

David Flynn, James Myers, and Seng Toh

This chapter addresses the approaches and methodologies appropriate to energy-constrained SoC design, implementation and verification using standard multi-voltage Electronic Design Automation tools, rather than resorting to full-custom circuit approaches. The Physical-IP libraries, memories and power-management components required to address both active-mode energy and deep-sleep state retention power are introduced, followed by a case study addressing the specific challenges of optimizing a micro-processor subsystem for Near- and Sub-Threshold Voltage operation. As well as system level power management the implementation and verification of clock distribution and system timing closure are covered in detail.

9.1 Example Activity Profiles for IoT Sensor Nodes

For IOT “edge-nodes” such as Wireless Sensor Nodes the typical activity and power profile is shown in Fig. 9.1.

The height of the bar indicates the relative current consumed or power dissipated and the width is indicative of the duration. The system is typically optimized for minimum residual cur-

rent in between periodic sensing, data processing and data transmission: this is annotated STANDBY in the figure, and in many systems this is the predominant impact on battery life.

1. The sensing activity is normally periodic and triggered by some form of real-time sample request at a controlled data collection rate. The height and width are conceptually marked as the activities labeled SENSE in the figure.
2. Some form of data processing step, such as filtering, or anomaly or limit detection, is often initiated after a certain number of samples have been buffered. The duration and current profile may be data-dependent, and is shown annotated as PROCESS.
3. A Wireless Sensor Node will typically package or compress data to minimize the energy required to transmit the data, and in many systems the transmission time-slots are pre-scheduled at specific times dependent on the wireless access protocol and scheduler (maybe in a sensor hub or base-station) at a rate that is independent of the data-sampling rate. This is shown with arbitrary power and duration as TRANSMIT in the figure.

Regardless of the specific waveform amplitude and activity profiles the key elements required to be minimized in design and implementation are:

D. Flynn (✉) • J. Myers • S. Toh
ARM Ltd, Cambridge, UK
e-mail: David.Flynn@arm.com

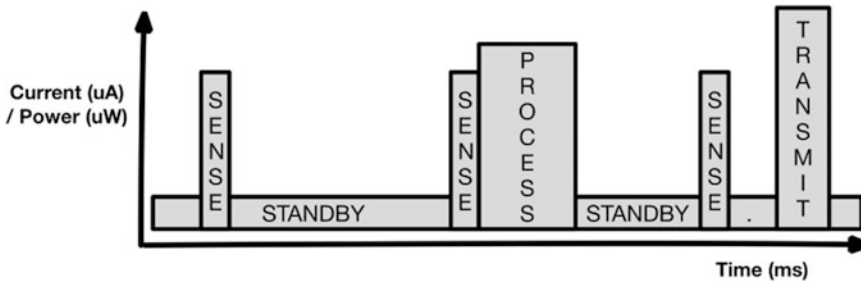


Fig. 9.1 Example active and standby profile for wireless sensor node

- Leakage and state retention energy—the integration of power over time for the Standby component when the main circuitry is inactive.
- Dynamic energy consumption when clocks are enabled to specific components required for active computation or communication.
- Peak active current—which is often the limiting factor in small on-chip power regulation schemes.
- Both active and leakage power consumption for the “always-on” circuitry such as the timer or Real-Time Clock (RTC) that provides the wake-up event scheduling.
- The PMOS and NMOS power gates are optimized for I_{ON}/I_{OFF} ratio, but the series on-resistance typically impacts circuit performance despite the off-current savings so in industrial usage only one is typically used. Footer NMOS power gating is shown in (b).

To mitigate peak current inrush when turning on the power gate there are various ways to build resistive turn-on networks but one approach is to support a threshold-voltage drop using a PMOS transistor in the case of footer-switched VVSS rail, as shown in (c). A logic-0 drive on both **PWR** and **nDROWSE** controls enables this mode of operation.

9.2 Static Power Reduction

9.2.1 Power Gating

The primary technique for static power reduction is power gating (Mutoh et al. 1996), which is well supported in Multi-Voltage EDA tools. Figure 9.2 illustrates the theory and practice:

- Early academic research focused on Multi-Threshold CMOS power gating, MTCMOS (a), where high-threshold “header” and “footer” switches are added to create gated “virtual” VDD and VSS rails, labeled VVDD and VVSS. The logic is powered when **PWR** control to footer is logic-1 and **nPWR** to header switch is logic-0.

9.2.2 Power Gating and Well-Bias

In the case of full MTCMOS power gating, as shown in Fig. 9.3a with the P- and N-wells are explicitly annotated, the VVDD and VVSS virtual switched rails collapse towards a mid-rail voltage with symmetric reverse bias to the switch P- and N-channel logic transistors. With the addition of “drowsy” threshold-voltage transistors it is possible to provide a mode which holds the logic sub-threshold with symmetric well bias (**DROWSE**=1, **nDROWSE**=0) and can support quick wake from sleep with $3\times$ lower wake energy (Mistry et al. 2014).

This is a special case of well-bias for standby which is simple to implement without requiring multi-dimensional standard cell characterization

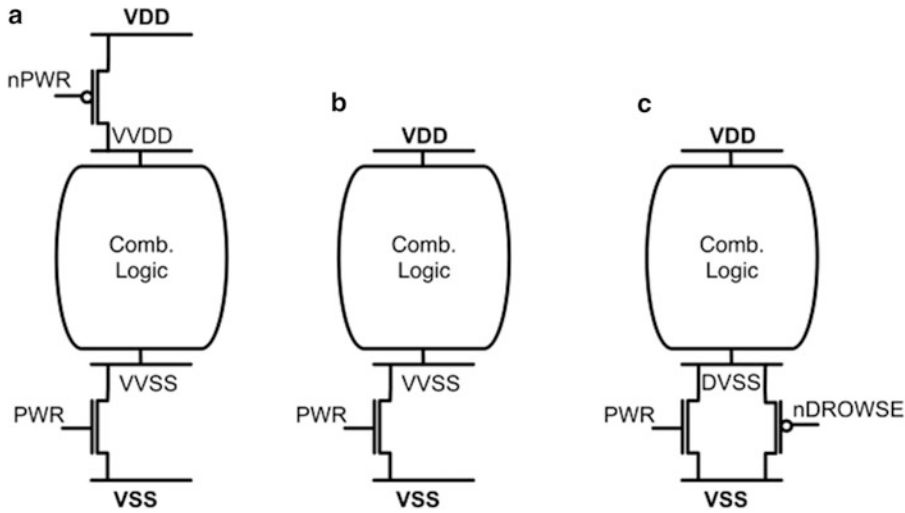
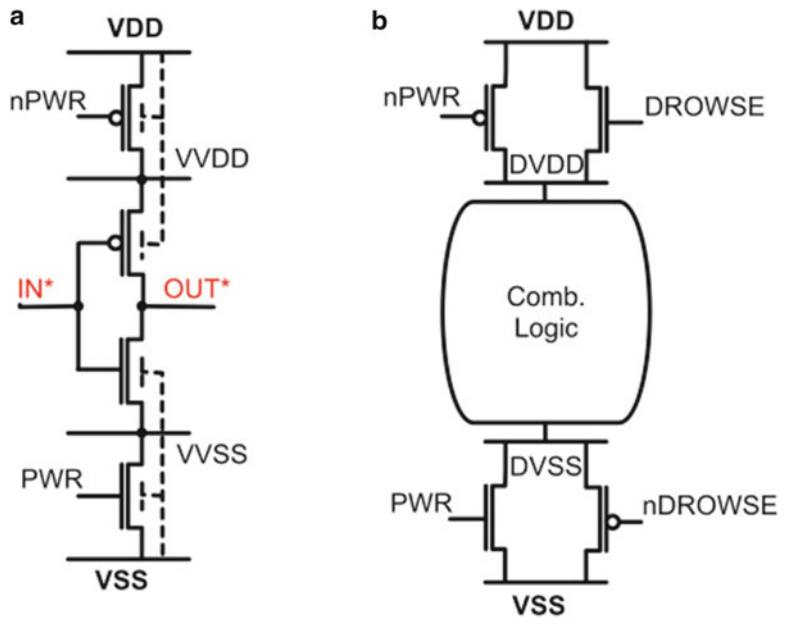


Fig. 9.2 Power gating (a) MTCMOS, (b) footer, (c) “drowsy” scaling

Fig. 9.3 (a) Power gating and well-bias, (b) drowsy rail power gating



as needed for forward-body-bias active modes. While traditional reverse-body-bias techniques are effective in older bulk process nodes it is a challenge in smaller IoT designs to generate boosted P-well and N-well voltages without expending more active power than the leakage that is saved.

9.2.3 Boosted-Gate Switches for Low-Voltage Power Gating

As discussed in Sect. 5.5, in near- and sub-threshold designs the voltage headroom to drive power gate controls with logic-level voltage swings results in compromised I_{ON} switch

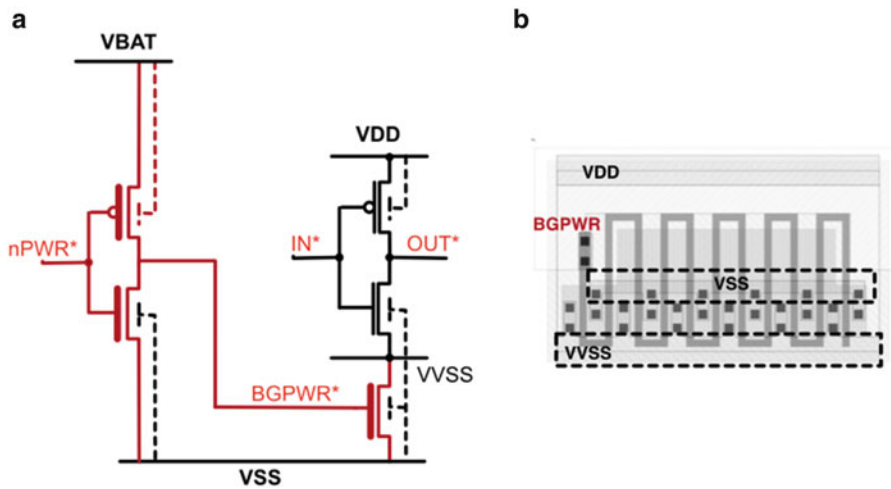


Fig. 9.4 (a) BG-CMOS footer power gating, (b) footer power gating cell

behavior. An effective technique to ensure highly effective power gating for reduced voltage logic rails is to use high threshold power gates with boosted gate voltage (Stan 1998). An optimal implementation can be achieved by building the control buffering and the footer power gates with Thick-Gate-Oxide (TGO) devices which can be operated from the unregulated battery (or super-capacitor in the case of battery-less systems) voltage rail, shown as **VBAT** in Fig. 9.4a. This requires care in the implementation flow to handle the extra higher-voltage rail distribution, but this is all low-current from a control signal perspective and results in highly effective distributed sub-circuit power gating.

Figure 9.4b also shows an example of the standard cell abstraction used for power gates that can be cleanly deployed in EDA implementation flows. The switch shares the **VDD** supply row architecture but connects the standard-cell ground rail as a switched virtual **VVSS** track. The global **VSS** supply is via-ed down to the power gate from the thick-metal ground mesh, and the switch is laid out as multiple fingers of switch where device length and width are tuned for best I_{ON}/I_{OFF} ratio.

9.2.4 Clamping and Isolation

Power gating provides effective leakage current reduction when sub-circuits are powered down,

but signals at the boundaries collapse to non-logic levels. In order to prevent crow-bar currents flowing in logic down-stream of a power gated block or region, specialized isolation or clamp cells are provided which are powered from the global **VDD** and **VSS** rails and provide the equivalent of AND- or OR-gate signal clamping. IEEE1801 power intent supports explicit association of high or low isolation signals to interface nets (IEEE Standard for Design and Verification of Low Power Integrated Circuits). Example cells are shown in Fig. 9.5.

9.2.5 State Retention with Power Gating

Power gating provides leakage power savings by switching off a sub-circuit, but any state is lost and the sub-circuit needs to be reset or reinitialized after the power is turned back on. For circuits such as fixed function accelerators with only transient state this is usually acceptable. But in many cases the loss of all current state is too costly such that either the architectural state must be saved away before power gating and restored after power is turned on (which may have considerable latency and energy cost), or the state is preserved in-place and maintained for a much smaller leakage cost.

Fig. 9.5 (a) Clamp-low cell, (b) clamp-high cell

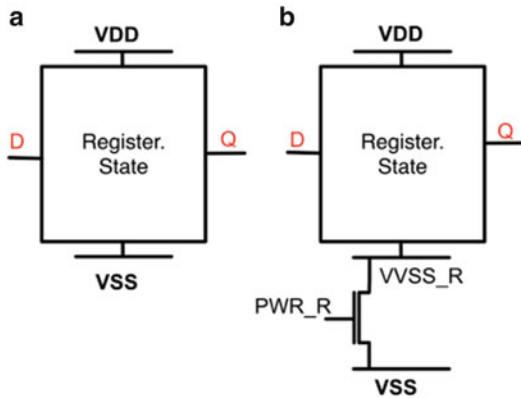
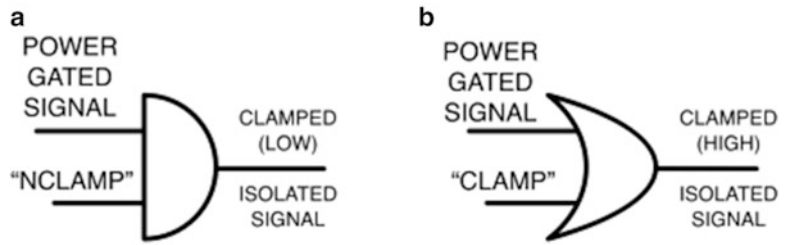


Fig. 9.6 (a) Always-on register, (b) independently gated state retention

Figure 9.6 illustrates the basic approach of either providing “always-on” power to register state (not supplied from the gated virtual standard cell rails) in (a), or adding a second gated retention rail “VSS_R” which can be independently controlled by the PWR_R control signal to allow state to be retained for certain periods and turned off fully for deep sleep modes.

9.2.6 State Retention with Power Gating Optimizations

To minimize the state retention currents incurred in standby mode states when retaining state is required there are optimizations that can be appropriate depending on the voltage headroom and register stability at scaled voltages (Kumagai et al. 1998). Figure 9.7 shows the concept applied to standard master-slave registers (a). A drowsy state virtual ground rail, labeled DVSS is shown

which can be either fully on (PWR_R=1), drowsy voltage scaled (PWR_R=0, nDROWSE_R=0) or fully off (PWR_R=0, nDROWSE_R=1). Careful validation of the voltage scaled retention reliability must be evaluated for the technology used, but this can provide valuable retention current savings for designs with a large number of registers.

Figure 9.7b illustrates a further optimization where only the slave latch portion of the master-slave flip-flop register is retained, while the input stage, master latch and output driver are all powered from the switched standard-cell VVSS rail, but a separate drowsy-voltage-scaled retention virtual ground rail is provided to the slave latch only (Flynn et al. 2012). While retention registers and power gating have been shown to save 95% of the standby leakage power, drowsy retention can achieve a further 50% reduction.

In both cases there is a need to add minimal clamping circuitry around the register or slave latch to protect this for floating inputs, especially clocks and resets, and this can be added to the cell or supported across a group of registers controlled by a “retention isolation” signal.

9.2.7 Physical Considerations for Independently Gated State Retention

While power intent formats such as CPF and UPF (IEEE 1801) support an arbitrary number of switched supplies, there is an assumption of a single default supply pair for each power domain. Power aware EDA tools expect this to be routed as the standard cell main rail. While always-on buffers (for buffering power gate, clamp or retention controls) are a well understood exception

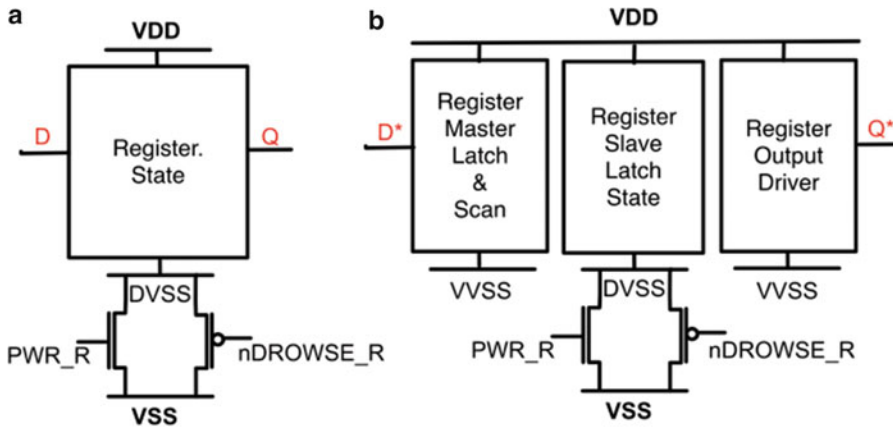


Fig. 9.7 (a) Drowsy voltage retention, (b) slave latch drowsy retention

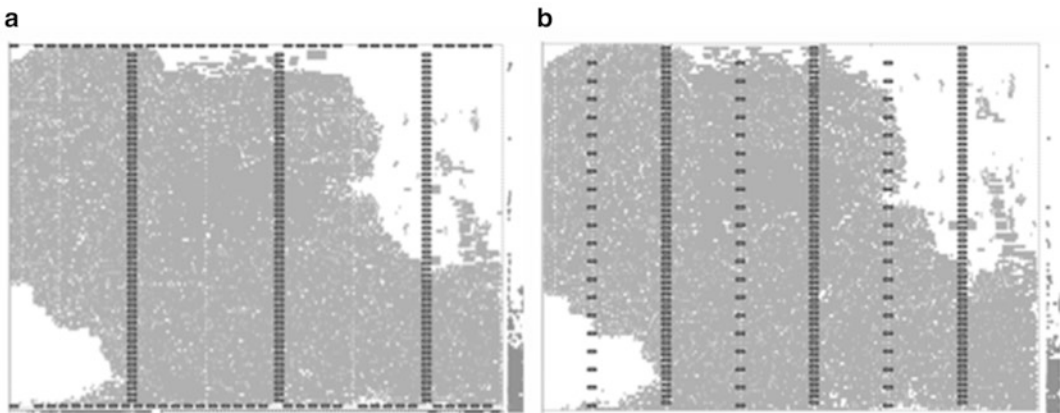


Fig. 9.8 ARM Cortex-M0+ power domain cell placement highlighting power gates (a) retention power gates implemented as *top/bottom* rows, (b) retention power gates implemented as columns with dedicated standard cell

where power supply effectively bypasses the local power gates, this is more complex for independently gated state retention as two or more switches must be placed within a single floorplan region.

The scheme illustrated in Fig. 9.6b can be implemented in two ways. The main logic power gate is usually distributed throughout the floorplan as many small power gate cells, such that the default VVSS for stateless logic can be driven onto the standard cell rails most effectively. If the retention power gate is to be implemented with the same library power gate cell, then it must be confined to dedicated rows to

prevent VVSS and VVSS_R from shorting together. For smaller power domains this is best implemented as top and bottom rows shown in Fig. 9.8a, but larger domains may also require pairs of dedicated rows through the center to reduce power grid voltage droop.

The other option is to use an alternate power gate cell, which does not drive onto the standard cell main rail but a retention rail instead. This has the advantage of allowing distributed placement as with the logic power gate, shown as additional half-density columns in Fig. 9.8b.

In power domains with a high concentration of retention registers it may be beneficial to route

this retention rail across the entire design, although this consumes an entire routing track and may not be feasible with the highest density six or seven track standard cell libraries where pin access is a challenge. In other cases it may be best to connect power gate `VVSS_R` directly up to a power grid and use on-demand routing to connect down to individual standard cells as required.

Note that the drowsy schemes illustrated in Fig. 9.7 intentionally short the outputs of the PMOS & NMOS footers, so the floorplan does not increase in complexity beyond the switched retention case.

9.2.8 Sequencing of Power Gating Controls

Power gating typically has to be controlled by a state machine powered in a relatively always-on voltage domain. The control signals that are required to drive the power gating and isolation clamping described previously need to have explicit sequencing and these are the ports that get connected to the IEEE 1801 inferred power controls (Keating et al. 2007). Figure 9.9 illustrates the standard sequencing into power-gated mode and then waking up back to active mode. On a request to sleep, first the clock is stopped, then the reset asserted (not strictly necessary but maintains symmetry), the isolation clamping of outputs is asserted and then the power-gating network turned off.

In the figure a power-gating acknowledge signal is shown which is valuable to ensure that the timing required to power down and back up the switched network is handled correctly by design (e.g. to avoid the condition when a wake up occurs just after the power gating is turned off and power is un-driven momentarily). Although comparators or Schmitt circuits may be used to assert this `PWR_ACK` when the virtual rail has been charged to a target voltage, in practice it is usually the output of a delay line, synchronous counter, or power gate daisy-chain (Shi et al. 2006).

Larger designs may suffer from high in-rush currents and ground bounce when powering up too quickly, which can cause corruption of retained state or timing errors in active blocks. Analysis of this in-rush current is therefore an important sign-off step for low power designs and is well supported by EDA tools. Where in-rush currents are found to be unacceptably high (a typical target is no greater than active peak currents, although tighter constraints may be required for some classes of design), the most common mitigation approach is to stagger power gate enables, such that a fraction of power gates are used to initialize the virtual rail more slowly. Drowsy power gates (as mentioned in Sect. 9.2.2) are another good option.

On a request to wake-up, power is requested, and only when valid is the reset de-asserted, the isolation clamping turned off and the clock finally re-enabled. In the case of state retention power gating the retain control signal timing is often similar to the isolation `NCLAMP` control waveform.

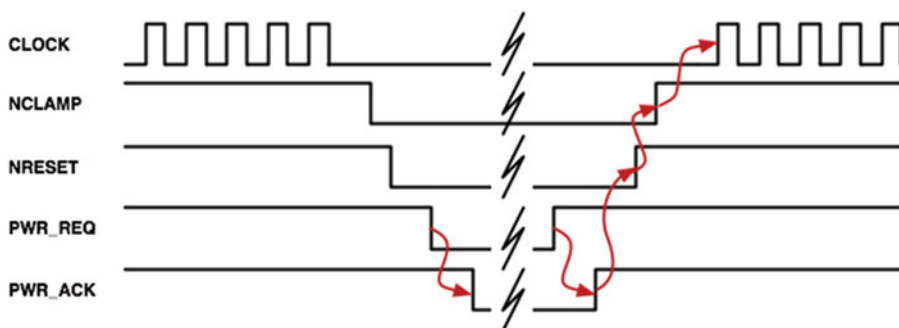


Fig. 9.9 Example clock, clamp, reset and power gate control sequencing

9.3 Active Power Reduction

The management of active power is mainly focused on addressing terms in the familiar CMOS dynamic power proportionality to CV^2F equation. The capacitance term, C , is minimized by striving to keep the circuit small and simple, balancing drive strength and keeping signal routing capacitance to a minimum. The frequency term, F , is addressed by optimizing the circuit implementation for peak required performance and then factoring in both architectural and inferred clock gating to suppress dynamic power dissipation whenever possible. The voltage term, V , is the most valuable control knob given the square-law contribution; in IoT applications the primary focus is to work with low voltage technology, with under-driven super-threshold libraries and memories, or more specialized near-threshold or sub-threshold robust physical IP.

9.3.1 Clock Gating

EDA tools are able to provide transparent clock gating where common sub-expressions in the enable terms of synchronous logic are coded in a clean synthesizable style. Figure 9.10 illustrates the basic scheme for determining groups of registers that share an enable term where the state is defined as sampling or re-circulating values. The multi-bit registers are shielded from the high-toggle-rate clock, marked in red by the inference of a latch and AND gate structure that suppresses clock pulses in cycles when the EN term is inactive (Fig. 9.10b). Figure 9.10c shows the cell abstraction for an Integrated Clock Gate (ICG) that provides the timing and clock

balancing attributes to EDA tools to support clean static timing analysis and clock tree balancing with such gated clocks.

Such ICG elements can also be instantiated in designs to support high-level architectural clock gating where the designer can determine where clock segments can be individually gated explicitly at system level.

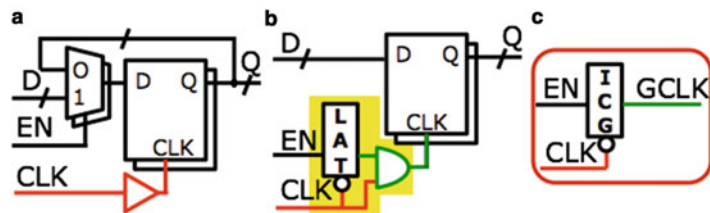
9.3.2 Voltage Scaling

Today's production microcontrollers rarely support dynamic voltage and frequency scaling (DVFS) due to the complexities of lightweight OS/SW scheduling, interfacing to off-chip voltage regulators, managing transition periods, identifying optimal voltage-frequency pairs, limited super-threshold voltage headroom, and more. But it is clear that this will be a key area of improvement for future IoT edge-node applications, enabled by integrated voltage regulators (see Chap. 10) and the increased versatility offered by near- and sub-threshold designs.

The physical IP to support this includes cell libraries that are optimized for the constrained voltage headroom. For near- and sub-threshold operation this usually implies constrained cell architectures, which avoid small length/width devices and minimize transistor stack depths, and register latch and memory bit-cells in particular need to be designed with increased robustness.

The only additional cells required are in the form of level-shifters that manage the voltage drive from low voltages up to higher voltage domains or input/output interface drivers. There are three types of interface with specialized level

Fig. 9.10 Clock gating inference and abstraction



shifters for each: guaranteed low voltage input to high voltage output; guaranteed high voltage input to low voltage output (which may be simply re-characterized buffer cells to enable STA tools to estimate timing correctly); and interfaces where input/output voltages scale independently and may be high/low, low/high, or the same. EDA tools can infer the appropriate level shifters from IEEE 1801 power intent definition and cell-library attributes.

9.3.3 Wake-Up and Power Management Circuits

A special case of active circuits that need to be always-on relative to the processing subsystems are the power management state machines and wake-up sources such as Real-Time-Clock (RTC) alarms. RTC circuits are typically clocked as low as 32 KHz, which dramatically reduces dynamic power compared to other logic running at MHz. Always-on leakage is still a concern and should be reduced by aggressive gate-count reduction and implementation with the lowest leakage devices available. Simple libraries of TGO gates and registers can be beneficial here as these demonstrate up to 100× reduction in leakage compared to regular threshold thin-oxide devices (Taki et al. 2011). The most compelling benefit of TGO libraries however is that they can run directly from unregulated battery voltages, thereby allowing all voltage regulators to be shutdown in deep sleep modes and saving regulator losses which can be significant under very light loading.

9.4 Automated Minimum Energy Design

Conventional EDA tools for automated synthesis, place, and route are usually optimized to produce designs with maximum performance or minimum power. Minimum energy design in general requires achieving both minimum

power without sacrificing performance, especially at the minimum energy point where leakage energy is strongly dependent on performance. This sub-chapter describes how conventional EDA flows may be adapted to achieve a minimum energy design. The impact of key decisions such as standard cell choice and clock design methodology on minimizing energy and cost are also evaluated. Results in this section are derived from a 65 nm R&D sub-threshold ARM[®] Cortex[®]-M0+ WSN processing sub-system with prototype 300 mV physical IP.

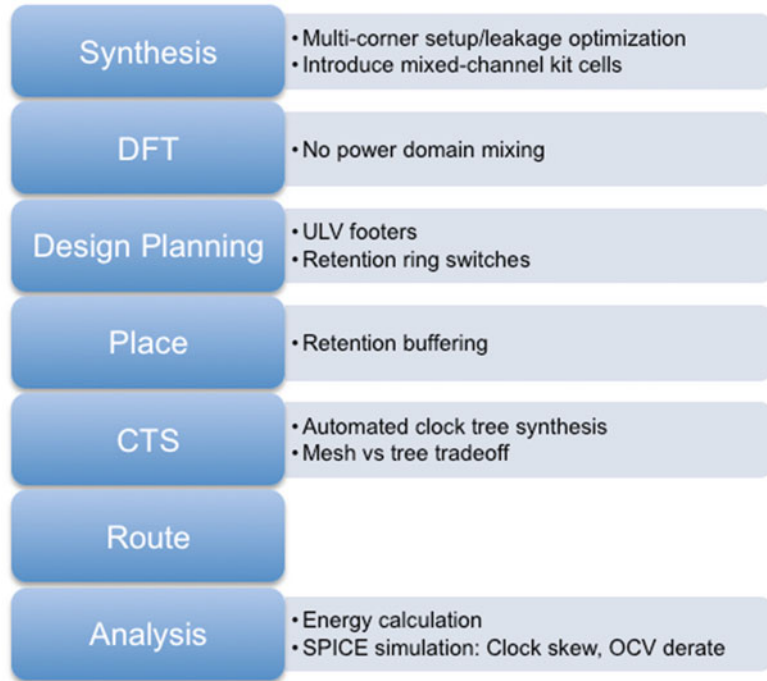
9.4.1 Implementation Flow

The majority of the implementation methodology is unchanged from a standard EDA flow and no custom tools are required at any stage. Power aware verification of the design is performed using a gate level simulator together with UPF power intent. The flow modifications identified in Fig. 9.11 will be covered in detail below, with the exception of design-for-test and placement steps that are not unusual for a highly power gated design.

9.4.2 Energy Reporting

Energy is the most important metric in this design and needs to be reported in all optimization steps. The tools however only report power. The calculation is simple so custom reporting can be implemented. Leakage energy/cycle is leakage power integrated over the minimal clock period. The subtlety here is that increased leakage power is acceptable, if the corresponding speedup is greater than or equal in magnitude. Dynamic energy/cycle is simply dynamic power divided by clock frequency. The libraries in this example were characterized at five voltages, which allow the majority of the voltage-energy curve to be interpolated.

Fig. 9.11 EDA flow with key modifications for minimum energy design



9.4.3 VT Selection Leakage/Performance Tradeoff

Regular VT (RVT) and low VT (LVT) gates exhibit an $8\times$ difference in performance when operating at sub-threshold voltages while leakage power scales by almost $20\times$ (Fig. 9.12). The amplified performance difference deviates from observed behavior at nominal voltages where performance typically improves by 50% with a $10\times$ leakage power increase by switching to a lower VT choice. Traditional leakage recovery flows that trade-off timing slack on each timing path for leakage reduction are not effective in sub-threshold design as the number of cell swaps that can be taken on each path are limited. Our front-end and back-end flows utilize only RVT gates in order to minimize system leakage. We also utilize an RVT mixed-channel kit (MCK) which has higher performance gates, achieved by optimizing the gate lengths of the transistors. The MCK library cells achieve an average 12% performance improvement at $3\times$

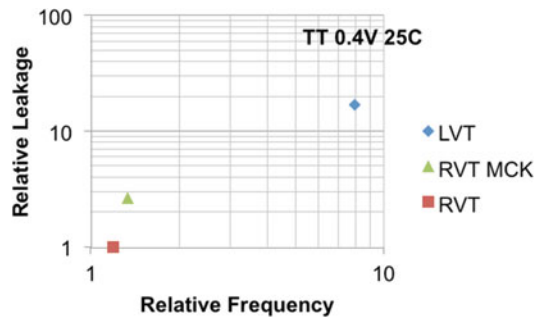


Fig. 9.12 Leakage vs. frequency comparison of various standard cell choices

higher leakage. Using these MCK cells sparingly on the design helps improve performance, resulting in lower leakage energy.

9.4.4 Cell Choices During Optimization Flow

This design ends up with 9.27% of MCK cells (by area) when no constraints are placed on

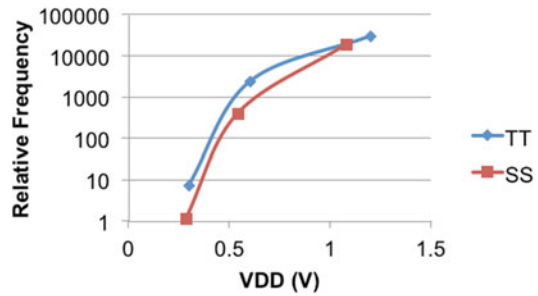
Table 9.1 Comparison of unconstrained vs. 5% MCK usage constraint

	Unconstrained	5% MCK constraint
Normalized area	1	1.025
Normalized dynamic energy	1	1.060
Normalized leakage energy	1	1.029
Normalized performance	1	1.013
MCK cell area	9.27%	5.95%

percentage of MCK cell usage. MCK cell usage constraints applied during synthesis results in a design that has higher area and energy (Table 9.1). This is likely caused by the leakage optimization algorithms in the synthesis tool that try as hard as possible to first implement the design using high VT cells before enabling a minimal amount of low VT cells. This high effort algorithm results in area increase and worse dynamic power. We also experimented with ECO leakage recovery in timing signoff tools but only observed a handful of cell swaps, resulting in miniscule leakage recovery. Based on these experiments, we decided to adopt a simple flow and enable unconstrained use of MCK cells beginning from synthesis.

9.4.5 Optimization Corner Selection for Minimum Energy and Runtime Impact

Battery powered WSN are required to scale voltage and frequency actively in order to meet throughput or latency requirements as well as to minimize energy during low periods of activity. These circuits are therefore required to be functional as VDD is scaled from nominal voltages down to sub-threshold voltages. Figure 9.13 plots the relative frequency of a design at TT and SS global process corners as supply voltage is scaled. Frequency degrades by 4000 \times across the span of operating voltages while performance degrades by 5 \times between TT and SS corners. DVFS across such wide operating conditions is

**Fig. 9.13** Leakage vs. frequency comparison of various standard cell choices

expected to require multi-corner optimization to ensure good performance across corners.

A study was conducted into multi-corner setup optimization vs. single corner setup optimization to determine whether multi-corner optimization is an absolute necessity. Hold timing was still optimized across all corners in these experiments. Figure 9.14 presents the resulting leakage and dynamic energy observed (normalized to single corner setup at SS 1.08 V). Good correlation is observed between the choice of setup corner and optimized leakage energy especially at voltages below 0.6 V. For example, SS 0.54 V minimizes leakage energy around 0.54 V while TT 0.3 V minimizes leakage at 0.3 V. The better performance at the respective voltages results in lower leakage energy, as leakage power is integrated over a shorter period of time. Multi-corner setup optimization appears to produce results that are close to minimum leakage energy across voltages because multi-corner optimization strives to optimize performance across all voltages. Figure 9.14 also plots the Synthesis runtime for the various single corner and multi-corner optimizations. Multi-corner optimization incurs a 4 \times runtime over single corner, which is quite reasonable considering the various corners that are being considered simultaneously.

9.5 Clock Distribution

Clock distribution is extremely challenging in sub-threshold voltage design due to increased

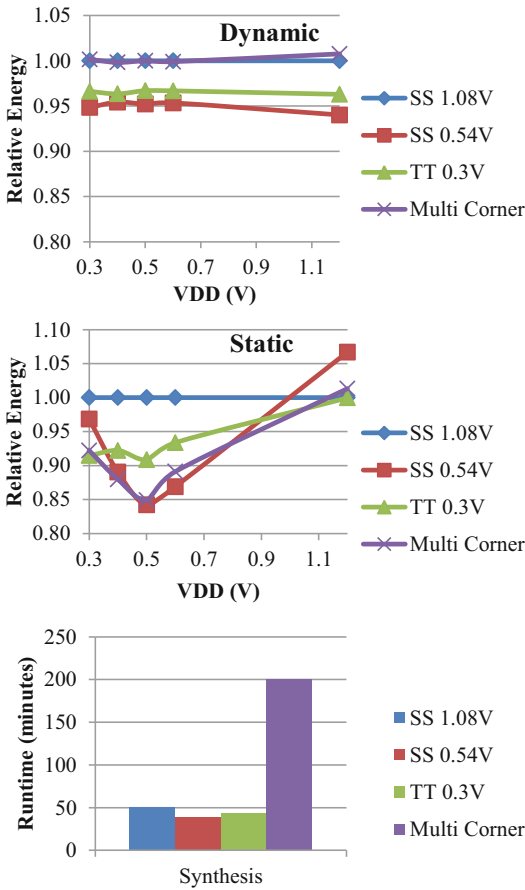


Fig. 9.14 Comparison of power and runtime effects of single corner vs. multi-corner optimization

on-chip-variation (OCV), larger clock latency due to slow buffers, and the requirement for minimizing energy.

9.5.1 OCV Characterization

OCV impacts clock distribution by introducing variation in arrival times to registers and memories. Margins are used to design for this variation, resulting in performance degradation or data-path upsizing to meet performance targets.

Figure 9.15 presents variation in delay through a chain of inverters across different dies. This analysis was performed using SPICE simulation of an extracted netlist of inverters and

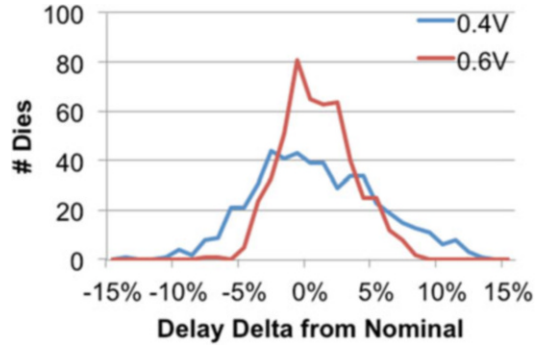


Fig. 9.15 Comparison of single corner vs. multi-corner optimization

Monte Carlo analysis. The spread in delay increases as voltage is scaled down to sub-threshold voltages because transistor performance is exponentially dependent on threshold voltage. Foundries typically specify OCV derates that are relevant to $V_{nom} \pm 10\%$. Larger OCV derates are required at sub-threshold voltages in order to margin for the worse variation. An OCV derate is derived from statistical data by multiplying the worst observed sigma by 3.

Clocks are typically distributed using synthesized tree structures or more structured networks like H-trees (Jain 2012) or meshes (SolvNet 2014). Synthesized clock trees are designed and optimized automatically by the tools but tend to exhibit worse performance compared to more structured networks. Clock meshes are quite attractive in sub-threshold design because they can be used to distribute the clock signal across the entire chip without OCV impact. Figure 9.16 illustrates the clock mesh structure that was investigated. Clock meshes essentially distribute a clock source across an entire portion of the design. This is usually achieved using a pre-tree. The outputs of the leaf nodes of the pre-tree are shorted together on the clock mesh to reduce the skew of the clock signal distributed by the pre-tree, creating an almost-ideal clock signal that spans the entire design area. The key to achieving best results from a clock mesh is to reduce the number of gates between the mesh and the clock sinks. We used all clock gates to accomplish this final step, forcing dummy clock gates on clock sinks that

Fig. 9.16 Clock mesh structure

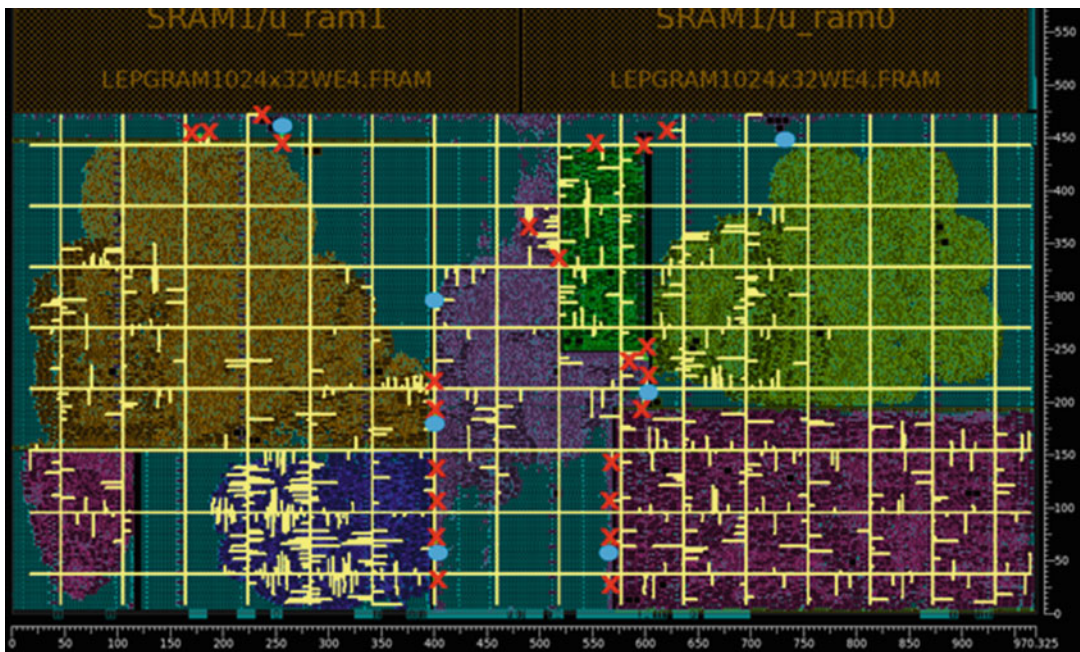
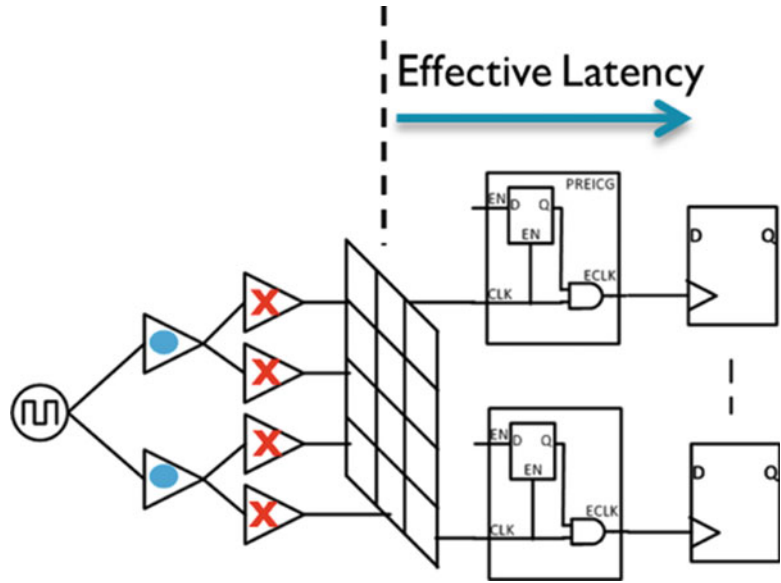


Fig. 9.17 Sub-system standard cell area floorplan, annotated with mesh structure

are not gated. The effective latency of this clock structure is therefore only one gate. Figure 9.17 is a layout view of the clock mesh with respective drivers indicated by respective symbols corresponding to Fig. 9.16. All pre-tree drivers

were placed in the middle section of the floorplan because this area corresponds to a power domain that is on in all active modes but can be switched off during retention and power down modes. This ensures that the clock mesh is always alive

regardless of the power state combinations of the system. Unfortunately, this also implies that the clock mesh is always consuming dynamic and leakage power.

9.5.2 Clock Tree Synthesis Methodology

Clock trees are the easiest way to distribute clocks with minimal dynamic power and area. Clock trees however suffer from more variability due to the lack of regularity in the structure. This section demonstrates best practices for clock tree synthesis, especially targeting sub-threshold minimum energy clock trees.

Transistor variability is inversely proportional to gate area. Low drive-strength clock cells are therefore expected to exhibit larger variability. Figure 9.18 plots results obtained from OCV characterization of the clock tree. Clock arrival sigma of up to 12% was observed in a clock tree that was not optimized (equating to 36% OCV derate!). Eliminating X1 drive inverters, buffers,

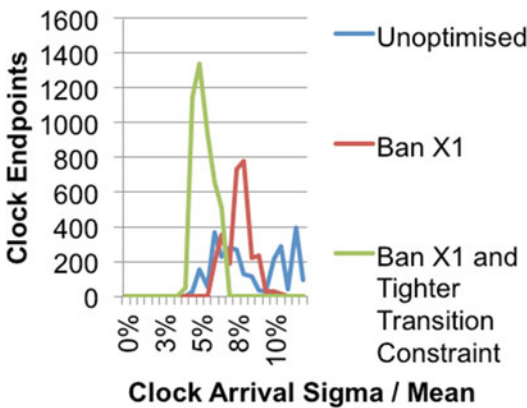


Fig. 9.18 Clock end-point variability across different optimizations (TT 0.4 V)

and ICG from the clock cell list reduced sigma to 9%. The area penalty incurred from not using X1 cells is worth the reduction in sigma because the clock tree accounts for less than 1% of total area. Sigma of clock arrival time is further reduced by introducing tighter max-transition constraints on the clock tree.

Automated clock tree synthesis algorithms are typically designed to improve clock-related metrics (latency and skew) at a particular corner. Reducing skew is usually accomplished by adding additional gates to balance delays between paths. We analyzed clock tree metrics observed when using SS 1.08 V CTS corner vs. SS 0.54 V CTS corner (Table 9.2) to determine which strategy produces the best results. Building the clock tree at SS 0.54 V provides minimum clock skew at sub-threshold voltages, compared to the SS 1.08 V CTS corner. This tighter skew was achieved by padding with more clock cells, resulting in 2× larger area. This clock tree however does not scale well with voltage and exhibits a clock skew of 8.9% clock period when measured at the SS 1.08 V corner. The clock tree constructed at the SS 1.08 V corner results in a clock tree that exhibits consistent skew across operating voltages and minimizes area. We have also limited clock tree fanout to 32 which minimizes interconnect delay and helps ensure consistent scaling of all clock endpoints across all operating conditions.

LVT clock cells present an interesting option especially for sub-threshold design due to the 8× better performance (only ~50% better performance is typically observed at nominal voltages). These faster cells will result in up to 8× lower clock latency, which reduces the impact of OCV by the same magnitude. An LVT clock tree will exhibit up to 10× higher leakage power than an RVT clock tree but the improvement in performance can potentially offset the leakage power

Table 9.2 Comparison of clock tree metrics synthesized at different CTS corners

Target CTS corner	Clock skew at respective corner as % of clock period			Area (μm^2)	Depth
	SS 1.08 V	SS 0.54 V	TT 0.4 V		
SS 1.08 V	1.5	2.5	2.1	1587	5–14
SS 0.54 V	8.9	0.9	1.2	3414	13–18

increase, resulting in a net reduction in leakage energy.

Figure 9.19 presents the dynamic energy and leakage energy of the clock distribution network implemented using different strategies. The LVT tree has lower dynamic energy compared to the

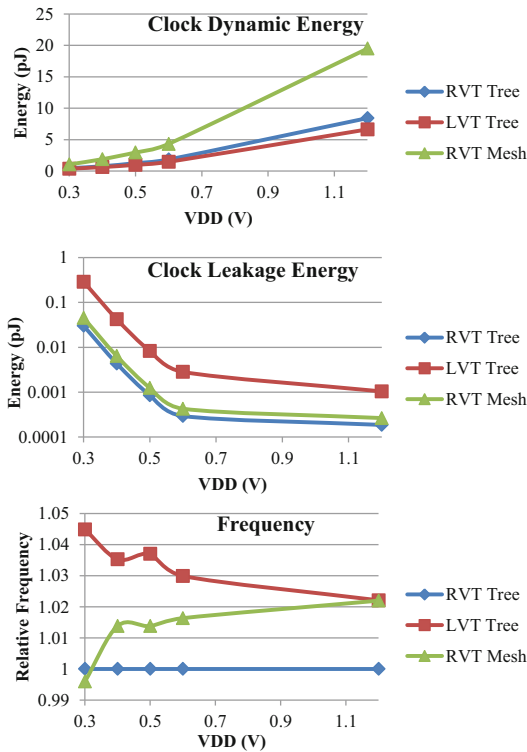


Fig. 9.19 Comparison of clock energy and frequency across clock tree implementations

RVT tree because the transitions are much sharper, resulting in lower short-circuit current. Leakage energy of the clock network however is almost 10× higher than the RVT tree. Table 9.3 presents some additional metrics measured from our WSN sub-system implemented using different clock strategies. The clock latency of the LVT tree is 4% of the clock period while the RVT tree latency is 18% of the clock period. Note that the LVT tree only achieves a design that is at-most 4% faster than the RVT tree, even though clock latency is much lower. The WSN sub-system we have designed is too small to realize the benefits of an LVT clock tree with lower latency. We expect larger designs, where clock latency is a larger fraction of clock period, to exhibit higher performance improvements from an LVT tree due to reduced impact of OCV. Another thing to consider with LVT tree is the cost of the additional VT implant. This could potentially tip the scale in favor of an RVT tree especially in IOT systems where cost is also important.

Our analysis of an RVT clock mesh implemented on the sub-system indicates that the clock mesh consumes significantly more dynamic energy than clock trees. This is because a large portion of the clock structure is always running and can never be gated. Leakage energy of the mesh is slightly higher than an RVT tree due to all the clock gates driving the final stage. Effective clock latency of the mesh is comparable to an LVT tree. The OCV derate of the mesh

Table 9.3 Comparison of WSN sub-system energy and performance across clock implementations

TT 0.4 V 25C	RVT tree	LVT tree	RVT mesh
Frequency (kHz)	835	865	847
Dynamic energy (pJ)	8.075	8.025	9.448
Leakage energy (pJ)	0.284	0.311	0.289
Total energy (pJ)	8.359	8.336	9.737
Clock latency (ns)	216	43	163
Effective latency (ns)	216	43	48.1
Clock skew (ns)	77.6	14.7	17.7
Total clock cell area (μm ²)	5940	4999 (0.8% total cell area)	9640
Number of stages	5–13	7–15	1
OCV derate % (TT 0.4 V)	19.5	22.5	66.0
Clock leakage energy (pJ)	0.00436	0.0425	0.00640
Clock dynamic energy (pJ)	0.757	0.628	1.89

(applied only to final ICGs) is much larger than the trees due to the shallow effective clock depth and poor transition on the clock mesh. The RVT clock mesh could be a lower cost alternative to a LVT tree especially for larger designs.

9.6 Perspectives and Trends

The current trend in the industry is towards enabling near-threshold SoCs to be easily and safely designed, implemented, verified and optimized. This is not an easy task however, nor can it be done in isolation. IP providers will have to offer new logical and/or physical IP, EDA tools will need enhancement to support energy optimization and handling of large variation, and silicon foundries will have to provide qualified models. The challenge is in coordinating all of these elements, but there is a real desire and demand for progress, such that the authors are confident that key barriers will be overcome in the next few years.

Looking beyond near threshold, it is clear that there are many innovative and exciting approaches for optimized IoT designs (sub-threshold, adaptive systems, asynchronous, drowsy power gating, non-volatile logic, etc. . .). Similarly to near-threshold there is often an IP & EDA barrier to the adoption of these cutting edge techniques. Unlike near-threshold however, there can also be an analysis barrier—the system-level cost/benefit tradeoffs of such techniques can be complicated to predict and model. Without progress in system-level exploration and design methodology, otherwise very beneficial technology will continue to be overlooked and fail to gain critical mass or wide adoption.

References

- D. Flynn, High performance state retention with power gating applied to CPU subsystems—design approaches and silicon evaluation, HotChips24 2012 Archives HC24.30.p10 (Poster) (2012)
- IEEE Standard for Design and Verification of Low Power Integrated Circuits, <https://standards.ieee.org/findstds/standard/1801-2013.html>
- M. Keating, D. Flynn, A. Gibbons, R. Aitken, K. Shi, *Low Power Methodology Manual—For System-on-Chip Design* (Springer, New York, 2007). ISBN 978-0-387-71818-7
- S. Jain, A 280 mV-to-1.2 V wide-operating-range IA-32 processor in 32 nm CMOS, in *ISSCC*, 2012
- K. Kumagai, H. Iwaki, H. Yoshida, H. Suzuki, T. Yamada, S. Kurosawa, A novel powering-down scheme for low VT CMOS circuits, in *VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on*, Jun 1998 (1998), pp. 44–45
- J.N. Mistry, J. Myers, B.M. Al-Hashimi, D. Flynn, J. Biggs, G.M. Merrett, Active mode subclock power gating. *IEEE Trans. VLSI Syst.* **22**(9), 1898–1908 (2014)
- S. Mutoh et al. A 1v multi-threshold voltage CMOS DSP with an efficient power management technique for mobile phone applications, *ISSCC1996* (1996), pp. 168–169
- K. Shi, D. Howard, Challenges in sleep transistor design and implementation in low-power designs, in *Design Automation Conference, 2006 43rd ACM/IEEE* (2006), pp. 113–116
- M. Stan, Low-threshold CMOS circuits with low standby current, in *Proceedings of the International Symposium on Low-Power Electronics and Design* (IEEE/ACM, Monterey, CA, 1998), pp. 97–99
- SolvNet: Power Compiler I-2013.12 Update Training (ID: 1506611) (Synopsys Inc., 2014)
- D. Taki, T. Shiozawa, K. Ito, Y. Shiba, K. Horisaki, H. Kajihara, T. Yamagishi, M. Sekiya, A. Yamaga, T. Fujita, H. Hara, M. Kuwahara, T. Fujisawa, Y. Unekawa, A 7uW deep-sleep, ultra low-power WLAN baseband LSI for mobile applications, in *Cool Chips XIV, 2011 IEEE*, 20–22 April (2011), pp. 1–3