Design techniques for variability mitigation

Shady Agwa*

Center of Nanoelectronics and Devices (CND),
American University in Cairo,
Zewail City of Science and Technology,
Sheikh Zayed District, 6th of October City, 12588, Giza, Egypt
E-mail: shady_agwa@aucegypt.edu
*Corresponding author

Eslam Yahya

Center of Nanoelectronics and Devices (CND),
American University in Cairo,
Zewail City of Science and Technology,
Sheikh Zayed District, 6th of October City, 12588, Giza, Egypt
and
Benha Faculty of Engineering,
Benha University,
Fareed Nada St., Benha, Qaliobia, Egypt
E-mail: eslam.yahya@aucegypt.edu

Yehea Ismail

Center of Nanoelectronics and Devices (CND),
American University in Cairo,
Zewail City of Science and Technology,
Sheikh Zayed District, 6th of October City, 12588, Giza, Egypt
E-mail: y.ismail@aucegypt.edu

Abstract: As the fabrication technology migrated towards the nanometre scale, 22 nm and beyond, yield enhancement has become one of the challenges facing the integrated circuits design community. Delay and power consumption of the manufactured chips deviate from their predesigned values due to process, voltage and temperature (PVT) variations. This deviation can lead to a considerable loss in yield and reliability. In this paper, we classify and survey the approaches developed to mitigate the PVT variations on the circuit and architectural levels.

Keywords: yield enhancement; variability mitigation techniques; correlated clock skewing; thermal induced time variability; domino logic; adaptive voltage and frequency scaling; programmable clock.

Reference to this paper should be made as follows: Agwa, S., Yahya, E. and Ismail, Y. (2013) 'Design techniques for variability mitigation', *Int. J. Circuits and Architecture Design*, Vol. 1, No. 1, pp.20–40.

Biographical notes: Shady Agwa received his BSc and MSc degrees in Computer Engineering from Electrical Engineering Department, Assiut University, Egypt, in 2006 and 2011, respectively. He is currently working toward his PhD degree at the School of Sciences and Engineering, Electronics Engineering Department, The American University in Cairo (AUC), Cairo, Egypt. In September 2012, he joined the Center of Nanoelectronics and Devices (CND), American University in Cairo, Zewail City of Science and Technology, Egypt, where he is currently involved as a Research Assistant. His research interests include self-adjusting architectures, variability mitigation, retiming techniques, thermal and power management for VLSI systems, and reconfigurable architectures.

Eslam Yahya received his Engineering BSc and MSc in Microelectronics in 2000 and 2005 from Benha University, Egypt. He joined IRISA, INRIA, France as a Research Fellow in 2002. In 2005, he joined TIMA Laboratory, France for his PhD; where he received his PhD in Micro-Nano Electronics in 2009. He joined TIMA as a Research Associate. In 2010, he joined Nile University and Benha University as an Assistant Professor. In 2011, he joined the Center of Nanoelectronics and Devices (CND), American University in Cairo, Zewail City of Science and Technology, Egypt, as an Assistant Research Professor and he established the asynchronous research team.

Yehea Ismail is the Director of the Center of Nanoelectronics and Devices at Zewail City and the American University in Cairo. He is the Editor-in-Chief of the IEEE Transaction on Very Large Scale Integration (TVLSI) Systems. He is the Distinguished Lecturer of IEEE CASS. He is an IEEE Fellow. He is on the editorial board of many international journals. He has chaired many conferences. He has several awards such as the USA National Science Foundation Career Award, the IEEE CAS Outstanding Author Award, and Best Teacher Award at Northwestern University. He has published more than 170 papers in top refereed journals and conferences.

1 Introduction and fundamentals

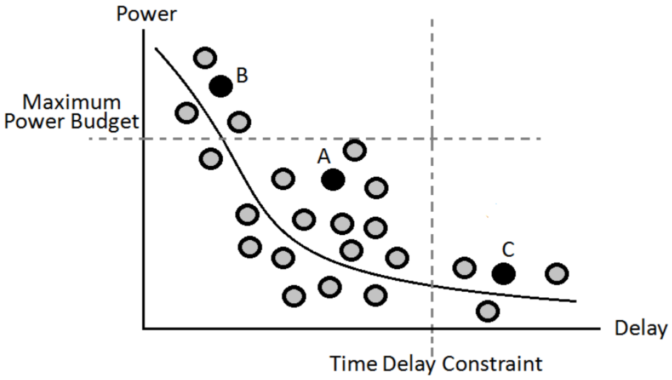
Yield can be simply defined as the amount of working chips that are satisfying the predesigned performance conditions or specifications compared to the total number of developed chips. Yield can be expressed using the following formula:

$$Y \alpha e^{-(DAKr)} \tag{1}$$

where *Y* is the yield, *D* is the defect density representing the number of defects per unit area, *A* is the area of the chip, and *Kr* is the kill ratio representing the fraction of total area that can be affected by defects (Pan, 2009). This formula shows that yield is affected exponentially by the area and this strongly shows the importance of yield enhancement techniques in the era of many-core chips. Loss in yield can be caused by three mechanisms (Pan, 2009): random yield loss mechanism, systematic yield loss mechanism and parametric yield loss mechanism. Random yield loss mechanism constitutes of random particulate and contaminant induced defects. Systematic yield loss mechanism is a function of specific layout patterns causing spatial or temporal correlation faults. Parametric yield loss mechanism occurs when a functional chip fails to meet its predesigned performance specifications. This type of loss could be a result of either manufacturing or environmental factors. This paper is particularly concerned about the parametric yield loss and the different approaches to solve it.

Focusing on parametric yield loss, the manufactured chips can be classified into three areas: A, B and C as Figure 1 shows. Area A encompasses the chips which meet the predesigned power budget and time delay constraints. Chips in area B consume more than the maximum power budget. These chips are considered malfunctioning only because they exceed the maximum power consumption limit. Finally, chips in area C are working on the functional level; however they do not meet the required delay constraints. There are different design approaches introduced in the literature to move chips from both area B and C to area A for increasing the yield. The chip excluded because of its high delay latency can be repaired at the cost of power consumption by scaling up the supply voltage. On the other hand, chips with high power consumption can be repaired at the cost of speed by scaling down the operational frequency.

Figure 1 Classification of chips based on parametric yield loss



Notes: A – working chips, B – loss due to power constraints, C – loss due to delay constraints; the idea is to move chips from areas 'B and C' to area 'A'.

The recent advancement in fabrication technology significantly increases the device-parameter variations; that dramatically decreases the reliability of the devices. Reliability is one of the urgent needs of modern electronic products and chips especially for medical and military applications. Unreliability of the modern devices is caused by many factors; for example: high temperatures, high current densities, thinner gate oxides. These factors can lead to unpredicted data latency which leads to unreliable results. Reliability defects are not detectable during manufacturing process as they may appear under specific voltage, temperature, frequency, and work load conditions.

2 Variability mitigation techniques

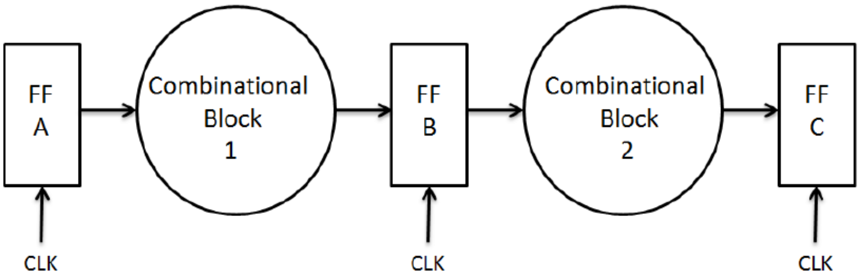
Process, voltage and temperature (PVT) variations are serious problems in synchronous circuit design. Different doping concentrations, different gate-lengths (which affects the threshold voltage),unexpected voltage drops in the power supply network and temperature fluctuations can cause temporary or permanent defects that lead to decreasing yield and reliability. PVT variations can cause unpredicted latency which lead to a miss-synchronisation. Because of these unpredicted latencies, data is delayed causing the registers to read wrong data. A traditional costly approach, used to reduce the error rate, is to increase the supply voltage margins to ensure the correctness of all operations under all possible variations. This conservative approach is not valid in the modern trends which try to reduce the power density. High power density and heat dissipation threaten the Moore's law and the trend of increasing operating frequency and number of transistors per processor so that increasing the voltage margins is not a valid solution.

Efficient techniques have been developed to solve PVT variations' problems including retiming and/or power and thermal management. In this paper these approaches are categorised into two main categories: fine grain mitigation techniques (which are working on the register/pipeline level) and coarse grain mitigation techniques (which are working on the module/partition level).

3 Fine grain (circuit level) mitigation techniques

Fine grain solutions are concerned about preventing the timing errors in sequential circuits by using retiming techniques with minimal addition of fine grained hardware.

Figure 2 Basic architecture of sequential circuit



In sequential circuits, as shown in Figure 2, combinational logic clouds are isolated by flip-flops and computed data should be ready before the active edge of the flip-flops' clock. This synchronisation should be ruled by two constraints.

$$X_j + T_{Hold} < X_i + d(i, j) \quad (2)$$

Formula (2) presents the short path constraint as data is going from flip-flop (*i*) to (*j*). $d(i, j)$ is the combinational logic minimum delay and T_{Hold} is the hold time. A positive clock edge arrives at flip-flop (*i*) at X_i then data races through the short path and it may change the data at flip-flop (*j*) before the positive clock edge arrives there, then wrong data will be clocked at flip-flop (*j*) so that the short path constraint eliminates this problem (Sathyamurthy et al., 1995).

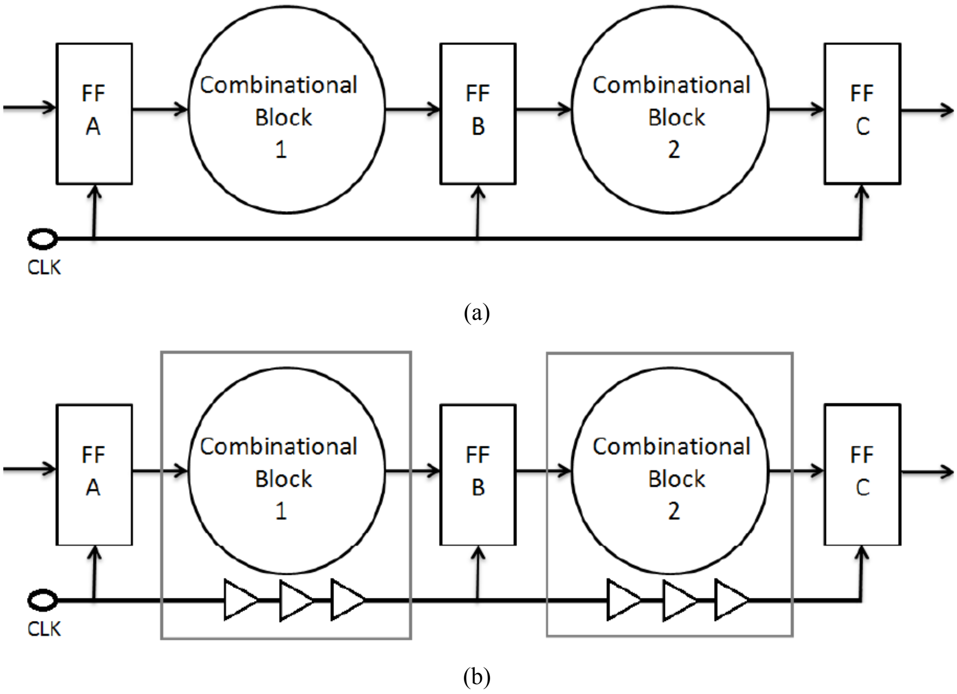
$$X_j + T_{CP} > X_i + D(i, j) + T_{SetUp} \quad (3)$$

Formula (3) presents the long path constraint as data is going from flip-flop (*i*) to (*j*). $D(i, j)$ is the combinational logic maximum delay, T_{CP} is the clock period and T_{SetUp} is the setup time. If a positive clock edge arrives at flip-flop (*j*) at X_j while data races through the long path and data has not been ready yet at flip-flop (*j*) then wrong data will be clocked at flip-flop (*j*) so that the long path constraint eliminates this problem (Sathyamurthy et al., 1995).

3.1 Correlated clock skewing

Process variations or voltage and temperature fluctuations can cause an unexpected delay for the data or the clock. Clock skewing and data latency can affect the reliability of the chip by increasing the error rate. Also because of differences in interconnection delays of the clock distribution network, there will be a skew among the clock signals reaching different flip-flops. Designing clock distribution network with zero clock skew should be beneficial in case of ignoring PVT variations but nowadays it is useful to use clock skew intentionally not only to improve performance, in addition to gate sizing (Sathyamurthy et al., 1995), but also to compensate the effects of PVT variations. At design time, safety margins should be considered to ensure that the chip will operate correctly in the presence of unpredicted PVT dependent skew variations. Because of variations, integrated circuits design should be based on the worst case timing principle as clock frequency should verify all of timing constraints under the worst conditions. A new compensation mechanism especially for environmental parameters fluctuation (voltage and temperature) was introduced in Andrade et al. (2009). Complex microprocessor architecture was modelled by a structure of pipelined stages as shown in Figure 3.

Figure 3 (a) Conventional and (b) proposed synchronising mechanism for n-stage pipeline circuit



(a)

(b)

Note: Cascaded delay chains are used to locally compensate the delay.

Source: Andrade et al. (2009)

All registers are driven by a common clock signal. Because of the heterogeneity of the clock distribution network, a skew can affect the clock signals. In the conventional mechanism, skew dependent on voltage or temperature fluctuation is not considered. The proposed compensation mechanism depends on the correlation between combinational logic stages and skewing buffers. As shown in Figure 3(b), the first register is driven by the system clock. Clock signals of other registers are generated through cascaded delay chains of buffers. Both combinational logic stages and skewing buffers are assumed to experience the same delay variations because of spatial correlation which will offer the same voltage and temperature fluctuations. This should decrease not only the error rate but also the total processing time of the circuit.

$$t_{CHAIN}^{W} = nT_{CLK} = n\max\left(t_p\right) + nK\sigma_f \tag{4}$$

$$t_{CHAIN}^{C} = n\max\left(t_p\right) + K\sqrt{n}\sigma_f \tag{5}$$

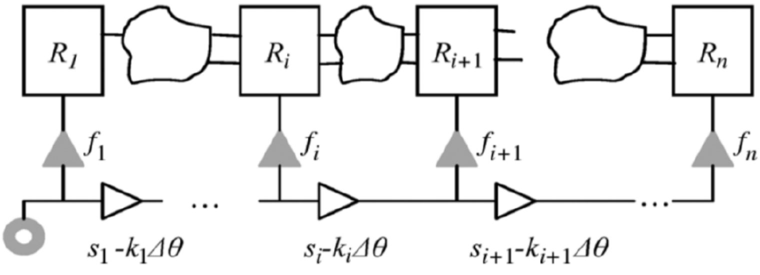
Equations (4) and (5) show the total processing time of the worst-case conditions for both conventional (t_{CHAIN}^{W}) and compensation (t_{CHAIN}^{C}) mechanisms (Andrade et al., 2009). Where max(t_p) is the worst-case delay of any combinational logic between two registers, σ_f is the standard deviation, K is a given number of standard deviations, and n is the number of stages (Andrade et al., 2009). The reduction of the propagation time for the whole pipeline chain is expressed by equation (6).

$$t_{CHAIN}^{W} - t_{CHAIN}^{C} = K\sigma_f\left(n - \sqrt{n}\right) \tag{6}$$

3.2 *Mitigation of thermal induced time variability*

Self-adjusting clock tree architecture (SACTA) (Long et al., 2010), is another approach which is more concerned about thermal-induced delay variations because thermal effects have a great impact on leakage power, device life time and circuit timing. SACTA uses clock skew to steal time from adjacent pipeline stages to keep the performance in the presence of variations with minimum hardware overhead. Temperature-dependent dynamic clock skew scheduling was discussed with detailed models in Long et al. (2010). The problem was defined as to design a clock tree that can change skew values to pipeline registers dynamically as these values should be linear functions of temperature.

Figure 4 Self-adjusting clock tree architecture



Note: Temperature adjustable buffers (white triangles) are used to mitigate the thermal induced time variability.

Source: Long et al. (2010)

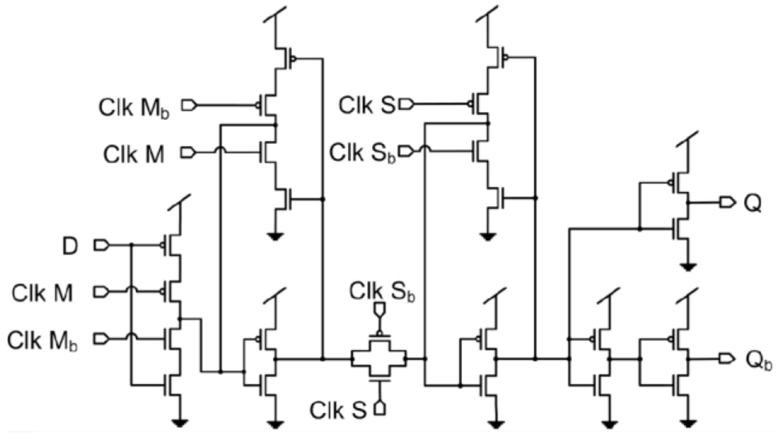
Figure 4 shows a pipeline with SACTA. The white triangles are Automatic Temperature Adjustable skew buffers and the gray triangles are fixed skew buffers which are designed at the zero temperature coefficient point (ZTC) to be temperature insensitive. These fixed buffers have a base delay F_i. The ZTC point is the point in which the effect of thermal change on the threshold voltage cancels the effect of the mobility, and the drain current of the transistor will be independent of temperature. The relationship between the delay and temperature is expressed as $s_i - K_i\Delta\theta$, where s_i is the delay of the skew buffer at the worst-case temperature; K_i is the temperature sensitivity coefficient; $\Delta\theta$ is the difference between the worst-case temperature and the operating temperature. According to Long et al. (2010), SACTA can improve the temperature tolerance of both single-V^{th} and multi-V^{th} designs depending on the assumption of spatial correlation which implies that temperature of the combinational logic stage and its surrounding registers are nearly the same. SACTA has a powerful advantage over the static clock skew scheduling techniques as static techniques will only satisfy some temperature profiles selected during design time while SACTA changes skew values dynamically as linear functions of temperature.

3.3 Soft-edge flip-flop

Soft-edge flip-flop was discussed as a potential solution for timing yield enhancement (Wieckowski et al., 2008) as it keeps synchronisation at the clock edge, in addition to a transparency window around the edge with minimal overhead in complexity, area and power. Transparency window reduces the sensitivity to clock skew and jitter and allows time borrowing among pipeline stages.

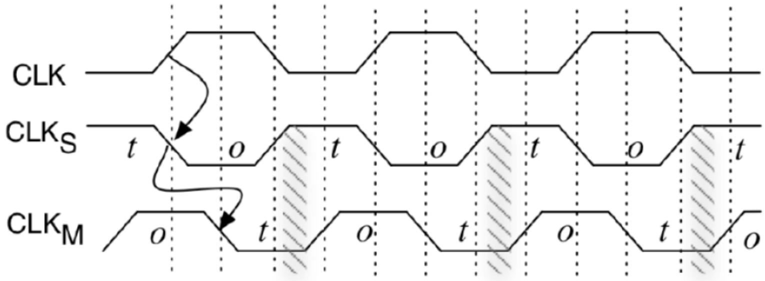
Figure 5 shows the soft-edge flip-flop design with two different clocks, one for the master latch stage and the other for the slave stage. The clock of the master stage is delayed with respect to the slave stage's clock to create the transparency window as illustrated in Figure 6. Because of delayed master clock, even delayed data could be latched and transferred to the slave latch.

Figure 5 Soft-edge flip-flop schematic design



Source: Wieckowski et al. (2008)

Figure 6 The delayed master clock (CLK_M) relative to the slave clock (CLK_S) to create a window of transparency

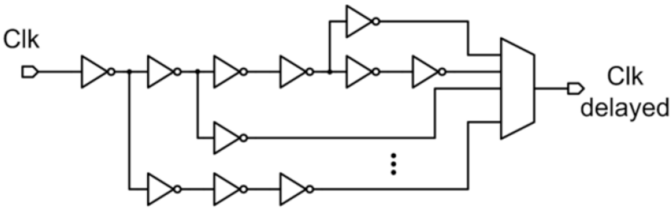


Notes: *t* = transparent an *o* = opaque.

Source: Wieckowski et al. (2008)

The size of the transparency window could be dynamically adjusted by choosing different delays for the master clock. Figure 7 presents a series of different inverter chains and a scan controlled multiplexer used to select different values of delay for the master clock.

Figure 7 Programmable delayed clock



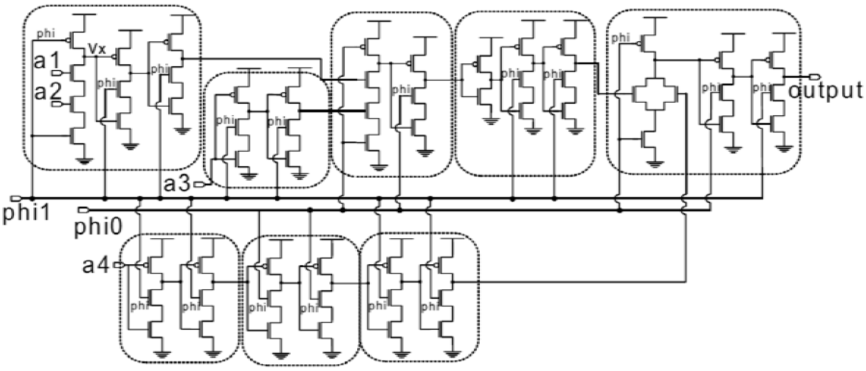
Source: Wieckowski et al. (2008)

Although increasing window size results in a performance enhancement, this enhancement will saturate if the window size reaches 10% of the cycle time. Another issue related to soft-edge flip-flop approach can be detected in case of cascaded critical stages that may produce delayed data. If there is a delayed computed data from more than one critical stage the delay will be accumulated as every stage will steal time from the next one and this can lead to a time violation in one or more stages.

3.4 Domino logic

It is known that static CMOS gates are not fast enough because of pull-up and pull-down networks. Domino logic runs 1.5–2× faster than static CMOS logic gates (Harris and Horowitz, 1997), so that domino circuits are widely used in high performance chips and skew-tolerant domino circuits can introduce a good solution for retiming issues (Harris and Horowitz, 1997). Based on domino circuits, a dual-phase pipeline circuit design, using domino latch, was introduced (Tsai et al., 2011) with a built-in performance adjusting mechanism. Although the designed domino latch circuit uses two clocks with opposite signal directions, it still has a similar operation to that of pipeline circuits.

Figure 8 Dual-phase domino latch circuit

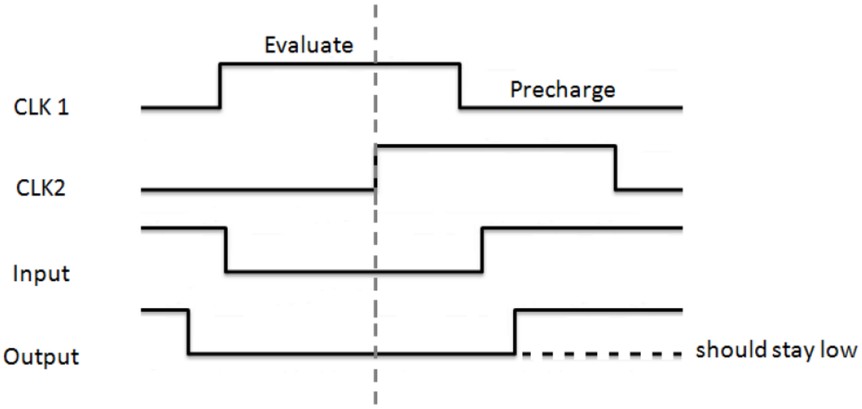


Source: Tsai et al. (2011)

As Figure 8 shows, the gates are divided into two groups and a different clock signal is assigned to each group. Additional dynamic latch gates are inserted for retiming process. As process variations can change the duty cycle pulse width of the clock signal, a duty cycle pulse generator was designed (Tsai et al., 2011) to provide an adjustment mechanism to ensure the correctness of high speed chips' operations. This adjustment mechanism should cooperate with a built-in self-testing (BIST) mechanism. Skew-tolerant domino latches with inverted clock signals, presented in Tsai et al. (2011), can be used instead of static CMOS standard flip-flops to withstand different variations.

Clock-logic domino was introduced (Sung and Elliott, 2007) as a solution to tolerate skew with small area overhead and lower power consumption. Sequencing overhead including clock skew, latch overhead and pipeline imbalances is removed by using overlapping clock phases for the different domino logic stages.

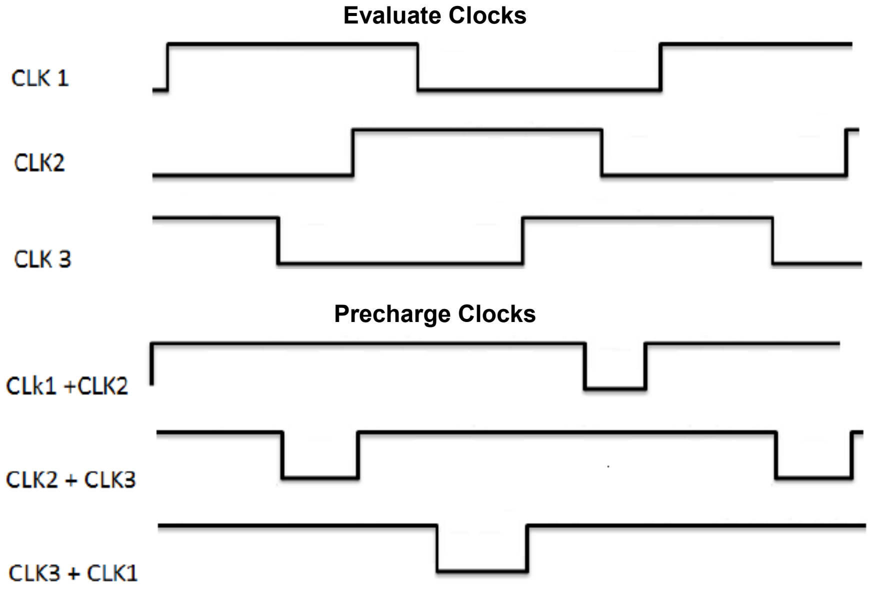
Figure 9 Domino logic failure due to clock overlapping



Source: Sung and Elliott (2007)

The overlapping clock signals, shown in Figure 9, are useful for retiming. They permit stealing time from adjacent stages in case of unpredicted delays. Using overlapping signals may create a new problem, illustrated by Figure 9, as an input to a dynamic gate may change, because of the precharge of the previous stage, before the end of the current evaluate cycle and the output will not be able to preserve the correct result. An approach was introduced in Sung and Elliott (2007) to solve this problem by using two different clocks one for the evaluation and the other for precharge. This eliminates the problem by delaying the precharge of the stage until the end of the evaluation of the next stage. This was called or-precharge/domino-evaluate approach, which is illustrated by Figure 10.

Figure 10 Or-precharge/domino-evaluate clocks



Source: Sung and Elliott (2007)

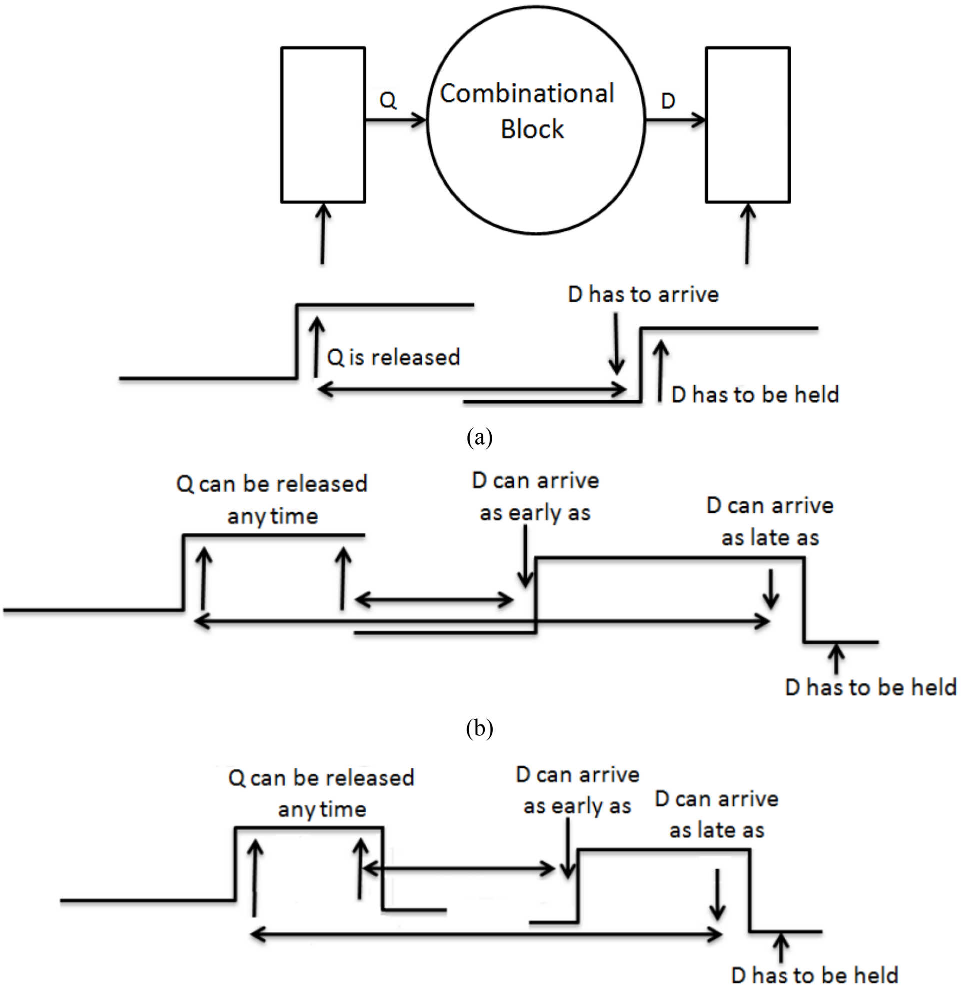
As shown in Figure 10, precharge clocks are created by a logical or function of evaluate clocks of different cascaded domino logic phases. Time borrowing will not be affected by or-precharge/domino-evaluate approach as evaluation time will not be affected and local clock generation is recommended to be at the dynamic gates to simplify the clock distribution network.

Another type of retiming using pulsed-latch circuits with regulating pulse width (Paik et al., 2011) can give enough flexibility to manage timing issues. Paik et al. (2011) introduced pulsed-latch as an ideal sequencing element for high performance integrated circuits due to its reduced sequencing overhead. Time borrowing could be achieved by using more than one pulse width. Figure 11 shows the difference of timing models among edge-triggered flip-flops, level-sensitive latches and pulsed-latches.

Pulsed-latch is driven by a clock with a small pulse width and there is a time variation available for combinational block but it is still less than that of level-sensitive latches. This flexibility can be used to borrow time from adjacent stages by making different clock pulse widths generated by a physically closed pulse generator to preserve the pulse shape. Figure 12 illustrates an easy way for retiming and time borrowing between two pipeline stages.

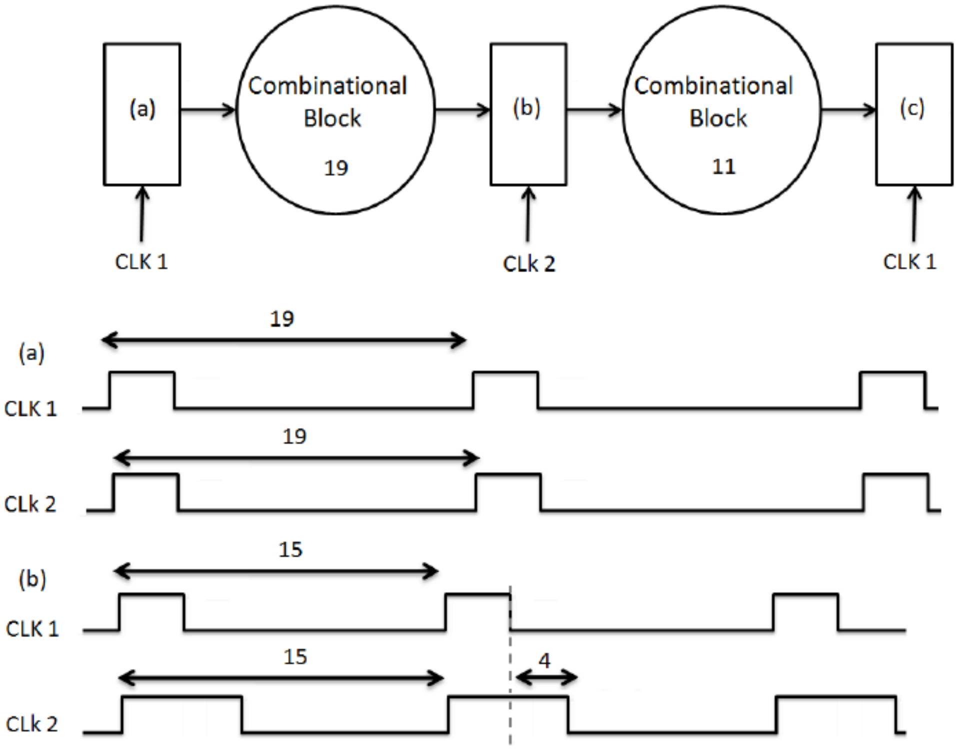
Three pulsed-latches a, b, and c, driven by the same clock, are shown in Figure 12. The maximum delay of the combinational logic stage between a and b is assumed to be 19 time units and that between b and c to be 11. In case of ignoring sequencing overhead, the clock period has to be 19 time units to preserve timing constraints. However, if b is driven by a new clock with wider pulse by 4 time units, then the clock period can be reduced to 15 time units because the combinational logic stage between a and b will be able to borrow 4 time units from the adjacent combinational logic stage between b and c.

Figure 11 Timing models of (a) edge-triggered flip-flop, (b) level-sensitive latch and (c) pulsed-latch



Source: Paik et al. (2011)

Figure 12 Time borrowing using different pulse widths



Source: Paik et al. (2011)

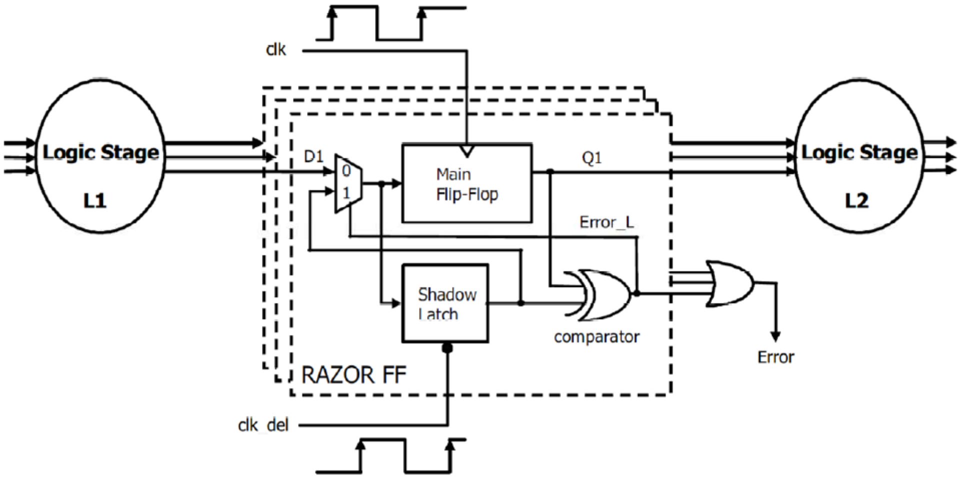
All these retiming approaches are very useful to overcome timing issues induced by PVT variations. However, these techniques do not offer any indication about the error rate as it does not manage the main problem but its symptoms. As they do not compute the error rate, they will not be able to manage power consumption and heat dissipation which can be monitored basically by error rate feedback.

3.5 Razor architecture

Razor architecture depends on detecting and correcting the error instead of preventing or hiding it. The error rate is used for power or thermal management through dynamic frequency and voltage scaling (Ernst et al., 2003). Razor (Ernst et al., 2003; Lee et al., 2004) is one of the most common approaches used for power savings by scaling down the supply voltage as low as possible while ensuring correct operation. In contrast to conservative designs, which need conservative margins for the supply voltage, Razor does not need margins as it depends on monitoring the error rate dynamically to adjust the required supply voltage.

Figure 13 shows a pipeline stage of Razor approach in which a shadow latch with delayed clock is used to capture a delayed version of data, comparing it with the main flip-flop to ensure the correctness of propagated data. The result of the comparison will be used to compute the error rate, and then the error rate will be used as a feedback to adjust the supply voltage through dynamic voltage scaling. In case of timing error, the correct data will be restored from the shadow latch and all instructions behind the faulty one will be invalidated and the pipeline is restarted after the faulty instruction (Lee et al., 2004).

Figure 13 Pipeline stage with razor latch

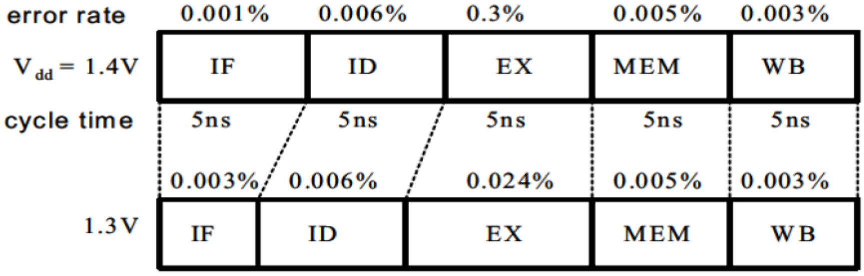


Note: The error is detected by comparing data in the main flip-flop with the delayed version of the data in the shadow latch.

Source: Ernst et al. (2003)

Dynamic retiming is augmented to the Razor approach (Lee et al., 2004) to permit pipeline stages with higher error rates to borrow time from low error rates stages as shown in Figure 14 by using negative and positive clock skew values generated by a programmable local clock controller.

Figure 14 Dynamic retiming of pipeline stages



Source: Lee et al. (2004)

Although razor introduced a good solution for both retiming and power saving through error detection and correction, it wastes two cycles to detect the error and restore the correct data, and then it flushes all potential incorrect instructions out of the pipeline. It would be better if razor could take the decision earlier not to permit the propagation of wrong data through pipeline stages.

4 Coarse grain (architecture level) mitigation techniques

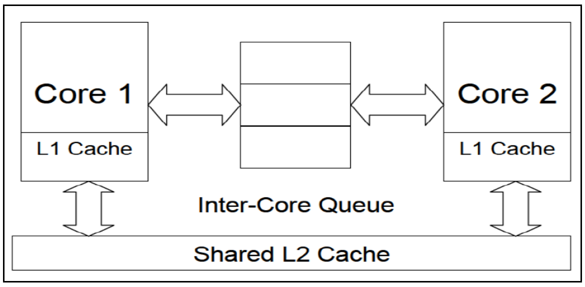
Coarse grain techniques can be introduced by another term as system level solutions. These techniques are widely used especially in many-core chips. The basic idea depends on the built in redundancy of the many-core chips. This redundant nature helps designer to make use of idle or not fully busy units to make error detections and corrections with minimal area and power overhead especially in homogeneous many-core chips. Manufacturing spare cores is a potential solution in case of large numbers of cores on the same chip. If one or more cores have been defected, spare cores can be used to compensate this loss and to maintain an acceptable yield's ratio.

4.1 Inter-core queue

A multi-core chip, using inter-core queue, was introduced (Pan, 2009) as a solution to improve manufacturing yield at the cost of performance degradation.

As shown in Figure 15, the basic idea is task swapping between the nearest cores through a storage unit called inter-core queue. If a faulty core has been detected a migration flag will be set to identify the faulty task or instruction then the faulty instruction will be transferred to the helper core through the inter-core queue. The result will be sent back to the faulty core and every core will continue its normal execution flow. If the transferred instruction stayed in the inter-core queue for a specific period of time, called idle interval, an emergency bit will be set to raise its priority of execution. According to Pan (2009), if the chip has a defective core it will continue its operation with marginally decrease in performance.

Figure 15 Multi-core chip with inter-core queue



Note: The queue is used for task swapping between the adjacent cores in case of a faulty instruction.

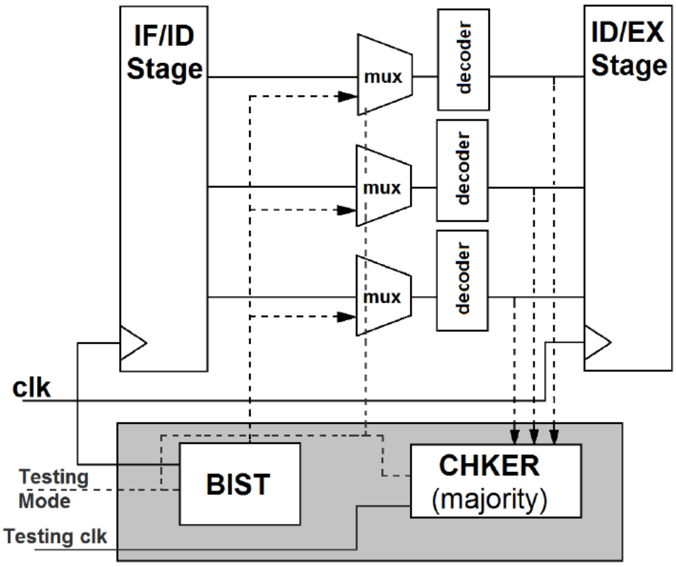
Source: Pan (2009)

This approach suffers from an overhead of complexity and hardware used for inter-core queue and its buses, hardware counters to calculate idle intervals, data flow controllers and also fault detection/isolation circuits.

4.2 Defect protection technique

A defect protection technique was introduced (Shyam et al., 2006) as a low cost defect protection system by checking the hardware functionality each period of computational instructions called computational epoch. This technique relies on the natural redundancy of instruction level parallelism processors. The main idea is that system's health will be monitored continuously. When the first defect is detected a low performance operational level will be the solution.

Figure 16 Online BIST circuit is used to check the functionality of decoders



Source: Shyam et al. (2006)

As shown in Figure 16, an online BIST circuit is used to check the functionality of decoders. The functionality of register file and arithmetic and logical unit (ALU) also can be checked by using BIST circuits. In case of decoders, the same test vector is sent to multiple decoders and if their outputs do not match, this will be an indication that one of these decoders is defected. Another test vector will be sent to the main ALU and another 9-bit mini-ALU is used to verify the functionality of the main ALU. In case of failure detection in any component, the defected component will be disabled for a period of time and the computational epoch will be flushed then the last trusted system state will be restored.

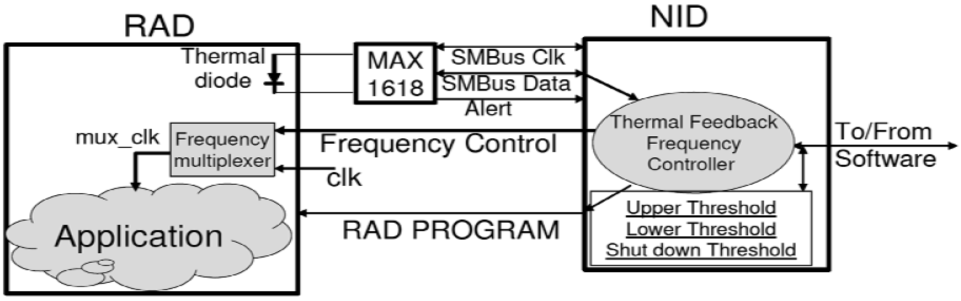
In this approach, coarse grain epochs can lead to high penalty in case of flushing them. If the checking process completed before a fault manifests, this fault will manifests in the next epoch and keeping backups for the last two epochs will be a must. Also depending on BIST circuits with test vectors to check hardware while components are idle is power and area consuming.

4.3 Thermal management

Heat dissipation is a clear indication about high power consumption. Controlling the temperature of the operating chip is very important to increase the life time of the chip and to also preserve its reliability. The target of any thermal management system is to operate at a safe operational temperature. This could be done by automatically adjusting the operating frequency of the managed chip as introduced in Jones et al. (2006, 2007). In

Jones et al. (2006), the target was determined to achieve high level of performance for a maximum junction temperature. The junction temperature of the device should be measured and the whole system should be monitored to get feedback. This feedback will be used to reconfigure temperature by dual frequency switching system. This technique makes use of the fast measurement of junction temperature changes versus the slow rate of system temperature changes.

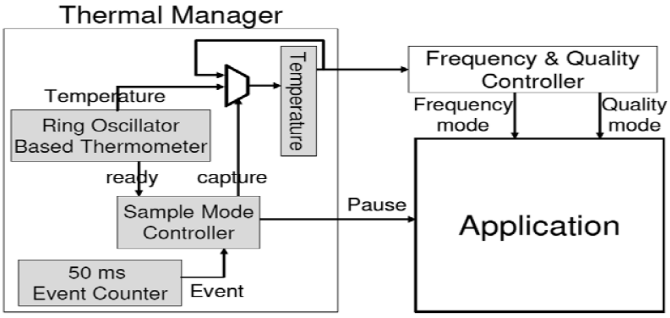
Figure 17 Temperature measurement and threshold frequency control mechanism



Source: Jones et al. (2006)

As shown in Figure 17, there are two main units: network interface device (NID) which is static; and reconfigurable application device (RAD) which is connected to an onboard Maxim temperature measurement device MAX1618. MAX1618 will measure the temperature and then will send it to NID which will compare it by upper and lower thresholds. If the temperature exceeds the upper threshold, the operating frequency will be reduced to cool the chip. On the other hand, if the temperature is lower than the lower threshold, the frequency can be increased. Only two clock frequencies are used: a low frequency for high temperatures and a high one for low temperatures. This approach controls the operational temperature for systems especially for those in temperature changing environments and also for systems with multiple modes of operations that impact the thermal budget.

Figure 18 Thermal-management system architecture

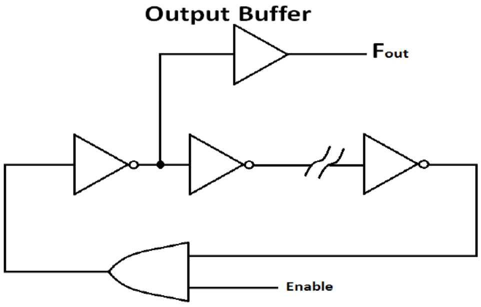


Source: Jones et al. (2007)

Another implementation for thermal management was introduced (Jones et al., 2007). The target was to scale voltage and frequency to lower power usage before the device overheats. Based on Jones et al. (2006), the approach is improved to give more flexibility through voltage and frequency scaling. As shown in Figure 18, a ring oscillator thermometer is used as a temperature sensor. A counter will trigger an event every 50 msec to pause the application and to take three different thermometer measurements. These measurements should be passed to the frequency and quality controller and then the application will continue its operation. The temperature values will be used to adjust the required frequency. The quality also can be adjusted by controlling the number of operating cores and features.

Ring oscillators, shown in Figure 19, and thermal diodes can be used to measure temperature (Buedo et al., 2000). Ring oscillators are easier to implement and they could be dynamically reconfigured if used by FPGAs with no restrictions about their positions on the chip. On the other hand, using thermal diodes will face extra wiring and hardware positioning problems.

Figure 19 Ring oscillator used to measure temperature



Source: Buedo et al. (2000)

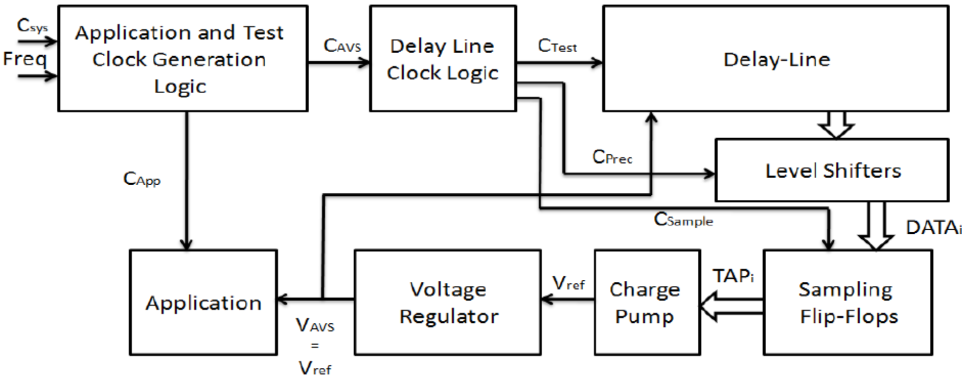
Although thermal management systems are very important to preserve the temperature within acceptable margins, they do not care about the timing issues caused by the unpredicted changes in temperature which may lead to unpredicted data latency or clock skewing. This may threaten the reliability of the chip by introducing wrong results.

4.4 Adaptive voltage scaling

An adaptive voltage scaling (AVS) technique was introduced (Dhar et al., 2002) to overcome the variability issues. This technique uses a closed-loop AVS controller to adjust the chip dynamically to the minimum required voltage for the desired speed.

As shown in Figure 20, a chain of logic buffers, operates as a ring-oscillator, is the reference circuit used to measure the delay as its frequency depends on the supply voltage. This ring-oscillator is used to model the critical path. If the supply voltage is not sufficient to make the test signal propagate, charge pumps will be used to increase the supply voltage. Increasing supply voltage will speed up the propagation of signals through the critical path. The charge pump takes a number of digital inputs to produce an analogue reference voltage based on the error rate. The reference voltage will be buffered by the voltage regulator to produce the supply voltage and the system clock is generated by voltage controlled oscillator (VCO) which is controlled by the supply voltage.

Figure 20 AVS scheme



Source: Dhar et al. (2002)

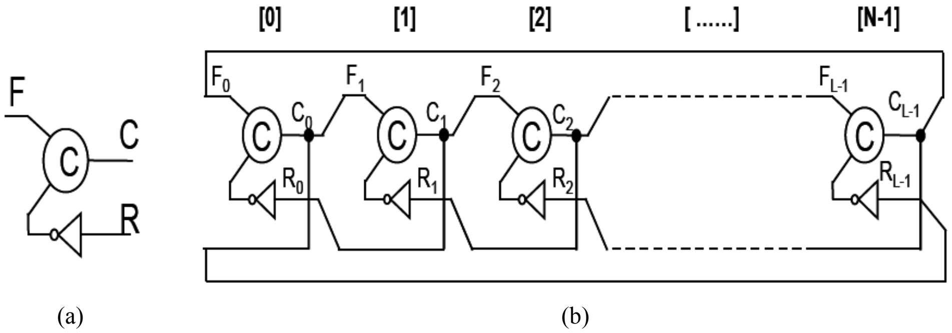
Although closed-loop adaptive voltage scaling can be an effective technique, it only cares about the critical path and its variations in general. This technique still needs voltage margins to ensure its correctness and reliability.

4.5 *Programmable clock generator*

In case of dynamic frequency scaling, clocks generated from analogue VCOs are based on dynamic voltage scaling. However, VCO have a limited operating range and it also requires a stabilisation time when changing the frequency. A programmable/stoppable oscillator based on self-timed rings approach was developed to generate high resolution timing signals (Yahya et al., 2009). This approach has a good robustness against process variability in comparison to inverter rings.

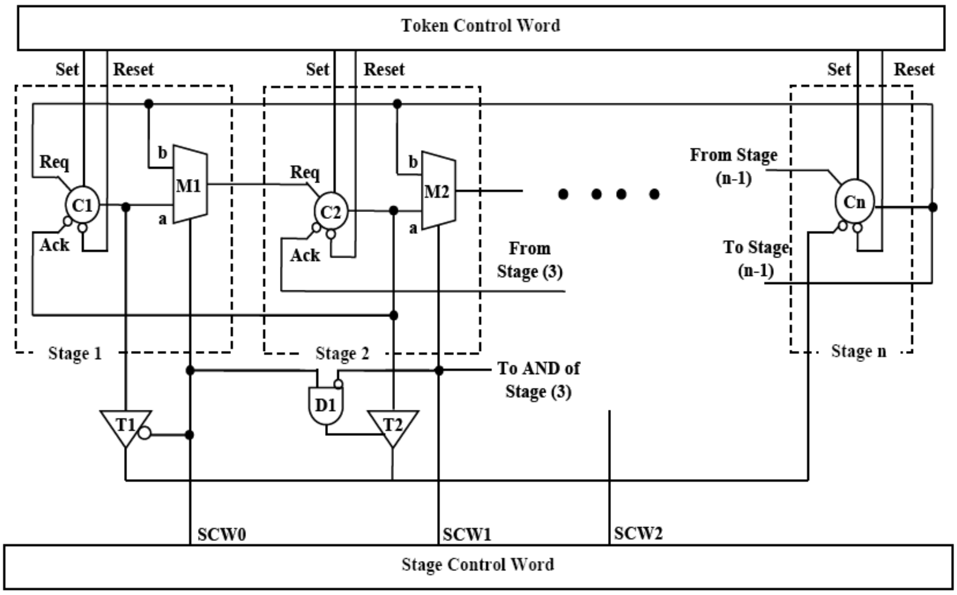
Figure 21(a) shows the structure of a ring stage. It consists of a C-element and an inverter. As shown in Figure 21(b), one of the C-element inputs is connected to the previous stage and it is marked F (forward), the second input is connected to the next stage and it is marked R (reverse) while C represents the output of the stage (Yahya et al., 2009).

Figure 21 Self-timed ring, (a) The basic stage composed of C-element and inverter (b) Ring oscillator composed of n-stages



Source: Yahya et al. (2009)

Figure 22 Programmable self-timed ring



Note: Token control word is used to control the initialisation and stage control word is used to change the number of stages.

Source: Yahya et al. (2009)

By using the architecture shown in Figure 22, the authors introduced digitally controlled programmability to self-timed rings. The 'token control word' controls the set\reset of the C-elements which enables the architecture to program the ring by the required initial conditions which in return controls the output clock frequency. By using a set of multiplexers and tri-state buffers, the architecture controls the number of stages in the self-timed ring which affects the output clock frequency.

This architecture combines the initial state control with the number of stages to implement a multi-frequency digitally controlled clock generator which is stable and robust against process variability. This design can be used to generate different clock frequencies for dynamic voltage/frequency scaling applications.

5 Conclusions and future directions

Process variations – result from perturbations in the fabrication process – and environmental variations – due to changes in the circuit operating environment – result in changes in the timing and power characteristics of the chip (Sapatnekar, 2004). Many techniques were developed to overcome the effects of both process and environmental variations, some of them are concerned about power/thermal management and other approaches are concerned about retiming to prevent time violations induced by different types of variations. Domino logic gates (Sung and Elliott, 2007) and domino latches (Tsai et al., 2011) especially pulsed-latches (Paik et al., 2011) could be used to introduce optimised retiming solutions. Using pulsed-latch instead of Razor latch (Ernst et al., 2003; Lee et al., 2004) may lead to a faster decision about the correctness of propagated data. Augmenting clock skewing techniques (Andrade et al., 2009; Long et al., 2010) for time borrowing can preserve the sequence of pipeline without flushing. For power saving and thermal management, detecting the error rate is recommended to scale both frequency and voltage dynamically. Using dynamic programmable latches to retime the data critical paths by resizing critical combinational logic stages is another trend of research.

Acknowledgements

This research was partially funded by Zewail City of Science and Technology, AUC, the STDF, Intel, Mentor Graphics, and MCIT.

References

Andrade, D., Martorell, F., Calomarde, A., Moll, F. and Rubio, A. (2009) 'A new compensation mechanism for environmental parameter fluctuations in CMOS digital Ics', *Microelectronics Journal*, June, Vol. 40, No. 6, pp.952–957.

Buedo, S.L., Garrido, J. and Boemo, E. (2000) 'Thermal testing on reconfigurable computers', *IEEE Design & Test of Computers*, Vol. 17, No. 1, pp.84–91.

Dhar, S., Maksimovi'c, D. and Kranzen, B. (2002) 'Closed-loop adaptive voltage scaling controller for standard-cell ASICs', *Proc. of the 2002 International Symposium on Low Power Electronics and Design, 2002, ISLPED '02*.

Ernst, D., Kim, N.S., Das, S., Pant, S., Rao, R., Pham, T., Ziesler, C., Blaauw, D., Austin, T., Flautner1, K. and Mudge, T. (2003) 'Razor: a low-power pipeline based on circuit-level timing speculation', *The 36th Annual IEEE/ACM International Symposium on Micro-architecture, 2003, MICRO-36*.

Harris, D. and Horowitz, M.A. (1997) 'Skew-tolerant domino circuits', *IEEE Journal of Solid-State Circuits*, November, Vol. 32, No. 11, pp.1702–1711.

Jones, P.H., Cho, Y.H. and Lockwood, J.W. (2006) 'An adaptive frequency control method using thermal feedback for reconfigurable hardware applications', *IEEE International Conference on Field Programmable Technology (FPT)*.

Jones, P.H., Moscola, J., Cho, Y.H. and Lockwood, J.W. (2007) 'Adaptive thermo regulation for applications on reconfigurable devices', *International Conference on Field Programmable Logic and Applications, FPL 2007*.

Lee, S., Das, S., Pham, T., Austin, T., Blaauw, D. and Mudge, T. (2004) 'Reducing pipeline energy demands with local DVS and dynamic retiming', *Proc. of the 2004 International Symposium on Low Power Electronics and Design, 2004, ISLPED '04*.

Long, J., Ku, J.C., Memik, S.O. and Ismail, Y. (2010) 'SACTA: a self-adjusting clock tree architecture for adapting to thermal-induced delay variation', *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, September.

Paik, S., Lee, S. and Shin, Y. (2011) 'Retiming pulsed-latch circuits with regulating pulse width', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, August, Vol. 30, No. 8, pp.1114–1127.

Pan, A. (2009) *A Hardware Framework for Yield and Reliability Enhancement in Chip Multiprocessors*, MSc thesis, September, Graduate School of the University of Massachusetts Amherst.

Sapatnekar, S. (2004) 'Statistical static timing analysis', *Timing*, Kluwer Academic Publishers, Boston; created in the USA.

Sathyamurthy, H., Sapatnekar, S.S. and Fishburn, J.P. (1995) 'Speeding up pipelined circuits through a combination of gate sizing and clock skew optimization', *IEEE/ACM International Conference on Computer-Aided Design, 1995, ICCAD-95*, Digest of Technical Papers.

Shyam, S., Constantinides, K., Phadke, S., Bertacco, V. and Austin, T. (2006) 'Ultra low-cost defect protection for microprocessor pipelines', *Proc. of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems*.

Sung, R.J.H. and Elliott, D.G. (2007) 'Clock-logic domino circuits for high-speed and energy-efficient microprocessor pipelines', *IEEE Transactions on Circuits and Systems II*, May, Vol. 54, No. 5, pp.460–464.

Tsai, Y.T., Tsai, C.C., Chien, C.A., Cheng, C.H. and Guo, J.I. (2011) 'Dual-phase pipeline circuit design automation with a built-in performance adjusting mechanism', *16th Asia and South Pacific Design Automation Conference, ASP-DAC2011*.

Wieckowski, M., Park, Y.M., Tokunaga, C., Kim, D.W., Foo, Z., Sylvester, D. and Blaauw, D. (2008) 'Timing yield enhancement through soft edge flip-flop based design', *IEEE Custom Integrated Circuits Conference, 2008, CICC 2008.*

Yahya, E., Elissati, O., Zakaria, H., Fesquet, L. and Renaudin, M. (2009) 'Programmable/stoppable oscillator based on self-timed rings', *Proc. of the 15th IEEE Symposium on Asynchronous Circuits and Systems, 2009, ASYNC '09.*