATOR BASED DEVICE SECURITY



(57) Abstract: Subsets of multiple signal generator circuits embodied in a device are selected, and then a volatile value for the device is generated from the selected subsets. The volatile value may be used for authentication of the device and/or for cryptographic procedures performed on the device. The signal generator circuits may each comprise an oscillator circuit, and the selection of the subsets may be according to a comparison of the outputs of the subsets of circuits, for example, according to a comparison of output oscillation frequencies.

WO 2007/087559 A2

FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SIGNAL GENERATOR BASED DEVICE SECURITY

Cross-Reference to Related Applications

[001] This application claims the benefit of U.S. Provisional Application Serial No. 60/761,821, "RING OSCILLATOR BASED DEVICE IDENTIFICATION," filed January 24, 2006, which is incorporated herein by reference.

[002] This application is related to the following pending U.S. Applications: Serial No. 11/273,920, "VOLATILE DEVICE KEYS AND APPLICATIONS THEREOF," filed November 14, 2005, published on September 21, 2006, as US2006/0210082A1; Serial No. 11/272,995, "SECURELY FIELD CONFIGURABLE DEVICE," filed November 14, 2005, published on September 21, 2006, as US 2006/0209584A1; and Serial No. 10/407,603, titled "AUTHENTICATION OF INTEGRATED CIRCUITS," filed on April 4, 2003, published on October 30, 2003, as US2003/0204743A1, which are also incorporated herein by reference.

Background

[003] This invention relates to use of signal generators for device security.

[004] Secret keys or other values embedded in devices such as integrated circuits (chips) may be used to authenticate the devices. An example of such an approach uses a secret key embedded in a smart card. This secret key is not intended to be read out, but can be used to decrypt messages sent to the smart card. A service provider who knows the secret key can authenticate the smart card by asking it to decrypt an encrypted message or using other cryptographic protocols that verify that the smart card possesses the key. The secret key needs to remain secret so that an adversary cannot duplicate the key and assume a false identity.

[005] Authentication of integrated circuits has been suggested using device characteristics that vary from device to device based on variation in the fabrication process. Some such approaches, which are based on circuit delay are described in U.S. Application Publication US2003/0204743A1, titled "Authentication of Integrated Circuits," which is incorporated herein by reference.

[006] One approach to generating a key that is described in US2003/0204743A1 uses a measurement of a frequency of a self-oscillating circuit. Another approach uses a ratio of the measured frequencies of two self-oscillating circuits. Use of a ratio provides some compensation for variations that are due, for example, to environmental conditions of the circuit.

Summary

[007] In one aspect, in general, subsets (e.g., pairs, three, or more) of multiple signal generator circuits embodied in a device are selected, and then a volatile value for the device is generated from the selected subsets. The signal generator circuits may each comprise an oscillator circuit, and the selection of the subsets may be according to a comparison of the outputs of the subsets of circuits, for example, according to a comparison of output oscillation frequencies.

[008] In another aspect, in general, an integrated circuit includes multiple signal generator circuits, and selection circuitry coupled to the signal generator circuits for outputting a selection of subsets of the plurality of signal generator circuits. The circuit also includes a volatile value generator coupled to the signal generator circuits for outputting a value according to the selected subsets.

[009] In another aspect, in general, software stored on computer-readable media enables a device to perform functions including selecting subsets of a plurality of signal generator circuits embodied in a device, and generating the volatile value for the device from the selected subsets. The software may include data for configuring programmable circuitry to perform the functions (e.g., a personality matrix for and FPGA), or may include instructions (e.g., Verilog) for generating a device specification for fabricating the device.

[010] In another aspect, in general, subsets of a set of signal generator circuits embodied in a device are selected. A volatile value is then generated for the device from the selected subsets.

[011] Aspects can include one or more of the following features.

[012] The signal generator circuits comprise oscillators.

[013] Each subset consists of a pair of circuits.

[014] Each subset comprises a pair of predetermined groups of two or more circuits.

[015] Information characterizing the selection of the pairs is stored, for example, in non-volatile form on or off a device hosting the circuits.

[016] Error correction information is computed from the generated volatile value. The error control information is stored, for example, either on or off the device.

[017] The device is authenticated using the volatile value. The device may also be repeatedly authenticated using repeated re-generation of the volatile value. A cryptographic procedure can also be applied to data at the device using the volatile value.

[018] The volatile value for the device is re-generated from the selected subsets. Re-generating the volatile value can include using stored information characterizing the selection of the subsets of circuits. Re-generating the volatile value can include using stored error correction information computed from the generated volatile value.

[019] Selecting the subsets of circuits and generating the volatile value are performed as part of an enrollment procedure.

[020] Storing information characterizing the selection of the subsets is performed as part of an enrollment procedure.

[021] Generating the volatile value includes generating a different portion of the value from each of the selected subsets. For example, a different bit of the value is generated from each of the selected subsets.

[022] The subsets are selected according to an anticipated error rates associated with corresponding subsets of the circuits.

[023] The subsets are selected according to a comparison of the outputs of the subsets of circuits. For example, the subsets are selected according to a comparison of oscillation frequencies of the subsets of circuits. The subsets with oscillation frequencies that differ according to a predefined criterion may be selected.

[024] In another aspect, in general, an integrated circuit includes a set of signal generator circuits, and selection circuitry coupled to the signal generator circuits for outputting a selection of subsets of the signal generator circuits. The circuit also includes a volatile value generator coupled to the signal generator circuits for outputting an identifier according to the selected subsets.

[025] Aspects can include one or more of the following features.

[026] The signal generator circuits comprise oscillators.

[027] The integrated circuit includes an interface to the circuit for providing and for retrieving the selection of subsets of the plurality of signal generator circuits. For example, the interface is an interface to an external storage to the circuit.

[028] The integrated circuit includes a storage for the selection of subsets of the plurality of signal generator circuits.

[029] The integrated circuit includes an error correction module coupled to the volatile key generator configured correct the value outputted from the volatile value generator according to error control information. The circuit can include an interface to the circuit for providing

and for retrieving the error control information, for example, from an external storage, and the circuit can include a storage for the error control information.

[030] In another aspect, in general, software is stored on computer-readable media for enabling a device to perform all the steps of any method set forth above.

[031] In another aspect, in general, software is stored on computer-readable media for enabling or causing a device to select subsets of a plurality of signal generator circuits embodied in a device, and to generate a volatile value for the device from the selected subsets.

[032] The software may include data for configuring programmable circuitry (e.g., an FPGA) to perform the functions, or for generating a device specification for fabricating the device (e.g., and ASIC).

[033] The software may include instructions for causing a computing system to perform the functions.

[034] In another aspect, in general, a device is configured to perform all the steps of any of methods set forth above.

[035] Aspects can have one or more of the following advantages.

[036] A unique of relatively unique and unpredictable value can be generated on a device and later re-generated without disclosing the value. This permits authentication and application of cryptographic procedures on the device in a secure manner. The value does not necessarily have to identify the device, but rather provides a degree of unpredictability of the value that is used as a basis for the security of the device.

[037] Other features and advantages of the invention are apparent from the following description, and from the claims.

Description of Drawings

[038] FIG. 1 is a block diagram of a bit generator.

[039] FIG. 2 is a block diagram of a multiple-bit generation approach.

[040] FIG. 3 is a flow chart of a pair selection process.

[041] FIGS. 4A and 4B are block diagrams related to enrollment and run-time use, respectively, of a multiple bit generator.

[042] FIGS. 5 and 6 are embodiments of a bit generator.

Description

[043] An approach to generation of a volatile value on a device makes use of a set of signal generators. The volatile value may, for example, be used for authentication of the device and/or for cryptographic procedures (including authentication and data encryption). In some examples, the volatile value is used as a volatile key for cryptographic purposes. An example of signal generators are self-oscillation circuits (e.g., ring oscillators), which are discussed in more detail below. A "volatile" value, such as a binary key, refers to a value that is transient, for example, not being retained when power is removed from a device on which the value is generated or temporarily stored. That is, a volatile value is not stored in a binary form, for example, using fusible links or in a electrically writeable read-only memory, such that the value can be determined from such a stored form. The frequencies of the self-oscillation circuits can be used to generate and repeatedly re-generate a volatile value associated with the device, without that value being discoverable from non-volatile values stored on the device or from values that are exposed outside the device.

[044] In some examples, each of the ring oscillators has the same design, such that absent fabrication-related or environmental variations, each of the ring oscillators would be expected to have the same frequency. For example, each ring oscillator may be formed from a loop that includes a fixed number (e.g., five) inverters that oscillate at approximately 2 GHz.

[045] Referring to FIG. 1, some approaches to generating a volatile key makes use single bit generator 100 that includes an oscillator circuit 110 that includes N separate ring oscillators 112. Due to fabrication variation, each ring oscillator (k) outputs a periodic signal S_k 114 with a frequency that is generally different from the frequencies of the other oscillators. A selector 120 accepts control inputs 122 that identify two of the ring oscillators, i and j, and passes the corresponding pair of periodic signals S_i and S_j to a frequency comparator 130. The frequency comparator 130 outputs a quantity characterizing a difference Δ_{ij} in frequencies of the signals (e.g., subtraction $S_i - S_j$, or alternatively division S_i / S_j). A one bit output b_{ij} is set to 1 if $S_i > S_j$ and 0 otherwise.

[046] Referring to FIG. 2, a multiple-bit generator 200 makes use of multiple selections of pairs of the ring oscillators, with each pair generating one bit 222 of a multi-bit output B 220, which can be used as a volatile key value. In particular, a control input R 210 specifies the sequence of pairs of oscillators to use to generate each of the corresponding bits of the multi-bit output. The control input can be determined, for example, from a bit mask of the N(N-1) possible pairs of inputs, or some equivalent data representation.

[047] Referring to FIG. 3, as is discussed further below, the selection of which pairs to use for the control input R may be based on an enrollment time generation of some of all of the N(N-1) quantities Δ_{ij} 310 characterizing the difference in frequencies of the ring oscillators for a particular device. A pair selector 300 is applied to these differences to generate the control input R 210 or an equivalent specification of the pairs to use in generating the multiple-bit output.

[048] Referring to FIG. 4A, in an enrollment phase, enrollment circuitry on the device includes the pair selector 300 as well as the multi-bit generator 200, which depends on the selection R of pairs of oscillators. The selection R, or a quantity that allows regeneration of R without disclosing R in a non-volatile manner, is stored on or off the device. The volatile key B is used on the device, for example, by security enrollment logic 440. Optionally, an error correcting code (ECC) module generates error correction information E, which is also stored on or off the device.

[049] Referring to FIG. 4B, in a use phase (i.e., at "run time"), the quantity R is passed to the multi-bit generator 200, which generates $\tilde{B}$ which is a version of B with possible bit errors. This generated value is passed to an error correction module that makes use of the saved error correction information E to generate $\hat{B}$ which typically matches the value of B generated during the enrollment phase. This error corrected value is then used, for example, by security run-time logic 450.

[050] In general, given N oscillators on a device, there are N! (N factorial) different orderings of the frequencies of the N oscillators. Each ordering is equally likely assuming that the frequencies are independent. Therefore, the N(N-1) pairwise comparisons of frequencies provides $\log_2(N!)$ independent bits of information. For example:

N	Pairs (=N(N-1))	bits (=log2(N!))
35	595	133
128	8128	716
256	32640	1687

[051] In practice, for a particular device, bits generated from comparison of different pairs of oscillators exhibit different probabilities of error (i.e., flipping of that bit) between enrollment and run-time use. For example, two oscillators that have almost identical frequencies may on one use (e.g., enrollment) have one ordering, and on another use (e.g., at run time) have the opposite ordering. One approach to handling this phenomenon is to use the error correction approach outline above to correct such errors. Because a limited number of pairs of oscillators will have such close frequencies, a limited number of such bit errors will have to typically be corrected.

[052] Another approach to address this effect essentially uses a predicted error rate for particular pairs of oscillators determined during the enrollment phase and uses this prediction in the selection process for pairs of oscillators. For example, the frequency differences Δ_{ij} are used as proxies for the error rate, with small differences between frequencies being uses as a prediction of a high error rate.

[053] Selection approaches, some of which are described below, can reduce the error rate and make it possible to avoid the need for an error correction phase altogether, or at least allow use of a simplified error correction approach, for example, that may be simpler than use of a BCH error correction approach.

[054] Referring to FIG. 5, an implementation 500 of a bit generator makes use of N ring oscillators 512. The outputs of all the oscillators are fed to two multiplexers 520, which together select the outputs of a pair of the oscillators. The output of each of the multiplexers is fed to a counter 530. At the end of a sampling interval in which the counters start at a common count (e.g., zero), the outputs of the counters are passed to a comparator 540, which determines the output bit value according to which counter has a larger value. The implementation 500 can be used to time-serially generate multiple bits with different control input values selecting the oscillator signals to feed to the counters for each bit to be generated. Alternatively, multiple instances of the selectors and counters can be used to generate multiple bits concurrently.

[055] Referring to FIG. 6, in another implementation 600, oscillators 612 are placed on a device in pairs (e.g., adjacent or in close proximity to one another), and it is a difference (e.g., absolute difference) in frequency between the two oscillators of such a pair of oscillators, rather than an absolute frequency of one oscillator, that is compared with the difference of another pair with the pairs being selected according to the control inputs. For example, oscillation frequency may have an overall trend across a fabricated chip that may be due to operating temperature variations across the chip. Such a trend can result in a degree of predictability (or equivalently lower probabilistic entropy) of the relative frequencies of different oscillators. The systematic skew may imply that more ring oscillators are needed to generate a certain number of unpredictable bits.

[056] Some fabrication approaches may be more susceptible to systematic skew. For example, systematic skews may be less significant in ASIC implementations than in FPGA implementation. In examples of ASIC implementations, each dimension of the ring oscillator array may be 100-1000 times smaller than in FPGA implementations, which may reduce significance of the skew effect as compared to other localized factors.

[057] In some examples, the mechanism illustrated in FIG. 6 essentially eliminates systematic skew such as may be found in FPGA implementations. This mechanism reduces/eliminates low frequency systematic skew by using the distance D = |F1 – F2| between frequencies of adjacent ring oscillators instead of ring oscillator frequencies themselves. These distances, or "delta" frequencies are compared to generate key bits.

[058] The implementation shown in FIG. 6 has N pairs of oscillators 612. A control input causes the signals from one pair (e.g., from ring oscillators j-A and j-B) through one of the selectors 620 to two counters 630 and the signals from another of the pairs (e.g., k-A and k-B) to two other counters 630. A summer 635 takes the difference between the difference between the counters from each of the pairs. That is, if C[j-A] represents the count from the selected A counter from the jth pair, then the summer computes |C[j-A]-C[j-B] | – |C[k-A]-C[k-B] |.

[059] Several examples of approaches to selecting the pairs of oscillators (or in the case of approaches such as shown in FIG. 6, pairs of localized oscillator pairs) for generation of successive bits in a key value make use of measurements related to the oscillators' characteristics on a particular device. This selection can be done internally on the chip without necessarily disclosing the selected pairs outside the device.

[060] In a first example, an ordering of the frequencies of all the ring oscillators is identified on a chip to generate the bits of the volatile key. In examples in which there are N ring oscillators, the frequency of the i^{th} oscillator is denoted f_i. To find the ordering of the oscillators, a processing element on the chip compares all possible pairs of ring oscillators (i,j) and generates 1 if $f_i > f_j$ $(i \neq j)$ otherwise 0. This scheme generates (N*(N-1)/2) bits from N ring oscillators, which effectively represent N! (N factorial) possible orderings or $\log_2(N!)$ independent bits.

[061] In an experiment in which all pairs were compared, the probability of a flip of a bit was compared according to the frequency separation of the oscillators used to generate the respective bits. For oscillators with 5 inverters the bits generated by oscillator pairs that have frequencies more than 2MHz apart were essentially error free.

[062] Another example generates bits only from pairs of oscillators that have frequencies that are sufficiently different (e.g., different by at least 2MHz) such that they generate response bits that do not necessarily require additional error correction schemes. During an enrollment phase, the frequencies are compared, and data representing the list of sufficiently separated pairs is stored on or off the device. A challenge to applying this example of only using "robust" pairs to generate response bits is to ensure a device will use the same set of pairs even with environmental variations without leaking information in the data stored on

or off the device. For example, a device can generate a bit-vector (i.e., one bit for each (i,j) oscillator pair) during an enrollment phase, which indicate whether to use each pair or not. However, depending on how this bit-vector is stored and accessible, information about which ring oscillators are close to each other may be exposed.

[063] In another example, the frequency of each ring oscillator is compared to an average frequency of all the ring oscillators. If an oscillator is faster than the average, the response associated with that oscillator is 1, otherwise 0. Therefore, in this example, N bits are generated from N oscillators. A subset of the oscillators is selected according to how different their frequencies are from the average, with the oscillators close to the average not being used to generate the volatile key. The identification of the oscillators with frequencies sufficiently different than the average can be stored on or off the device and later used to regenerate the volatile value. Note that this identification information does not reveal any information whether any particular oscillator is faster or slower than the average frequency.

[064] One possible limitation of this example that may outweigh advantages (such as less revealed information) in some applications is that response bits may not be uniformly distributed. Because a bit is generated by comparing a ring oscillator frequency with the average obtained on the chip, response bits from different oscillators may be correlated. For example, if there are only two ring oscillators (n=2), then the average frequency is always between the two ring oscillator frequencies, resulting in the response of either 10 or 01, but not 00 or 11.

[065] Some examples follow the general approach illustrated in FIGS. 4A-B, but have a goal to replace the error correcting code or to reduce the complexity of the error correcting code by reducing the error rate. For calibration, a chip generates error reduction (correction) information (E) as well as a response (B). To re-generate the same response, the chip is given the same error reduction information.

[066] In some examples, for calibration of a device with N oscillators, an N-bit mask is generated that indicates whether to use each oscillator or not. For example, a bit in the mask is set if the corresponding oscillator frequency is too close to the average or the corresponding (e.g., adjacent) pair consists of oscillators with very close frequencies. For re-generation, only oscillators or pairs whose mask bit is not set are used.

[067] In some examples, for calibration of a device with N oscillators, an M=N*(N-1)/2 bit mask is generated that indicates whether to use each pair of oscillators or not. The bit is set if the corresponding pair consists of oscillators with very close frequencies. Optionally, in order to prevent adversaries from guessing the ordering of oscillators from the mask, some additional bits of the M-bit mask are randomly set so that it is difficult to guess if a

certain oscillator pair is close or not. For re-generation, only oscillator pairs whose mask bit is not set are used. The mask bit can be compressed to save storage space.

[068] In some examples, rather than storing a bit vector or a compression of the bit vector, a quantity that can be used to generate a bit vector is stored, and the bit vector is generated when needed. For example, the quantity can be a seed for a pseudo-random number generator, such as a Linear Feedback Shift Register (LFSR). In order to generate a suitable bitvector, a number of seeds can each be tested according to the quality of the resulting bit vector measured according to the quality of the volatile value that is generated. For example, for a particular seed value, M oscillator pairs are generated from the seed, and the probability of each bit error on re-generation of the associated M-bit volatile value is estimated using the difference between the compared oscillator frequencies. The error probabilities for the bits in the volatile value are summed to get the expected number of bit errors for the value as a whole. The initial seed is changed, and the process is repeated. The seed that produces M response bits with lowest expected number of errors is chosen. This seed is publicly stored on or off-chip, essentially serving the function of the pair selections (R) in FIG. 4A. An ECC is optionally chosen based on the expected number of errors.

[069] In some examples, the possible N*(N-1)/2 oscillator pairs are split in a predetermined manner into P groups, each of which has K oscillators (i.e., N*(N-1)/2 = P*K). During calibration, a (K-1) bit vector is generated per group. The (K-1) bits in the vector for a group is obtained by XORing a bit generated from the first oscillator pair of the group with each of the 2^{nd} through K^{th} bits generated from the other pairs in the same group. These P (K-1)-vectors are publicly stored on- or off-chip. During re-generation, the bits from each ring oscillator pair are generated. For a given group, the (K-1) bit vector is again generated and XORed with the stored (K-1) bit vector for that group. Majority voting over the resulting (K-1) bit vector is used to determine if the bit generated from the first pair of the group flipped or not.

[070] To elaborate, consider that we have K = 4, and a particular group corresponds pairs that generate bits r1, r2, r3, r4. We generate r1 xor r2, r1 xor r3, r1 xor r4, and store it away during calibration. During re-generation, the originally generated bits r1, r2, r3, r4 are not known with certainty, and the same pairs generate new bits r1', r2', r3', r4'. Rather than using r1' in place of r1, we then compute (r1 xor r2 xor r2'), (r1 xor r3 xor r3'), and (r1 xor r4 xor r4') and use the majority vote of these three bits as the re-generated value of r1.

[071] In some examples, the N oscillators divided to form M groups, each of which consists of K oscillators (N ≥ M*K). Groups are determined at an enrollment time in a way that oscillator frequencies across different groups are as far apart as possible. The oscillator frequencies within a group are close to each other. In order to achieve this, some oscillators

can be eliminated (not included in any group). To generate bits, the groups are compared by pair-wise comparisons and majority voting. That is, the first oscillator in a group A is compared to the first in another group B, the second oscillator in group A is compared to the second in group B, etc. A majority vote is taken in order to determine the response bit. That is, each pair of groups generates one bit of the volatile value according to whether the group as a whole has a higher or lower frequency.

[072] In some examples, an error correction approach is tailored to account for the different probability of error of each of the bits in a re-generated volatile value. For example, more redundant bits are used to protect bits that are more likely to be erroneous. This optimization can lead to a smaller total number of syndrome bits (E).

[073] Given that an enrollment time or at re-generation time it is known which bits are likely to have errors and that only a small number of bits are likely to flip, errors can be corrected using a searching approach. During enrollment, a one-way hash of the volatile value is stored. To re-generate bits of the volatile value (B), we first re-generate the value ($\hat{B}$), which could have errors in a few positions in comparison to the calibrated value. Given that we have knowledge at re-generation time of the positions that are likely to have errors in the re-generated values, different 0/1 combinations of the potentially erroneous bits are tried, for each try computing the hash and compare with the stored hash to see if we have obtained the value B. This search continues until the value B is re-generated.

[074] In some examples, error control information (E) and/or selection information (R) is not required to be stored. Such examples, may have advantages because device specific information is not retained after calibration.

[075] In some examples, the bits generated from pairs of oscillators are determined based on a comparison of their frequencies with an offset. In one such example, a pair of oscillators (i,j), for $i \neq j$ may generate a 1 $f_j - t < f_i < f_j + t$ otherwise 0. In another such example, a pair of oscillators (i,j), for $i \neq j$ may generate a 1 $f_i > f_j + t$ otherwise 0.

[076] Embodiments of approached described above can use a variety of chip technologies. For example, oscillators can be placed on a FPGA (field programmable gate array) or designed into an ASIC (application specific integrated circuit). The characteristics of the oscillators, such as the within-chip, across-chip, or temperature or voltage dependent variation of oscillator frequencies can be used to predict performance. For example, bit error rate of the volatile keys can be predicted based on fabrication process characteristics of the devices hosting the oscillators. The predicted performance can then be used to select a number of oscillators or a degree of error correction that is suitable to achieve desired characteristics of the volatile values that are regenerated on the devices.

[077] Approaches described above can be combined with approached described in the copending applications that are incorporated into this document by reference. For example, the volatile value generated by the oscillators can be combined with a challenge presented to the device to produce a response to the challenge. As another example, the volatile value can be used to generate public/private cryptographic key pair, such that the private key is not disclosed outside the device.

[078] The approaches described above for selection of pairs or ring oscillators can be used for selection of subsets of more than two oscillators or for selection of signal generators other than oscillators. Also, other characteristics than frequency of signal generators may be used.

[079] Approached described above can be implemented on a device in dedicated logic circuitry, using a processor controlled by stored instructions, or a combination of dedicated circuitry and processors. The instructions for a processor may be stored in a machine-readable medium (e.g., ROM) on or off the device. For example, the processor may comprise a general-purpose processor core that uses instructions that are stored off the device to control oscillator selector circuitry and processes oscillation counts to determine the bits of a volatile value. Storage of values, such as selection information or error control information, can use a variety of technologies, including flash memory, electrically writeable read-only memory, fusible links, etc. Some implementations can be represented as instructions or data for configuring a FPGA, or in instructions or data (e.g., Verilog) that is used to specify a circuit design of an ASIC.

[080] It is to be understood that the foregoing description is intended to illustrate and not to limit the scope of the invention, which is defined by the scope of the appended claims. Other embodiments are within the scope of the following claims.

What is claimed is:

1. A method comprising:

selecting subsets of a plurality of signal generator circuits embodied in a device; and

generating a volatile value for the device from the selected subsets.

2. The method of claim 1 wherein the signal generator circuits comprise oscillators.

3. The method of claim 1 wherein each subset consists of a pair of circuits.

4. The method of claim 1 wherein each subset comprises a pair of predetermined groups of two or more circuits.

5. The method of claim 1 further comprising:

storing information characterizing the selection of the subsets.

6. The method of claim 1 further comprising:

storing error correction information computed from the generated volatile value.

7. The method of claim 1 further comprising:

authenticating the device using the volatile value.

8. The method of claim 1 further comprising:

applying a cryptographic procedure to data at the device using the volatile value.

9. The method of claim 1 further comprising:

re-generating the volatile value for the device from the selected subsets.

10. The method of claim 9 wherein re-generating the volatile value includes using stored information characterizing the selection of the subsets of circuits.

11. The method of claim 9 wherein re-generating the volatile value includes using stored error correction information computed from the generated volatile value.

12. The method of claim 1 wherein selecting the subsets of circuits and generating the volatile value are performed as part of an enrollment procedure.

13. The method of claim 12 further comprising storing information characterizing the selection of the subsets as part of the enrollment procedure.

14. The method of claim 1 wherein generating the volatile value includes generating a different portion of the value from each of the selected subsets.

15. The method of claim 14 wherein generating the volatile value includes generating a different bit of the value from each of the selected subsets.

16. The method of claim 1 wherein selecting the subsets includes selecting the subsets according to an anticipated error rates associated with corresponding subsets of the circuits.

17. The method of claim 1 wherein selecting the subsets includes selecting the subsets according to a comparison of the outputs of the subsets of circuits.

18. The method of claim 17 wherein selecting the subsets includes selecting the subsets according to a comparison of oscillation frequencies of the subsets of circuits.

19. The method of claim 18 wherein selecting the subsets according to a comparison of oscillation frequencies of the subsets of circuits includes selecting subsets with oscillation frequencies that differ according to a predefined criterion.

20. An integrated circuit comprising:

a plurality of signal generator circuits;

selection circuitry coupled to the signal generator circuits for outputting a selection of subsets of the plurality of signal generator circuits; and

a volatile value generator coupled to the signal generator circuits for outputting a value according to the selected subsets.

21. The integrated circuit of claim 20 wherein the signal generator circuits comprise oscillators.

22. The integrated circuit of claim 20 further comprising:

an interface to the circuit for providing and for retrieving the selection of subsets of the plurality of signal generator circuits.

23. The integrated circuit of claim 20 further comprising:

a storage for the selection of subsets of the plurality of signal generator circuits.

24. The integrated circuit of claim 20 further comprising:

an error correction module coupled to the volatile key generator configured correct the value outputted from the volatile value generator according to error control information.

25. The integrated circuit of claim 24 further comprising:

an interface to the circuit for providing and for retrieving the error control information.

26. The integrated circuit of claim 24 further comprising:

storage for the error control information.

27. Software stored on computer-readable media comprising data for causing a device to perform functions including:

selecting subsets of a plurality of signal generator circuits embodied in a device; and

generating a volatile value for the device from the selected subsets.

28. The software of claim 27 wherein the data comprises data for configuring programmable circuitry to perform the functions.

29. The software of claim 27 wherein the data comprises instructions for generating a device specification for fabricating the device.

30. The software of claim 27 wherein the data comprises instructions for causing a computing system to perform the functions.

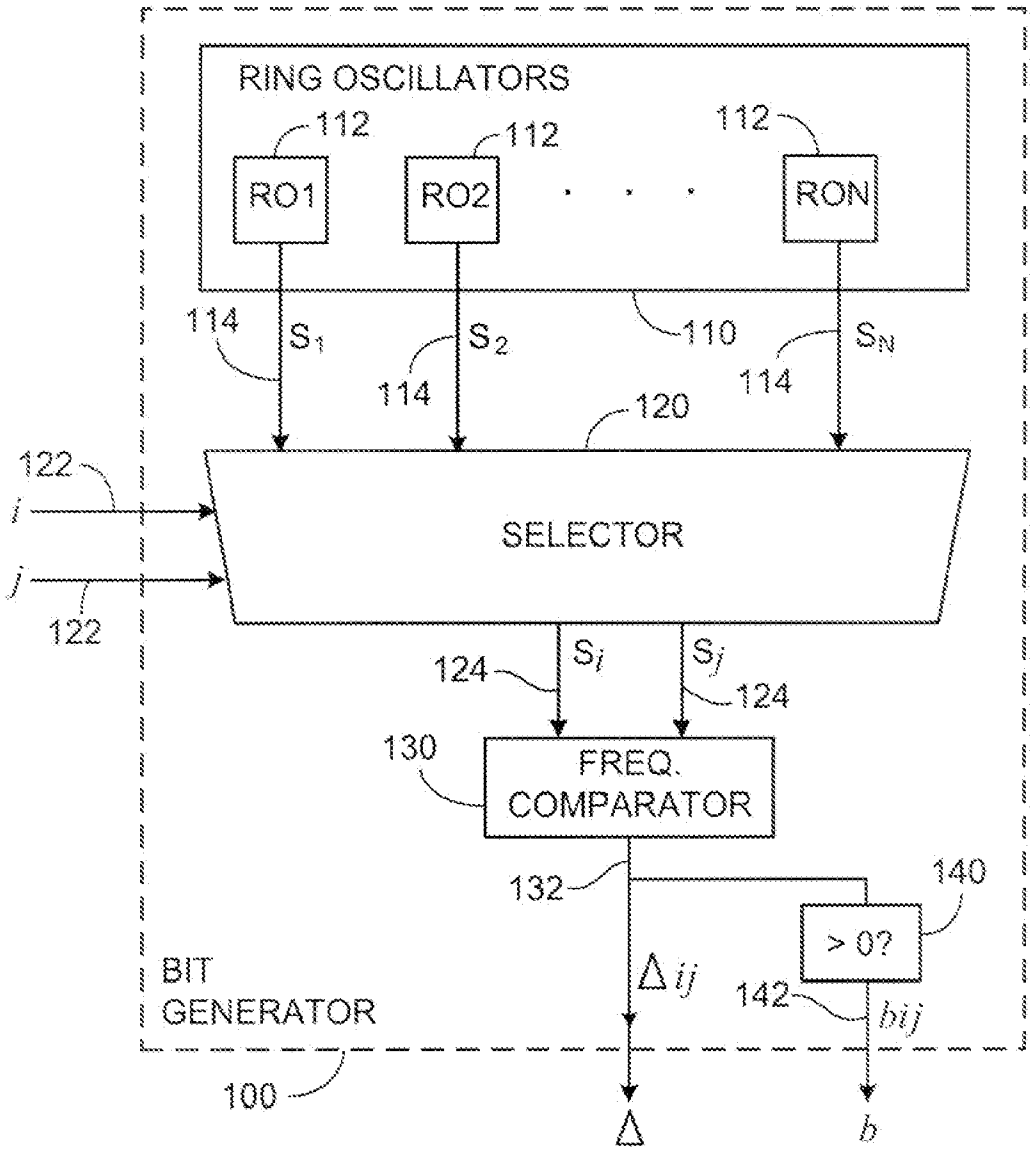

RING OSCILLATORS
112
112
112
RO1
RO2
RON
114
S_1
S_2
S_N
114
114
110
120
122
i
j
122
SELECTOR
124
S_i
S_j
124
130
FREQ. COMPARATOR
132
140
> 0?
Δ_{ij}
142
bij
BIT GENERATOR
100
Δ
b

FIG. 1

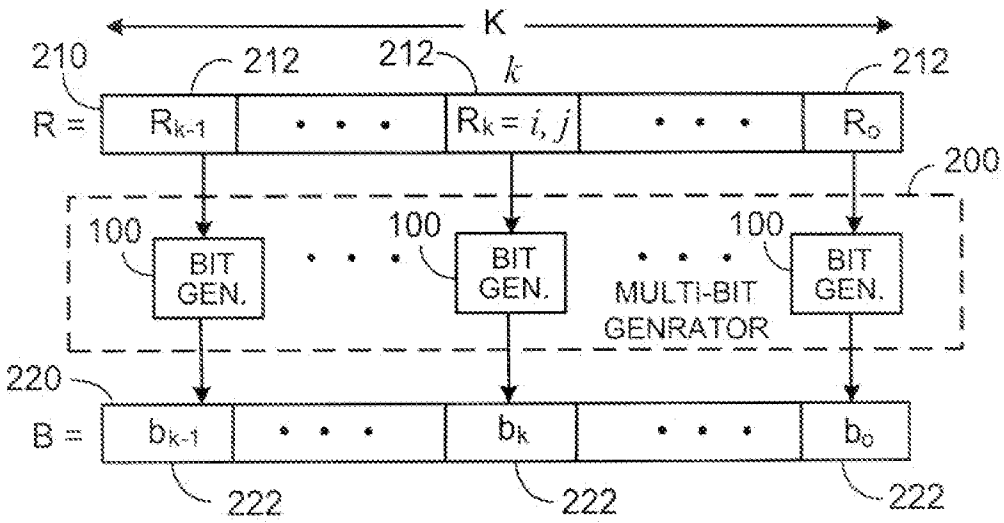

K
210
212
212
k
212
R =
R_{k-1}
$R_k = i, j$
R_0
200
100
100
100
BIT GEN.
BIT GEN.
MULTI-BIT GENRATOR
BIT GEN.
220
B =
b_{k-1}
b_k
b_0
222
222
222

FIG. 2

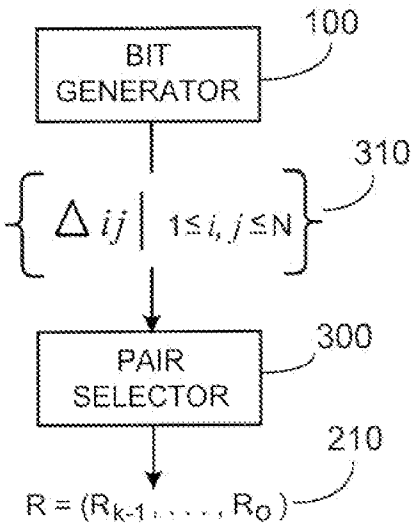

BIT GENERATOR
100
310
$\Delta ij \mid 1 \le i, j \le N$
PAIR SELECTOR
300
210
$R = (R_{k-1}, \ldots, R_0)$

FIG. 3

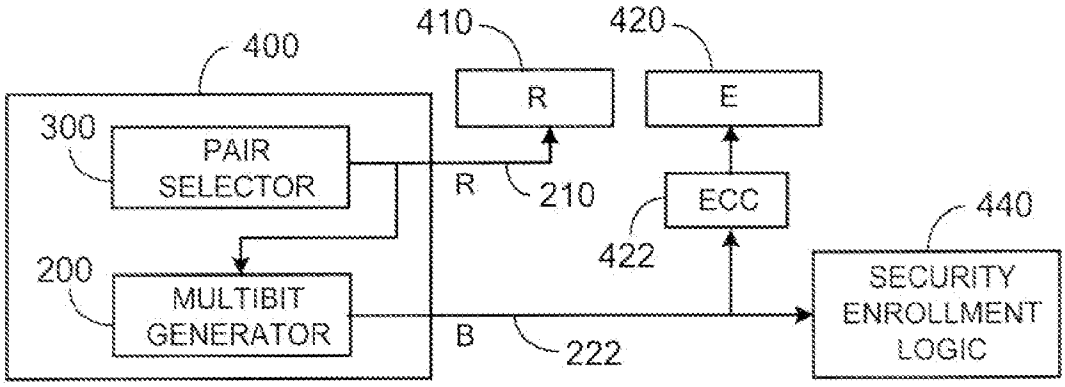

400
410
420
R
E
300
PAIR SELECTOR
R
210
ECC
422
440
200
MULTIBIT GENERATOR
B
222
SECURITY ENROLLMENT LOGIC

FIG. 4A

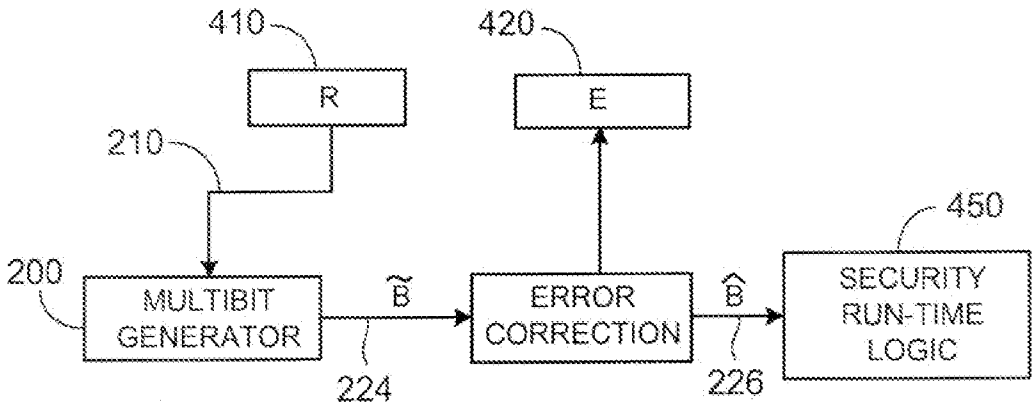

410
420
R
E
210
450
200
MULTIBIT GENERATOR
$\widetilde{B}$
ERROR CORRECTION
$\widehat{B}$
SECURITY RUN-TIME LOGIC
224
226

FIG. 4B

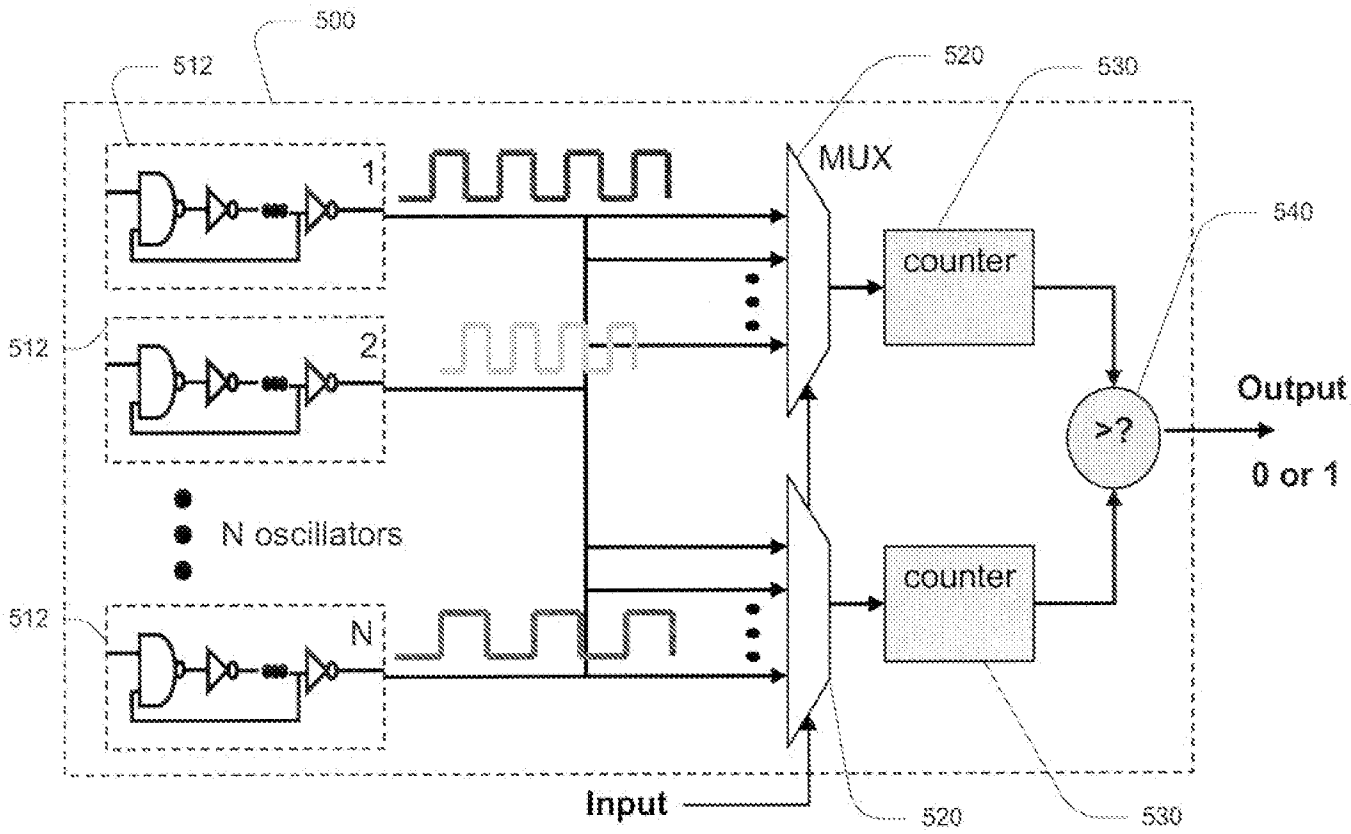

500
512
520
530
MUX
1
2
N
512
512
N oscillators
counter
counter
540
>?
Output
0 or 1
Input
520
530

FIG. 5

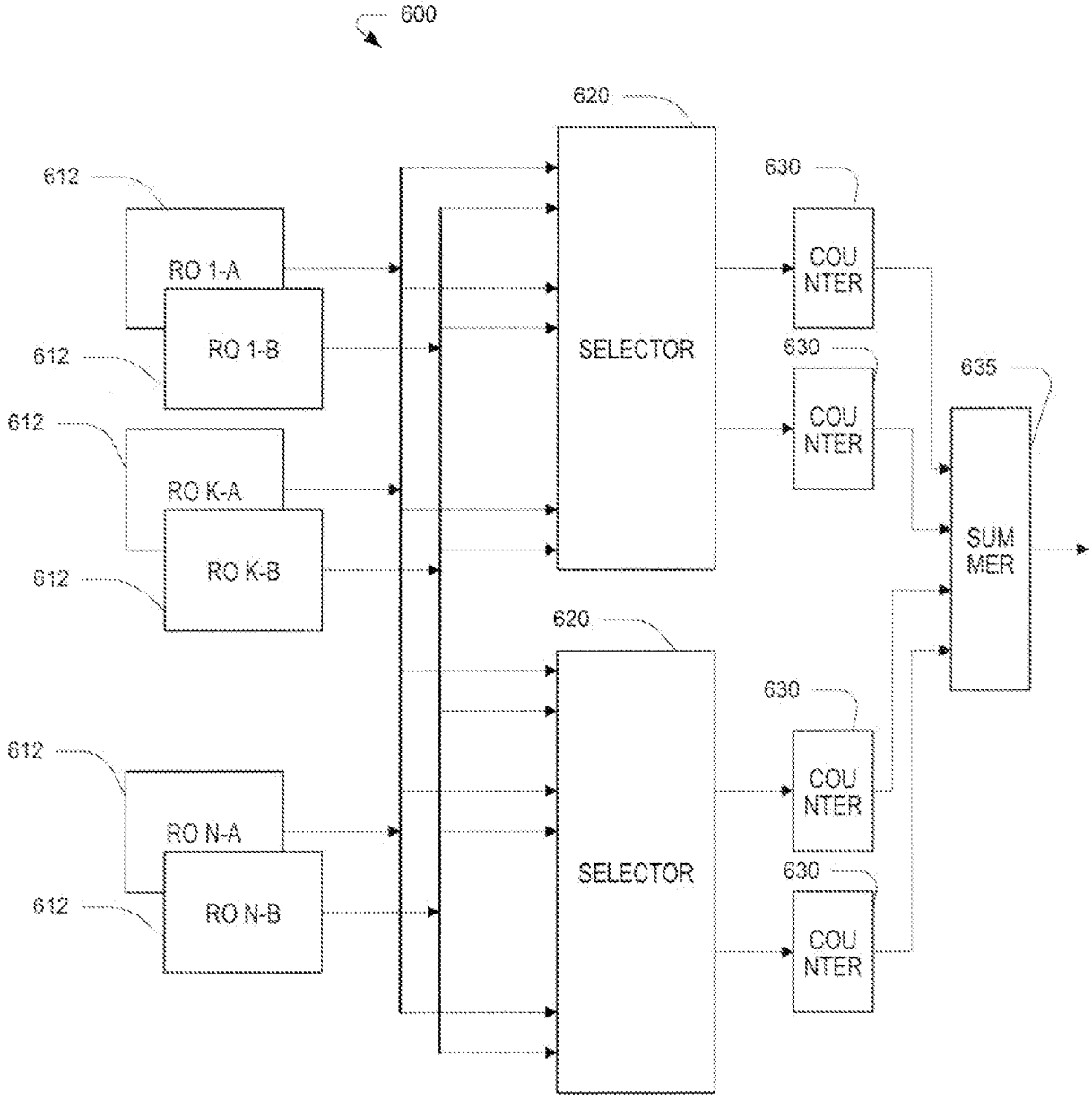

600
620
612
RO 1-A
RO 1-B
612
612
RO K-A
RO K-B
612
SELECTOR
630
COU
NTER
630
COU
NTER
635
SUM
MER
620
612
RO N-A
RO N-B
612
SELECTOR
630
COU
NTER
630
COU
NTER

FIG. 6