Aggressive Leakage Management in ARM Based Systems

John Biggs - ARM
Alan Gibbons - Synopsys

ABSTRACT

The management of power consumption for battery life is widely considered to be the limiting factor in supporting the concurrent operation of high performance, complex applications on mobile platforms. At 65nm and below, minimizing the static power dissipation through aggressive techniques such as coarse grain MTCMOS power gating and threshold voltage scaling can yield these significant reductions in power consumption that are necessary. ARM and Synopsys have jointly developed a comprehensive low power technology demonstrator that employs these advanced low power techniques. Various alternative approaches to MTCMOS power gating and threshold voltage scaling are discussed together with a detailed description of the implementation flow and the results.

Table of Contents

Table of Figures

1 Introduction

Managing the power dissipation of complex SoCs has been a prime design consideration for some years now. It is well understood that reducing both the peak and average power consumption will reduce the manufacturing and packaging costs as well as improve the reliability and battery life.

However the thriving market for ever more sophisticated mobile wireless devices such as cell phones, media players, PDAs and cameras is placing ever increasing demands on the battery. Consumers want more and more features in their mobile devices but still demand a convenient form factor and long battery life. Unfortunately battery technology is not developing fast enough to meet this demand and this shortfall is what is driving the demand for cheap, low power, energy efficient SoCs.

1.1 Power Dissipation

There are three major sources of power dissipation in digital CMOS circuits and they can be broken down in to dynamic power dissipation ($P_{switching}$+$P_{short\text{-}cicuit}$) and leakage power dissipation ($P_{leakage}$) as summarized by equation (1).

$$P_{average} = \underbrace{\alpha C_L V_{DD}^2 f_{clk}}_{P_{switching}} + \underbrace{V_{DD} I_{SC}}_{P_{short-circuit}} + \underbrace{V_{DD} I_{leak}}_{P_{leakage}} \quad (1)$$

1.2 Dynamic Power

In order to minimise the dynamic power dissipation term of equation (1) then not only should the clock frequency (f_{clk}) be lowered but also the switching activity (α) and where possible the supply voltage (V_{DD}) should be reduced too.

One of the simplest ways to reduce the switching activity (α) is to inhibit registers from being clocked when it is known that their output will remain unchanged. In a typical SoC as much as 30% of the switching power is dissipated in the clock tree so this technique, known as Clock Gating (CG), can yield a significant saving in both power dissipation and energy consumption[1].

As power is the rate of doing work then the average power dissipation of a system can be reduced by slowing the rate at which work is done. In practice this means lowering the clock frequency (f_{clk}) when the maximum system performance is not required. This technique, known as Dynamic Frequency Scaling (DFS), leads to a linear reduction in average power dissipation but unfortunately does not reduce the energy consumption for a given task as the work done remains a constant. For some very "leaky" processes the total energy consumption may in fact increase due to spending longer in active mode.

However, if at the same time as reducing the clock frequency, the voltage is also reduced to a level that is just high enough to support this lowered clock frequency, then there is less work to do in charging the internal capacitances to the supply voltage (V_{DD}) and so less energy is consumed. This technique, known as Dynamic Voltage and Frequency Scaling (DVFS), leads to a quadratic reduction in energy consumption and a cubic reduction in average power dissipation[2]. It should be noted that, as it is not possible to dynamically scale the voltage and

frequency instantaneously, there is some energy overhead in moving between the various performance levels.

1.3 Leakage Power

The other source of power dissipation is leakage power which is predominantly due to the fact that transistors are not perfect switches and so can never be completely turned off.

Although leakage power used to be considered insignificant when compared to dynamic power at 90nm, it has become significant and at 65nm, it is dominant and so can no longer be ignored.

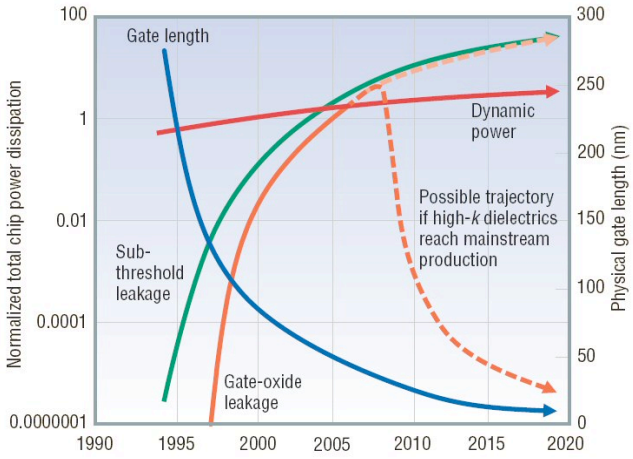
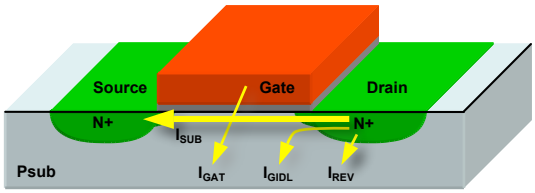


Figure 1 - Trends in Power Dissipation[3]

Leakage power is dissipated in both active mode and standby mode and the currents which go to make up the total leakage are increasing fast (Figure 1). In some applications it may be more energy efficient to run fast and stop rather than to lower the voltage and frequency due to the high active leakage currents.

There are four main sources of leakage currents in a CMOS transistor (Figure 2)

1. Sub-threshold Leakage (I_{SUB}): the current which flows from the drain to the source current of a transistor operating in the weak inversion region.
2. Gate Leakage (I_{GATE}): the current which flows directly from the gate through the oxide to the substrate due to gate oxide tunneling and hot carrier injection.
3. Gate Induce Drain Leakage (I_{GIDL}): the current which flows from the drain to the substrate induced by a high field effect in the MOSFET drain caused by a high $V_{DG.}$
4. Reverse Bias Junction Leakage (I_{REV}): caused by minority carrier drift and generation of electron/hole pairs in the depletion regions.



$$I_{LEAK} = I_{SUB} + I_{GATE} + I_{GIDL} + I_{REV} \quad (2)$$

Figure 2 - Components of leakage current in an NMOS transistor

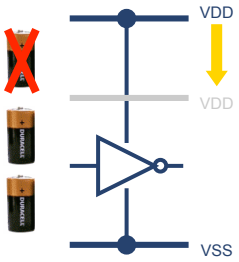
Of the various components which go to make up the total leakage current (I_{LEAK}) it is currently the sub-threshold leakage (I_{SUB}) which is dominant. However, the gate leakage (I_{GATE}) is becoming significant but may yet be mitigated by high K dielectric material such as TiO_2 and TaO_5[4]

The most effective techniques for mitigating sub-threshold leakage are Power Gating and VTCMOS, both of which will be described later.

1.4 Leakage Power Mitigation Techniques

There are a number of leakage mitigation techniques available to reduce the various leakage currents in both active and standby mode[5]. Some techniques such as Dual V_T and VTCMOS rely on additional support in the manufacturing process to lower the leakage whilst others such as Power Gating and Stack Effect are stand-alone circuit techniques.

1.4.1 Lower V_{DD}



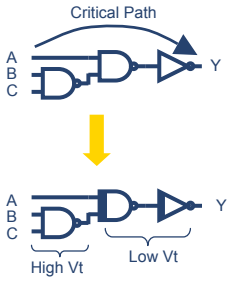
Again by referring to equation (1), it can be seen that leakage power will reduce with the lowering of the supply voltage (V_{DD}). However, any reduction in V_{DD} also reduces V_{GS} which impacts the MOSFET gate drive (V_{GS}-V_T). It can be seen from equation (3) that a reduction in (V_{GS}-V_T) significantly reduces the MOSFET's drive strength (I_{DS}).

$$I_{DS} = \mu C_{ox} \frac{W}{L} \cdot \frac{(V_{GS} - V_T)^2}{2} \quad (3)$$

Some of this loss in performance can be regained by lowering the threshold voltage (V_T) to restore the loss in gate drive, however lowering the threshold voltage (V_T) results in an exponential increase in the sub-threshold leakage current (I_{SUB}) and hence overall the leakage power increases - see equation (4). So, in order to manage the overall leakage power, the number of high leakage low V_T transistors should be kept to a minimum

$$I_{SUB} = \mu C_{ox} V_{th}^2 \frac{W}{L} \cdot e^{\frac{V_{GS} - V_T + \eta V_{DS}}{n V_{th}}} \cdot \left(1 - e^{\frac{-V_{DS}}{V_{th}}}\right) \quad (4)$$

1.4.2 Dual V_T



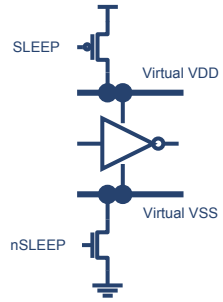
It is now quite common to use a "Dual V_T" flow during synthesis to ensure that the total number of low V_T transistors is kept to a minimum by only deploying low V_T cells when required.

This usually involves an initial synthesis targeting a prime library in the conventional manner followed by an optimization step targeting one (or more) additional libraries with differing thresholds[5].

As more often than not there is a minimum performance which must be met before optimizing power then in practice this usually means targeting the high performance, high leakage library first and then relaxing back any cells not on the critical path by swapping them for their lower performing, lower leakage equivalents.

If however minimizing leakage is more important than achieving a minimum performance then this process can be done the other way around by targeting the low leakage library first and then swapping in higher performing, high leakage equivalents in speed critical areas.

1.4.3 Power Gating



A far more aggressive and effective technique for leakage mitigation is to simply cut the power supply to any inactive transistor.

Fundamentally, this is done by placing switches in the power network, the ground network, or both. However, the exact placement and sizing of these switches must be done with great care so as not to have an adverse impact on performance. These switches are known as "power gates" and can be distributed throughout the power/ground network in either a "coarse gain" or a "fine gain" manner.

Fine Grain power gating is when the switch is placed locally inside every standard cell in the library (Figure 3). Since this switch must supply worst case current required by the cell, it has to be quite large in order to not impact performance. In order to keep this area overhead to a minimum, fine grain power gates are usually implemented as "footer" switches in the ground as NMOS transistors have a lower on-resistance than PMOS and so will be smaller.

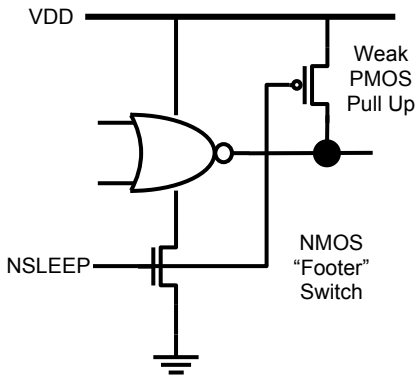


Figure 3 – Fine Grain Power Gating

Although the area overhead of each cell is quite large (often 2x-4x the size of the original cell), overall the area overhead of fine grain power gating will be much less as it is only necessary to power gate the high leakage, low threshold cells.

As not all cells are power gated and some will remain powered, it is important to ensure that the inputs to these cells don't float in order to avoid crowbar currents. This means that every power gated cell must have additional circuitry to "clamp" its outputs to a valid CMOS logic level. In the case of fine grain power gates this means adding a weak PMOS pull up on each output (Figure 3).

The key advantage of fine grain power gating is that the timing impact of the IR drop across the switch and the behavior of the clamp is easy to characterize as they are contained within the cell. This means that it is still possible to use a traditional design flow to deploy fine grain power gating although care must be taken over the routing of the sleep signal. However, the larger footprint of the power gated cells means that swapping between high threshold (non power gated) and low threshold (power gated) cells is more complex than that of the traditional Dual V_T flow.

Coarse Grain power gating is when the switch is placed such that it is shared amongst a number of cells (Figure 4). The sizing of a coarse grain switch is much more difficult than a fine grain switch as the exact switching activity of the logic it supplies is not known and can only be estimated. Also, it is common to have distributed coarse grain power gating where the outputs of all the switches are joined to create a "virtual" power or ground. This just complicates the switch sizing calculations still further as each power gated cell is in fact fed by a number of switches connected in parallel.

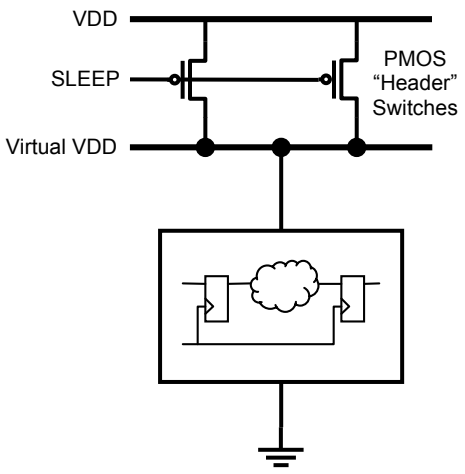


Figure 4 - Coarse Grain power Gating

The size of a course grain switch will be much less than the sum of the equivalent fine grain switches of the logic it supplies. This is because for a given block of logic the switching activity will not only be far less than 100% but due to the propagation delay through the cells the

switching activity will be distributed in time. As coarse grain power gating switches do not have the same area overhead as fine grain it is possible to use the slightly larger "PMOS "header" switch in the power supply instead. This not only has the advantage of a common ground plane but also means that the outputs of power gated blocks can be clamped to this common ground, which is convenient for multi voltage design. Also, with coarse grain power gating, not as many clamps are needed as they are only required at the block outputs rather than on every cell.

Unlike fine grain power gating, when the power is switched in coarse grain power gating, the power is disconnected from all logic, including the registers, resulting in the loss of all states. If state is to be preserved whilst the power is disconnected then it must be stored somewhere which is not power gated. Most commonly this is done locally to the registers by swapping in special "retention" registers which have an extra storage node that is separately powered. There are a number of retention register designs which trade off performance against area. Some use the existing slave latch as the storage node whilst others add an additional "balloon" latch storage node. However, they all require one or more extra control signals to save and restore the state[7].

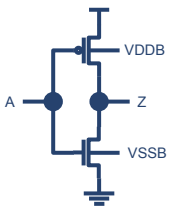
The key advantage of retention registers is that they are simple to use and are very quick to save and restore state. This means that they have a relatively low energy cost of entering and leaving standby mode and so are often used to implement "light sleep". However in order to minimize the leakage power of these retention registers during standby, it is important that the storage node and associated control signal buffering is implemented using high threshold low leakage transistors.

If very low standby leakage is required then it is possible to store the state in main memory and cut the power to all logic including the retention registers. However, this technique is more complex to implement and also takes much longer to save and restore state. This means that it has a higher energy cost of entering and leaving standby mode and so is more likely to be used to implement "deep sleep".

One of the key challenges in power gating is managing the in-rush current when the power is reconnected. This in-rush current must be carefully controlled in order to avoid excessive IR drop in the power network as this could result in the collapse of the main power supply and loss of the retained state.

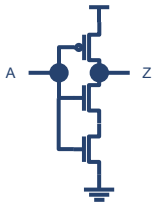
In summary, although fine grain power gating is easier to implement, it has the disadvantage of requiring a completely new cell library with the integrated power gates which have a significant area impact. Coarse grain power gating on the other hand is more complex to implement and verify[7] . It may require special tooling but has the advantage of less area overhead and only requires the addition of retention registers, isolation clamps and power gates to the library.

1.4.4 VTCMOS



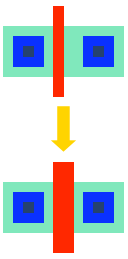
Variable Threshold CMOS (VTCMOS) is another very effective way of mitigating standby leakage power. By taking advantage of the body effect and reverse biasing the substrate, it is possible to reduce the standby leakage by up to three orders of magnitude. However, VTCMOS adds complexity to the library views and requires two additional power networks to separately control the voltage applied to the wells. Unfortunately, the effectiveness of reverse body bias has been shown to be decreasing with scaling technology[9].

1.4.5 Stack Effect



The Stack Effect, or self reverse bias, can help to reduce sub-threshold leakage when more than one transistor in the stack is turned off. This is primarily because the small amount of sub-threshold leakage causes the intermediate nodes between the stacked transistors to float away from the power/ground rail. the reduced body-source potential (more This results in a slightly negative gate-source drain voltage (which reduces the sub-threshold leakage) as well as a reduced drain-source potential (less DIBL) which, together with body effect), increases the threshold, again lowering leakage. The leakage of a two transistor stack has been shown to be an order of magnitude less than that of a single transistor[10]. Also this stacking effect makes the leakage of a logic gate highly dependant on its inputs and so there is a minimum leakage state for a particular circuit which could be applied just prior to halting the clocks.

1.4.6 Long Channel Devices



Using non-minimum length channels will reduce the active leakage as well as standby leakage by avoiding the V_T roll off that occurs in short channel devices. Unfortunately, long channel devices have larger area and therefore greater gate capacitance which has an adverse effect of performance and dynamic power consumption. This means that there may not be a reduction in total power dissipation unless the switching activity of the long channels is low. Therefore, switching activity must be taken in to account when choosing gates whose transistor lengths are to be increased. However, the properties of long channel devices make them very suitable for the implementation of power gates.

2 Synopsys ARM Leakage Technology Demonstrator

Synopsys and ARM have a long history of working together on lowering the barriers to the adoption of advanced methodologies for the rapid deployment of ARM synthesizable IP with Synopsys tools[2][3][11][12][13].

The **S**ynopsys **A**RM **L**eakage **T**echnology demonstrator known as "SALT" was an R&D collaboration implemented in TSMC90G to explore the practical details of implementing some of the more aggressive leakage mitigation techniques described above. Specifically we chose to implement Coarse Grain Power Gating together with Dual V_T and VTCMOS as these techniques are the most effective at combating standby leakage power dissipation in the 90nm node.

2.1 SALT Design

The design of the SALT technology demonstrator was based on an established ARM926EJS reference system[3] with the addition of a prototype next generation Intelligent Energy Controller ("IEC") for leakage control and an Synopsys DesignWare OTG PHY (Figure 5). The ARM926EJS was partitioned into two voltage domains to allow the RAMs to remain powered whilst the core logic was switched off. The design also implemented in-rush current management with a "soft-start" to avoid any adverse IR drop in the power supply during start up.

The SALT design has support for four levels of standby leakage power management:

1. **Halt** – simple stopping of the clocks.

2. **Light Sleep** – the CPU is power gated and the state retained in retention registers.
3. **Deep Sleep** – the CPU is switched off and the state retained in RAM
4. **Shutdown** – both CPU and RAM are switched off so the state is not retained.

The sequencing of the various control signals for entering and leaving these sleep modes is managed by the Intelligent Energy Controller ("IEC").

The implementation of Deep Sleep uses a novel scan based technique together with a dedicated AMBA bus master to store the state in any AHB connected memory. This will be described in more detail later.

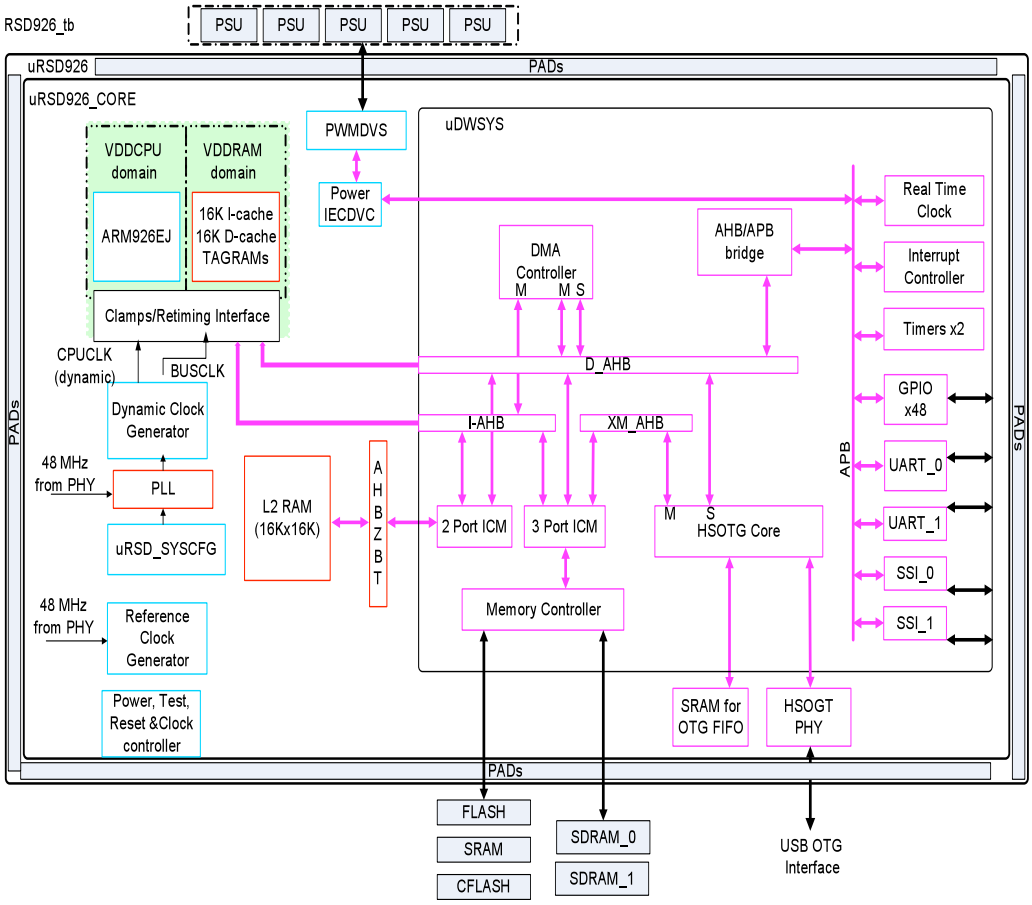


Figure 5 - SALT Architecture

2.2 SALT Library

The SALT technology demonstrator targeted an experimental "R&D" library based on Artisan's SAGE-X standard cell library in TSMC90G process. In order to support VTCMOS it was necessary to target a triple well process and add deep nwell to each cell. Also it was decided to add an extra 10th track supplying true V_{DD} to the top of each cell in the library in order to simplify the distribution of the un-switched power to the "always on" buffers and retention registers. In addition to these modifications, a power management kit consisting of the following cells was also created, drawn to the same standard cell rules:

- **Power gates** to disconnect the power from the logic.

- **Isolation Clamps** to preserve CMOS logic levels on the power gated outputs
- **Always On Buffers** to drive power management signals, clocks and reset.
- **Retention Registers** to retain the state whilst power gated.
- **Schmitt Trigger** for in-rush current management.
- **Well Ties and Deep nwell End Caps** for VTCMOS support

The power gates were implemented as PMOS "headers" in order to have a common ground plane and also so active high power gated outputs get clamped inactive when isolated.

To use the deep nwell layer, it was necessary to also create a set of physical only deep nwell capping cells. These must be placed around the standard cell region to ensure that there is sufficient nwell overlap of deep nwell at the ends of each standard cell row to meet the design rules.

Finally this new "R&D" experimental library was recharacterised at lower voltage to account for the estimated IR drop across the PMOS "header" switches.

This collection of additional cells became known as a "Power Management Kit" and formed the basis of a prototype library which has now been productized (without the 10th track!) as ARM's PMK.

2.3 SALT Implementation

The implementation employed the 2005.09 XG Galaxy design platform and the flow was largely based on the ARM Synopsys IEM Reference Methodology[2] which extends the standard ARM Synopsys Galaxy Reference methodology to have support for multiple voltage domains and dual V_T. The only additional functionality that was required over and above this flow was the ability to perform state retention synthesis, size and place the power gates, implement the in-rush current management circuitry and add deep nwell capping cells.

Although there was full support for state retention synthesis in the tools, the placement of the power gates, implementation of the in-rush current management and support for deep nwell were all somewhat manual steps.

In order to minimize the impact on the tools and flow it was decided to implement these power management cells as "physical only" cells which could be placed in Jupiter during the floor planning stage. This will be described in more detail later.

3 Key Implementation Challenges

3.1 Power Gating

In coarse grain power gating there is a clear trade off between the size, number and spacing of the switches, simplistically the fewer there are the bigger they need to be. However, it is not quite as simple as that as some subtle short channel effects come in to play. For example, increasing the gate length by a small percentage can significantly reduce the leakage current and the leakage per unit width generally goes down as the transistor width is reduced (Figure 6). After much simulation it was decided that a switch transistor of width 0.55µm and length 0.13µm

(TSMC90G) provided a good trade-off between area and the I_{ON} to I_{OFF} ratio and so the power gates were built using multiple transistors of this size in parallel.

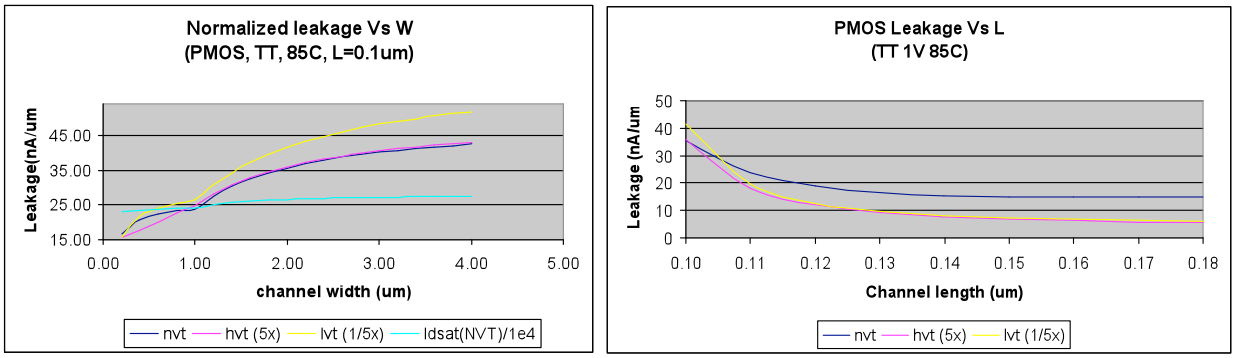


Figure 6 – Leakage Current vs. Gate Width and Length (TSMC90G)

For several reasons, not least layout convenience, the number of transistors in a power gate was chosen to be 30, so now the resistance (R_{ON}) was fixed the spacing could be determined. This was again done by running many HSPICE simulations on a representative test circuit varying the number of headers and the load that they were supplying. The effects on signal delay, IR drop and leakage were then measured. It was found that a power gate was required approximately every 50µm in order to have less than a 5% IR drop in the switched power supply at 250MHz.

The power gates were laid out as double height cells in such a way that they would easily stack in columns with all the vertical connectivity done by abutment. This meant there was enough room to integrate all the necessary re-buffering of the control signals. A script was then written to place these power gates in columns every 50µm throughout the VCPU placement region (The 40 or so columns of header cells can be clearly seen in Figure 7).

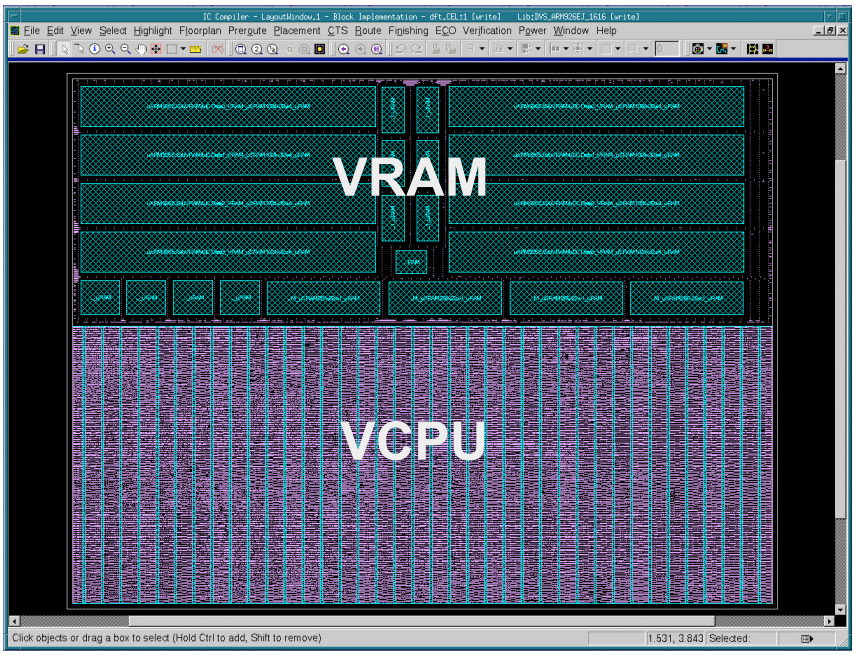


Figure 7 - SALT926 CPU Floor Plan Showing Power Gates in Columns

Once the power gate network was sized and placed, extensive PrimeRail analysis was performed to verify the IR drop from the pads through the VDD mesh, and across the power gates. It was found to be 18mV, well within the 50mV budget (assuming a 20% switching activity).

3.2 In-Rush Current Management

The "soft start" was implemented by building two networks of power gates, a daisy chain of weak "starter" power gates and a network of full power gates. These were then linked by a Schmitt trigger which senses the level of the switched "virtual" V_{DD} and, when the level reaches approximately 90% of the un-switched "true" $V_{DD,}$ it engages the main power gate network and asserts a "ready" signal (Figure 8). The Schmitt trigger cell in the R&D experimental library also had an integrated AND function to gate in the SLEEP signal to ensure that the READY signal is de-asserted as soon as SLEEP is asserted with out having to wait for the virtual V_{DD} network to discharge.

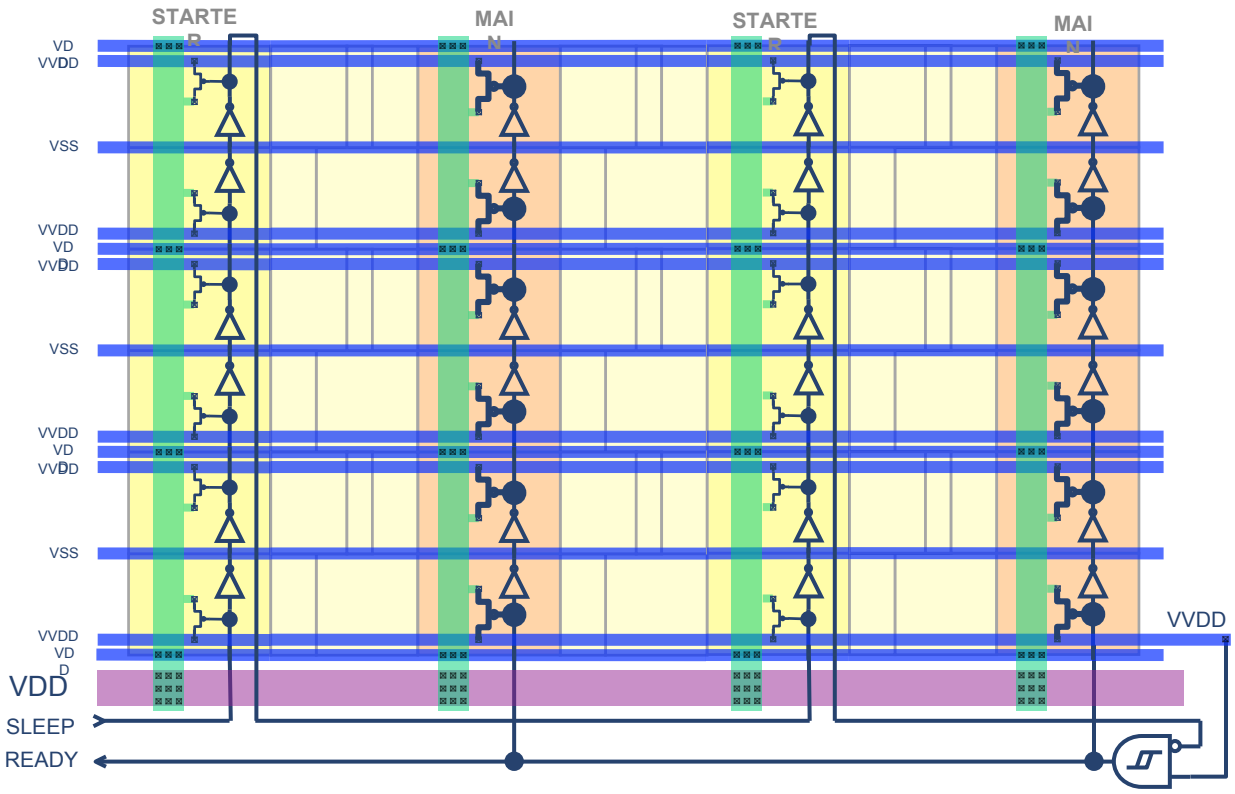


Figure 8 – Conceptual Representation Of In-Rush Current Management Circuit

The whole circuit was simulated using NanoSim to verify the in-rush current and switch on times. It was found that the maximum in-rush current was no more than 80mA and it took just under 100nS from de-asserting SLEEP to bring the switched "virtual" V_{DD} up to operating voltage and for the Schmitt trigger to fire and assert READY(Figure 9).

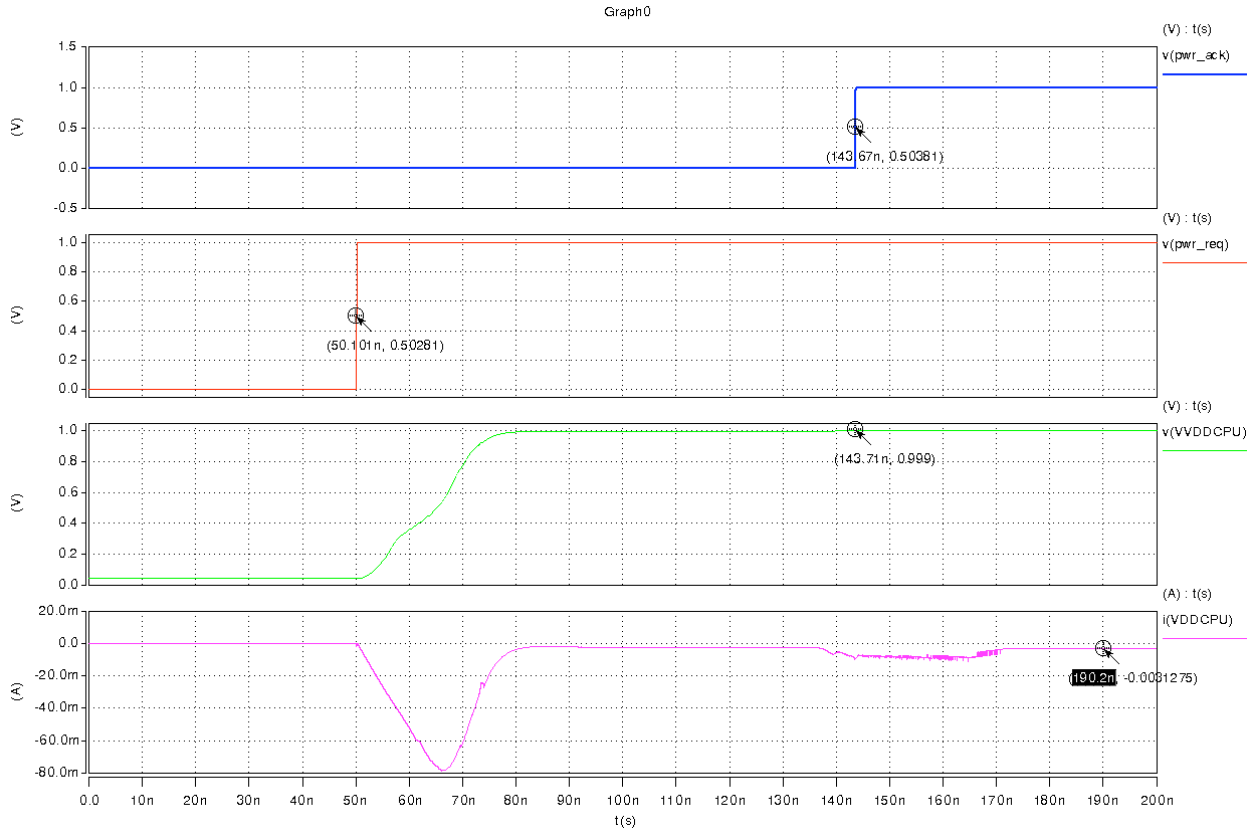


Figure 9 - Soft Start

3.3 State Retention

The design of SALT employed aggressive coarse grain power gating to disconnect the power from both the ARM926EJS processor and the OTG USB core when in standby mode. However, to ensure a quick return from standby back into active mode, it is necessary to preserve the state whilst the power is gated. Two state retention techniques were implemented in SALT: one for "light" sleep, where the state was stored locally in retention registers, and the other for "deep" sleep, where the state was scanned out and stored in memory.

3.3.1 Retention Registers

For most designs it is not strictly necessary to preserve the contents of every storage element whilst in standby as only the salient "architectural" state needs to be preserved. In the case of the ARM926EJS, this essential state is in effect the state relating to the programmer's model. However, unless this essential state is explicitly marked in the source RTL, it is very difficult to infer during implementation. In the SALT implementation it was decided to simply convert every register in the ARM926EJS into a retention register to ease the verification process.

This was done using Power Compiler in the following manner:

```
set power_enable_power_gating true
set_power_gating_style -type DRFF
set_power_gating_signal -type DRFF nrestore
compile_ultra -scan
hookup_power_gating_ports -type DRFF -port_naming_style nrestore
```

As previously mentioned there are a number of styles of retention register design which trade off speed, power and area. The simplest design uses the existing slave latch as the storage node which must be kept powered during power gating and should be implemented using high threshold, low leakage transistors. However, although this design has a minimal area overhead and only requires one additional control signal, it does unfortunately suffer from a loss in performance due to the high threshold transistors of the storage node being on the data path. This performance impact can be avoided by keeping the high threshold, low leakage transistors off the data path by adding a "balloon" latch storage node off to one side. Although this design results in minimal impact on performance, there is an area overhead and unfortunately it requires two additional control signals.

The retention register used in SALT was a prototype of the one that is now available in ARM's Power Management Kit. The design of this "PMK" retention register manages to retain the performance of the "balloon" style whilst having the same simple control as the "live slave" (Figure 10).

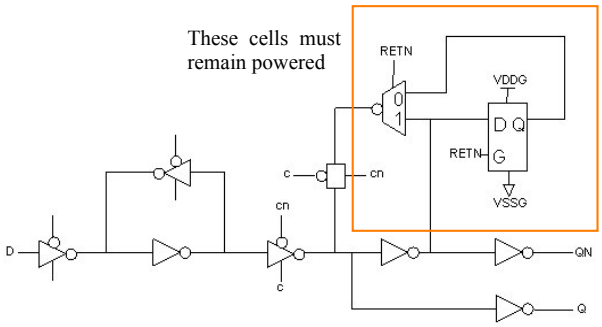


Figure 10 - PMK Retention Register

3.3.2 Scan Hibernate

In order to reduce the leakage still further the power to the ARM926EJS can be shut off completely. In this case, the state can not be stored locally in retention registers and must be stored elsewhere before the power is disconnected.

A novel bus transaction based technique was developed to save and restore state to any AHB connected memory. This technique (called "Scan Hibernate") involved padding out the number of retention registers to ensure that the number was a multiple of 32 so that the state could be scanned out and presented in a series of 32 bit words to a dedicated AMBA bus master to be saved to memory (Figure 11). The design of this dedicated bus master included an implementation of the "CRC-32" algorithm to check the integrity of the restored the data.

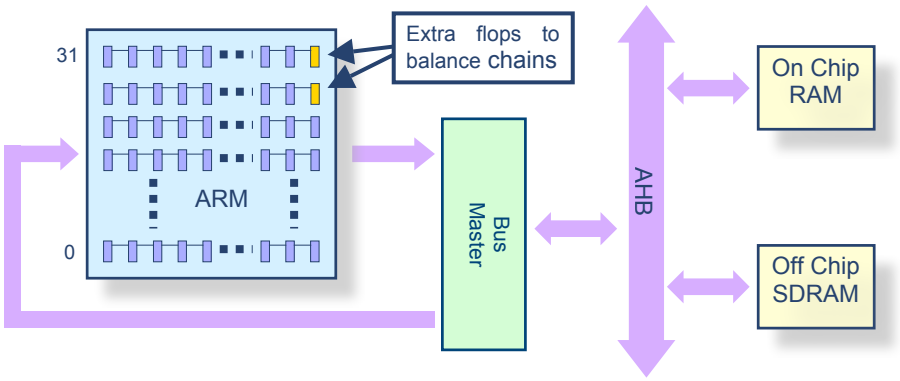


Figure 11 - Scan Hibernate

An interesting use of this "Scan Hibernate" system is to verify the integrity of the state restored from the retention registers. This can be done by storing the state to memory as well as the retention registers before entering light-sleep mode and then storing the restored state to memory immediately after return to active mode. By comparing the two images of the state from before and after power gating it is possible to verify whether any state got corrupted. This is a very useful diagnostic technique which can be used to explore the low voltage operation of the retention registers as well as the effects of in-rush current induced IR drop.

3.4 Variable Threshold CMOS (VTCMOS)

The implementation of VTCMOS requires a triple well process so that (assuming a p-type substrate) "deep" nwells can be placed under the pwells in order to isolate them from each other so that they can be held at different potentials. In addition to this extra process step, VTCMOS also requires a "tapless" library with floating wells so that special cells which have independent contact with the wells can be placed at regular intervals to set the body bias. These special well bias cells then need to be all connected together with two extra power meshes, V_{DDB} for the nwell and V_{SSB} for the pwell.

As all the power gates in SALT were arranged in columns placed at regular intervals, it was convenient to make the well bias connections by incorporating them into the layout of each power gate cell. This meant that the implementation of VTCMOS almost came for free as all the vertical connectivity of these extra power nets was done by abutment between each power gate cells just like the SLEEP signal in the in-rush current management.

To complete the VTCMOS implementation, it was necessary to place a ring of special deep nwell "capping" cells around the standard cell region in order to meet the minimum nwell overlap of deep nwell as prescribed by the TSMC90G rules.

4 Conclusions and Future Work

As we move down the process generations leakage currents are fast becoming a significant source of power dissipation in both active and standby modes. Various techniques for mitigating leakage power were investigated and Power Gating, Dual V_T and VTCMOS were found to be the most effective. These techniques were then explored in further in practical detail through an

ongoing collaborative R&D program with Synopsys to investigate aggressive leakage mitigation techniques on ARM based systems. The first phase of this program was focused on understanding the technology and yielded the design described in this paper (which at the time of writing was still in fab). Being an R&D project, the demands of this design were a little ahead of the capabilities in both the tools and the library and so certain back roads had to be taken in order to complete the tape out. However many valuable lessons have been learned, some of which have already been factored into the latest releases of ARM's Power Management Kit and Synopsys tools.

When the silicon comes back, we plan to investigate the real time impact and entry/exit energy costs of the various sleep modes in order to further develop the next generation "Intelligent Leakage Controller". Also, we plan to verify the effectiveness of the in-rush current management by using the diagnostic features of the "scan hibernate" system as well as benefits of VTCMOS in both forward and back bias modes.

The second phase of this collaborative program is focused on defining a set of best practices for the rapid deployment of ARM IP with Synopsys tools to provide a complete low power design solution based on open industry standards for our mutual customers.

5 Acknowledgments

The authors would like to express a special thanks to both ARM and Synopsys for their continued support with this collaboration. In addition they would like to specifically thank Dave Flynn, Mike Keating, Dave Howard, Sachin Idgundi and last but not least Rich Goldman.

6 References

[1] Greenhalgh P. "Power Management Techniques for Soft IP" SNUG Europe 2004

[2] Biggs, J. Uttley, P. "Rapid implementation of an IEM enabled ARM1176JZF-S with Galaxy Power" SNUG 2005

[3] Kim, N.S. Austin, T. Baauw, D. Mudge, T. Flautner, K. Hu, J.S. Irwin, M.J. Kandemir, M. Narayanan, V. "Leakage current: Moore's law meets static power" IEEE Computer Vol. 36, Issue 12, 2003

[4] S. Borkar, "Design Challenges of Technology Scaling" IEE Micro Vol. 19, Issue 4, 1999.

[5] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits", IEEE proceedings 2003

[6] V. Sundararajan, P. Keshab "Low Power Synthesis of Dual Threshold Voltage CMOS VLSI Circuits" ISLPED1999.

[7] Kao, J. Chandrakasan, A "MTCMOS Sequential Circuits" ESSCIRC 2001

[8] Calhoun, B. Honore, F. and Chandrakasan, A. "A leakage reduction methodology for distributed MTCMOS," IEEE J. Solid-State Circuits, vol.39, no.5, pp.818–826, May 2004.

[9] A. Keshavarzi, C. F. Hawkins, K. Roy, and V.De, "Effectiveness of reverse body bias for low power CMOS circuits" Symp. VLSI Design 1999.

[10] Y. Ye, S. Borkar, and V. De, “New technique for standby leakage reduction in high-performance circuits,” Symp. VLSI Circuits, 1998.

[11] Whitfield, T. Gent, C. Christy, R, “Implementation of an ARM Core with Performance Scaling for Intelligent Energy Management” SNUG Europe 2004

[12] Biggs, J. Gibbons, A. “Enabling Core Based Design” SNUG Europe 2002

[13] Flynn, D. Flautner, K. Patel, D. Roberts D. “IEM926: An Energy Efficient SoC with Dynamic Voltage Scaling” DATE 2004