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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 October 2009 (15.10.2009)

PCT

(10) International Publication Number
WO 2009/126322 A2

(51) International Patent Classification:
*G06F 11/30* (2006.01) *G06F 1/26* (2006.01)

(21) International Application Number:
PCT/US2009/002268

(22) International Filing Date:
10 April 2009 (10.04.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/101,598 11 April 2008 (11.04.2008) US

(71) Applicant *(for all designated States except US)*: **ADVANCED MICRO DEVICES, INC.** [US/US]; One AMD Place, Mail Stop 68, P.O. Box 3453, Sunnyvale, CA 94088-3453 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **NAFFZIGER, Samuel, D.** [US/US]; 3749 Ashmount Drive, Fort Collins, CO 80525 (US).

(74) Agent: **DRAKE, Paul, S.**; Advanced Micro Devices, Inc., 7171 Southwest Parkway, Mail Stop B100.3.341, Austin, TX 78735 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, 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, LY, MA, MD, ME, 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, ST, 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, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR),

*[Continued on next page]*

(54) Title: SAMPLING CHIP ACTIVITY FOR REAL TIME POWER ESTIMATION

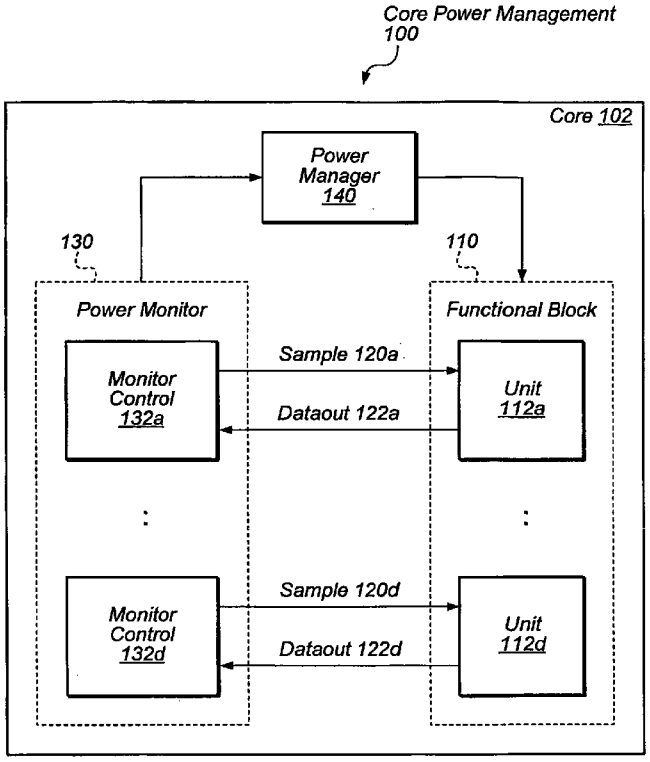


FIG. 1

(57) Abstract: A system and method for real-time power estimation. A core may be divided into units. Each unit is simulated to achieve a real power consumption characterization. The power consumption is sampled. Statistical anaylsis is performed that assumes the core has node capacitance switching behavior that is approximated by a stationary random process with a Poisson distribution. The statiscial analysis determines the number of samples to take during a sample interval. The operational frequency, sample interval, and number of samples are used to determine the number of signals to sample. Signals are chosen that have a high correlation with the node capacitance switching behavior, such as clock enable signals on the last stage of a clock distribution system. Weights with tuned values are assigned to each sampled signal. Sampling occurs during every predetermined number of clock cycles. The weights of asserted sampled signals are summed in order to determine a repeatable power estimation value.

WO 2009/126322 A2

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 (Rule 48.2(g))*

# TITLE:SAMPLING CHIP ACTIVITY FOR REAL TIME POWER ESTIMATION

## BACKGROUND OF THE INVENTION

### Technical Field

[0001] This invention relates to electronic circuits, and more particularly, to an efficient method of repeatable accurate real-time power estimation.

### Background Art

[0002] The power consumption of modern integrated circuits (IC's) has become an increasing design issue with each generation of semiconductor chips. Increased IC power consumption directly affects the cost of building a system. As power consumption increases, more costly cooling systems such as larger fans and heat sinks must be utilized in order to remove excess heat and prevent IC failure. However, cooling systems increase the system cost. The IC power dissipation constraint is not only an issue for portable computers and mobile communication devices, but also for high-performance superscalar microprocessors, which may include multiple processor cores, or cores, and multiple pipelines within a core.

[0003] The power consumption of IC's, such as modern complementary metal oxide semiconductor (CMOS) chips, is proportional to the expression $\alpha fCV^2$. The symbol $\alpha$ is the switching factor, or the probability a node will charge up or discharge during a clock cycle. The symbol f is the operational frequency of the chip. The symbol C is the equivalent capacitance, or the switching capacitance, to be charged or discharged in a clock cycle. The symbol V is the operational voltage of the chip. In an attempt to reduce the power consumption of the chip, and to reduce the electrostatic fields within the transistors, the power supply voltage V may be decreased. There is a limit to the power supply voltage reduction, since this reduction deceases the amount of current that may flow through a transistor and, thus, increases the propagation delays through transistors. If the threshold voltages are reduced in order to turn-on the transistors at an earlier time and aid in maintaining performance, then transistor leakage current increases. An increase in transistor leakage current both increases power consumption and the potential for logic failure.

[0004] The geometric dimensions of devices and metal routes on each generation of cores are decreasing. Superscalar designs increase the density of integrated circuits on a die with multiple pipelines, larger caches, and more complex logic. Therefore, the number of nodes that may switch per clock cycle significantly increases. Operational frequencies have been increasing with each generation too. The switching factor, $\alpha$, may decrease for certain blocks or units in the chip by disabling the clock to these areas during periods of non-use. Therefore, although the operational voltage and switching factor may be decreasing with each generation of ICs, the other terms in the IC power consumption expression are increasing and cause an overall increase in power consumption. Some solutions to this power consumption increase include both microarchitectural and circuit-level techniques.

[0005] These design techniques are aided by efforts to estimate the power consumption in real-time. For example, while running an application or applications, if the real-time power consumption of a core is both known and conveyed to a power manager, then the power manager may alter the operational supply voltage, the operational frequency, or both in order to increase performance during low power consumption periods, or to decrease power consumption during high power consumption periods.

[0006] Real-time power estimation may be achieved by a monitor measuring the switching capacitance on a die during a particular clock cycle. A node capacitance, Cac, comprises both the switched, or ac, capacitance, and the effective capacitance resulting from crossover current. For a given part, leakage current is fixed when operation has reached the temperature limit. This leakage value can be accounted for by a fused-in power offset based on measurements at test time. Attempts to take advantage of temperature induced variations in leakage current is not encouraged due to repeatability reasons discussed below. Therefore, what is needed in order to obtain an accurate real-time estimation of the chip's power is the operational supply voltage, which is set digitally, the operational frequency, which is known, the fused-in leakage value found during testing, and a measurement of the number of nodes in the design switched in a particular clock cycle along with the node capacitance, Cac. The latter term, Cac, is not a straight-forward value to measure on a semiconductor chip.

[0007] An on-chip monitor may attempt to measure the switching node capacitance, Cac. The monitor may measure the instruction issue rate within a core or within a

processor with multiple cores. Accuracy increases if the behavior of major components such as the instruction issue queue, the reorder buffer, the physical register files, the execution units, and other, and their respective interactions are monitored. However, sampling the hundreds of signals to accomplish this task provides a huge overhead in additional metal routes, increased circuitry, and added power consumption.

[0008] Further, a monitor and subsequent response mechanism, such as a power manager, need to be deterministic. In other words, the monitor and mechanism need to provide the same results, or frequency changes, from part-to-part, system-to-system, and run-to-run. The reason for this requirement is Original Equipment Manufacturers (OEMs) using the processor in one of their systems need to provide performance guarantees to customers. Customers and OEMs need to replicate the benchmarks and performance measures at different times and in different locations. If this replication can not be done within a tight tolerance (i.e. +/- 1.5%), then the monitor and response mechanism need to be turned off. Now the monitor and mechanism take up area on the die without performing useful work to improve performance and power savings.

[0009] Analog sensors, such as temperature sensors and/or an ammeter, are able to provide accurate power estimations, but all of them are environment dependent. Variations in fabrication processes, ambient temperature, power supplies, and the quality of the heat removal solution alter the measurements of analog sensors.

[0010] In view of the above, efficient methods and mechanisms for providing an accurate digital real-time power estimation of an IC are desired.

## DISCLOSURE OF INVENTION

[0011] Systems and methods for accurate real-time power estimation are contemplated.

[0012] In one embodiment, a system is provided comprising a functional block, a power monitor, and a power manager. The power manager is configured to, in response to a power estimate number, convey operational voltage and frequency adjustments to the functional block in order to reverse power consumption behavior within the functional block. The functional block is configured to store a logic value for each of a predetermined number of sampled signals and convey the logic values to the power monitor. The power monitor is configured to store a weight value for each of the

predetermined number of sampled signals, to sum the weight values of the only the sampled signals that have a stored logic value of a logic high value, and to convey the power estimate number to the power manager, wherein the power estimate number is translated from the sum of the weight values.

[0013] In another aspect of the invention, a method is provided to, in response to a power estimate number, convey operational voltage and frequency adjustments in order to reverse power consumption behavior. There is storage of a logic value for each of a predetermined number of sampled signals. There is also storage of a weight value for each of the predetermined number of sampled signals. An accumulation follows of the weight values of only the sampled signals that have a stored logic value of a logic high value. The power estimate number is translated from the sum of the weight values and is conveyed to a power manager.

[0014] In another aspect of the invention, a power monitor is provided comprising an interface configured to communicate with a functional block and a power manager. A weight table stores a weight value for each of the predetermined number of sampled signals. A weight accumulator sums the weight values of only the sampled signals that have a stored logic value of a logic high value. A chain control conveys a power estimate number to the power manager, wherein the power estimate number is translated from the sum of the weight values.

[0015] These and other embodiments will be appreciated upon reference to the following description and accompanying figures.

## BRIEF DESCRIPTION OF DRAWINGS

[0016] FIG. 1 is a general block diagram illustrating one embodiment of an IC, such as a core, with power management.

[0017] FIG. 2A is a generalized diagram illustrating one embodiment of a real power measurement of an application executed on a core during a sample interval.

[0018] FIG. 2B is a generalized diagram illustrating one embodiment of a digital sampling of a real power measurement of an application executed on a core during a sample interval.

[0019] FIG. 3 is a generalized diagram illustrating one embodiment of circuitry for sampling signals and serially outputting the values of the sampled signals.

[0020] FIG. 4 is a generalized diagram illustrating one embodiment of a timing diagram for outputting sampled signal information from a unit.

[0021] FIG. 5 is a general block diagram illustrating one embodiment of a power estimation monitor circuit.

[0022] FIG. 6 is a general block diagram illustrating one embodiment of a method for accurate real-time power estimation of a core.

[0023] FIG. 7 is a general block diagram illustrating one embodiment of a method for determining weights to assign to sampled signals for accurate real-time power estimation of a core.

[0024] While the invention is susceptible to various modifications and alternative forms, specific embodiments are shown by way of example in the drawings and are herein described in detail. It should be understood, however, that drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

## MODE(S) FOR CARRYING OUT THE INVENTION

[0025] In the following description, numerous specific details are set forth to provide a thorough understanding of the present invention. However, one having ordinary skill in the art should recognize that the invention may be practiced without these specific details. In some instances, well-known circuits, structures, and techniques have not been shown in detail to avoid obscuring the present invention.

[0026] Referring to FIG. 1 one embodiment of a core power management 100 is shown. Here, core 102 may be any integrated circuit (IC). In one embodiment, core 102 may be a processor core. A processor core may have an on-die instruction and data cache. The processor core may be a superscalar processor with a single pipeline or multiple pipelines. In another embodiment, core 102 may be an application specific IC (ASIC). Any transistor family may be use to implement core 102. Examples include metal oxide semiconductor field effect transistors (MOSFETs) and bipolar junction transistors (BJTs).

[0027] A functional block 110 may include transistors configured to perform logic functions, data storage, or other. For power management purposes, functional block 110 may

be divided into units 132a-132d. As used herein, elements referred to by a reference numeral followed by a letter may be collectively referred to by the numeral alone. For example, units 132a-132d may be collectively referred to as units 132. In one embodiment, units 132 may not correspond to functional components, such as a reorder buffer, a memory management unit, an execution unit, or other of a processor. Rather, units 132 may be selected based on the types of signals that need to be sampled for power management purposes. For example, clock enable signals routed to local clock distribution blocks may be the signals chosen to be sampled.

**[0028]** The selection of which signals to sample during a particular clock cycle may correspond to how well the selection correlates to the amount of switching node capacitance, Cac. A further description of the sample signal choice is provided later. The selected clock enable signals may overlap functional blocks in the floorplan. Therefore, the division separating, for example, unit 132a and 132b may not correspond to a division in the floorplan. Units 132 are units that consume power and this power is to be measured in real-time.

**[0029]** In one embodiment, Power Monitor 130 may be used to collect data from units 132, such as the logic values of all the predetermined sampled signals. After collecting the data, Power Monitor 130 may calculate a power consumption estimation. One Monitor Control 132 may correspond to each Unit 132. In alternative embodiments, a Monitor Control 132 may collect data for two or more Units 112 and calculate a total power estimation for those Units 112. In yet another embodiment, one Monitor Control 132 (i.e. Control 132a) may have a signal interface with one or more other Monitor Controls 132 (i.e. Controls 132b-132d) in order to collect data from the one or more Monitor Controls 132 (i.e. Controls 132b-132d). Then a power estimation for the one or more Monitor Controls 132 may be calculated.

**[0030]** Signals Sample 120 and Dataout 122 are control and data signals used for power management purposes. The interface signals between Power Monitor 110 and Functional Block 130 may comprise any necessary number of signals and communication protocols. In one embodiment, the control signal Sample 120 may be asserted for a single clock cycle only during a chosen repeating interval, such as every 100 clock cycles. In one embodiment, at a predetermined number of clock cycles after the control signal Sample 120 is asserted, the data signal Dataout 122 may begin providing a logic value for a different sampled signal each clock cycle. In

other words, the data signal Dataout 122 may be used to scan out a chain of values comprising the logic values of the sampled signals at a particular cycle. Also, in other embodiments, there may not be a single pair of signals between each Monitor Control 132 and Unit 112 pair. In an alternative embodiment, additional signals may be included in order for a Monitor Control 132 to poll a Unit 112, for a Unit 112 to acknowledge to a Monitor Control 132 that it is ready to convey output data.

**[0031]** Once Power Monitor 130 calculates an accurate real-time power estimation from the data received from Functional Block 110, then Power Monitor 130 has determined the power profile of the currently running application(s). This determination is conveyed to Power Manager 140. Now the Power Manager 140 has an ability to decrease (or increase) power if the application is above (below) a threshold limit.

**[0032]** If a high power application or virus is executed on core 102, Power Manager 140 comprises circuitry configured to make adjustments to the operating voltage, the operational frequency, or both. For applications with an average power above a limit set for Cac, the operational frequency needs to be reduced. However, lower limits set for Cac allow the part to be binned at a higher frequency since it's guaranteed that no application will switch more than the Cac limit.

**[0033]** As mentioned before, one manner to estimate the core switching power is measuring the instruction issue rate. Unfortunately, for some embodiments of a core that may have many specialized operations, it is unlikely to have a few simple methods for estimating node capacitance, Cac, switching in a clock cycle due to data widths, performance enhancing circuitry, cache behavior uncertainties, one of many branch prediction methods, and so forth. In order to accurately estimate Cac from a few uncorrelated signals, those signals are not known until a decent power model is designed and a broad set of applications have run through it for signal power correlation checks. By that time it's too late to modify the design by trying to add the requisite signal sampling circuitry in the needed locations.

**[0034]** On the other hand, it is possible to sample a broad set of signals in the design that have a significant impact on power estimation. This approach may over-sample signals since many of the signals are likely correlated, and some signals may only have a small impact on power. However, if the design cost of somewhat over-sampling is small, such as with an efficient implementation, then an increase in the confidence of an accurate Cac monitor may increase significantly with this approach.

[0035] As mentioned before, sampling hundreds of signals in the design that have a non-trivial impact on Cac would be a significant design cost in terms of overhead in wires, circuitry, and even power consumption if the monitoring were done at full operational frequency. However, it is possible to take advantage of the fact that a Cac estimation that leads to a power estimation only needs to respond in thermal time frames, which is in the range of milliseconds. This time frame allows an estimation to use millions of cycles over which to determine an average Cac. In fact, if an estimation and subsequent response occurred too quickly, and responded to short term periods of high activity in an application, then the response may throttle the core. Then the performance may be reduced unnecessarily. An estimation needs to monitor the average power of the application over longer periods of time upon which the subsequent response and possible throttling is based. However, if the sample interval is extended to millions of cycles, the sigma, or standard deviation, of samples will increase significantly since the application has a decent probability of moving to a completely unrelated section of code. Based on power model data, a good correlation of activity for samples may be found in a certain range. For example, in one embodiment, a good correlation of activity for samples for a typical application may be in the 50K clock cycle domain where sigma is small.

[0036] If a few hundred known signals are chosen to be sampled upfront and the signal values are scanned out in a serial fashion, then a few hundred clock cycles are required to get a single sample of the activity related signals. Then from standard statistical analysis, a few samples are required of these few hundred signals before a confidence is reached (i.e. 99%) that the mean and sigma of the samples have a close correlation to the mean and sigma of the actual population of signals.

[0037] It is assumed that the switching behavior of the application can be approximated by a stationary random process with a Poisson distribution. Also, the central limit theorem tells us that a large population of data can be accurately approximated with a normal distribution. This statistical approach is similar to an approach used to produce opinion polls from a small sample of the population, statistical census numbers, etc. The characteristics of the broader population is reconstructed by taking a sufficient number of random samples. Two parameters that need to be determined are the number of necessary samples and which signals to sample.

[0038] For the number of necessary samples, in one embodiment, a spreadsheet may be generated using statistical analysis and measurements of both the real power consumption of an application and estimated power consumption from a sampling.

[0039] FIG. 2A illustrates one embodiment of a real power measurement 200 of an application executed on a core during a sample interval. In an alternative embodiment, this power measurement may be performed on a subdivision of the core, such as a unit, a functional component, or other. In one embodiment, this measurement may be performed by a power model executed on a pre-silicon model of the core. Peak power 210 may be defined by functional failure of the core or by executing a high power virus on the core, such as Zuraski. The consumed power during the sample interval is shown by Real Power Waveform 220. In one embodiment, a sample interval may be in the range of tens of microseconds.

[0040] In modern processors, a thermal design power (TDP) may be set by a de-rated form of the high power virus Zuraski. This de-rated, or derived, TDP code and its corresponding power consumption is used to set the operational voltage and frequency of the core for binning purposes. During most of a modern processor's lifetime however, the applications that are executed, consume no more than 60% to 80% of this TDP value.

[0041] With an accurate real-time Cac estimation, and following power estimation, it is possible to exploit the 20% to 40% gap between an executed application and the TDP. This may be done by turning the gap into either a TDP reduction, wherein the new TDP is the application power, or a frequency increase, wherein the TDP is the same as before, but is consumed by typical applications. The TDP would not be consumed by a virus, since the virus would get throttled by a subsequent response by a power manager following the Cac estimation.

[0042] FIG. 2B illustrates one embodiment of a digital sampling 230 of a real power measurement of an application executed on a core during a sample interval. A Sampled Datapoint 240 is taken during each number of Intermittent Cycles 250. In the illustrated embodiment, a total of N samples are taken during the sample interval beginning with Sampled Datapoint 240a and ending with Sampled Datapoint 240n. The determination of the number of intermittent cycles between sampling a power estimate value may depend on The determination of the values of the number of

sampled datapoints, N, and the number of intermittent cycles depends on the desired accuracy and confidence.

**[0043]** As stated above, a spreadsheet, or look-up table, may be generated using both statistical analysis and measurements of both the real power consumption of an application and estimated power consumption from a sampling as shown in FIG. 2A-2B. Next a confidence and an error rate need to be chosen.

**[0044]** For example, in a first example, design goals may require a 95% confidence that the mean of a sample, such as one illustrated in FIG. 2B, over an entire core is within 12% of a sigma, or standard deviation, of the mean of the real application, such as one illustrated in FIG. 2A. The term 0.12*sigma is the error due to statistical sampling. Using the look-up table or spreadsheet, it may be determined that at least 500 samples, for example, are needed to achieve these design requirements. Recall that a Cac estimation that leads to a power estimation only needs to respond in thermal time frames, which is in the range of milliseconds. This time frame allows an estimation to use millions of cycles over which to determine an average Cac. A sample interval may be 25μs. In one embodiment, the operational frequency may be 3GHz. For this example, then the number of intermittent clock cycles between samples would be (3GHz)*(25μs)/500 = 150 clock cycles. In one embodiment, the values of the sampled signals may be scanned out in a serial manner. Therefore, 150 signals may be sampled in a single clock cycle and serially scanned out before the next sample is performed.

**[0045]** In this example, an absolute worst case estimation scenario may include a clipped normal distribution of +/- 2 sigma, wherein one sigma is 25% of the range. Therefore, a worst case error is 0.12*0.25 = 3%. A more typical maximum sigma value is approximately 10%. With an error term of 12% sigma, it is possible to have a 95% confidence that the error due to statistical sampling is 0.12*0.1 = 1.2%.

**[0046]** In a second example, design goals may require a 99% confidence that the mean of a sample, such as one illustrated in FIG. 2B, over an entire core is within 10% of a sigma, or standard deviation, of the mean of the real application, such as one illustrated in FIG. 2A. The term 0.10*sigma is the error due to statistical sampling. Using the look-up table or spreadsheet, it may be determined that at least 650 samples are needed to achieve these design requirements. Further, in one embodiment, the core may be divided into 10 units, wherein each unit samples the

same number of signals. The reason for this division is to increase accuracy as will be shown shortly. For each unit, only a 90% confidence is required. The sum of the confidences yields a 99% total confidence as $0.9^{1/10}$ is 0.9895. Using the look-up table or spreadsheet again, it may be determined for 650 samples and 90% confidence, the deviation from sigma is no longer 10%, but it is reduced to 6.5%. Although for an individual unit, the confidence level has decreased as its accuracy increased, the overall confidence level for the core remains unchanged at 99% and now its accuracy has improved from 10% to 6.5%.

**[0047]** If this second example has the same sample interval as above for the first example of 25μs, and the operational frequency is the same at 3GHz, then the number of intermittent clock cycles between samples would be (3GHz)*(25μs)/650 = 115 clock cycles. In one embodiment, as in the first example, the values of the sampled signals may be scanned out in a serial manner. Therefore, 150 signals may be sampled in a single clock cycle from each of the 10 units and serially scanned out before the next sample is performed.

**[0048]** Referring to FIG. 3, one embodiment of circuitry 300 for sampling signals and serially outputting the values of the sampled signals is shown. In one embodiment, in order to capture the chosen signals to sample within a clock cycle, D flip-flop circuits 320 and multiplexers 310 may be placed as shown only for the signals to be sampled, which are represented by Signal 350. This added circuitry provides a low overhead for obtaining sampled data for accurate real-time Cac estimation. When the selection line, Sample 120, is asserted to a logic high value, Signal 350 is selected by the multiplexer 310 and stored by the D flip-flop 320.

**[0049]** The signal Sample 120 may be received from a control block such as Monitor Control 132 of FIG. 1. In one embodiment, the number of signals to sample in a unit on the core may be 100. In alternative embodiments, the number of signals to sample may be any number that provides an accurate power estimation for the particular core design. Sample 120 may be asserted for a clock cycle every 100 cycles, which allows the captured values of the sampled signals to be serially output between samples.

**[0050]** The clock signal Clk 330 that is routed to the D flip-flops 320 may be an ungated clock signal. When the signal Sample 120 is deasserted to a logic low value, the multiplexers 310 convey the output value of the previous D flip-flop 320. In this manner, the values during the sampled clock cycle of the 100 sampled signals, Signal

350, are output in a serial manner. Depending on the routing distance to the Monitor Control 132, inverter buffers may be placed following D flip-flop 320a.

**[0051]** In one embodiment, a signal chosen to be sampled may be the clock gater enable signals. A gater circuit may be the last stage of a clock distribution system that generates a clock signal. Typically, the gater circuits are in local blocks of a floorplan of a modern processor and local designers choose from a clock library a gater circuit to drive a certain clock load in the block.

**[0052]** The Gater Circuit 340 in FIG. 3 receives a Common Clock 360 from a previous stage in the clock distribution system and an enable signal, Clk Enable 362. Clk Enable 362, in the embodiment shown, is the signal to be sampled and is an input to multiplexers 310. The Gater Circuit 340 may also be coupled to scan input and output signals among other types of signals.

**[0053]** As discussed above concerning the statistical analysis, the error of a Cac estimation due to statistical sampling may be small. However, there may be errors due to an inaccurate correlation of the sampled signal activity to the node switching capacitance, Cac. The clock gater enable signals, Clk Enable 362, are relatively easy to locate by local designers. Also these signals directly determine whether downstream circuitry is active or not, and thus, whether or not the downstream node capacitances may switch or not.

**[0054]** For a particular Clk Enable 362 that is sampled, a corresponding stored value of a logic high value implies at least a minimum capacitance is being switched. An additional amount of capacitance is conditionally switched, which is determined by other signals and configurations that are downstream of the clock gater enable, Clk Enable 362. Some examples of signals downstream of the gater clock enable signals that determine how Cac changes in various ways are valid bits, such as the valid bit in a load-store queue, for one example; an operand bit width; and a cache hit or miss. Although not shown in FIG. 3, in alternative embodiments, some of these signals and other signals may be logically combined with the clock gater enable signal to generate a new signal to be sampled and input to multiplexers 310. This new signal may determine the incremental capacitance to add to the minimum capacitance switched.

**[0055]** Also, there are signals whose assertion indicates an amount of switched capacitance aside from the clock gater enables. Some examples are bus driver enables,

mismatches in content-addressable memories (CAM), and CAM word-line (WL) drivers. A transition detector may be needed to determine the additional switched capacitance for these signals.

**[0056]** FIG. 4 illustrates one embodiment of a timing diagram 400 for signals Clock 330, Sample 120, and Dataout 122 in circuit 300 of FIG. 3. Clock 330 is an ungated clock signal. When Sample 120 is asserted, the sampled signals, Signals 350, are stored in D flip-flops 330. When Sample 120 is deasserted, the sampled signals, Signals 350, are serially output in a chain on the line Dataout 122.

**[0057]** Once the sampling of signals is done accurately, this information may be turned into a Cac estimate using a weighted average sum. Referring to FIG. 5, one embodiment of circuitry 500 for determining a weighted average sum is shown. This circuitry may reside in Monitor Control 132. Chain control 502 may convey Sample 120 to a unit within Functional Block 110. Also, chain control 502 may receive Dataout 122. After Sample 120 has been asserted in order to sample data and has been deasserted, chain control 502 may track which signal, or bit position in the serial chain, currently has its value being received. In one embodiment, a counter may be used to distinguish the sampled signals. The output of such a counter may be used to index Weight Table 504. The value conveyed by chain control 502 may be compared to entries Bit Positions 506 in Weight Table 504. When an entry hits, or matches, the index, its corresponding weight stored in Weights 508 is conveyed to Weight Accumulator 510. Here, the received weight is summed with the previously received weights.

**[0058]** For example, 100 signals may be chosen to be sampled. A new sampling period may have begun, and the current value received on the line Dataout 122 is the $37^{th}$ received value of this new sampling period. If this $37^{th}$ sampled value has a logic low value, then no more work needs to be done until the $38^{th}$ sample value arrives. If the $37^{th}$ sampled value has a logic high value, then the value 37 is used to index Weight Table 504. The corresponding weight for the $37^{th}$ sampled signal is conveyed to Weight Accumulator 510. This corresponding weight is summed with the accumulated weights of the 36 previous received values that had a logic high value. The method used for choosing values for the weights is described shortly below.

**[0059]** FIG. 6 illustrates one embodiment of a method 600 for accurate real-time power estimation of a core. Method 600 may be modified by those skilled in the art in order

to derive alternative embodiments. Also, the steps in this embodiment are shown in sequential order. However, some steps may occur in a different order than shown, some steps may be performed concurrently, some steps may be combined with other steps, and some steps may be absent in another embodiment. In the embodiment shown, a semiconductor chip is floorplanned and individual blocks and cells are placed according to the floorplan in block 602. A core may be the entire chip or a portion of the chip. As stated earlier, the core may be any IC. In one embodiment, the core may be a processor core. In another embodiment, the core may be an ASIC or other semiconductor chip.

**[0060]** The core may be divided into units for sampling purposes in block 604. Each unit may provide a serial chain of values of predetermined sampled signals. In block 606, a power model is executed on the core over a broad set of applications in order to characterize power consumption over the die during sample intervals. In block 608, a number of samples to perform within a sample interval may be determined from a look-up table derived from statistical analysis using a Poisson distribution or other, and characterizing the node capacitance switching behavior of the core while executing applications. Then the number of signals to sample may be determined from the operational frequency, the sample interval, and the determined number of samples to take.

**[0061]** In block 610, a weight value is assigned to each sampled signal. A more detailed description of this process is given below. A sum of the weights of the asserted sampled signals is used to determine an accurate real-time power estimation of the core in a particular clock cycle. The sum is an estimation of the node switching capacitance, Cac, in the particular clock cycle and this sum may be translated to a power number by indexing a look-up table. This look-up table may have been generated by a characterization process when the power model was run on the core.

**[0062]** In block 612, the real-time power estimate generated from the sum of the weights of sampled signals is compared to the power estimate generated from the power model for each application in a predetermined application suite. If the difference between these two values is less than or equal to a predetermined minimum accuracy (conditional block 614), then design of the power estimate circuitry is complete and the chip may tape-out with the Monitor Control in the core in block 616. If the minimum accuracy is not met (conditional block 614), then control flow is returned

to block 608. A new number of samples and a new number of signals to sample are chosen.

**[0063]** FIG. 7 illustrates a method 700 for determining weights to assign to sampled signals for accurate real-time power estimation of a core. Method 700 may be modified by those skilled in the art in order to derive alternative embodiments. Also, the steps in this embodiment are shown in sequential order. However, some steps may occur in a different order than shown, some steps may be performed concurrently, some steps may be combined with other steps, and some steps may be absent in another embodiment. In the embodiment shown, a power model is executed on the core over a broad set of applications in order to characterize power consumption over the die during sample intervals in block 702.

**[0064]** In block 704, an initial equal weight value is assigned to each sampled signal. In order to determine this initial value, a particular application is executed on the core and a sum of the asserted sampled signals in that clock cycle is recorded. The number of asserted sampled signals is recorded for each sampled clock cycle in the sample interval. There may be hundreds of sampled clock cycles in a sample interval. The actual power estimation for this sample interval found from the execution of the power model in block 702 is used to determine an actual Cac value. This actual Cac is divided by the total sum of asserted sampled signals over the sample interval. This derived value may be the equal initial weight value assigned to each sampled signal.

**[0065]** The above steps may be performed for each application in an application suite in block 706. Each application will have its own derived initial equal weight value for the sampled signals. In block 708, the average of all of these values may be the initial equal weight value used for the sampled signals.

**[0066]** For each application, the calculated Cac from the initial weights will have an error term. The Total Error is calculated for all of the applications in block 710. In one embodiment, this Total Error may be a sum of the errors for each application. In an alternative embodiment, this Total Error may be a sum of the squares of the errors for each application. In alternative embodiments, other equations may be used to calculate the Total Error.

**[0067]** In block 712, for a particular sampled signal, its corresponding initial equal weight value may be increased or decreased in a direction that diminishes the Total Error. If

the Total Error has not stopped diminishing (conditional block 714), then control flow return to block 712. If the Total Error stops diminishing (conditional block 714), then in block 716, the current weight value for that particular sampled signal is saved. This value may be saved in a table such as Weight Table 504.

**[0068]** If all sampled signals have not been investigated for a possible value change (conditional block 718), then control flow returns to block 712. If all sampled signals have been investigated (conditional block 718), then the one or more tables storing the weight values have the final weight values in block 720.

**[0069]** Although the embodiments above have been described in considerable detail, numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.

Industrial Applicability:

**[0070]** This invention may generally be applicable to electronic circuits.

## WHAT IS CLAIMED IS

1. A system comprising:

a functional block (110) configured to convey data corresponding to each of a predetermined number of sampled signals;

a power monitor (130) configured to:

receive said data and store a corresponding weight value for each of the predetermined number of sampled signals, wherein the weight values are determined using statistical analysis that assumes the functional block has node capacitance switching behavior that is approximated by a stationary random process;

sum the weight values of the sampled signals that have a corresponding logic high value; and

determine a power estimate number based on said sum;

a power manager (140) configured to convey voltage and frequency adjustments to the functional block to alter power consumption behavior within the functional block, responsive to the power estimate number.

2. The system as recited in claim 1, wherein the power estimate number is a repeatable value independent of environment variations.

3. The system as recited in claim 2, wherein said stationary random process has a Poisson distribution.

4. The system as recited in claim 2, wherein the storing of the weight values occurs once every predetermined number of clock cycles.

5. The system as recited in claim 4, wherein a predetermined number of samples within a sample interval is determined from the statistical analysis.

6. The system as recited in claim 5, wherein the predetermined number of sampled signals is determined from the operational frequency, the sample interval, and the predetermined number of samples.

7. The system as recited in claim 5, wherein the predetermined number of sampled signals comprises one or more clock gater enable signals.

8. The system as recited in claim 7, wherein the functional block comprises one or more units, wherein for a same predetermined number of samples and a same predetermined number of sampled signals, each of said one or more units comprise a lower confidence level and a higher accuracy than the functional block.

9. A method comprising:

conveying data from a functional block, the data corresponding to each of a predetermined number of sampled signals;

receiving said data in a power monitor and storing a corresponding weight value for each of the predetermined number of sampled signals, wherein the weight values are determined using statistical analysis that assumes the functional block has node capacitance switching behavior that is approximated by a stationary random process;

summing the weight values of the sampled signals that have a corresponding logic high value;

determining a power estimate number based on said sum; and

conveying voltage and frequency adjustments to the functional block to alter power consumption behavior within the functional block, responsive to the power estimate number.

10. A power monitor (130) comprising:

an interface configured to communicate with a functional block (110) and a power manager (140);

a weight table (504) configured to store a weight value for each of the predetermined number of sampled signals, wherein the weight values are determined using statistical analysis that assumes the functional block has node capacitance switching behavior that is approximated by a stationary random process;

a weight accumulator (510) configured to sum the weight values of the sampled signals that have a logic high value; and

a chain control (502) configured to convey a power estimate number to the power manager, wherein the power estimate number is translated from the sum of the weight values.

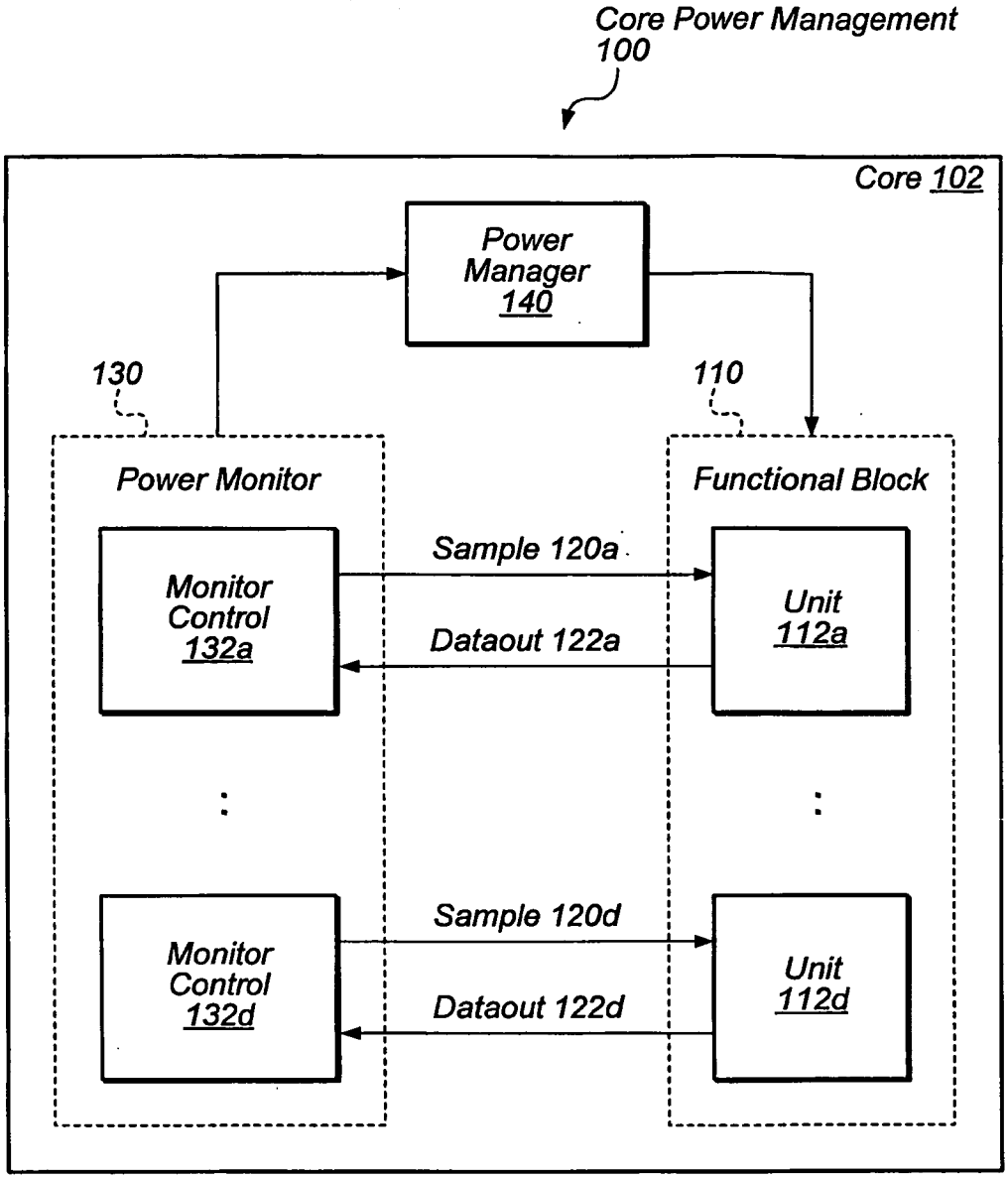
Core Power Management
100
Core 102
Power
Manager
140
130
110
Power Monitor
Functional Block
Monitor
Control
132a
Sample 120a
Dataout 122a
Unit
112a
Monitor
Control
132d
Sample 120d
Dataout 122d
Unit
112d

FIG. 1

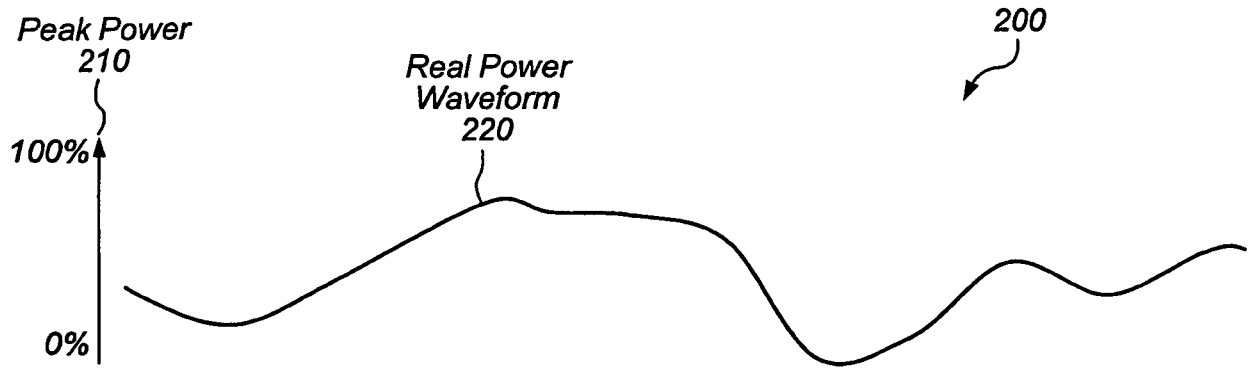
Peak Power
210
100%
0%
Real Power
Waveform
220
200

FIG. 2A

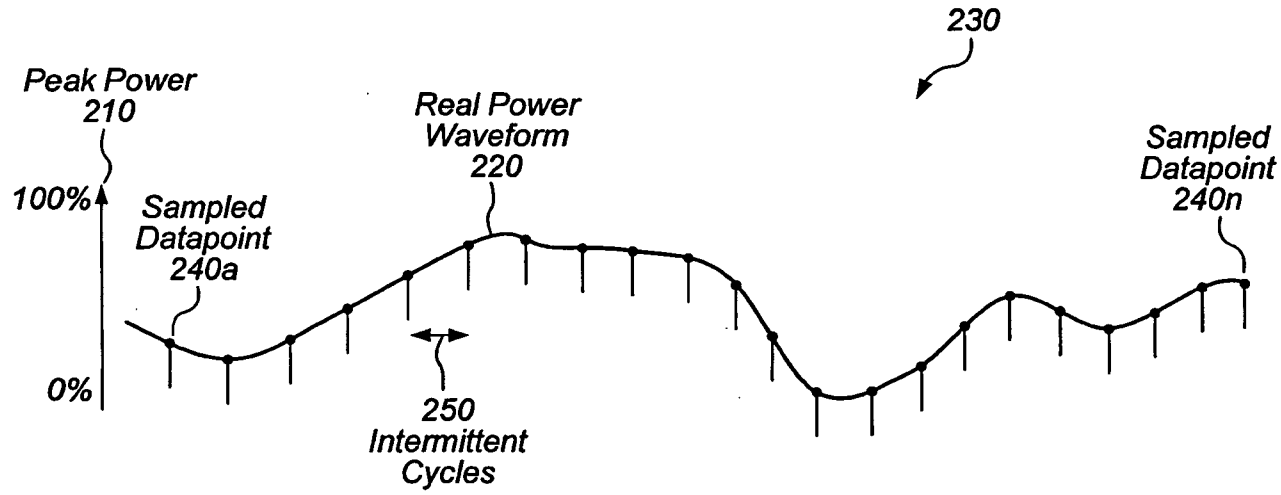
230
Peak Power
210
100%
0%
Sampled
Datapoint
240a
Real Power
Waveform
220
Sampled
Datapoint
240n
250
Intermittent
Cycles

FIG. 2B

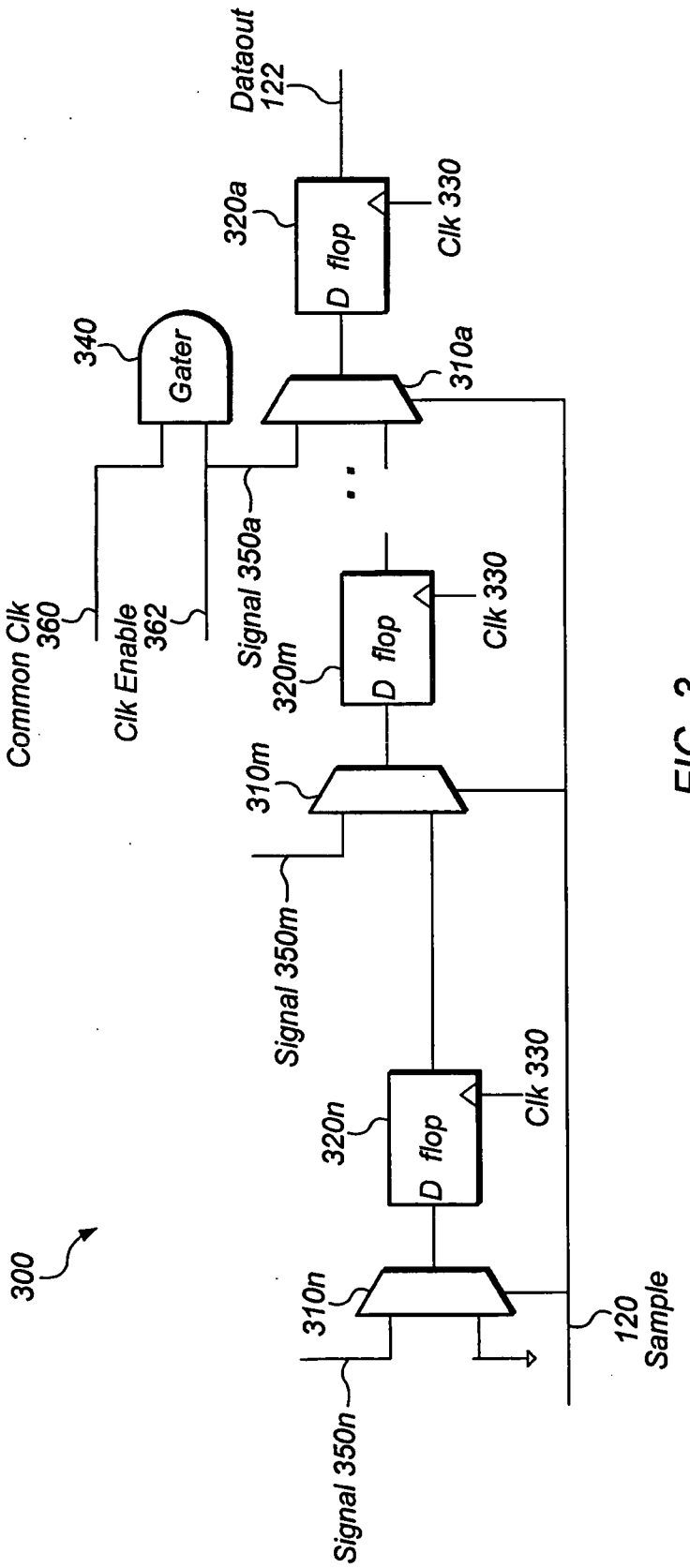
Dataout
122
320a
D flop
Clk 330
340
Gater
310a
Common Clk
360
Clk Enable
362
Signal 350a
320m
D flop
Clk 330
310m
Signal 350m
320n
D flop
Clk 330
300
310n
120
Sample
Signal 350n

FIG. 3

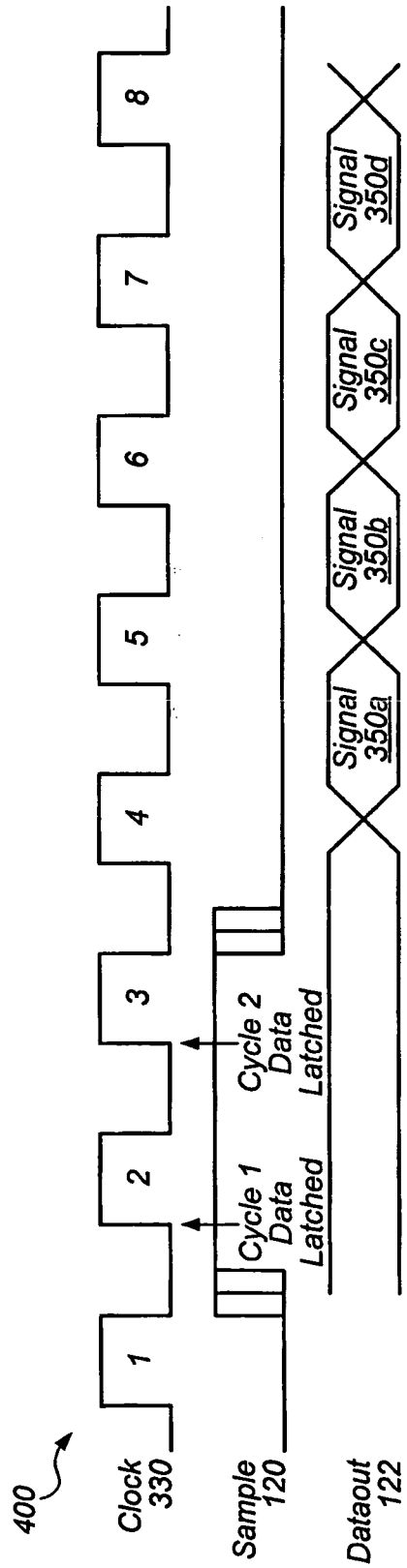
400
Clock 330
Sample 120
Dataout 122
1
2
3
4
5
6
7
8
Cycle 1 Data Latched
Cycle 2 Data Latched
Signal 350a
Signal 350b
Signal 350c
Signal 350d

FIG. 4

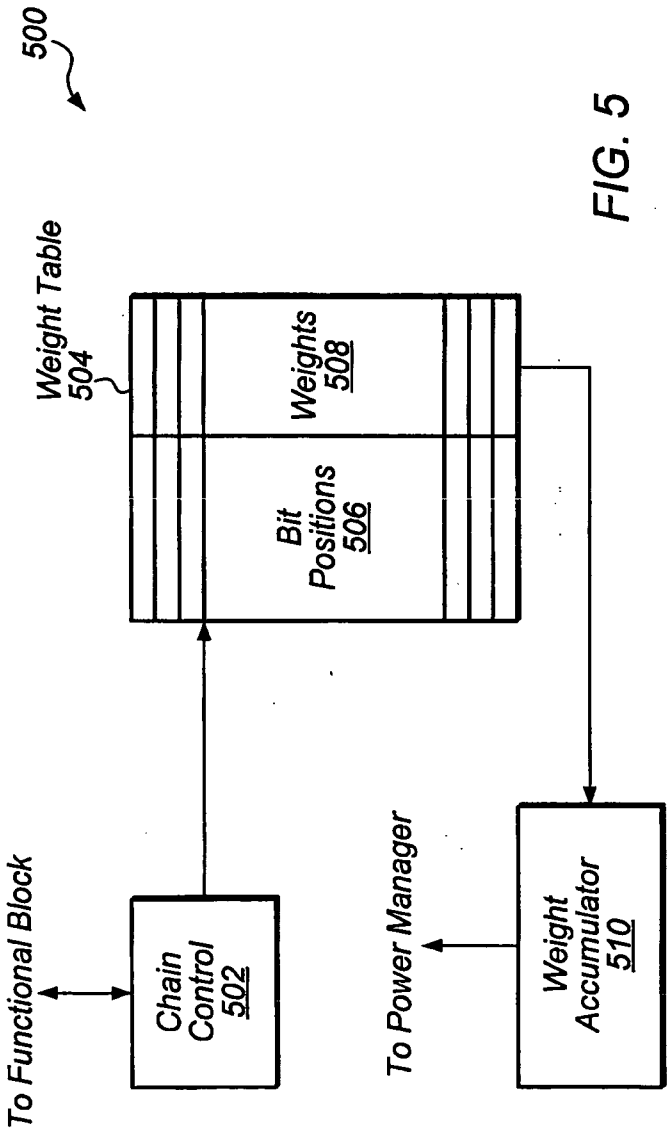
500
Weight Table 504
Bit Positions 506
Weights 508
To Functional Block
Chain Control 502
To Power Manager
Weight Accumulator 510

FIG. 5

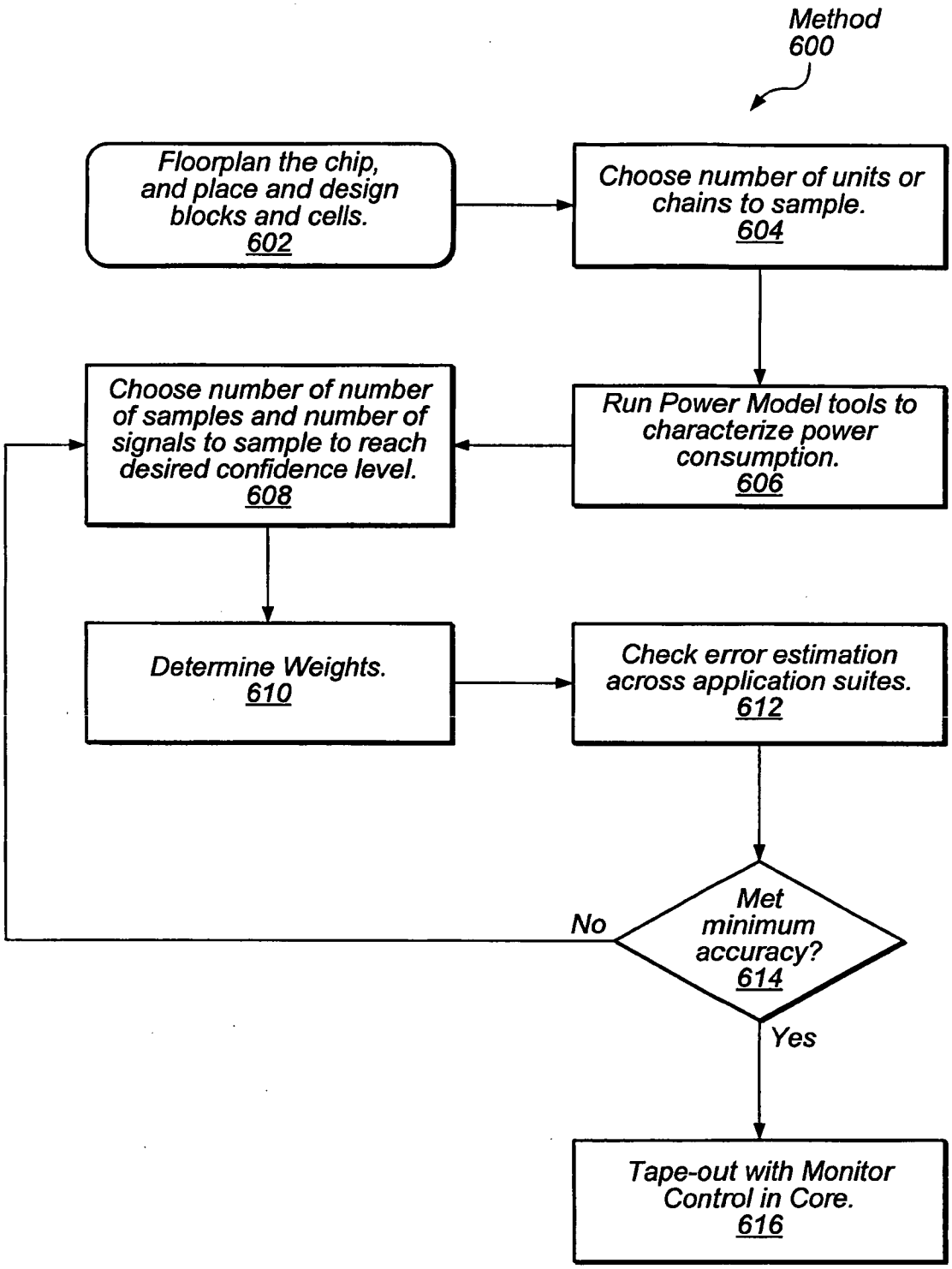
Method
600
Floorplan the chip, and place and design blocks and cells.
602
Choose number of units or chains to sample.
604
Run Power Model tools to characterize power consumption.
606
Choose number of number of samples and number of signals to sample to reach desired confidence level.
608
Determine Weights.
610
Check error estimation across application suites.
612
Met minimum accuracy?
614
No
Yes
Tape-out with Monitor Control in Core.
616

FIG. 6

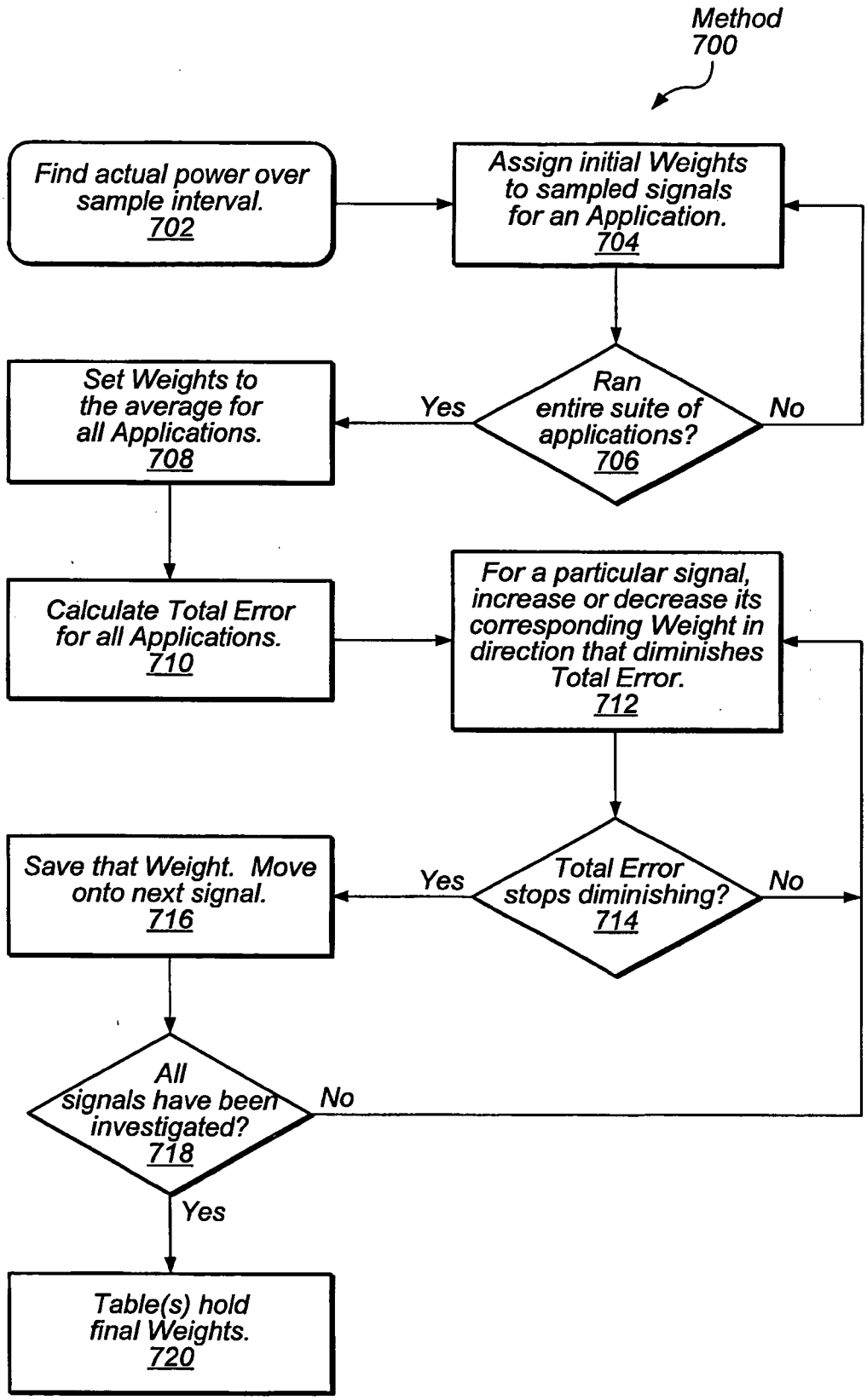
Method 700
Find actual power over sample interval. 702
Assign initial Weights to sampled signals for an Application. 704
Ran entire suite of applications? 706
Yes
No
Set Weights to the average for all Applications. 708
Calculate Total Error for all Applications. 710
For a particular signal, increase or decrease its corresponding Weight in direction that diminishes Total Error. 712
Total Error stops diminishing? 714
Yes
No
Save that Weight. Move onto next signal. 716
All signals have been investigated? 718
No
Yes
Table(s) hold final Weights. 720

FIG. 7